Robots Waiting for the Elevator: Integrating Social Norms in a Low-Data Regime Goal Selection Problem

Mattia Racca, Jutta Willamowski, Tommaso Colombino, Gianluca Monaci, and Danilo Gallo

***Abstract*— As robots increasingly share spaces with people, it becomes important for them to behave according to our social norms. In this paper, we explore the problem of finding socially acceptable locations for a robot to wait for a shared elevator by learning from expert annotations. Access to relevant, unlabeled data is however scarce in this setting and annotations expensive to gather, as they require explicit knowledge about the social norms, the robot, and the service it carries out. We tackle this low-data regime as follows. First, we use Procedural Content Generation to generate plausible waiting scenes to be annotated. Second, we leverage available sociological studies and operationalize relevant social norms as feature maps. We train a variety of models with only 125 procedurally-generated expert-annotated scenes, testing the impact of the proposed feature maps. In our ablation study, the feature maps help the models' performance and their generalization capabilities to non-synthetic, real scenes. We inspect the decisions taken by the best models, probing their strengths and weaknesses, and identifying general issues and discussing potential solutions.**

I. Introduction

Robots are poised to become increasingly present in our day-to-day life, co-inhabiting with humans while providing services. Regardless of their level of autonomy or the nature of their embodiment, robots are perceived as social agents [1] and expected to comply with social norms [2]. Interacting with people can be a secondary task for such robots; yet their behaviors greatly impact people's perception of and attitude towards them [3].

Our work builds on the ongoing development of an indoor service robot capable of sharing the use of elevators with bystanders. Elevators are everywhere, and we want robots to access the vertical mobility they provide while minimizing dedicated infrastructure and discomfort of human passengers. With ethnographic studies aimed at understanding how people routinely wait for and board elevators, we investigated the requirements for socially acceptable navigation behaviors in office buildings [4] and designed non-verbal behaviors aimed at reducing disruptive behaviors over repeated encounters for a robot with limited social features and expressiveness [5]. In a Wizard-of-Oz experiment, we studied the use of waiting positions that adapt to the number of people and the available space by mimicking social conventions like *e.g.*, proxemics [6] and queuing principles [7], as opposed to a fixed position next to the elevator's door. Although we found that a fixed position has advantages in terms of predictability, consistency of behavior, and ease of implementation, a fixed position may not always be available in practice or may not satisfy other technical requirements of the service carried out by the robot. In this work, we therefore automate the selection of socially acceptable waiting positions for a robot in a shared elevator scenario.

Enumerating the social norms and technical requirements we want our robot to follow, we realized (a) how the intersection or average of them would often produce no acceptable waiting areas and (b) how people using elevators will non-trivially pay less attention or fully ignore certain norms when the situation at hand requires it. We therefore deemed necessary to train a model that, given a representation of the robot's surroundings, would classify positions as socially acceptable or not. As for other practical Human-Robot Interaction (HRI) problems, annotated data is scarce, and labeled samples are costly to gather because of the level of expertise required from the annotator. We attenuate the data scarcity of this setup in two ways. First, we rely on a Procedural Content Generation (PCG) technique to create plausible scenes with people waiting for the elevator. These scenes are then labeled by experts with socially acceptable robot waiting positions for two target missions[1], based on the robot's level of urgency: *Priority*, when the robot seeks to enter the elevator as fast as possible (even before bystanders), and *Yielding*, when the robot willingly forfeits its priority. Second, we operationalize the prior knowledge from the available literature in a set of feature maps applicable to any configuration of the space shared by one or multiple bystanders, the robot, and the elevator. As an example, Fig. 1j shows how one of our maps, the *Avoid transactional space* map, disincentivizes the robot from waiting in the transactional space created by the human passengers and their focus of attention, *i.e.* the elevator.

By training a variety of models, we aim at answering the following research questions:

- RQ1: is learning from data necessary? and if that is the case, can a satisfactory classifier (*i.e.* achieving a Jaccard index of .4) be trained in this low-data regime, *e.g.*, with less than 150 annotated scenes?,
- RQ2: do the proposed feature maps, encoding the prior sociological knowledge, help improve the classifier's performance?, and
- RQ3: can the trained models generalize to unseen real scenes, considering that the totality of the training data is procedurally generated?

Our evaluation confirms the impact of the proposed feature

*All authors are with NAVER LABS Europe, 38240 Meylan, France. name.surname@naverlabs.com

[1]Annotated dataset available at https://europe.naverlabs.com/research/publications/robots-waiting-for-the-elevator/

