Exploiting Proximity-Aware Tasks for Embodied Social Navigation

Enrico Cancelli[1*] Tommaso Campari[2*] Luciano Serafini[2] Angel X. Chang[3,4,5] Lamberto Ballan[1]

[1] University of Padova [2] Fondazione Bruno Kessler (FBK) [3] Simon Fraser University
[4] Canada-CIFAR AI Chair [5] Amii

Abstract

Learning how to navigate among humans in an occluded and spatially constrained indoor environment, is a key ability required to embodied agent to be integrated into our society. In this paper, we propose an end-to-end architecture that exploits Proximity-Aware Tasks (referred as to Risk and Proximity 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. 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.

In Embodied AI, common tasks such as PointGoal [4, 9, 12] or ObjectGoal [1, 2, 7] navigation, 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 (doing so without harming others). This social element is included in the Social Navigation Task (SocialNav) [6, 10], where an agent must tackle PointGoal Navigation in simulated indoor environments. To tackle this task, Yokoyama et al. [13] introduced a simple but quite effective model. However, the approach does not explicitly encode any social behavior in its navigation policy. We believe that a clear encoding of human-agent interactions, as well as social behaviors, are required for safe navigation and interaction with humans. By modeling the movement of humans, 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 *Proximity-Aware Tasks*, referred as *risk* and *proximity compass*. These auxiliary tasks model the present and future danger for the agent's action. Finally, we also introduce a dataset of episodes on top of HM3D [8] for Embodied Social Navigation to assess our agents in different environments.

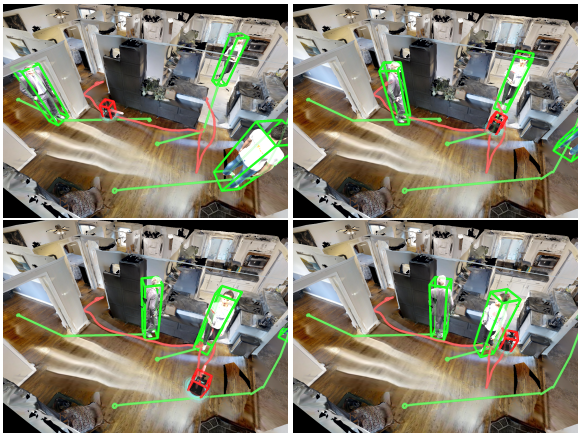

Figure 1. Example episode. From top-left to bottom-right: *i*) episode starts; *ii*) the agent sees a person; *iii*) it moves back to avoid a collision; *iv*) it reaches the goal by avoiding the person.

2. Method and Experiments

Overview. Our framework comprises two modules: (i) *Proximity feature extraction*, and (ii) *Policy architecture*. The *Proximity feature extraction* module refines proximity information to extract features of social interactions (ground truth proximity features). The *Policy architecture* extracts from the RGB-D and GPS+Compass sensors an embedding that is given as input to our Proximity-Aware tasks. These tasks refine this embedding to n task embeddings (one per task) which are then fused through state attention. An action is sampled from the output.

Policy Architecture Our network comprises the following modules: *i)* two encoders that create an embedding from input sensors; *ii)* a *Recurrent State Encoder* that accumulates

*Both authors contributed equally. The full paper is on arXiv.

Name	Sensors		Aux Tasks			Proximity Tasks		Metrics (Gibson4+)			Metrics (HM3D-S)		
	RGB	Depth	CPCA	GID	CPCA/B	Risk	Compass	Success	SPL	H-Collisions	Success	SPL	H-Collisions
Baseline [13]		✓						72.65±1.6	47.43±1.2	24.35±1.9	62.76±2.2	36.69±1.1	29.29±2.2
Baseline + RGB [13]	✓	✓						74.28±1.8	44.84±0.7	23.78±1.3	61.43±0.5	34.84±0.6	29.23 ± 0.7
Aux tasks [12]	✓	✓	✓	✓	✓			73.4±2.0	52.08±1.4	23.40±1.5	63.62±1.6	42.27±1.2	24.79±2.2
Risk only	✓	✓				✓		74.90±1.7	50.25±1.1	22.56±1.2	66.22±1.2	45.26±0.8	24.47±1.7
Compass only	✓	✓					✓	75.08±1.5	50.55±1.0	22.49±1.1	67.32±1.7	45.74±1.0	23.54±1.7
Aux + risk	✓	✓	✓	✓	✓	✓		75.61±1.8	51.43±0.2	21.04±1.4	68.16±0.8	45.64±0.2	22.00±1.6
Aux + compass	✓	✓	✓	✓	✓		✓	75.63±1.2	52.60±1.6	23.17±1.2	67.94±1.4	45.76±1.0	23.78±2.0
Proximity tasks	✓	✓				✓	✓	76.6±1.8	52.81±1.2	20.47±0.4	68.35±0.5	45.83±0.5	**21.72±1.2**
Proximity + Aux tasks	✓	✓	✓	✓	✓	✓	✓	**77.24±1.1**	**55.23±1.4**	**19.50±1.0**	**70.16±1.1**	**47.60±1.0**	22.09±1.3

Table 1. Social Navigation evaluation. For each model are listed used *Sensors* and type of self-supervised *Aux tasks* or *Proximity tasks*.

such embedding through a series of recurrent units; *iii)* a *State Attention* module that fuses the outputs.

