Exploiting Socially-Aware Tasks for Embodied Social Navigation

Enrico Cancelli1*  Tommaso Campari1,2*  Luciano Serafini2  Angel X. Chang3  Lamberto Ballan1

1 University of Padova  2 Fondazione Bruno Kessler (FBK)  3 Simon Fraser University

Abstract

Learning how to navigate among humans in an occluded and spatially constrained indoor environment, is a key ability required to embodied agents to be integrated into our society. In this paper, we propose an end-to-end architecture that exploits Socially-Aware Tasks (referred as to Risk and Social Compass) to inject into a reinforcement learning navigation policy the ability to infer common-sense social behaviors. To this end, our tasks exploit the notion of immediate and future dangers of collision. Furthermore, we propose an evaluation protocol specifically designed for the Social Navigation Task in simulated environments. This is done to capture fine-grained features and characteristics of the policy by analyzing the minimal unit of human-robot spatial interaction, called Encounter. We validate our approach on Gibson4+ and Habitat-Matterport3D datasets.

1. Introduction

Navigating safely in a dynamic scenario populated by humans who are moving in the same environment is necessary for embodied agents such as home assistants robots. To do so, as depicted in Figure 1, the agent should be able to dynamically and interactively navigate the environment by avoiding static objects and moving persons.

Recently, the development of photorealistic 3D simulators [18, 31, 32] has provided the tools to train embodied agents and experiment in large-scale indoor environments [8,15,28]. Thanks to these frameworks, several tasks, and challenges have been presented [1, 14, 46]. In particular, in the PointGoal Navigation task (where an agent is required to reach a specific location in an environment), an agent without any sensor/actuation noise trained for billions of steps can obtain almost perfect performance [39]. Other approaches [23,43] obtained impressive results even in the presence of noise. Another relevant task is Object Goal Navigation, where an agent is required to find and navigate to a specific object instance. This task requires both semantic and navigation capabilities; to this end, modular approaches based on semantic-maps [6,9,27], as well as end-to-end models [29,42] have been presented lately. High-level semantic understanding is even more critical in Vision-Language Navigation [30,36,45].

However, although challenging and despite encouraging progress, all the previously mentioned tasks frame navigation in a fundamentally static environment. The dynamic element introduced by sentient, moving human beings in the scene forces us to rethink how the current models are designed. A good navigation policy must not be just effective (i.e., able to achieve its goal) and efficient (i.e., able to achieve the objective through a close-to-optimal path) but also safe (reaching the destination without harming others). This social element is included in the Social Navigation Task [25,40], where an agent must tackle PointGoal Navigation in simulated, indoor, and crowded environments. In this scenario, [44] recently introduced a simple but quite effective model, although the approach can not explicitly encode any social behavior in its navigation policy. We be-
lieve that a clear encoding of human-agent interactions, as well as social behaviors, are required in complex scenarios in which the embodied agent cooperates and interacts with humans. In this way, the agent could prevent collisions or dangerous behaviors and adapt its path to the dynamic environment in which it is navigating. We encode these “signals” by introducing two Socially-Aware Tasks, referred as risk and social compass. These tasks model the present and future danger connected to the agent’s action.

Additionally, we define an extensive evaluation protocol for the Embodied Social Navigation task in order to better analyse the performances in case of human-agent interactions. This is inspired by a similar attempt that has been recently introduced in robotics [26], consisting in collecting statistics about specific encounters between humans and a robot (through questionnaires). We propose an automated procedure for fine-grained human-agent interactions. To this end, given a specific episode, we extract short subsequences of interest in which a social interaction becomes the predominant factor influencing navigation, called encounters. Each encounter is associated with a corresponding category based on the type of human-agent interaction occurring, following a set of predetermined rules. Finally, we also created a dataset for Embodied Social Navigation to assess our agents in different environments. This dataset was built on top of HM3D [28].

In summary, the contributions of this work are threefold:

- A novel architecture for embodied social navigation which is based on Socially-Aware tasks; we prove the effectiveness of the model on two public datasets.
- A new Encounter-based evaluation protocol for social navigation models.
- An extended dataset for embodied social navigation based on the popular HM3D dataset (called HM3D-S).

2. Related Work

Socially-Aware Navigation. Socially/Human Aware representations and models have been studied by several researchers in the field of robotics, computer vision and human social behavior analysis [22]. Some works focused on collision avoidance algorithms that, similarly to earlier models like ORCA [5] or RVO [35], try to enable collaborative, collision-free navigation of non-communicating agents in a shared space. Modern approaches employ Deep Reinforcement Learning (DeepRL) to learn a motion policy that can produce a safe path to a goal for every agent, by enabling efficient online prediction of future states of its neighbours [13, 20].