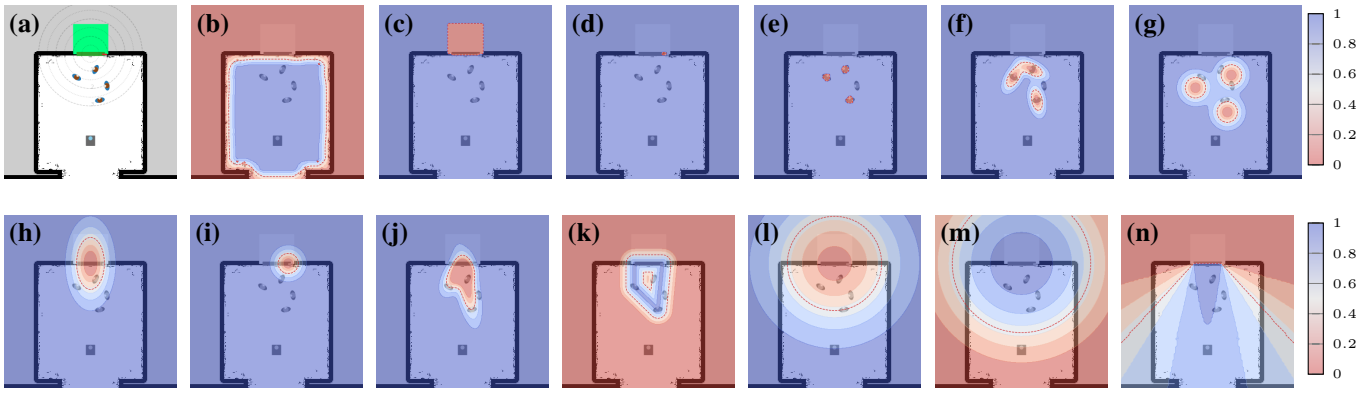

Fig. 1: Top-view of a procedurally-generated elevator waiting scene with 2 groups of 2 people each and a robot (a), along with the proposed feature maps applied on the area surrounding the robot: (b) is the occupancy map, (c-e) are Basic feature maps (Elevator location, Commands location, People detection), while (f-n) are Norms feature maps (Proxemics, Avoid standing behind people, Avoid blocking the door, Avoid blocking the commands, Avoid transactional space, Communicative space, As far as furthest person, Efficient boarding, Visibility cone).

maps on the models' performance and their generalization capabilities. The prior knowledge embedded in the maps overcomes the data scarcity for the Yielding mission, while, for the more complex Priority mission, the models show promising but not-deployment ready results. Going beyond the performance metrics, we have an expert annotator inspect models' decisions to further discuss strengths and weaknesses of the proposed pipeline.

II. Related work

Social robot navigation: Enabling robots to navigate among humans is a long-standing research problem [8]. While early approaches solely focused on avoiding collisions with bystanders [9], subsequent methods encoded the presence of nearby humans in their decisions, either by forecasting human trajectories while planning [10], [11], [12] or by modeling crowd interaction with attractive and repulsive force fields [13].

With the goal of injecting prior knowledge about social norms, Kothari *et al.* [14] proposed a hybrid approach to trajectory forecasting using hand-coded social behaviors, such as collision avoidance and leader following, in a Discrete Choice Model. This approach is reminiscent of the pioneering work of Kirby [15], where paths are computed on a cost map that combines different task- and interaction-specific factors.

In this work, we address the related but distinct problem of identifying socially acceptable waiting positions for robots. Decoupling goal selection from navigation allows to (a) use the selected waiting positions with any social navigation pipeline, and to (b) prevent waiting-specific requirements (*e.g.*, the need to see the inside of the elevator's car) from over-constraining navigation towards the chosen spot.

Social aspects of robot waiting: Research on the social norms of waiting for robots as opposed to navigating is quite limited [16], [17], [18], [19]. Tackling the problem of searching the best pose for a robot to join a conversation, the approach of [20] operationalises the concepts of proxemics [6] and F-formations [21] with two losses, and performs an optimization procedure on their weighted average. We here operationalize a larger number of social norms and technical requirements (as shown in Fig. 1) which often do not yield acceptable poses when the constraint to satisfy all of them or their average is imposed.

There are of course many possible approaches to getting a robot to successfully board an elevator, including using vocal interactions to ask people to move out of the way [22]. This may be acceptable in certain scenarios where interactions with the robots are episodic (*e.g.*, hotels), and less so in others (*e.g.*, office spaces). With this in mind, we focus on the waiting position as a non-disruptive social cue to convey the robot's intention to board the elevator. Since waiting positions have a strong communicative function [7], [23], selecting a good waiting position is therefore crucial.

III. Method

We formalize the problem of robots finding a socially acceptable waiting position in a shared elevator scenario as follows. Given (a) static information about the environment the robot operates in, *i.e.* an occupancy map M, and the location of the elevator E and its commands C_e, (b) the pose p_p of people $p \in \mathcal{P}_v$ detectable by robot-mounted sensors, (c) the robot's estimated pose p_r, and (d) the robot's target mission T, the robot classifies waiting positions in its proximity as acceptable or not for that mission. We consider the sensory information about the robot's surroundings $\mathcal{S} = \langle M, E, C_e, p_r, p_p \forall p \in \mathcal{P}_v \rangle$ as available to the robot at all times. However, we constrain $\mathcal{S}$ to what is generally needed to operate a robot in a human-inhabited environment. For example, we assume the robot to be able to reliably detect the people's pose p_p (position and main facing orientation) through robot-mounted vision systems instead of, for example, relying on connected ceiling-mounted cameras.

Following from [5], we consider two target missions, Priority and Yielding. As explained in Section III-B, different social norms are more or less relevant depending on the robot's urgency, and we expect the feature maps to ease the learning of both missions. While raw video data of people navigating socially is available on the web, few annotated datasets contain elevator scenes [24]. Consequently, an annotated dataset with socially acceptable robot waiting positions is not available. We therefore procedurally generate plausible elevator waiting scenes within a simple simulator, providing us the unlabeled data. We then design feature maps, combining the environmental information available to the robot with social norms observed in previous studies, injecting prior knowledge in the learning process. Given procedurally generated scenes, annotated by an expert with socially acceptable waiting locations according to the robot's target mission, we learn a classifier for these acceptable locations given, as input, a set of feature maps including (a) information about the environment mentioned above (Basic feature maps) and (b) feature maps about social norms and service requirements (Norms feature maps). The expert annotators in our case are researchers with sociological expertise, but this task could be performed as part of a service design pipeline by someone with knowledge of the desired robot behaviors for a specific deployment.

