Pedestrian Motion Pattern Prediction for Social Navigation Via Self-Supervision

Felipe Felix Arias, Marco Morales and Nancy M. Amato
University of Illinois at Urbana-Champaign

***Abstract*—Recent advances in machine learning have unlocked new possibilities in human modeling, self-supervision, and lifelong learning. This work addresses relevant applications of these fields to navigation amongst pedestrians by using a self-supervised segmentation neural network to predict motion patterns from traversability maps. Our proposed model learns to predict how frequently pedestrians may visit a given region from an environment's topology and geometry. We extend graph-theoretic concepts to crowd simulation in real-world buildings for supervision and image segmentation for motion pattern prediction, resulting in a novel application of existing techniques. We show that the proposed model and annotation algorithm generalize to environments not seen during training, can generate reasonable predictions for environments of any size, and capture intuitive patterns such as the locations of chokepoints. We also study the model's applicability to social navigation and formally define the problem of predicting the long-horizon motion of pedestrians for path planning.**

I. Introduction

In robotics, navigation refers to the process by which an agent computes a feasible and collision-free trajectory to arrive at a desired location given a state representation and set of obstacles [15]. Adding other agents or people (i.e., pedestrians) to the environment adds considerable complexity and requires additional computation to account for their motions. There have been many solutions proposed for so-called social navigation [17]. Nonetheless, further work is necessary to find a solution capable of reliably arriving at a given goal with minimal social costs (e.g., personal space violations) and within a reasonable amount of time. Of particular interest to this work are the additional challenges that arise when deploying robots in constrained pedestrian environments like the one seen in Fig. 1a (i.e., houses or apartments). Spatial constraints such as doorways affect maneuverability and create layout-dependent patterns of pedestrian interaction that continue to challenge navigation systems [22]. To this end, we seek to develop a model that can predict long-horizon motion patterns in constrained environments. Endowing agents with the ability to predict where they may encounter others and how they are likely to move would enable preemptive collision avoidance and plans that better minimize potential social and path costs.

Recent works on sampling-based motion planning [10], [1] have used neural networks to learn to characterize motion patterns from local environment features to sample in regions that are relevant motion planning. By using an approximation of *betweenness centrality* [7], a graph-theoretic measure of how frequently a vertex is visited in shortest paths through a graph, to compute the labels for a regression or classification model, it has been shown that neural networks can effectively identify relevant sampling regions (e.g., chokepoints). In contrast, our method models the prediction as segmentation and computes a task-agnostic feature representation that better captures the long-horizon relationship between centrality and topology over the entirety of the environment.

a) Map b) Simulated Data c) Predictions

Fig. 1: Merom 0 environment a) map, b) heat map from hundreds of simulated pedestrian trajectories and c) motion pattern predictions using our approach. As seen, the predictions capture the narrow passages and frequently visited regions of the environment.

The contributions of this paper are threefold. First, we define a self-supervised methodology, dataset, and map annotation strategy for learning to predict pedestrian motion patterns from maps of real-world buildings through segmentation. Second, we formally define the problem of predicting the future motion of pedestrians to minimize future social costs and provide an algorithm for estimating future social costs using our model. Third, we demonstrate that our proposed model and strategy can be used for motion pattern and social cost prediction in previously unseen environments.

II. Related Work

Human Trajectory Prediction. Our proposed approach is distinct from existing human and traffic modeling research in two fundamental ways. First, our model does not explicitly predict the future trajectories of pedestrians. Instead, it estimates a region's occupancy probability based on previous experience. Second, our model only requires a traversability map as an input and generalizes to topology for which observations are not yet available (e.g., a house the robot has not visited before).

In the context of navigation, active collision avoidance would require additional trajectory prediction for collision

avoidance. Various systems estimate human activity and flow from stationary sensors like security cameras by tracking pedestrians and their trajectories (e.g., [6], [35], [11], [30]). However, they are seldom used for navigation and instead focus on surveillance and monitoring. In general, most trajectory prediction in robotics leverages information about the recent motion of pedestrians within the sensing range to predict where each agent will be in the near future [28]. Most relevant to this work is the relatively small subset of methods that explicitly account for environment geometry and topology (e.g., [24], [26], [27], [13]).

The existing approaches for motion pattern prediction do not scale well for robotics applications due to their assumed access to a sensor network with good coverage and the known susceptibility to errors when deployed in perspectives or scenes not seen during training [28]. Unlike such systems, we abstract out trajectory prediction and active monitoring by assuming access to a data set of traversability maps annotated with traffic flow heat maps computed by an arbitrary sensor network or robot(s). Therefore, we model motion pattern prediction as the probability of occupancy over a long time frame rather than future trajectory prediction.