Other works explicitly tackle the problem of motion planning and navigation in environments with dynamic obstacles [3] and/or humans [11, 12, 16, 21]. [12] employs collision avoidance algorithms like CADRL [13] and introduces common-sense social rules to reduce uncertainty while minimizing the risk of incurring in the Freezing Robot Problem [34]. [11, 16], instead, try to model human-agent interaction using techniques like Spatio-temporal graph [21]. These methods have been widely tested on minimalist simulation environments that provide complete knowledge and simple obstacles and often assume collaboration between moving agents. Limited tests have been conducted on real-world scenarios, but they often require a large set of sensors for free-space detection, mapping obstacles, and sensing humans [12].

Recently, an increasing amount of works have been focusing on exploiting egocentric visual data to learn navigation policy to operate in social environments with partial information. [33] used depth maps to train an agent using imitation learning and studied the policy behavior in a set of simulated interactions. [25], instead, focuses on constrained indoor environments and uses a combination of a global planner with complete map access and a low-level RL policy that exploits data from a LIDAR. This approach was tested on a set of simulations and evaluated using both standard metrics and domain-specific like Human Collision Rate. However, this approach requires prior knowledge about the environment where it operates for the path planner to work.

Embodied Navigation. Embodied Navigation had a surge in the last years [14]. Mainly, this was possible thanks to large-scale datasets consisting of 3D indoor environments [8, 28, 32], and to simulators that allow researchers to simulate navigation inside these 3D environments [18, 31, 32]. In this context were proposed many tasks [1] such as: PointGoal Navigation [39], ObjectGoal Navigation [4], Embodied Question Answering [38], and Vision and Language Navigation (VLN) [2, 19]. To tackle these problems, where an agent operates in static, single-agent environments, modular approaches were proposed [7, 9, 10, 27], exploiting SLAM, Path Planning and exploration strategies, and end-to-end RL-trained policies [6, 23, 37, 42, 43], without exploiting any explicit map. In this paper, we will focus on the Social Navigation Task [25, 40], where the agent has to navigate in environments with moving humans. This adds new challenges to Visual Navigation since social capabilities are required to avoid collisions. Modular approaches are harder to adapt in this context since humans are constantly moving and therefore harder to track.

In [44], Yokoyama et al. proposed an end-to-end RL-trained policy for Social Navigation. This model extracts embeddings from the Depth and the GPS+Compass sensors and feeds them to a GRU, together with the previous action. However, this model did not exploit any Social information and was evaluated only using success rate, which is limiting since the agent is dealing with humans and it is preferable
to navigate safely in order to avoid collisions, even if this means having lower success rate. There have been some attempts to do a fine-grained evaluation for Social Navigation. For example, [26] defined an evaluation protocol for social agents based on human questionnaires. This cannot be easily applied to simulations, where test sets contain thousands of episodes. In this paper, we will propose an automatic evaluation protocol to measure the social capabilities of our models.

3. An evaluation protocol for SocialNav

SocialNavigation Task. In Social Navigation [25,40,44], as in PointGoal Navigation, the agent aims to reach a target location, but a collision with a human subject constitutes a failure and will terminate the episode. An episode $e$ is characterized by the agent trajectory $\alpha$, the tuple of human trajectories $(p_i)$, and the target goal $g$.

The agent trajectory $\alpha$ is a sequence of positions and rotations of the agent from the beginning to the end of the episode $t_{\text{end}}$. Formally, $\alpha = \{\alpha_t\}_{t \in [0,t_{\text{end}}]}$ where $\alpha_t \in SE(2)$

\[ \text{is the 2D translation and rotation of the agent with} \]

\[ \text{respect to the origin at time } t. \]

Similarly, the trajectories of humans in the episode are sequences of positions and rotations associated with the $i$-th human. Formally, $p_i = \{p_i^t\}_{t \in [0,t_{\text{end}}]} \forall i \in \mathcal{P}$ with $p_i^t \in SE(2)$. In our simulation, the movement of each person is constrained by an associated starting point and an endpoint, with the person moving back and forth between those two points following the shortest path.

The target goal $g \in \mathcal{G}$ is specified by the 2D position in world coordinates. The agent must at any point in time provide an action $(\text{lin}_\text{vel}, \text{ang}_\text{vel}) \in [-1,+1]^2$, representing the normalized linear forward velocity and the normalized clockwise rotational angular velocity (where $+1$ is the maximum velocity and $-1$ the maximum backward/counter-clockwise velocity). The stop action is automatically called when the agent is within 0.2 meters from the target goal point. The agent has 500 actions (or steps) to reach the target location. If it collides with a human, the episode terminates immediately.

