Learning Human-Aware Path Planning with Fully Convolutional Networks*

Noé Pérez-Higueras[1], Fernando Caballero[1] and Luis Merino[1]

Abstract— This work presents an approach to learn path planning for robot social navigation by demonstration. We make use of Fully Convolutional Neural Networks (FCNs) to learn from expert's path demonstrations a map that marks a feasible path to the goal as a classification problem. The use of FCNs allows us to overcome the problem of manually designing/identifying the cost-map and relevant features for the task of robot navigation. The method makes use of optimal Rapidly-exploring Random Tree planner (RRT*) to overcome eventual errors in the path prediction; the FCNs prediction is used as cost-map and also to partially bias the sampling of the configuration space, leading the planner to behave similarly to the learned expert behavior. The approach is evaluated in experiments with real trajectories and compared with Inverse Reinforcement Learning algorithms that use RRT* as underlying planner.

I. INTRODUCTION

Nowadays, mobile robots are becoming part of our daily lives. Robots must be prepared for sharing the space with humans in their operational environments. Therefore, the comfort of the people and the human social conventions must be respected by the robot when is navigating in the scenario. Besides, the robots have to be coherent and maintain the legibility of their actions [1]. This is called human-aware navigation.

First approaches for including human-awareness into robot navigation were based on hard-coding some specially-designed constraints in the motion planners in order to modify the robot behavior in the presence of people [2], [3]. However, the task of "social" navigation is hard to define mathematically but easier to demonstrate. Thus, a learning approach from data seems to be more appropriate [4].

In the last years, several contributions have been presented regarding the application of learning from demonstrations to the problem of human-aware navigation [5], [4]. One successful technique to do that is Inverse Reinforcement Learning (IRL) [6]: the observations of an expert demonstrating the task are used to recover the reward (or cost) function the demonstrator was attempting to maximize (minimize). Then, the reward can be used to obtain a similar robot policy.

Several IRL approaches for the robot navigation task can be found in the literature. In [7] a experimental comparison of different IRL approaches is presented. Also in [8], IRL

is applied to transfer some human navigation characteristics into a robot local planner. A similar work is done in [9], where the densities and velocities of pedestrians observed via RGB-D sensors are used as features in a reward function learned from demonstrations for local robot path planning.

Most of the IRL approaches frame the problem as a Markov Decision Process (MDP) where the goal is to learn the reward function of the MDP. The use of MDPs presents some drawbacks, as the difficulty of solving the MDP at each learning step in large state spaces. This limits the applicability of most of these approaches to small states spaces. Some authors have tackled these MDP limitations from different points of view. A graph-based approximation is employed in [10] to represent the underlying MDP in a socially normative robot navigation behavior task. Another example is [11] where IRL is used to learn different driving styles for autonomous vehicles using time-continuous splines for trajectory representation. In [12] the cooperative navigation behavior of humans is learned in terms of mixture distributions using Hamiltonian Markov chain Monte Carlo sampling. Other authors have tried to replace the MDP by a motion planner. In [13], an approach based on Maximum Entropy IRL [14] is employed to learn the cost function of a RRT* path planner. Also in [15] an adaptation of the Maximum Margin Planning approach [16] to work with RRT* planner is presented.

Moreover, the IRL techniques present other problems when the task to be learned is complex, like the human-aware navigation task. In these cases we have to manually design the structure of the reward and features involved in the task. In many cases, the designed reward probably is not able to recover the underlying demonstrated behavior properly. Even though the weights of the cost function can be learned from expert demonstrations, we still need to define a set of features functions that describe the space states and the relation between them. While for small problems this is doable, for many realistic and more complex problems, like human-aware navigation, it is hard to determine.