Learning Sampling Distributions. In recent years, there has been an increased interest in using machine learning to bias sampling in sampling-based motion planning. This family of approaches sample configurations in free space and connects them with traversable edges to construct a graph that reduces the problem of motion planning to pathfinding. While most existing machine learning-enhanced methods focus on Ad-Hoc models for learning sampling distributions or heuristics (e.g., [21], [8], [3], [29], [9], [37], [36], [19]), others propose a task-agnostic approach for identifying regions critical to solution paths through regression [10] or classification [1]. Unlike in our work, these task-agnostic models produce a single prediction for a given region and are evaluated solely on the downstream application improvements and not prediction accuracy.

[10] tracks *criticality*, a measure of how often each vertex is visited in many one-to-all problems in various roadmaps, and learns a regression model to predict criticality based on local topology. This model is then used to construct single-agent probabilistic roadmaps that identify a subset of the critical configurations through pre-sampling and connect all subsequent samples to the critical samples, effectively finding solutions with fewer samples. [1] computes criticality in dynamic environments and uses waiting in place as a metric for determining whether a region is helpful for obstacle avoidance. With a classification network trained to identify relevant areas in coarse, randomly generated environments, [1] uses a sliding window approach to identify critical areas and constructs a learned skeleton for multi-agent motion planning that includes configurations that enable collision avoidance.

[10] and [1] provide an intuition for how learning to predict criticality can improve the scalability of sampling-based methods with respect to environment complexity, size, and the number of agents. However, they solely focus on the sample efficiency and runtime improvements to sampling-based motion planning. We argue that criticality-annotated maps are useful representations of how other agents move more generally. To this end, we focus on novel methods for annotating traversability maps of real-world apartments and seek to capture pedestrian motion patterns that span more significant swaths of the environment.

Image Segmentation. Given that motion patterns span multiple regions and that the criticality of a region is not independent of those around it, we formulate the prediction of motion patterns as a segmentation problem. Image segmentation is a quintessential computer vision problem that partitions images into clusters of pixels corresponding to segments or objects. In recent years, deep learning has become the most common method for segmentation due to its state-of-the-art performance in various vision tasks [18]. We utilize fully convolutional deep neural networks (FCNs) [16], which consist of only convolutional layers, use skip connections for resolution robustness, and generate segmentation maps of the same size as the input. Notably, the loss functions of FCNs accumulate the error in predictions for all pixels within the input image, a desirable feature when the class of each pixel is not independent of those around it. In our implementation, we use U-Nets [25] a commonly used and simple architecture that has shown success in multiple domains.

Social Navigation. Safe navigation amongst humans is essential for deployment in the real-world and has been one of the most studied problems in robotics. Early on, solutions treated motion planning and control as independent submodules and resulted in the robot "freezing" in scenarios where the planner could not find feasible solutions given the assumed pedestrian trajectories ([32], [31]). These and more modern solutions generally rely on predicted pedestrian trajectories or positions as inputs and may lead to planning based on incorrect assumptions or predictions. Recent approaches that leverage imitation and reinforcement learning instead couple prediction and planning to compute policies that match human navigation patterns (e.g., [20], [2], [23]) or maximize an expected reward (e.g., [5], [4], [22]) respectively. While these approaches provide a framework for learning rather than encoding all of the complexities of social navigation, they are limited to the regions of the state space covered during training. Consequently, learning-based approaches lack safety guarantees and do not provide a scalable and flexible life-long learning framework. In this work, we study how machine learning can extract relevant features for learning and non-learning techniques based on experience. By accumulating data throughout a robot's lifetime, we believe that models like the one proposed in this paper can learn patterns that can more easily enable preemptive collision avoidance, life-long learning, and context-dependent behaviors (e.g., account for the time of day).

III. Problem Statement

In order to best express and understand the most formal application of our model to social navigation, we use a general

formulation for social navigation from [17]. We extend the formulation and instead express the optimization in terms of the expected cost given an initial trajectory. Let $\mathscr{Q}$ be a given planar configuration space for the robot of interest. Given some configuration $c \in \mathscr{Q}$, we denote by $A_R(c)$ the area the robot takes at that position in workspace. At some time $t \geq 0$, the agent has to navigate amongst a set of $\bar{n}_p \geq 1$ pedestrians. Given some configuration $c \in \mathscr{Q}$ for pedestrian $j \in [\bar{n}_p]$, we denote by $A_j(c)$ the volume it takes at that position in workspace. The agent and pedestrians intend to reach their destinations g_R and $(\bar{g}_j)_{j\in[\bar{n}_p]}$ respectively while avoiding collisions with the a set of static obstacles $\mathscr{C}_{obst}$ and abiding to social norms (e.g., respecting the personal space of others). Let $\mathscr{T} = \{\tau : [0,1] \to \mathscr{Q} \mid \tau(0) = q_R, \tau(1) = g\}$ be the set of all possible trajectories over the configuration space (parameterized as a curve) that start at point $q_R \in Q$ and finished at the goal point $g \in \mathscr{Q}$. At planning time, the agent generates an initial intended trajectory, or global plan, $\tau^*_{\text{ini}} \in \mathscr{T}$ by solving the following optimization problem:

$$\tau^*_{\text{ini}} = arg\,min_{\tau_{\text{ini}}\in\mathscr{T}}\ \mathbb{E}_{\tau_f,\overline{\mathscr{T}}\sim\phi(\tau_{\text{ini}},\mathscr{E})}\left[c(\tau_f) + \lambda c^s(\tau_f,\overline{\mathscr{T}})\right] \quad (1)$$

$$\begin{aligned} s.t.\ A_R(\tau_i(t)) &\notin \mathscr{C}_{obs} \forall t \in [0,1] \\ \tau_{\text{ini}}(0) &= \tau_f(0) = q_R \\ \tau_{\text{ini}}(1) &= \tau_f(1) = g_R \end{aligned}$$

where $\phi(\tau_{\text{ini}}, \mathscr{E})$ is a simulation of potential navigation episodes given the initial trajectory τ_{ini} and the environmental context $\mathscr{E}$ (e.g., environment geometry/topology, time of day, semantics, etc.). This simulation is stochastic due to the stochastic nature of the pedestrian trajectories. Since the initial trajectory may require additional replanning due to the presence of an arbitrary number of pedestrians with stochastic trajectories, the simulation must compute multiple final trajectories $\tau_f \in \mathscr{T}$ and pedestrian trajectories $\overline{\mathscr{T}} = (\bar{\tau}_1, \ldots, \bar{\tau}_{n_p})$, where $\bar{\tau}_i : [0,1] \to \mathscr{Q}$, $i \in [\bar{n}_p]$.

Since there are quasi-infinite potential scenarios, the agent must consider a large number of potential scenarios when choosing its initial path τ_{ini}. In order to quantify the cost of its trajectory the agent must consider $c : \mathscr{T} \to \mathbb{R}$, a cost corresponding to the nature of the path (e.g., how long it took the robot to traverse it) and $c^s : \mathscr{T}^{1+n_p} \to \mathbb{R}$, a λ-weighted cost that captures social considerations and takes into account predictions about the future behavior of others $\overline{\mathscr{T}} = (\bar{\tau}_1, \ldots, \bar{\tau}_{n_p})$. In general, the robot does not have access to the navigation costs of the pedestrians, which includes $\bar{c}_j$, $\bar{c}^s_j$, trajectory $\bar{\tau}_j$, or weight λ of the pedestrians, though each is computing a similar optimization independently. In this work, we refer to this type of navigation as social navigation.

The objective above captures the most general form of social navigation. However, it is intractable because the social costs are not strictly defined. Therefore, we focus on the cost of proximity. We model the social cost as the average of the distances to the pedestrians that are within a distance threshold δ through a zero mean Gaussian density, a common formulation for social costs ([14]). In other words, the social costs are exponentially larger the closer the pedestrian is to the robot. We sum over the Gaussian density from the beginning to the end of the discretized final trajectories sampled. Let $\hat{Q}$ be a discretization of the configuration space. Then, we can define the discrete set $\mathscr{B}_\delta(p) := \{x \in \mathscr{Q} \mid \|x - p\| \leq \delta\} \cap \hat{Q}$. We also define $\hat{\mathscr{T}}$ the set of trajectories taking values in $\hat{Q}$ that start in $q_r \in \hat{Q}$ and finish in $g \in \hat{Q}$ and $\hat{\tau} \in \hat{\mathscr{T}}$, the set of elements in $\hat{Q}$ that are occupied by the trajectory τ. Finally, we discretize the time and assume the robot agent and pedestrians move in a total time interval of $N+1$ steps, from time step 0 to N.

$$c^s(\tau_f, \overline{\mathscr{T}}) = \sum_{n=0}^{N} \frac{1}{|\bar{\Gamma}_\delta(\hat{\tau}_f(n), n)|} \sum_{x\in\bar{\Gamma}_\delta(\hat{\tau}_f(n),n)} e^{-\frac{\|\hat{\tau}_f(n)-x\|^2}{2\sigma^2}} \quad (2)$$

where $\bar{\Gamma}_\delta(p,n) = \mathscr{B}_\delta(p) \cap \{q \mid q \in \{\bar{\tau}_j(n)\}_{j\in[\bar{n}_p]}\}$.