3.1. Evaluation Protocol

Given an episode $e$, we define an encounter as follows:

**Definition 3.1.** An encounter taking place in episode $e$ between the agent and a specific pedestrian $i \in \mathcal{P}$, is defined as a subsequence of trajectories $\alpha$ and $p_i$ in a given timeframe $[t_1,t_2] \subseteq [0,t_{\text{end}}]$ such that the following conditions are met:

- **Time Constraint**: the timeframe $[t_1,t_2]$ is larger than a threshold $T_{\text{min}}$;

- **Spatial Constraint**: the geodesic distance between the agent and person $i \forall t \in [t_1,t_2]$ is less than a threshold $D_{\text{max}}$;

- **Heading Constraint**: person $i$ is in front of the agent for the first $T_{\text{front}}$ timesteps. That is, given the agent’s heading angle $\theta_i^n$, $\theta_i^{\text{agent}}$ the angle of the segment connecting the agent to person $i$ and $\Theta_{\text{max}}$ a threshold, $|\theta_i^n - \theta_i^{\text{agent}}| \leq \Theta_{\text{max}}$ holds $\forall t \in [t_1, t_1 + T_{\text{front}}]$.

**Encounter classification.** To distinguish the encounters between the agent and a human subject, we devise a heuristic called inclusion rule (IR). The IR is defined according to the following parameters:

- $\Delta_i, \Delta_t^a$ represent, respectively, an approximation of the general direction of the trajectory of the agent and a person $i$ in the timeframe $[t_1,t]$ with $t \in [t_1,t_2]$, where $t_1, t_2$ are the start and the end steps of an encounter;

- intersect is a binary value that is 1 if robot and pedestrian paths intersect and 0 otherwise;

- $\text{blind}(t)$ is a time-conditioned binary value indicating whether the agent can see the person at time step $t$;

- $\text{d_diff}(t)$: difference between the geodesic and the euclidean distance between the agent and the person.

Subsequently, we defined four categories (inspired by [26]), and their respective inclusion rule (see Figure 2):

- **Frontal approach**
- **Intersection**
- **Blind corner**
- **Person following**

Figure 2. A scheme representing the four different classes of encounter. The dashed line represents the general direction of the agent and the person involved. The red area represents the agent’s field of view at the beginning of the encounter.
**Frontal approach:** The robot and the human come from opposite directions and have trajectories that are roughly parallel. In this context, the agent must deviate slightly to avoid a frontal collision. IR: \((-\text{blind}(t) \forall t \in [t_1, t_1 + T_{\text{view}}]) \land \pi - \Delta_{\text{slack}} \leq |\Delta_{t_2} - \Delta_{t_2}| \leq \pi + \Delta_{\text{slack}},\) where \(\Delta_{\text{slack}}\) is a slack value (in radians) on the angle \(\pi\) and \(T_{\text{view}}\) is the number of initial timesteps in which the person must be visible by the agent.

**Intersection:** The robot and the human’s trajectory intersect at approximately 90°. In this situation, the agent may want to stop and yield to the human or decrease its linear velocity and slightly deviate. IR: \((-\text{blind}(t) \forall t \in [t_1, t_1 + T_{\text{blind}}]) \land d_{\text{diff}}(t_1) \leq 0.5\) where \(T_{\text{blind}}\) is the number of initial timesteps in which the person must not be visible by the agent.

**Blind Corner:** An agent approaches a person from an initially occluded position, like a corner or a narrow doorway. In situations with limited visibility like this, the agent should act cautiously to avoid sudden crashes. IR: \((-\text{blind}(t) \forall t \in [t_1, t_1 + T_{\text{blind}}]) \land |\Delta_{t_2} - \Delta_{t_2}| \leq \frac{\pi}{2} + \Delta_{\text{slack}}\land \text{intersect}.

**Person following:** A person and the agent travel in the same direction. The agent must maintain a safe distance from the person and a relatively low linear velocity. IR: \((-\text{blind}(t) \forall t \in [t_1, t_1 + T_{\text{blind}}]) \land |\Delta_{t_2} - \Delta_{t_2}| \leq \Delta_{\text{slack}}\).

**Metrics.** For each encounter category, we compute the following metrics:

- **Encounter Survival Rate (ESR)** is the percentage of encounters (in a specific category) without a human collision (e.g., in the Blind Corner the agent collided with a human in the 20% of the cases, the ESR will be 80%);
- **Average Linear-Velocity (ALV)** is the average linear velocity of the agent in an encounter;
- **Average Distance (AD)** is the average distance of the agent w.r.t. the human in an encounter.