A. Procedural generation of waiting scenes

To overcome the lack of unlabeled data, we take a page from the field of PCG, in which algorithms are used to generate content based on a mix of human-designed rules and assets, and computer-generated randomness [25]. Historically developed within the video game industry, PCG has recently been used as a tool to generate and augment data for training ML models [26].

Inspired by work on PCG of crowds [27], [28], we develop a generator of elevator waiting crowds, based on the rejection sampling technique [29]. With rejection sampling, new entities (in our case, people waiting the elevator) are iteratively sampled from a provided *sampling function* S and evaluated against a set $\mathcal{R}$ of handcrafted *rejection functions*. If an entity e is rejected by any $R \in \mathcal{R}$, it is discarded and a new entity is sampled, up to a provided iteration limit. Otherwise, the accepted entity is added to the scene, often triggering the addition of new rejection functions to $\mathcal{R}$. Newly sampled entities will therefore need to respect these additional rejection functions as well.

Provided with a map M with an elevator E, our rejection sampling method generates a desired number of groups g of people by sampling from a *group* sampling function S_g and a *person in a group* sampling function S_p. While the scene is empty, the rejection functions only take care that people are not spawned inside occupied areas of the map. When groups are being populated, rejection functions are added, avoiding people from spawning (a) too close to each other, (b) and further than a hand-tuned distance from their group. Once a group is fully populated, more rejection functions are added, avoiding further groups from spawning (a) too close to already generated groups and (b) in the space between the elevator and previous groups. Fig. 2 shows a top-view comparison of generated scenes in two different maps and with different amounts of people, and two scenes extracted from video recordings for comparison.

It is worth mentioning that the PCG approach and the learning pipeline are disjoint: a generated scene is solely used to construct the robot's sensory information $\mathcal{S}$. This is a deliberate choice, as we want the pipeline to work outside of our PCG simulator. Furthermore, we consider visibility constraints and omit people hidden behind walls or behind other people when populating $\mathcal{P}_v$ from the set of all simulated people $\mathcal{P}$ (as shown in Fig. 1e, where one person is occluded and therefore not considered when computing the feature map).

B. From social norms to feature maps

In previous research, we examined video data to understand the specific practices of waiting for, entering, and exiting an elevator [4]. Based on those findings, we operationalize a number of norms into feature maps which give the robot the prior knowledge needed to select acceptable waiting positions. It is worth noting that, while some of the proposed maps are specific to the elevator scenario we tackle, most maps can be reused in other goal selection problems. The *Proxemics* map is a clear candidate and, less trivially, the *Avoid blocking the door* map could be repurposed to other threshold crossing situations, like *e.g.*, the shared use of a badge access gate.

We define a feature map as the application of a function $F : \mathbb{R}^2 \to [0, 1]$ to the space in front of the robot, scoring it based on a subset of the sensory information $\mathcal{S}$. In practice, the selected space is discretized in a $n \times n$ grid, and each cell receives a score from F, with scores closer to 1 being more desirable. In addition to the Occupancy map (fundamentally replicating the static layer of the ROS navigation stack), we design 12 feature maps, separated in two groups: three Basic and nine Norms feature maps. Each feature map is designed to represent only one aspect of the task at hand, allowing learning methods like Decision Trees to retain interpretability.

Basic maps replicate the information contained in one element of $\mathcal{S}$ with no processing, acting as the baseline input for the robot to perform the task. Norms maps encode instead the prior knowledge about the task, often using a subset of $\mathcal{S}$. Fig. 1 depicts each feature map applied to a waiting scene. The three Basic maps include

1) **Elevator location**: scores at 0 the space occupied by the elevator E, providing information about its size and location;
2) **Commands location**: scores at 0 the location of the elevator's commands C_e;
3) **People detection**: scores at 0 the location of the detected people $\mathcal{P}_v$.

The Norms maps were designed following an iterative process, balancing fidelity towards the target norm and complexity of computation. Norms maps make extensive

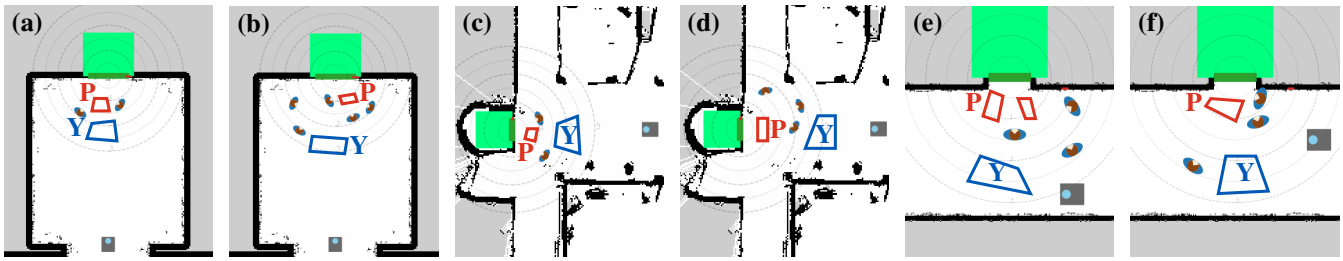

Fig. 2: Annotated scenes from the training set generated with our PCG technique (a-d), along two scenes from the Real Elevator set (e-f), manually recreated from video recordings. The elevator is depicted in green, its commands in red, and the robot as a gray square with a cyan dot. The circles mark the distance in meters from the elevator's door, helping annotators to better judge distances. The highlighted areas are the annotations given for the Yielding (**Y**) and Priority (**P**) missions.