Rather than simulate the set of trajectories $\overline{\mathscr{T}}$ given $\mathscr{E}$ we propose to model the probability of finding a pedestrian at any location $x \in \hat{Q}$ (independent of time) with a model $h_\theta(x)$. As discussed in Sec. I, this is the focus of our work. Our goal is to estimate an appropriate h_θ or similar representation that can be used to estimate the social costs relative to other options. Given such a model, we propose to estimate the social costs with the following expression:

$$c^s(\tau_f, \overline{\mathscr{T}}) \approx \sum_{n=0}^{N} \frac{1}{|\mathscr{B}_\delta(\hat{\tau}_f(n))|} \sum_{x\in\mathscr{B}_\delta(\hat{\tau}_f(n))} h_\theta(x) e^{-\frac{\|\hat{\tau}_f(n)-x\|^2}{2\sigma^2}} \quad (3)$$

IV. Methodology

A. Data Generation

The goal of our model is to learn to predict which regions of the map are most frequently visited by pedestrians from experience. We generate our training data by tracking the criticality of single and multiple agents over multiple episodes. Our multi-agent simulation uses reciprocal collision avoidance (ORCA) [33], a commonly used crowd simulation algorithm for social navigation simulations, to simulate trajectories in realistic reconstructions of apartments from the iGibson dataset [34]. We generate relevant ground-truth annotations by simulating multiple episodes and tracking the area swept by the pedestrians during execution. Once the visit counts per pixel are collected, we normalize and bin them into three classes, effectively constructing a segmentation mask where each class corresponds to the "level" of criticality of the region based on the uniform quantization. While three is an arbitrary hyper-parameter, we chose it since we observed that there are roughly three clusters in the motion patterns: chokepoints/narrow passages, regions frequently visited by pedestrians given their proximity to chokepoints and narrow passages, and open spaces. We use this intuition for the qualitative analysis of our results and propose it as a guideline for human-generated labels in the future. Fig. 1b shows the segmentation labels for one of the training environments (brighter colors depict more visited regions). This procedure

is computed in eight environments, five of which are used for training (shown in Appendix A) and three for evaluation.

The dataset Y consists of segmentation mask labels y and unannotated occupancy grids x. Where $y^i_{p,c} \in [0,1]$ for every pixel p in the i-th grid depending on whether it is labeled as class c or not. There are two special classes, 0 and -1 for pixels in $\mathscr{C}_{obst}$ and unlabeled free space ($\mathscr{C}_{free}$), respectively. These classes are necessary for calculating the loss and weights. The input features x are occupancy grids of size $(n \times n)$, where $x^i_p \in [0,1]$ for every pixel p in the i-th grid depending on whether the location is an obstacle or free space. We extract grids with overlaps from the environments, and rotate/mirror them to create eight unique transformations of each example and discard examples with few labeled pixels. Further details on the dataset can be found in Appendix A.

B. Segmentation U-Net

We use a U-Net [25] based implementation, an FCN that downsamples, upsamples, and skips connections to learn features at various resolutions. Using Y, we train h_θ. Which outputs $\hat{y} \in \mathbb{R}^{n \times n}$, a matrix consisting of the criticality predictions for each pixel. During training, we minimize pixel-wise cross-entropy loss and optimize with Adam [12]. Namely, we minimize:

$$\mathscr{L}(\theta) = -\sum_{p \in y} \sum_{c=1}^{C} w_c (\log \hat{y}_{p,c}) y_{p,c} \tag{4}$$

Where p is the pixel position in y and w_c is a weight that depends on the class. Since the training data is imbalanced, we use the inverse of the frequency of each class as its weight. We also have many unlabeled pixels, so we assume that the unlabeled pixels follow the same class distribution as the labeled critical pixels and add them to the labeled class counts to compute the weights. Without increasing the counts of critical pixels, the weights would practically ignore the obstacle class due to the considerably larger number of obstacle pixels relative to labeled free-space pixels.

During training, we set the weight of unlabeled pixels in $\mathscr{C}_{free}$ to 0. Therefore, the optimization does not include unlabeled free space in the loss and prevents the network from predicting pixels without a criticality value. More details on the network can be found in Appendix E. We found that the network benefits from predicting pixels in $\mathscr{C}_{obst}$, effectively learning to predict where obstacles are despite the obstacles being perfectly specified in the input. Notably, the model sometimes predicts obstacles in $\mathscr{C}_{free}$ due to it not being penalized for doing so over unlabeled $\mathscr{C}_{free}$. We assign an additional criticality level to those predictions to avoid creating misspecified traversability maps. The nature of the problem raises interesting research questions outside of the scope of this work. In the future, we plan to further explore model prediction over topology, where one may not want to predict $\mathscr{C}_{obst}$ nor treat $\mathscr{C}_{obst}$ and unlabeled $\mathscr{C}_{space}$ in the same way in the loss function.