However, in the last years, the use of deep neural networks in IRL problems is bringing good results. The neural networks, as a function approximators, permit the representation of nonlinear reward structures in domains with high dimensionality. Some authors have already applied deep networks to the problem of human-aware robot navigation. In [17], a Reinforcement Learning approach is applied to develop a socially-aware collision avoidance system where a deep network is employed to learn multi-agent crossing trajectories. Also, Deep Q-networks have been used in [18],

[1]Noé Pérez-Higueras, Fernando Caballero and Luis Merino are with School of Engineering, Universidad Pablo de Olavide, Crta. Utrera km 1, Seville, Spain `noeperez@upo.es, fcaballero@us.es, lmercab@upo.es`

to solve the MDP step of an IRL algorithm for the task of driving a car. Furthermore, the well-known Maximum Entropy IRL [14] algorithm is extended in [19] to use deep networks, and its application to path planning in urban environments is presented in [20].

In this work we consider a different approach. We propose a learning from demonstration strategy for the task of human-aware path planning that avoids the explicit representation and definition of the cost function and its associated features. The proposed method does not follow the classical IRL approach. Instead, the problem is formulated as a classification task where a Fully Convolutional Network (FCN) is used to learn to plan a path to the goal in the local area of the robot in a supervised way, using the demonstrations as labels. This general approach in then combined with an optimal path planner to solve efficiently the task and to ensure collision-free paths. Thus, the contribution of this paper is twofold: 1) a novel approach for learning human-aware path planning from demonstrations based on Fully Convolutional Networks and 2) the combination of this information with an RRT* planner to enhance the planner capabilities while it behaves similarly to the expert behavior.

The paper is structured as follows. Section II describes the methodology followed for learning the task of human-aware path planning proposed. Then, Section III shows the results of experiments for validation of the deep learning approach, and Section IV presents a set of experiments including a comparison with other learning algorithms in realistic situations. Finally, Section V summarizes the paper contribution and outlooks future work.

II. LEARNING TO PLAN PATHS WITH FCNs

In this section, we present our approach to learn from demonstrations to plan human-aware paths in 2D environments for robot social navigation. We propose using a Fully Convolutional Network to learn, from the expert's path demonstrations, to predict the correct path to the goal according to the actual obstacles and people in the vicinity of the robot, without explicitly defining the environment features used. Unlike Deep IRL approaches, that try to learn the cost function that the expert is following in a MDP framework, our goal is to directly predict the path to the goal given the information from obstacles and persons.

The path predicted by the FCN is then combined with a RRT* planner in order to perform the navigation task efficiently and to prevent the robot for collisions or prediction fails.

A. Input and output of the network

We consider a local navigation task in a 2D space centered in the robot of size $10 \times 10 \ m^2$. Thus, the sensor data together with the detected people position and orientation is used as input to the network. No other information is employed, so the network has to derive the features of the task based on the provided state information.

The sensor data based on laser scans and point-clouds is projected in a 2D grid of $200 \times 200 \ pixels$ with resolution
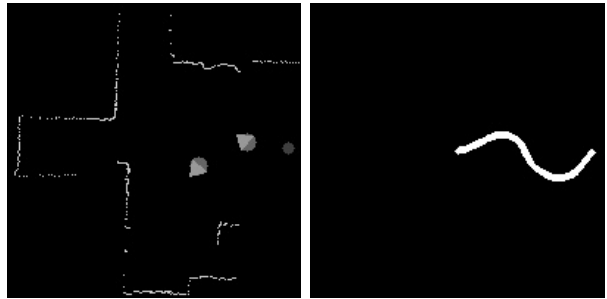

Fig. 1: Left. Example of input grid to the network. Obstacles, goal position and people positions and orientations are included. Right. Respective label used for learning with the path to the goal.

of $0.05 \ m/pixel$. As we are considering the task of social navigation, we need to be able to detect people. So, a people detection system is employed to provide information about the position an approximate orientation in the scenario. This data is also included in the 2D grid, where the people are marked using circles and triangles to indicate the orientation. The goal is also marked on the grid as a small circle. This grid is used as a gray-scale image where the background color is black and each element previously described takes a different gray intensity. These values are normalized in order to be in the range $[0, 1]$ before serving as input to the FCN. An example of the input gray-scale image of the network can be seen on the left image of the Fig. 1.

The output of the network is an approximate path from the center (robot position) to the goal marked on a grid, which has the same size as the input. We can see an example on the right image of the Fig. 1.