use of Bivariate Normal Distributions $\mathcal{N}(\cdot|\mu, \Sigma)$, Euclidean distance (denoted with $|\cdot|_2$), and logistic functions $\lambda^{\pm}_{\phi,x_0}(x)$, defined as

$$\lambda^{\pm}_{\phi,x_0}(x) = \frac{1}{1 + e^{\pm\phi(x_0 - x)}}, \tag{1}$$

with x_0 being the midpoint of the function (so that $\lambda^{\pm}_{\phi,x_0}(x_0) = 0.5$) and ϕ dictating the steepness of curve. The $\lambda^{+}_{\phi,x_0}(\cdot)$ goes from 0 to 1 as x grows; vice versa for $\lambda^{-}_{\phi,x_0}(\cdot)$. The maps and their hyper-parameters were manually tuned as part of the design process. The nine Norms maps are operationalized as follows.

Proxemics: based on the concept of proxemics [6], [30], it disincentivizes the close proximity of people in $\mathcal{P}_v$, penalizing each person's sides less than their front and back. This is achieved by placing an appropriately oriented Normal distribution $\mathcal{N}(\cdot|\mathbf{p_p}, \Sigma)$ on each person p, with the size of the discouraged area that can be adjusted by varying Σ. The score for location $\mathbf{x}$ is computed as

$$F(\mathbf{x}) = 1 - \min\Big(1, \sum_{p \in \mathcal{P}_v} \mathcal{N}(\mathbf{x}|\mathbf{p_p}, \Sigma)\Big), \tag{2}$$

ensuring the score to be $\in [0, 1]$.

Avoid standing behind people: similarly to the *Proxemics* map, it discourages the robot from waiting behind people, as it can cause discomfort. The radius of the area to be avoided is a hyper-parameter, and the scoring follows a scheme similar to (2).

Avoid blocking the door: this map scores low the entrance of the elevator E, discouraging the robot from potentially blocking people exiting the elevator. It is worth mentioning how people do not always respect this norm and rely on their ability to quickly side-step or step back – something that is however challenging for certain robots. Mathematically, we have

$$F(\mathbf{x}) = \min(0, 1 - \mathcal{N}(\mathbf{x}|E_p, \Sigma)), \tag{3}$$

where the size of the discouraged area varies with Σ.

Avoid blocking the commands: this map scores low areas close to the elevator's commands C_e, operationalizing the fact that the robot should not block human passengers from calling the elevator. The scoring follows the same scheme of (3).

As far as furthest person: As a service principle in the Yielding scenario, we want the robot to yield to all the people in the elevator's vicinity, as to mimic a queuing principle. The map therefore disincentivizes the space between the elevator E and the person standing the furthest from it p_{far}. The score for position $\mathbf{x}$ is computed as

$$F(\mathbf{x}) = \lambda^{+}_{\phi,|p_E - p_{\text{far}}|_2}(|p_E - \mathbf{x}|_2). \tag{4}$$

Avoid transactional space: Kendon [21] defines the space in front of one person engaged in a social context or activity as a transactional segment, over which the person endeavors to maintain some degree of jurisdiction or control. The location and orientation of a transactional segment are framed by the posture and orientation of the body in relation to the social context and activity. When the activity in question involves groups of people, transactional segments can overlap to create a transactional space. Maintaining this space requires cooperation (or negotiation), and previous work implemented it for robotics tasks [17], [18].

When a robot intends to yield to people entering an elevator, it is beneficial to avoid any emergent transactional space. The *Avoid transactional space* map therefore scores low the transactional space between the detected people $\mathcal{P}_v$ and the elevator E. We operationalize this concept by placing Normal distributions between each person $p \in \mathcal{P}_v$ and the elevator E's door, having the score for position $\mathbf{x}$ as

$$F(\mathbf{x}) = 1 - \min\Big(1, \sum_{p \in \mathcal{P}_v} \mathcal{N}(\mathbf{x}|\mu_{E,p}, \Sigma)\Big), \tag{5}$$

where $\mu_{E,p}$ is the midpoint between the elevator E and person p.

Communicative space: In Priority scenarios, the edge of transactional spaces can be used to position the robot strategically to indicate its intent and perform, for example, step-in gestures when the elevator doors open [5]. We operationalize this concept by scoring locations based on their distance from the border of a convex hull built on the people $\mathcal{P}_v$ and the elevator E door. The score for position $\mathbf{x}$ is computed as

$$F(\mathbf{x}) = \lambda^{-}_{\phi,d^*}(d_{x,\text{hull}}), \tag{6}$$

where $d_{x,\text{hull}}$ is the Euclidean distance between x and its closest point on the convex hull, and d^* is a suitably tuned distance, controlling how much area is set to 1.

Efficient boarding: this map incentivizes areas close to the elevator E, encoding the idea that the robot should not wait too far away from the elevator, as that would make the boarding slower and therefore deteriorate service's quality. Similarly to (4), the score for position $\mathbf{x}$ is

$$F(\mathbf{x}) = \lambda^{-}_{\phi,d^*}(|p_E - \mathbf{x}|_2), \tag{7}$$

with d^* being a suitable distance from the elevator.