### 4. Method

**Overview.** Figure 3 shows an outline of our framework. It comprises two main modules: (i) **Social feature extraction,** and (ii) **Policy architecture.** The Social feature extraction module refines social information obtained from the simulator to extract features that describe some aspect of social interactions (ground truth social features). The Policy architecture extracts from the RGB-D and the GPS+Compass sensors an embedding that serves as input for our Socially-Aware tasks. These tasks refine this embedding and create \(n\) embeddings (one per task). These embeddings are then fused together through state attention. From the state attention output is then sampled an action. In the following, we will detail the whole architecture.

#### 4.1. Policy Architecture

Our policy network comprises the following modules:

- i) two encoders (the Visual backbone and the Position Encoder) that create an embedding from the RGB-D and the GPS+Compass sensors;
- ii) a Recurrent State Encoder that accumulates such embedding through a series of recurrent units;
- iii) a State Attention module that fuses the outputs of such units through an attention mechanism to produce the action the robot has to perform.

Each RGB-D frame \(x_t\) is encoded in a \(\phi_t^g\) embedding using a CNN (Visual Backbone) \(f(.)\) such that \(\phi_t^g = f(x_t)\). To encode the position and rotation of the agent \(\alpha_t\) in a \(\phi_t^p\) embedding, we used a linear layer \(g(.)\) such that \(\phi_t^p = g(\alpha_t)\). Subsequently, the outputs of these two encoders are concatenated into the final embedding \(\phi_t^f = \phi_t^g \oplus \phi_t^p\). To accumulate embeddings over time, we decided, similarly to what has been done in [43] for PointGoal Navigation, to implement our state encoder as a stack of parallel recurrent units. Each unit at each timestep is fed \(\phi_t^f\), and it outputs its internal state, called belief.

The key idea of having multiple beliefs is that each recurrent unit focuses on a specific navigation aspect. The final decision about what action the robot should take is sampled by weighting each belief according to the situation. For this reason, all beliefs \(B\) are subsequently fused through the State Attention module to calculate the mean \(\mu_t\) and standard deviation \(\sigma_t\) of the normal distribution from which we sample the action \(a_t\). More formally, given \(\{RU^{(i)}\}_{i \in B}\) a set of recurrent units, the encoded beliefs \(h_t\) are defined as follows:

\[
\begin{align*}
  h_t := \{h_t^{(i)}\}_{i \in B} &\leftarrow \{RU^{(i)}(h_{t-1}^{(i)}, \phi_t^{f^i})\}_{i \in B} \\
  \text{Attention}(Q, K, V) &\rightarrow \text{Softmax}^2(V) \text{ and } FC_a^2 \\
  \mu_t, \sigma_t &\leftarrow \text{Attention}(Q, K, V) \rightarrow \text{Softmax}^2(V) \text{ and } FC_a^2 \
\end{align*}
\]

The fusion mechanism of the state attention module \(SA\) is defined as:

\[
\begin{align*}
  \mu_t, \sigma_t &\leftarrow \text{Attention}(h_t, FC_C(\phi_t^f), h_t) \\
\end{align*}
\]

where \(\text{Attention}(Q, K, V) \rightarrow \text{Softmax}(QK^T)\) and \(FC_C\) are two linear layers.

#### 4.2. Socially-Aware Tasks

With multiple beliefs, we can inject different signals in our embeddings, e.g., social dynamics occurring in an episode. To this end, during training, we condition each belief with a unique auxiliary loss jointly optimized with the
action and value ones during the optimization step of the policy network. This is done by processing each belief with a specific type of Social feature, through a Regressor network (see Fig. 4), that computes our Socially-Aware tasks predictions. Such tasks consist in the prediction of social features in $[t, t+k]$, conditioned by the corresponding belief $h_t^{(i)}$ and the sequence of performed actions $\{a_j\}_{j \in [t, t+k]}$, where $k$ is the number of future frames to predict. Formally, for a given sequence of social features $\{s_j\}_{j \in [t, t+k]}$, the task aims to optimize the following auxiliary loss:

$$L_f = \sum_{j \in [t, t+k]} \frac{MSE(s_j, s_j^*)}{k}$$

where $\{s_j\}_{j \in [t, t+k]} = M(h_t^{(i)}, \{a_j\}_{j \in [t, t+k]})$ and $M$ is the regressor network. We designed two types of social tasks corresponding to two social features: (i) Risk Estimation, and (ii) Social Compass. Such design has the benefit of being easily extensible with other, possibly more complex social tasks and to be also compatible with general purpose self-supervised tasks like the ones used in [43] (e.g., CPC/A [17] or ID [24, 42]).