Algorithm 1 Map Annotation

Input: Map X of dimensions $r \times c$, h_θ with input $(n \times n)$
Output: Map annotated with h_θ

1: output $\leftarrow 0$
2: **for** Every $(n \times n)$ segment around $x \in X$ **do**
3: output$_x \leftarrow$ output$_x$ $+h_\theta(x)$
4: **end for**
5: **if** EVALUATION **then**
6: output $\leftarrow$ quantize(output)
7: **else**
8: output $\leftarrow$ normalize(output)
9: **end if**
10: output $\leftarrow$ output $\circ X$
11: **return** output

C. Map annotation

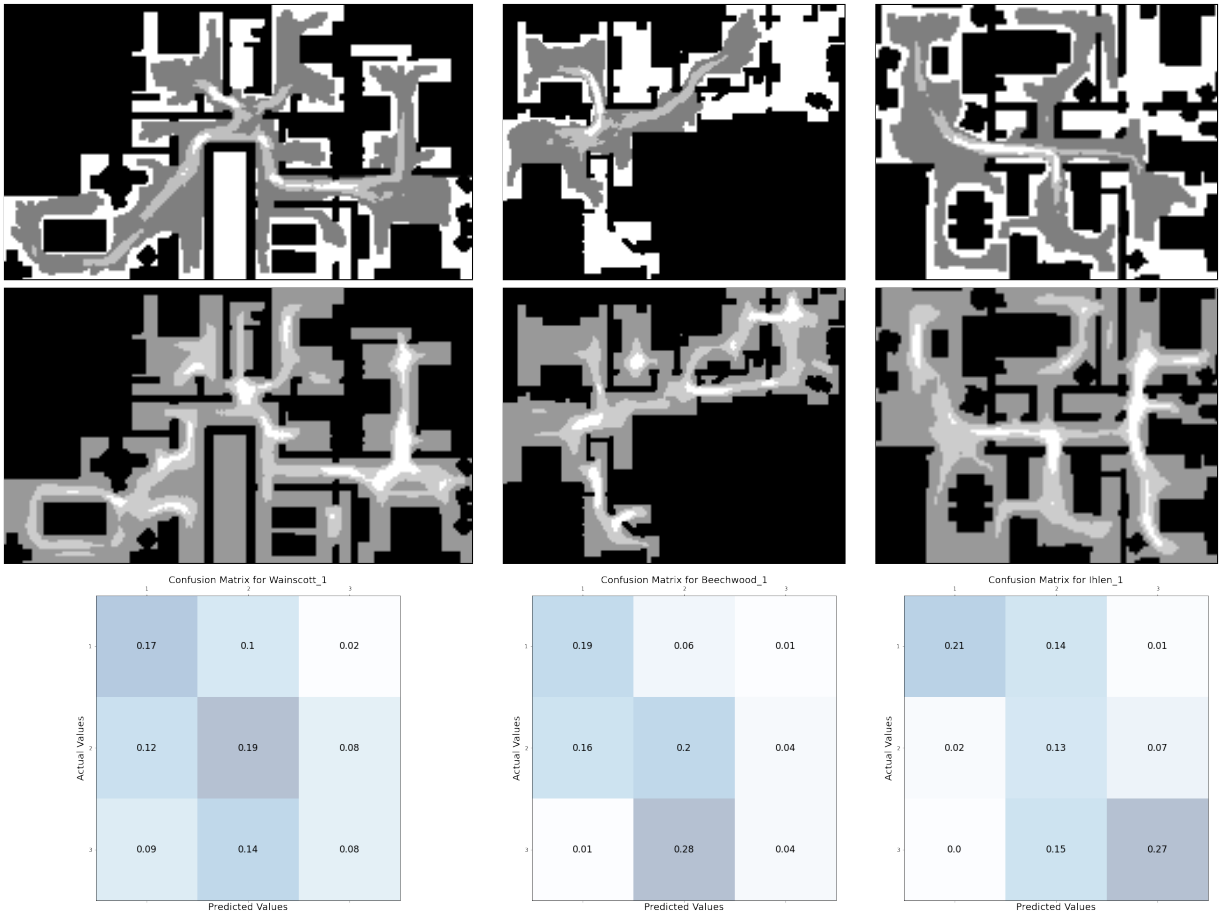
After training h_θ, we use a sliding window approach to compute multiple predictions per pixel and avoid artifacts from predictions with insufficient topological context. Alg. 1 breaks down our proposed map annotation strategy. First, The map is iterated over with a stride of one and the model is applied to every possible $n \times n$ occupancy grid that contains a non-obstacle pixel (padding if necessary) (line 2). The predictions are accumulated in an array (line 3). For evaluation, the sum of predictions is then quantized such that the distribution of predictions matches a given distribution (line 6). The quantization subroutine finds the binning thresholds (i.e., percentiles) for each class and assigns the class label for each pixel accordingly. If a discrete output is not necessary or desired, the output is simply normalized (line 8). Finally, the predictions are multiplied element-wise with the input traversability map so that the annotated map contains the same obstacle geometry as the input (line 10). Additional discussion on our map annotation methodology is included in Appendix D.

V. Experimental Results

We use three testing environments for evaluation. As with the training data, the ground-truth labels for segmentation evaluation are noisy and imbalanced due to the randomly sampled starts and goals of the trajectories, the limited number of simulated trajectories used to generate the training data, and the sparse nature of criticality. While other metrics like betweenness centrality may be deterministic, we argue that this approach is more applicable to predicting the movement of agents controlled by an unknown policy or from real-world data. We opted against hand labeling the environments according to the guidelines described in Sec. IV-A to prevent introducing any bias. For social cost prediction evaluation, we use the normalized annotations.