Visibility cone: this map incentivizes areas where the robot, once stopped, can see the interior of the elevator E. Similarly to the *Efficient boarding* map, the *Visibility cone* map does not directly encode a social norm but rather a technical dependency that can however impact bystanders, as the quicker the robot can see that the elevator's car is full, the quicker it will abort mission and avoid slowing down people. A technique akin to ray-tracing is used to compute the portion of visible elevator interior from a position $\mathbf{x}$ as score.

IV. Experiments

To answer our research questions, we present the results from training a suite of classifiers and performing an ablation study to discern the efficacy of the proposed Norms maps in encoding prior knowledge.

A. Datasets

We generate a training dataset of 125 waiting scenes. Aside from the variability provided by the proposed PCG technique, the training dataset used 3 different locations (2 of which shown in Fig. 2), a number of groups of people varying from 1 to 4, with a maximum of 6 people in the scene at a time. We also vary the dimensions of the elevators, along with the relative position of the commands.

As for test datasets, we have three sets. First, we have a PCG set of 15 scenes named the In-Distribution (ID) set, containing scenes generated with the same PCG hyper-parameters of the training set. Second, we have a procedurally generated set of 67 scenes named the Out-of-Distribution (OOD) set. These scenes have a larger number of people waiting and different elevator's sizes and locations, helping us test the classifiers' out-of-distribution performance. Third, we have the Real Elevator set, with 13 scenes extracted from 16 hours video recordings of an elevator lobby (with 1 to 4 people per scene) and manually recreated in simulation, allowing the tackling of RQ3.

Two authors[2] – a sociologist and an interaction designer, familiar with the service, the robot, and the social norms – annotated waiting areas on each scene, given one of the two target missions with the open-source Computer Vision Annotation Tool (CVAT) [31]. Fig. 2 shows examples of these annotated areas. Annotations are agnostic towards the current robot's location, meaning that the selected areas are socially acceptable regardless of the pose from where the robot will compute them. In other words, the annotators have access to the top-down scene, where all people are detected. The annotations present a high class imbalance, as the acceptable areas represent a small portion of the overall $n \times n$ grid. The average ratio of negative over positive samples is of 200 and 300 for target missions Yielding and Priority respectively. The experts annotated a subset of the scenes, with an overlap of 50 scenes and a Cohen's κ of 0.53, indicating moderate agreement. Each annotation (one scene, one target mission) took the experts an average of two minutes to produce.

[2]The annotators were purposely excluded from the development of the method to avoid unwanted bias, but were informed that their annotations would help automate the robot's waiting behavior.

B. Performance scores

For each model and scene in the test datasets, we compute the Jaccard index $\mathcal{J}(S, A)$, also known as Intersection over Union. The Jaccard index is a commonly used metric for evaluating segmentation performance, ranging from 0 (no overlap between the model's prediction and the ground truth) to 1 (perfect overlap). Fig. 4 presents a number of models' decisions, along with their $\mathcal{J}(S, A)$.

C. Trained classifiers

We want a classifier that takes as input the robot's sensory information $\mathcal{S}$ in the form of the proposed feature maps $\mathcal{F}$ (covering a 5 by 5 meters area in front of the robot, discretized with a $n \times n$ grid) and classifies them as being acceptable waiting position or not. We set n to 64, as a trade-off between decision resolution and training feasibility. We implement two categories of models, Grid-based and Cell-based models, differing in how they use the feature maps as input. Each model was trained in two fashions: with the Occupancy map and Basic maps as input (B) and with the Occupancy map and Norms maps (N), allowing us to address RQ2. We performed data augmentation by sampling the robot's pose for each scene, creating roto-translations of the feature maps and the corresponding annotations.

Grid-based models take the $n \times n$ grids of the feature maps as a hyper-spectral image input, and output a mission-specific mask. We employ a U-Net [32], a deep learning architecture for image segmentation originally proposed for the biomedical domain, where data is scarce and expensive to annotate. After preliminary experiments where we observed the vanilla U-Net and its more recent extensions [33] overfit to our training set with no generalization, we limited the network to a single contracting step, and reduced the number of channels after the first convolution operation from 64 to 16, leading to networks with about 27k parameters. The model is trained for up to 200 epochs by minimizing the Binary Cross Entropy loss, appropriately weighted to account for the aforementioned class imbalance. The best model was selected based on the Jaccard index $\mathcal{J}$ on a validation set, constructed by removing 10 scenes from the training set.

Cell-based models take instead a cell at a time from the $n \times n$ grid of each map as input, classifying each cell based on a vector of the features and their pairwise interaction. These models trade the geometric information about the scene for a more manageable feature space (11 dimensions for B inputs, 56 for N), allowing the adoption of

less data hungry models. We employ Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and Multilayer Perceptron (MLP), as implemented in `scikit-learn` [34]. Parameters' numbers for Cell-based models are 1 to 2 orders of magnitude lower compared to the 27k of the U-Net. For each model, we tune its hyper-parameters and classification threshold with the aforementioned 10 scenes as validation set.

To test our assumption that learning from data is necessary, we include a non-trained Baseline method that segments acceptable waiting areas by thresholding the element-wise minimum values of the input feature maps (implementing a *respect all norms* policy). We test the Baseline model with B maps and with an expert selection of N maps. As an example, the *As far as furthest person* map is removed from the training of the Baseline method classifying Priority areas. This selection is necessary to ensure a fair comparison: some feature maps are relevant for only one mission, but the Baseline method, unlike the trained models, cannot learn this from data.