Finally, the path prediction from the network and the resulting RRT* path for the same example presented in Fig. 1 are shown in Fig. 2. As can be seen, the resulting path is very similar to the demonstrated one.

B. Network architecture

Path planning problems are specially characterized by the existence of a goal destination in its formulation. This information is critical to optimize the path searching policies. The robot needs to know where to go prior to plan the best trajectory from its current position.

The importance of the goal destination has been considered into the design of the proposed network architecture, presented in Fig. 3. The architecture is inspired by the global-coarse to local-fine deep learning architecture presented in [21]. Thus, the proposed structure is divided into two major branches: a global-coarse estimation that sequentially subsample the input grid while applying larger kernels in order to extract global high level features of the input, and the local-fine branch that make use of the extracted global features and the original input grid to build the final path considering local information and global constraints.

Notice how the proposed network does not make use of fully connected layers, which significantly reduces the total
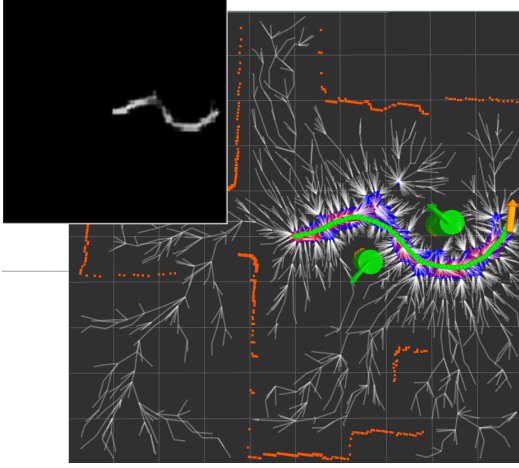
Fig. 2: Upper left. Example of output plan of the network. Down right. RRT* path using the network prediction with a bias of the sampling of 70 %. Robot position is the center of the image. Orange lines shows the obstacles. People position and orientation are indicated with green cylinders and arrows. The goal is represented with the golden arrow. The planner tree is drawn in white color.
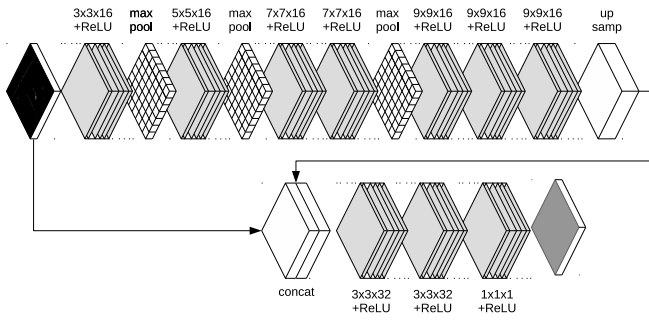

Fig. 3: Proposed network architecture. The network is composed by two branches, a global-coarse branch that extract high level features of the input grid and a local-fine branch that make use of these features and the original input grid to generate the final path.

number of parameters to one hundred thousand approximately. Instead, an output layer with 1x1 kernel size is used to generate the output grid with the path.

C. Integration with the RRT* planner

The final final step consist in combining the path predicted by the FCN with a RRT* planner in order to perform the navigation task efficiently and to prevent the robot for collisions or prediction fails.

The planner uses the path prediction in two ways: first, as a cost function to connect the tree nodes; and second, to partially bias the sampling of the state space, taking a higher percentage of samples from the areas where the plan is. This allows to guide the sampling process efficiently to areas of interest and thus, reaching an optimal path faster.

The percentage of samples drawn from the path prediction or from a uniform distribution is an important parameter.

TABLE I: Mean Squared Error (MSE) reached in the different stages with different set of trajectories

	Learning set	Validation set	Testing set
MSE	0.0073	0.0067	0.0094

Keeping a percentage of uniform sampling allows the planner to be still able to find a path to the goal when the prediction is not complete or fails. In this work, we have used a 70 % of samples from the path prediction.

III. DEEP NETWORK VALIDATION

We first validate the deep learning approach by testing the performance of the network prediction. A dataset of 10500 trajectories has been used to validate the capability of the proposed FCN to learn to plan paths. The dataset has been created by randomly generating positions and orientations for the robot, the goal and for the people around in different scenarios within a large map.