A. Segmentation Evaluation

Test Environments. Fig. 2 summarizes the results of our approach's segmentation predictions. The confusion matrices in Fig. 2c are scaled by the weights proposed in Sec. IV-B and show that our predictions are generally correct with respect



a) Wainscott 1 b) Beechwood 1 c) Ihlen 1

Fig. 2: Ground-truth, predictions, and confusion matrices for Wainscott 1, Beechwood 1, and Ihlen 1 environments

to the simulated data. Class value 1 corresponds to least frequently visited regions (darkest), while 2 and 3 correspond to frequently visited regions and the most frequently visited regions respectively. For Wainscott 1 and Ihlen 1, two of the predicted criticality values are mostly predicted correctly, while only one is mostly predicted correctly for Beechwood 1. Notably, whenever a prediction is wrong, it is almost always predicted as an adjacent criticality value. In other words, the model only slightly over or underestimates the criticality of a region. This is expected because of the noise in the ground-truth and reflects the continous nature of motion patterns. Qualitatively, the predictions in Fig. 2 capture the desired narrow passages and patterns. Overall, the correlation between the class labels and the predictions and our intuition of the desired output yield acceptable results.

Unlabeled Regions. There are additional opportunities to assess the performance of the network qualitatively. In over half of the training and testing environments, parts of the maps are not labeled in the simulated data. As seen in Figs. 2a-c, our model predicts the expected patterns and generalizes to unlabeled regions. This finding further motivates the use of machine learning for motion pattern prediction. While using the simulated training data or a non-learning-based method for pattern prediction may be more straightforward, the agent may not be able to accurately generalize the patterns to other rooms within the same environment.

B. Social Cost Prediction Evaluation

Lastly, we evaluate whether the output of our approach is a good estimator of social costs as described in our motivation (Sec. III). The goal is to approximate the social costs of a discrete trajectory by treating $h_\theta(x)$ as the probability of occupancy by a pedestrian in the annotated environments (See Appendix F). We simulate navigation episodes with more than four pedestrians each in the test environments and compute the ground-truth and predicted social costs of one trajectory per navigation episode. To depict the comparison between the predicted costs and ground-truth, we sort the episodes based on both the predictions and simulated data and measure how well the model predicted the social costs relative to other episodes by computing the offset between the ground-truth and predicted ranks.

In the experiments depicted in Fig. 3, we set the standard deviation σ to 1 meter and distance threshold δ to 2 meters. We use around 70 episodes per environment and paths of

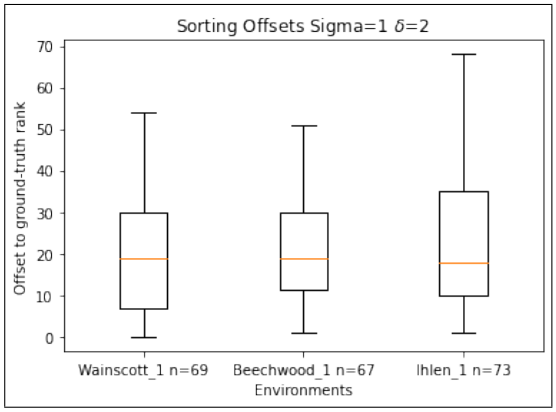


Fig. 3: Boxplots of sorting offsets when comparing the ground-truth rank and predicted rank of social costs of n navigation episodes.

equal length. The sorting offsets have high variance but are better than random and predict over half of the episodes within 25% of the right rank. These results are expected since our estimate always assumes that there is a non-zero probability of occupancy (as judged by the environment's topology). Therefore, the approximation inherently provides an incorrect estimate when no pedestrians are present. Generally, this stems from the stochastic nature of pedestrian trajectories and our predicted social cost is a conservative approximation when the robot does not encounter pedestrians during execution time.

Overall, it is difficult to assess whether a model is a reasonable estimate for social costs due to the stochasticity of the problem. Despite this, the most crucial aspect of such a model is its ability to estimate the social costs relative to other potential trajectories. In practice, robots can use incorrect estimates if the best trajectory has a lower estimate than the rest. As mentioned in Sec. I and is done in relevant related work, we intend to further evaluate the model by measuring its benefits to downstream applications in future work. Moreover, we note that, in practice, the robot would update its estimates as observations become available and that a robot could reasonably behave more cautiously than necessary in various situations when predicting social cost.

VI. Limitations

While there are various open research questions and some arbitrary design choices in our work, most stem from the stochastic nature of the problem and novelty of our approach. Ideally, our evaluation would use agreed-upon ground-truths and metrics for social navigation, but the nature of the problem leads to the need to propose new evaluation techniques. Moreover, our work is most useful for a subset of the potential scenarios, where there are various pedestrians in the environment or the agents must share relatively small regions of the environment. While these scenarios are not infrequent in the real-world, the usefulness of our approach is therefore most easily measured when it is used to improve planning.