D. Results and discussion

Table I presents the average Jaccard index $\mathcal{J}$ for each model, provided input (B and N), separated by test set. We first notice how the $\mathcal{J}$ of the non-trained Baseline cannot compete with the trained models, confirming our assumption that learning is indeed necessary. Second, we see how the prior knowledge injected via the N maps helps the models. The models trained with the B maps have low precision and high recall, and learn exclusively that the robot should wait in unoccupied space, missing the necessary nuances provided by the N maps. Overall, the models achieve better Jaccard indexes with the proposed Norms maps, positively answering RQ2. Furthermore, all models have comparable performances between the synthetic test datasets (ID and OOD) and the Real Elevator dataset, positively answering RQ3.

Entering the experiment, we expected the models to achieve Jaccard indexes of .4, based on an average of scores typically achieved in medical segmentation challenges [35] and on the fact that, unlike the annotators, the models' decisions depend on the robot's location and are potentially impacted by undetected people (because of the egocentric camera view and occlusions, as shown *e.g.*, in Fig. 4f).

For the Yielding mission, all trained models using the N maps achieve this target. However, for the Priority mission, all models are short of this number. The lower Jaccard

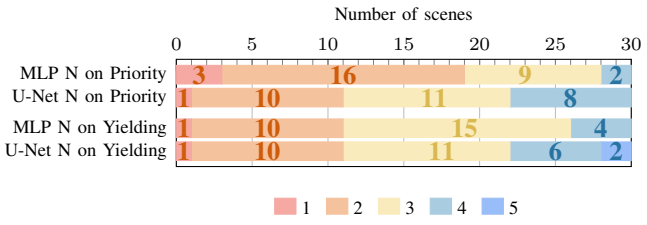

Fig. 3: Score distribution from the expert inspection.

indexes for the Priority mission can be explained by the fact that the areas annotated for this mission are generally small in size and even minor shifts between model decision and annotation can cause low indexes. We can therefore only partially answer RQ1. To investigate these differences in performance, we had an annotator visually inspect the decisions of two models that achieved the highest indexes for the Real Elevator set – MLP and U-Net, both using N maps – on 30 random scenes from the OOD and Real Elevator sets. The expert was prompted with the scenes as shown in Fig. 4 and asked to score them on a 1-5 Likert scale, from 1 *"the model's decision breaches too many social norms and technical dependencies to be considered"* to 5 *"the model's decision is better than the expert's annotation"*, and provide any relevant comment. Fig. 3 summarizes the expert's scores.

While the scores slightly favor the U-Net, we did not find statistically significant differences between the models' scores (Wilcoxon signed-rank test: for Priority T=36, p-value=.05; for Yielding T=22, p-value=.15). The expert pointed out how both models do respect most social norms and technical dependencies, but do not always respect the assigned mission, *i.e.* including yielding locations despite the Priority mission, and vice-versa.

The Cell-based MLP shows the tendency to include Yielding locations in Priority areas (in 16 out of 30 scenes), as shown *e.g.*, in Fig. 4d, where portions of the selected area do not claim priority over *all* the bystanders. The expert annotator commented in this regard how *"the [MLP] model seems to not have learned that the priority principle must be applied to everybody in the scene"*. Furthermore, MLP had the tendency to pay too little attention to the Proxemics norm when seeking Priority – a risky behavior that can however be mitigated by further filtering the model's decision based on norms that are deemed non-negotiable. The opposite issue (Priority positions in Yielding areas) was observed only in 8 of the 30 inspected scenes. An example is shown in Fig. 4a, where the MLP selects an area to the left of the central person, signaling priority over them. For comparison, the U-Net avoids this problematic area (see Fig. 4g), but selects additional areas to the left of the annotation, where visibility is scarce (hence the annotator giving a score of 3).

The U-Net presents fewer mission mismatches (8/30 for Yielding, 9/30 for Priority). As an example, in Fig. 4h the model selects the area behind the rightmost group, incorrectly signaling priority over the person on the left. One possible explanation for this difference is that, despite the prior knowledge injected by the Norms maps, there is still the need for a residual term capturing these details. The U-Net, having more representational capacity than the Cell-based models, is better at recovering this residual term from the annotations.

Overall, the expert's inspection favors the U-Net. Fig. 4i presents a case where the expert reconsidered its annotation after seeing the U-Net's decision, stating that *"the model does select a slightly different waiting area, but its area allows for greater visibility of the elevator's car while still achieving the mission"*. Similarly in Fig. 4j, the U-Net's

TABLE I: Jaccard index $\mathcal{J}$ of models on the test datasets, color-coded from .0 (red) to .5 (blue).

	ID set		OOD set		Real Elevator	
	B	N	B	N	B	N
Model	Target mission: Priority					
Baseline	.016	.136	.014	.108	.015	.122
LR	.010	.160	.019	.173	.015	.164
SVM	.014	.140	.018	.159	.017	.142
DT	.000	.148	.000	.151	.004	.134
RF	.014	.195	.018	.206	.016	.181
MLP	.014	.192	.018	.206	.017	.171
U-Net	.080	.220	.199	.267	.001	.200
Model	Target mission: Yielding					
Baseline	.066	.159	.044	.165	.075	.147
LR	.077	.364	.067	.317	.104	.380
SVM	.078	.352	.072	.302	.106	.350
DT	.078	.325	.072	.247	.105	.328
RF	.078	.349	.072	.297	.106	.365
MLP	.078	.362	.072	.316	.106	.400
U-Net	.059	.413	.128	.309	.045	.318

decision considers only the sides of the annotator's area, avoiding the central area that breaches the *Avoid blocking the door* norm – a norm that the annotator purposely decided to ignore, given the mission and the limited space.