We encode each RGB-D frame x_t using a CNN $f(\cdot)$ to a visual embedding $\phi_t^v = f(x_t)$. The position and rotation of the agent α_t are encoded using a linear layer $g(\cdot)$ to obtain the embedding $\phi_t^p = g(\alpha_t)$. The final embedding is $\phi_t^f = \phi_t^v \oplus \phi_t^p$. To accumulate embeddings over time, we follow [12]'s design for PointNav and implement our state encoder as a stack of parallel recurrent units. Each unit at each timestep is fed ϕ_t^f, and outputs its internal state, called *belief*.

The key idea is that each recurrent unit can focus on a specific navigation aspect and each belief is weighted according to the situation. The *State Attention* module computes the mean $\vec{\mu_t}$ and standard deviation $\vec{\sigma_t}$ of the normal distribution from which we sample the action a_t. Formally, given $\{RU^{(i)}\}_{\forall i \in \mathcal{B}}$ a set of recurrent units, the encoded beliefs h_t are defined as:

$$h_t := \{h_t^{(i)}\}_{\forall i \in \mathcal{B}} \leftarrow \{RU^{(i)}(h_{t-1}^{(i)}; \phi_t^f)\}_{\forall i \in \mathcal{B}}$$

The fusion mechanism of the state attention module is:

$$\vec{\mu_t}, \vec{\sigma_t} \leftarrow FC_a(\text{Attention}(h_t, FC_k(\phi_t^f), h_t))$$

with attention function $\text{Attention}(Q, K, V)$ and FC_a and FC_k are linear layers.

Proximity-Aware Tasks With multiple beliefs, we can inject different signals in our embeddings like information related to social dynamics. To this end, we condition each belief with a unique auxiliary loss. Each belief is processed with a specific type of *Proximity feature*, through a *Regressor network*, that computes our *Proximity-Aware tasks* predictions. Each regressor predicts the proximity features in the time range $[t, t+k]$, conditioned by the corresponding belief $h_t^{(i)}$ and the sequence of actions $\{a_j\}_{j \in [t, t+k]}$. Each task minimizes the MSE loss between such predictions and ground truth proximity features. The proximity features are only used in training and detached during evaluation.

We designed two proximity tasks and two social features: (i) *Risk Estimation*, and (ii) *Proximity Compass*. Also general purpose self-supervised tasks like the ones used in [12] (e.g., CPC|A [3] or ID [5, 11]) can be combined.

Risk Estimation Task. *Risk Estimation* deals with short-range social interactions, to inform the agent about imminent collision dangers. 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).

Proximity Compass Task. *Proximity Compass* models the long-distance component of social dynamics. This feature captures social interaction on a larger area with radius $D_c > D_r$ and also a weak indication of the direction a person may come. Such information is represented through a *Proximity Compass*. In the compass, north represents the direction the agent is looking, and the quadrant is partitioned into 8 sectors. We compute the risk value among people belonging to the same sector, for each sector. **Baseline models.** We compared our approach to two baselines: the model presented in [13] (referred as to *Baseline*) and a "simplified" version of our model that only uses 3 self-supervised auxiliary tasks (inspired by [12]): 2 CPC|A tasks (respectively using 2 and 4 steps) and GID (4 steps), referred to as *Aux tasks*.

Performance analysis and comparison to prior work. Table 1 reports the social navigation performance (on the test set) on different auxiliary task combinations. In both cases, *Aux tasks* appears as the strongest baseline (highest SPL and lowest Human-Collision), reaching comparable performances to single proximity task models but with a higher SPL.

Moreover, we notice that both models that use just one Proximity-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 proximity compass task deals with long-range proximity information and HM3D scenes are larger in size.

Adding self-supervised tasks significantly increases SPL and Success performances. It also appears to positively affect Human Collision when combined with Risk (-1.52% in Gibson4+, -2.47% in HM3D-S). Overall, the best results are obtained by combining all tasks.

Acknowledgements. TC and LS were supported by the PNRR project Future AI Research (FAIR - PE00000013), under the NRRP MUR program funded by the NextGenerationEU. EC and LB were supported by an UniPD BIRD-2021 Project. AXC was supported by funding from Canada CIFAR AI Chair and NSERC Discovery grant.

References

[1] 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] 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

[3] Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo A. Pires, and Rémi Munos. Neural predictive belief representations. *arXiv preprint arXiv:1811.06407*, 2018. 2

[4] Ruslan Partsey, Erik Wijmans, Naoki Yokoyama, Oles Dobosevych, 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

[5] 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. 2

[6] Claudia Pérez-D'Arpino, Can Liu, Patrick Goebel, Roberto Martín-Martín, 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

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

[8] Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, 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

[9] 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

[10] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, 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

[11] 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. 2

[12] 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

[13] 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