A more intrinsic limitation of this instance of our model stems from the simulated data. Though the approach should be robust to sim-to-real given that the model does not rely on physics simulation or simulated high-dimensional sensor inputs (e.g., lidar scans), the motion patterns the model learns are those of ORCA controlled pedestrians and not real humans. A real-world dataset would capture more nuanced semantic information and likely be more useful for deployment. However, compiling a dataset as thorough as ours without a simulator would require substantial work. We assume access to generally unknown information that would likely have to be collected from cameras and other sensors without coverage of the entire environment. Currently, no such real-world dataset exists, but the need for such data and better simulators would help ground the validation of this work.

VII. Conclusion

In this paper, we proposed a novel methodology for learning to predict motion patterns from traversability maps through self-supervision, discussed the implications of motion pattern predictions in social navigation, and demonstrated that our approach generalizes to new environments. Most importantly, our work sheds light on the need for predicting long horizon pedestrian motion for preemptive collision avoidance and social cost reduction. We also discuss various directions for future work and address the limitations of our proposed model. In the future, we plan to deploy the proposed model to improve social navigation solutions and further validate the usefulness of predicting the probability of occupancy from topology.

VIII. Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 46756.

References

[1] Felipe Felix Arias, Brian Ichter, Aleksandra Faust, and Nancy M. Amato. Avoidance critical probabilistic roadmaps for motion planning in dynamic environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10264–10270, 2021. doi: 10.1109/ICRA48506.2021.9560974.

[2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[3] Constantinos Chamzas, Zachary Kingston, Carlos Quintero-Peña, Anshumali Shrivastava, and Lydia E Kavraki. Learning sampling distributions using local 3d workspace decompositions for motion planning in high dimensions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1283–1289. IEEE, 2021.

[4] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware

robot navigation with attention-based deep reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6015–6022. IEEE, 2019.

[5] Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P How. Socially aware motion planning with deep reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1343–1350. IEEE, 2017.

[6] David Ellis, Eric Sommerlade, and Ian Reid. Modelling pedestrian trajectory patterns with gaussian processes. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1229–1234. IEEE, 2009.

[7] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[8] Jinwook Huh, Volkan Isler, and Daniel D Lee. Cost-to-go function generating networks for high dimensional motion planning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8480–8486. IEEE, 2021.

[9] Brian Ichter, James Harrison, and Marco Pavone. Learning sampling distributions for robot motion planning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7087–7094. IEEE, 2018.

[10] Brian Ichter, Edward Schmerling, Tsang-Wei Edward Lee, and Aleksandra Faust. Learned critical probabilistic roadmaps for robotic motion planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9535–9541. IEEE, 2020.

[11] Kihwan Kim, Dongryeol Lee, and Irfan Essa. Gaussian process regression flow for analysis of motion trajectories. In *2011 International Conference on Computer Vision*, pages 1164–1171. IEEE, 2011.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Julian FP Kooij, Fabian Flohr, Ewoud AI Pool, and Dariu M Gavrila. Context-based path prediction for targets with switching dynamics. *International Journal of Computer Vision*, 127(3):239–262, 2019.

[14] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743, 2013.

[15] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[17] Christoforos Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core challenges of social robot navigation: A survey. *arXiv preprint arXiv:2103.05668*, 2021.

[18] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[19] Marco Morales, Lydia Tapia, Roger Pearce, Samuel Rodriguez, and Nancy M. Amato. A machine learning approach for feature-sensitive motion planning. In M. Erdmann, D. Hsu, M. Overmars, and A. F. van der Stappen, editors, *Algorithmic Foundations of Robotics VI*, Springer Tracts in Advanced Robotics, pages 361–376. Springer, Berlin/Heidelberg, 2005. (WAFR '04).

[20] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann Cun. Off-road obstacle avoidance through end-to-end learning. *Advances in neural information processing systems*, 18, 2005.

[21] Keisuke Okumura, Ryo Yonetani, Mai Nishimura, and Asako Kanezaki. Ctrms: Learning to construct cooperative timed roadmaps for multi-agent path planning in continuous spaces. *arXiv preprint arXiv:2201.09467*, 2022.

[22] Claudia Pérez-D'Arpino, Can Liu, Patrick Goebel, Roberto Martín-Martín, and Silvio Savarese. Robot navigation in constrained pedestrian environments using reinforcement learning. *arXiv preprint arXiv:2010.08600*, 2020.

[23] Ashwini Pokle, Roberto Martín-Martín, Patrick Goebel, Vincent Chow, Hans M Ewald, Junwei Yang, Zhenkai Wang, Amir Sadeghian, Dorsa Sadigh, Silvio Savarese, et al. Deep local trajectory replanning and control for robot navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5815–5822. IEEE, 2019.