V. Limitations and Future Work

We have explored how the use of social norms maps as priors can help tackling a data scarce goal selection problem. Next, we discuss the observed limitations, possible ways of addressing them, as well as future research directions.

Additional components will be needed to deploy the proposed method in the real world and enable a robot to approach, wait for, and board an elevator. First, given the selected waiting areas, the robot will need to select a goal pose and navigate there. As the robot may need to respect a subset of the social norms we considered here while reaching the selected pose, there is the opportunity of re-using the proposed maps, potentially as rewards for Reinforcement Learning navigation techniques [36] or as costs for traditional planners. Furthermore, our method is modular and additional feature maps could be included as the context requires it, like *e.g.*, maps further describing the environment in which the robots operates. As an example, a *People flow* map could be included [37], discouraging the robot from waiting where people are known to navigate quickly.

Regarding instead the hyper-parameters of the feature maps, while with more data they could be learned from expert's annotations, in future work we want to explore whether modifications of the feature maps through their hyper-parameters (for example, changing the *Efficient boarding* map's d^* parameter to make the norm stricter) can influence the models without retraining. If the frozen model's decisions reflect these modifications, we would have access to a quick way to fine-tune the models to new user's requirements or new environments, without additional training and data annotation.

Once a waiting position is reached and the elevator finally arrives, the robot needs to negotiate the priority with its human bystanders. The drop in performance all models suffer for the Priority mission, and the reachability issues such areas will likely pose to the robot while navigating towards them, indicates that positioning alone may not be sufficient and that additional channels like *e.g.*, (non-)verbal communication need to be investigated.

Finally, embedding the proposed goal selection pipeline in a full navigation scenario is the logical next step. While we expect the U-Net to select goals that respect the investigated social norms, we are interested in studying how people behave around robots which mimic such norms while navigating towards the selected areas.

VI. Conclusions

We tackled the problem of selecting socially acceptable waiting positions for a robot sharing the use of elevators with people. The proposed feature maps, encoding prior knowledge about social norms and technical dependencies of the task, allowed for the training of a variety of models in this low-data regime scenario. Our results confirm the generalization capabilities of models trained solely on PCG scenes, as well as the benefits of injecting social norms directly into the training. As shown by the expert inspection, the models struggle to pick up some of the nuances between the two levels of urgency we considered, with the U-Net's results being more promising and therefore prompting us to further explore this direction. As we believe that many niche yet relevant HRI problems lay in a low-data regime, these results shed a positive light on the use of PCG and the practice of feature engineering for prior encoding.

References

[1] M. M. de Graaf, "An ethical evaluation of human–robot relationships," *International journal of social robotics*, vol. 8, pp. 589–598, 2016.

[2] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[3] S. Rossi, A. Rossi, and K. Dautenhahn, "The secret life of robots: perspectives and challenges for robot's behaviours during non-interactive tasks," *International Journal of Social Robotics*, vol. 12, pp. 1265–1278, 2020.

[4] D. Gallo, S. Gonzalez-Jimenez, M. A. Grasso, C. Boulard, and T. Colombino, "Exploring machine-like behaviors for socially acceptable robot navigation in elevators," in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2022, pp. 130–138.

[5] D. Gallo, P. L. Bioche, J. K. Willamowski, T. Colombino, S. Gonzalez-Jimenez, H. Poirier, and C. Boulard, "Investigating the integration of human-like and machine-like robot behaviors in a shared elevator scenario," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, 2023, pp. 192–201.

[6] E. T. Hall, R. L. Birdwhistell, B. Bock, P. Bohannan, A. R. Diebold Jr, M. Durbin, M. S. Edmonson, J. Fischer, D. Hymes, S. T. Kimball, *et al.*, "Proxemics [and comments and replies]," *Current anthropology*, vol. 9, no. 2/3, pp. 83–108, 1968.

[7] A. Furnham, L. Treglown, and G. Horne, "The psychology of queuing," *Psychology*, vol. 11, no. 3, pp. 480–498, 2020.

[8] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, 2023.

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

[10] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005.