Regarding the generation of demonstration paths to the goals in the scenarios, a RRT* planner with a pre-defined cost function has been employed. In particular, a weighted linear combination of five features is used as cost function similarly to the one employed in [13] for robot social navigation. These features are based on distance metrics as the Euclidean distance to the goal and to the closest obstacle. Also metrics regarding the people in the vicinity of the robot are taken into account. The Euclidean distances and orientations with respect to the people in the scene are used through three Proxemics functions placed in front, back and sides of each person.

From the dataset, 10000 trajectories are used in the learning process, while the remaining 500 trajectories are reserved for testing the network after learning finishes. Also, during the learning, a number of 100 trajectories were taken from the learning set as a validation set for overfitting checking. The results in terms of Mean Squared Error (MSE) for the different sets of trajectories, are presented in Table I. As can be seen, the error committed in the testing set keeps low regarding the error reached in the learning set.

Furthermore, a visual comparison of some network predictions corresponding to some of the 500 trajectories of the testing set are presented in Fig. 4. The images are presented in pairs with the demonstrated path in red in the left image, and the respective prediction on the right image in green. As can be observed, the predictions fit the expert's paths very well.

An interesting and unexpected outcome of the proposed network is its capability to generate more than one valid path when the goal can be similarly reached by following different homotopies, even when the learning has been made with a single trajectory per example. Figure 5 shows two examples of prediction in two valid homotopies. This information is easily handled by the RRT* planner which will select the shorter path.
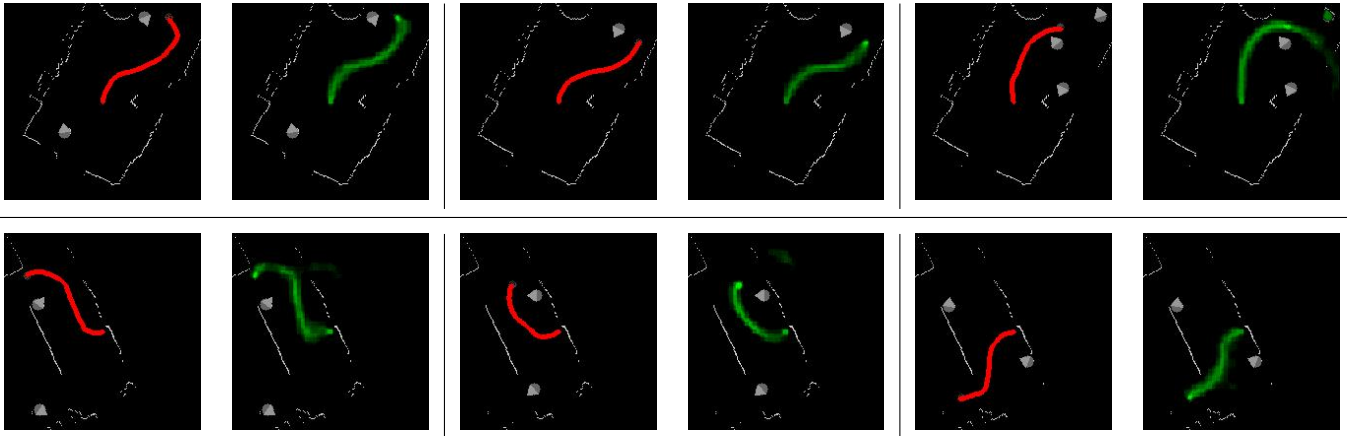
Fig. 4: Visual comparison of 6 pairs of labels and network predictions from the testing dataset. For each pair, the label is represented in red line on the left image, and the prediction is shown in green color on the right image.
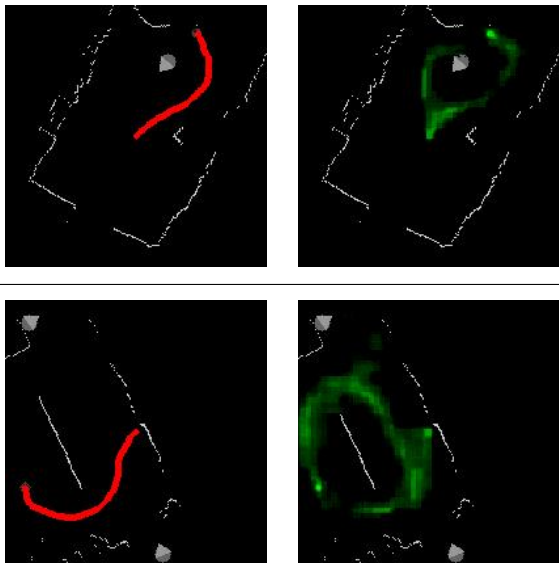