[24] Ewoud AI Pool, Julian FP Kooij, and Dariu M Gavrila. Using road topology to improve cyclist path prediction. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 289–296. IEEE, 2017.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[26] Andrey Rudenko, Luigi Palmieri, and Kai O Arras. Predictive planning for a mobile robot in human environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Works. on AI Planning and Robotics*, 2017.

[27] Andrey Rudenko, Luigi Palmieri, Achim J Lilienthal, and Kai O Arras. Human motion prediction under social grouping constraints. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3358–3364. IEEE, 2018.

[28] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020.

[29] Manish Saroya, Graeme Best, and Geoffrey A Hollinger. Roadmap learning for probabilistic occupancy maps with topology-informed growing neural gas. *IEEE Robotics and Automation Letters*, 6(3):4805–4812, 2021.

[30] Meng Keat Christopher Tay and Christian Laugier. Modelling smooth paths using gaussian processes. In *Field and Service Robotics*, pages 381–390. Springer, 2008.

[31] Pete Trautman, Jeremy Ma, Richard M Murray, and Andreas Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research*, 34(3):335–356, 2015.

[32] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 797–803. IEEE, 2010.

[33] Jur van den Berg, Stephen J Guy, Jamie Snape, Ming C Lin, and Dinesh Manocha. Rvo2 library: Reciprocal collision avoidance for real-time multi-agent simulation, 2011.

[34] 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. doi: 10.1109/LRA.2020.2965078.

[35] YoungJoon Yoo, Kimin Yun, Sangdoo Yun, JongHee Hong, Hawook Jeong, and Jin Young Choi. Visual path prediction in complex scenes with crowded moving objects. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 2668–2677, 2016.

[36] Clark Zhang, Jinwook Huh, and Daniel D Lee. Learning implicit sampling distributions for motion planning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3654–3661. IEEE, 2018.

[37] Matt Zucker, James Kuffner, and J Andrew Bagnell. Adaptive workspace biasing for sampling-based planners. In *2008 IEEE International Conference on Robotics and Automation*, pages 3757–3762. IEEE, 2008.

APPENDIX

A. Details

We calculate the following distribution of labels in our simulated training data (higher class values correspond to more frequently visited regions):

Class Distribution			
Class	1	2	3
Percentage	82.5%	14%	3.5%

We use the inverse of the frequency of each class as its weight. Below are the weight values calculated from the distribution of the training data:

Class Weights					
Class	-1	0	1	2	3
Weight	0	0.006	0.033	0.194	0.767

B. Discrete Training Environments

(a) Merom 0

(b) Pomaria 0

(c) Rs

(d) Benevolence 0

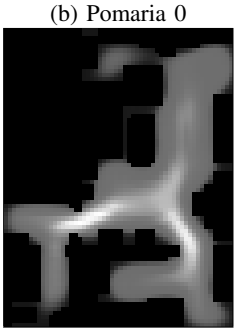
(e) Ihlen 0

Fig. 4: Segmentation masks used as training data.

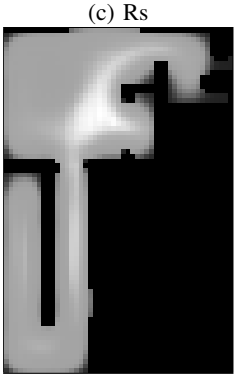
C. Criticality Predictions for Training Environments

(a) Merom 0

(b) Pomaria 0



(c) Rs



(d) Benevolence 0

(e) Ihlen 0

Fig. 5: Criticality predictions of training environments.

D. Map Annotation Ablation

a) Proposed Strategy b) Large Stride c) Per Pixel

Fig. 6: Ihlen 0 environment annotated with a) the proposed annotation strategy b) the strategy with a large stride/no overlaps, and c) a classification, prediction per pixel, strategy

Fig. 6a depicts the output of our strategy. Fig. 6b shows an output with large stride and no overlap that contains artifacts that stem from not averaging over predictions from multiple windows. There are noticeable differences around the boundary of the sliding window. This is because the window lacks the topological context that is beyond its boundary. On the other hand, Fig. 6c shows the output when the model only makes a single prediction per pixel during the sliding window annotation, practically approaching the annotation as classification rather than segmentation.

E. U-Net Details

The details of our model are as follows:

- Input: 64^2 pixel local image (6.4^2 meters)
- Architecture: $32^2 \times 3^2$ conv - max pooling - $64^2 \times 3^2$ conv - max pooling - $128^2 \times 3^2$ conv - max pooling - $128^2 \times 3^2$ conv - upsampling - $64^2 \times 3^2$ conv - upsampling - $32^2 \times 3^2$ - (with residuals across all resolutions)
- Data: 5 environments, 37240 samples
- Learning rate: 0.01
- Number of epochs: 30

F. Criticality Predictions for Test Environments

a) Wainscott 1 b) Beechwood 1 c) Ihlen 1

Fig. 7: Test environments annotated with criticality predictions.