To exploit different social features, we extract from the simulator the relative position of every person w.r.t. the agent. We refer to this data as Social Information:

$$SI_t \overset{\text{def}}{=} \{\delta_i := (\text{pos}(p^i_t) - \text{pos}(\alpha_t)) \in \mathbb{R}^2 \}_{\forall i \in P}$$

where the function $\text{pos}()$ extracts the position from an element of $\alpha$ or $p^i$.

**Risk Estimation.** Risk Estimation is a Socially-Aware Task designed to deal with short-range social interactions, to inform the agent about imminent collision dangers. Given $SI_t$, we define the Risk value as a scalar representing how close the agent and the nearest person are up to a maximum distance $D_r$. This value ranges from 0 (the nearest neighbor is further than $D_r$ meters away) to 1 (the agent and person are colliding). Formally:

$$\text{risk}_t = \text{clamp}\left(1 - \frac{\min\{|\delta_i^r| | \delta_i^r \in SI_t\}}{D_r}, 0, 1\right)$$

where $\text{clamp}(-, 0, 1)$ limits the value to the $[0, 1]$ range.

**Social Compass.** Complementary to Risk Estimation, this Socially-Aware Task deals with the long-distance component of social dynamics. This feature captures not only social interaction on a larger area with radius $D_c > D_r$ but also a weak indication of the direction a person may come. Much like humans can make guesses about people’s whereabouts based on previous observations, partial knowledge of the environment topology, and a person’s trajectory; we expect to provide similar knowledge at training time while being easy to infer at evaluation time.
Such information is represented through a **Social Compass**. In the compass, north represents the direction the agent is looking, and the quadrant is partitioned into a finite number of non-overlapping sectors. Given each person \( i \in \mathcal{P} \), \( \theta_{a \to i} \) represents the angle of the segment connecting the agent to that person w.r.t. the north of the compass. These angles are associated with a specific sector. We compute the risk value for each sector among people in the pass. These angles are associated with a specific sector. We equivalent sectors, the vector \( \Theta \) is defined as:

\[
\Theta_j = \left\{ \delta_i^t \in SI_t \mid \theta_{a \to i} \in \left[\frac{2\pi}{k}, j \cdot \frac{2\pi}{k}, (j + 1)\right), \forall j \rightarrow [0, k - 1] \right\}
\]

where \( \Delta_d \) is the potential reward based on the geodesic distance to the goal, \( r_{\text{slack}} \) is the slack reward, and \( I_{\text{coll}}, I_{\text{back}}, I_{\text{suc}} \) represent the indicator functions respectively of a collision with objects in the environment, the linear velocity being less than 0 and success. \( \beta_{\text{coll}} \) and \( \beta_{\text{suc}} \) are coefficients. We used the same parameters as in [44], that are -0.002 for \( r_{\text{slack}} \), 0.02 for \( \beta_{\text{coll}} \) and 10.0 for \( \beta_{\text{suc}} \).

### 5. Experiments

**Datasets and training procedure.** We performed our experiments using Gibson4+ and HM3D-S, a new dataset based on HM3D and adapted for social navigation. Gibson4+ contains 86 high-rated scenes taken from the original Gibson dataset [41]. For training, we used 64 scenes, while 8 and 14 environments were used for validation and test, respectively. HM3D-S is a dataset for Embodied Social Navigation that we generated on top of HM3D [28]. It consists of 900 scenes (800 used for training, 30 for validation, and 70 for test) with, on average, a larger walkable surface compared to Gibson4+. We have generated 8M episodes for the training set (10k per scene), 450 for the validation set (15 episodes per environment), and 490 for the test set (7 episodes per environment). Each episode is obtained by selecting a starting point and a goal point for the agent from the navigable area of the environment (such that it exists a navigable path from one to the other). Pedestrians are included as in [44]; namely, each person is positioned on a starting point and navigates back and forth to an endpoint with a random linear velocity between 0.45 and 0.5m/s. On Gibson4+, we trained each model for \( \approx 100 \)M steps of experience (2.5 days training). On HM3D-S, we fine-tuned our models for \( \approx 40 \)M steps (1-day training) starting from the final checkpoint obtained on Gibson4+. This was done to reduce the computational cost of each training.