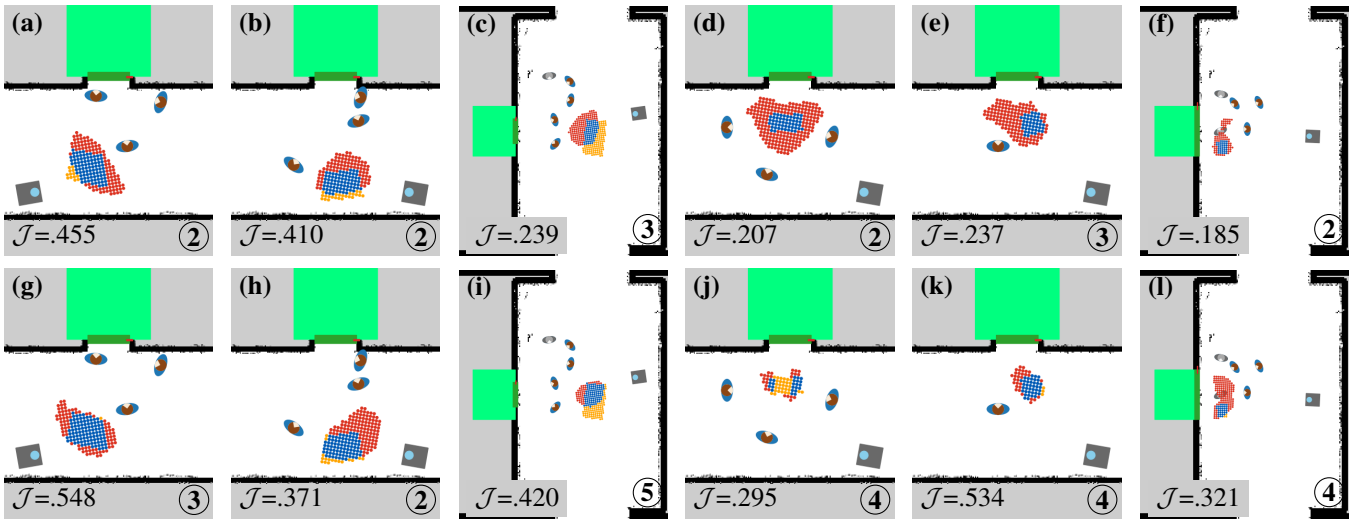

Fig. 4: Decisions on selected scenes from the OOD and the Real Elevator set from the annotators and the best models, MLP (first row) and U-Net (second row). Highlighted areas represent the true positives (in blue), false positives (in red), and false negatives (in orange). Columns 1-3 present Yielding targets, columns 4-6 Priority targets with Jaccard indexes $\mathcal{J}$ and the expert inspection score (circled) overlaid. Undetected bystanders are grayed out.

[11] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 797–803.

[12] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.

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

[14] P. Kothari, B. Sifringer, and A. Alahi, "Interpretable social anchors for human trajectory forecasting in crowds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 556–15 566.

[15] R. Kirby, *Social robot navigation*. Carnegie Mellon University, 2010.

[16] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, "How to approach humans? strategies for social robots to initiate interaction," in *Proceedings of the 4th ACM/IEEE International conference on Human Robot Interaction*, 2009, pp. 109–116.

[17] M. A. Yousuf, Y. Kobayashi, Y. Kuno, A. Yamazaki, and K. Yamazaki, "How to move towards visitors: A model for museum guide robots to initiate conversation," in *2013 IEEE RO-MAN*. IEEE, 2013, pp. 587–592.

[18] M. Vázquez, A. Steinfeld, and S. E. Hudson, "Parallel detection of conversational groups of free-standing people and tracking of their lower-body orientation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3010–3017.

[19] C. Shi, S. Satake, T. Kanda, and H. Ishiguro, "A robot that distributes flyers to pedestrians in a shopping mall," *International Journal of Social Robotics*, vol. 10, pp. 421–437, 2018.

[20] M. Vázquez, A. Lew, E. Gorevoy, and J. Connolly, "Pose generation for social robots in conversational group formations," *Frontiers in Robotics and AI*, vol. 8, p. 703807, 2022.

[21] A. Kendon, *Conducting interaction: Patterns of behavior in focused encounters*. CUP Archive, 1990, vol. 7.

[22] Yunji Technology, "YUNJI robot intelligent delivery service robot run," https://youtu.be/yCnRtHoOg3I, 2020.

[23] R. Ayaß, "Doing waiting: An ethnomethodological analysis," *Journal of Contemporary Ethnography*, vol. 49, no. 4, pp. 419–455, 2020.

[24] D. M. Nguyen, M. Nazeri, A. Payandeh, A. Datar, and X. Xiao, "Toward human-like social robot navigation: A large-scale, multi-modal, social human navigation dataset," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2023.

[25] G. Smith, "An analog history of procedural content generation," in *Proceedings of the 2015 Foundations of Digital Games Conference*, 2015.

[26] S. Risi and J. Togelius, "Increasing generality in machine learning through procedural content generation," *Nature Machine Intelligence*, vol. 2, no. 8, pp. 428–436, 2020.

[27] E. Cheung, T. K. Wong, A. Bera, X. Wang, and D. Manocha, "LCrowdV: Generating labeled videos for simulation-based crowd behavior learning," in *Computer Vision–ECCV 2016 Workshops*. Springer, 2016, pp. 709–727.

[28] O. Rogla, G. A. Patow, and N. Pelechano, "Procedural crowd generation for semantically augmented virtual cities," *Computers & Graphics*, vol. 99, pp. 83–99, 2021.

[29] A. Willmott, "Fast object distribution," in *ACM SIGGRAPH 2007 Sketches*. Association for Computing Machinery, 2007, p. 80.

[30] R. Mead and M. J. Matarić, "Autonomous human–robot proxemics: socially aware navigation based on interaction potential," *Autonomous Robots*, vol. 41, no. 5, pp. 1189–1201, 2017.

[31] OpenCV, "Computer vision annotation tool (cvat)," https://github.com/opencv/cvat, 2023.

[32] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[33] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE access*, vol. 9, pp. 82 031–82 057, 2021.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org

[35] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki, *et al.*, "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," *arXiv preprint arXiv:1811.02629*, 2018.

[36] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5671–5677.

[37] F. Verdoja, T. P. Kucner, and V. Kyrki, "Bayesian floor field: Transferring people flow predictions across environments," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 801–12 807.