### 5.1. Results

**Evaluation Metrics.** We used standard evaluation metrics for Point Goal Navigation such as **Success Rate** and **Success weighted by Path Length** (SPL) to evaluate the efficacy of the policy [1]. To evaluate its safety properties, we used **Human Collision Rate**, which is the percentage of episodes that end with failure by hitting a person. We run all our experiments with five runs to assess the mean and standard deviation for every metric, as done in [44].

**Baseline models.** We compared our approach to two baseline models: the model used by [44] (called Baseline) and a version of our model that only uses a set of 3 self-supervised auxiliary tasks: 2 CPC A tasks (respectively using 2 and 4 steps) and GID (4 steps) (called Aux tasks), starting from the one provided by [43]. Baseline only uses the depth channel as input. Since we believe that RGB input

---

2 Dataset, splits, code, and features will be publicly released.
| Name                          | Sensors | Aux Tasks | Social Tasks | Metrics (Gibson4+) | Metrics (HM3D-S) |
|-------------------------------|---------|-----------|--------------|-------------------|-----------------|
| Baseline [44]                | RGB     | ✓         |              |                   |                 |
| Baseline+RGB [44]            | ✓       | ✓         |              |                   |                 |
| Aux tasks [43]               | ✓       |           |              |                   |                 |
| Risk only                    | ✓       | ✓         |              |                   |                 |
| Compass only                 | ✓       |           |              |                   |                 |
| Aux + risk                   | ✓       | ✓         |              |                   |                 |
| Aux + compass                | ✓       | ✓         |              |                   |                 |
| Social tasks                 | ✓       |           |              |                   |                 |
| Social + Aux tasks           | ✓       | ✓         |              |                   |                 |

Table 1. Social Navigation evaluation on Gibson4+ and HM3D-S. For each model are listed the type of input data it uses (Sensors column) and, eventually, what kind of self-supervised Aux tasks or Social tasks the model employs. The metrics reported are Success rate, SPL and Human-Collisions Rate (H-collisions).

is fundamental for people recognition and trajectory prediction, we also experimented with an adapted version of the model that uses all the information from RGB-D frames (called Baseline+RGB).

Socially-aware models and auxiliary tasks. Firstly, to evaluate each social task contribution, we experimented with single-GRU models equipped with just one Socially-Aware task at a time (i.e., Risk-only or Compass-only). Since the tasks deal with two different aspects of social navigation (short-range and long-range), we then tried to combine them in a two-GRU model (referred to as Social tasks). Finally, we combined our approaches with the self-supervised auxiliary tasks presented in [43], which reported state-of-the-art performance on PointGoal Navigation. We have thoroughly investigated the benefit of combining them with single social tasks (Aux+Risk and Aux+compass), as well as combining them all in our final model (Social+Aux tasks). We now discuss our results and highlight the main takeaways.

Performance analysis and comparison to prior work. Table 1 reports the social navigation performance (on the test set) for both Gibson4+ and HM3D-S. In both cases, Aux tasks appears as the strongest of our baselines (highest SPL and lowest Human-Collision for both datasets), reaching comparable performances to single social task models while having a higher SPL. Our initial hypothesis that integrating the Baseline with an RGB signal could benefit performances was partially supported by the results on Gibson4+. However, the trend shifted on HM3D-S. This happens because of the higher quality of scene reconstruction in Gibson4+ (scenes have been manually rated and are among the best in the original Gibson dataset). Comparatively, HM3D-S has more reconstruction errors that, while leaving depth-only policies unaffected, may impair the performance of RGB-enabled models.

Moreover, we notice that both models that use just one Socially-aware task perform similarly on Gibson4+ (sub 0.5% of difference between metrics). However, this changes on HM3D-S, where Compass-only slightly outperforms Risk-only (+1.1% Success, -0.93% h-collisions).

This difference is expected since the social compass task explicitly aims to deal with long-range social information. Being HM3D scenes larger in size, the Social Compass role becomes more important.

Adding self-supervised tasks significantly increases SPL and Success performances (both for single-task and all Social-tasks models). It also appears to positively affect Human Collision when combined with Risk (~1.52% in Gibson4+, ~2.47% in HM3D-S). We hypothesize that self-supervised tasks, since they are either action-based contrastive tasks (CPC/A) or tasks that try to retrieve the inverse dynamics of navigation (GID), help socially-aware models to have smoother trajectories thanks to a more accurate linear and angular velocity dialing. This claim will be substantiated by the fine-grained analysis reported in the next section. Overall, the best results are obtained by combining all tasks together in the same model.

5.2. Fine-grained evaluation

We report the results obtained on the Gibson4+ dataset by applying our evaluation protocol (defined in Section 3)
to understand better how each model operates and their attitude towards social interactions. Figure 5 summarizes the statistics, in terms of number of encounters and ESR, collected for each encounter class by different models during 500 randomly sampled validation episodes.

**Policy behavior analysis.** Looking at the relationships between the number of encounters and ESR figures, there seem to be two types of policies: a first group with a high number of encounters and a high ESR, and a second group that tends to avoid encounters and has low/medium ESR. Those two types describe different approaches to social navigation: either risking to interact to access potentially more efficient routes to the goal or keeping a safe distance from humans (for example, prioritizing less populated areas, waiting for people to move away before crossing a room).

An example of a policy that avoids encounters is Risk only, which has one of the lowest ESR for Following (74.11%) and Intersection class (86.85%) and the lowest number of encounters (188 in total). The opposite is true for Compass only, which has a high overall ESR for every encounter class and one of the highest numbers of encounters (279 in total). It is interesting to note how the two best-performing policies, Social tasks and Social+Aux, adopt each of these different approaches while remaining comparable in coarse-grained metrics.

**Reacting to sudden danger.** A critical ability that a social policy must possess is the capability to react to immediate and sudden danger in situations with limited visibility. The class of encounters that better represents this is the **blind corner** class. To investigate how our two best models react compared to the baseline, in Figure 6, we plotted the AD and ALV values at a percentage of completion of all blind corner encounters. We can notice how the ALV curves are not smooth, reflecting uncertainty and high risk. However, while the baseline needs to brake and backtrack ($\approx -0.5$ ALV between 20% and 40% of the episodes), the other models tend to maintain a positive and proportionate ALV velocity throughout the episode. We can also notice that the Social+Aux ALV curve is smoother than the one of Social tasks. The same phenomenon is true for each single socially-aware task model and their self-supervised task counterpart (see supplementary material). This supports the claim that self-supervised tasks provide a smoothing effect on action dynamics under uncertainty.

**Qualitative results.** Figure 7 shows two qualitative examples of successfully managed encounters in Gibson4++. The first example (on top) depicts a Frontal encounter. After seeing people (first frame), the agent moves to the side and yields, letting them move away (second frame). Finally, the agent can reach the goal (third frame). In the second example (an intersection encounter), The agent sees a pedestrian (first frame), then it yields, letting the pedestrian pass (second frame), and finally, it continues on its path.

### 6. Conclusion

We introduced a model for Embodied Social Navigation based on two **Socially-Aware tasks**. Our experiments show that exploiting social signals, alone or in combination with self-supervised auxiliary tasks, is an effective strategy in complex and crowded scenarios. Our model can avoid the majority of encounters by using only Socially-Aware tasks. Furthermore, by combining Socially-Aware and auxiliary tasks [43], it can prevent human collisions in almost all the cases, despite a higher number of encounters. However, a major limitation of our setup, and more broadly of the Embodied Social Navigation task, resides in the simple movement of pedestrians. In future works we would like to focus on simulating more natural human behaviors and to experiment on sim2real domain transfer.
References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. arXiv preprint arXiv:1807.06757, 2018. 1, 2, 6

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2

[3] Georges S. Aoude, Brandon D. Luders, Joshua M. Joseph, Nicholas Roy, and Jonathan P. How. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. Autonomous Robots, 35(1):51–76, 2013. 2

[4] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav revisited: On evaluation of embodied agents navigating to objects. arXiv preprint arXiv:2006.13171, 2020. 2

[5] Jur van den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In Robotics Research, pages 3–19. Springer, 2011. 2

[6] Tommaso Campari, Paolo Eccher, Luciano Serafini, and Lamberto Ballan. Exploiting scene-specific features for object goal navigation. In Proc. of the European Conference on Computer Vision Workshops (ECCVW), 2020. 1, 2

[7] Tommaso Campari, Leonardo Lamanna, Paolo Traverso, Luciano Serafini, and Lamberto Ballan. Online learning of reusable abstract models for object goal navigation. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 2

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In Proc. of the International Conference on 3D Vision (3DV), 2017. 1, 2

[9] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In Proc. of Advances in Neural Information Processing Systems (NeurIPS), 2020. 1, 2

[10] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2

[11] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2019. 2

[12] Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P. How. Socially aware motion planning with deep reinforcement learning. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017. 2

[13] Yu Fan Chen, Miao Liu, Michael Everett, and Jonathan P. How. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2017. 2

[14] Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angela T. Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez D’Arpino, Kiana Ehsani, Ali Farhadi, et al. Retrospectives on the embodied ai workshop. arXiv preprint arXiv:2210.06849, 2022. 1, 2

[15] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihls, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, et al. Proctor: Large-scale Embodied AI using procedural generation. arXiv preprint arXiv:2206.06994, 2022. 1

[16] Gonzalo Ferrer, Anais Garrell, and Alberto Sanfeliu. Social-aware robot navigation in urban environments. In Proc. of the European Conference on Mobile Robots, 2013. 2

[17] Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo A. Pires, and Rémi Munos. Neural predictive belief representations. arXiv preprint arXiv:1811.06407, 2018. 5

[18] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihls, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474, 2017. 1, 2

[19] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In Proc. of the European Conference on Computer Vision (ECCV), 2020. 2

[20] Pinxin Long, Wenxi Liu, and Jia Pan. Deep-learned collision avoidance policy for distributed multiagent navigation. IEEE Robotics and Automation Letters, 2(2):656–663, 2017. 2

[21] Yazhou Lu, Xiaogang Ruan, and Jing Huang. Deep reinforcement learning based on social spatial–temporal graph convolution network for crowd navigation. Machines, 10(8):703, 2022. 2

[22] Ronja Möller, Antonino Furnari, Sebastiano Battiato, Aki Häräni, and Giovanni Maria Farinella. A survey on human-aware robot navigation. Robotics and Autonomous Systems, 145:103837, 2021. 2

[23] Ruslan Partsey, Erik Wijmans, Naoki Yokoyama, Oles Dobosevykh, Dhruv Batra, and Oleksandr Maksymets. Is mapping necessary for realistic pointgoal navigation? In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 1, 2

[24] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Proc. of the International Conference on Machine Learning (ICML), 2017. 5

[25] Claudia Pérez-D’Arpino, Can Liu, Patrick Goebel, Roberto Martín-Martin, and Silvio Savarese. Robot navigation in constrained pedestrian environments using reinforcement learning. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2021. 1, 2, 3

[26] Sören Pirk, Edward Lee, Xuesu Xiao, Leila Takayama, Anthony Francis, and Alexander Toshev. A protocol
for validating social navigation policies. arXiv preprint arXiv:2204.05443, 2022. 2, 3

[27] Santhosh K. Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Potential functions for object goal navigation with interaction-free learning. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 1, 2

[28] Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Makysmets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, et al. Habitat: matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. arXiv preprint arXiv:2109.08238, 2021. 1, 2, 6

[29] Ram Ramakhraya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 1

[30] Zhiwei Deng Karthik Narasimhan Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. In Proc. of Advances in Neural Information Processing Systems (NeurIPS), 2020. 1

[31] Manolis Savva, Abhishek Kadian, Oleksandr Makysmets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied AI research. In Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 1, 2

[32] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D’Arpino, Shyamal Bucb, Sanjana Srivastava, Lyne Tchapmi, et al. iGibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021. 1, 2

[33] Lei Tai, Jingwei Zhang, Ming Liu, and Wolfram Burgard. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2018. 2

[34] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010. 2

[35] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2008. 2

[36] Xin Wang, Qiyuan Huang, Aslı Celikyilmaz, Jianfeng Gao, Dihghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 1

[37] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. MultiON: Benchmarking semantic map memory using multi-object navigation. In Proc. of Advances in Neural Information Processing Systems (NeurIPS), 2020. 2

[38] Erik Wijmans, Samyak Datta, Oleksandr Makysmets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied Question Answering in Photorealistic Environments with Point Cloud Perception. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[39] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In Proc. of the International Conference on Learning Representations (ICLR), 2019. 1, 2, 6

[40] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micheal Edmond Tchampi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive Gibson benchmark: A benchmark for interactive navigation in cluttered environments. IEEE Robotics and Automation Letters, 5(2):713–720, 2020. 1, 2, 3

[41] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-World Perception for Embodied Agents. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 6

[42] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary Tasks and Exploration Enable ObjectNav. In Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 1, 2, 5

[43] Joel Ye, Dhruv Batra, Erik Wijmans, and Abhishek Das. Auxiliary Tasks Speed Up Learning PointGoal Navigation. In Proc. of the International Conference on Robot Learning (CoRL), 2020. 1, 2, 4, 5, 6, 7, 8

[44] Naoki Yokoyama, Qian Luo, Dhruv Batra, and Sehoon Ha. Learning Robust Agents for Visual Navigation in Dynamic Environments: The Winning Entry of iGibson Challenge 2021. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022. 1, 2, 3, 6, 7

[45] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 1

[46] Fengda Zhu, Yi Zhu, Vincent Lee, Xiaodan Liang, and Xiaojun Chang. Deep learning for embodied vision navigation: A survey. arXiv preprint arXiv:2108.04097, 2021. 1