Fig. 5: Example of two cases where the network learn to plan in different valid homotopies. The demonstrated path is shown in red color on the left images while the respective prediction is shown in green color on the right side images.

The implementation code of the network and all the datasets used in the experiments can be found in the Github repository of the UPO Service Robotics Lab, in the module *upo_fcn_learning*[1].

IV. PATH PLANNING EVALUATION

The feasibility of the proposed approach is demonstrated for human-aware planning by learning in a small dataset of real robot trajectories. Then we compare the resulting trajectories with a ground-truth set and two IRL algorithms of the state of the art that learn the cost function of a RRT* planner as a weighted linear combination of features.

[1]https://github.com/robotics-upo/upo_fcn_learning

A. Dataset

A dataset of 300 trajectories has been recorded in different controlled scenarios with static people around and where the robot was remotely controlled by a human expert who tried to perform a correct human-aware navigation to the goal.

This dataset is sorted in 3 sets, where 250 demonstration paths are used for learning and 50 are used for testing the results. The difference between sets is that the 50 paths chosen for testing (and thus the 250 for demonstration) are different for each set.

Is noteworthy that, in terms of deep learning with FCNs, a dataset of 250 demonstrations for learning human-aware navigation can be considered very small. We will show that even with this small information the presented approach is able to equal or overcome the result of two state-or-the-art IRL algorithms.

B. State-of-the-art algorithms

The performance of our approach is tested against two IRL algorithms of the state of the art: RTIRL [13] and RLT [15]. Both try to learn the weights of a linear combination of features used as cost function of a RRT* planner using the same set of demonstrated trajectories. The first one is based on a Maximum Entropy approach [14] that replace the MDP by an RRT* planner; while the second one is an adaptation of the Maximum Margin Planning algorithm (MMP) [16] to work with a RRT* planner.

We have used with these algorithms the extended set of features specifically designed for robot social navigation in [13]. The set is a compound by five features. Three of them are based on distances and orientations between the robot and the people in the scene and are coded through three Gaussian functions placed in front, back and side of each person in the scene. Then, two more features measuring the distance to the goal and distance to the closest obstacle are also considered. It is important to remark that our approach, on the contrary, is fed with just the raw laser data and the position and orientation of people in the form of an image, as explained in Section II-A.

The implementation of the IRL algorithms used for comparison can be found in the module *upo_nav_irl*[2] of the Github repository from the UPO Service Robotics Lab.

C. Metrics

In order to compare the planned paths with the ground-truth paths, we use a metric based on a directed distance measure between two paths:

$$D(\zeta_1, \zeta_2) = \frac{\sum_{i=1}^{N} d(\zeta_1(i), \zeta_2)}{N} \qquad (1)$$

where the function $d(\zeta_1(i), \zeta_2)$ calculates the Euclidean distance between the point i of path ζ_1 and its closest point on the path ζ_2, and N stands for the number of points of the path ζ_1. This distance in then combined to obtain a final metric of path comparison μ:

$$\mu(\zeta_1, \zeta_2) = \frac{D(\zeta_1, \zeta_2) + D(\zeta_2, \zeta_1)}{2} \qquad (2)$$

Moreover, a comparison between the difference in the feature counts of the ground-truth paths and the planned paths is also employed, as the feature counts play a key role in the IRL approaches [13], [15]. The feature count of a path is defined in Equation IV-C, where $f(\zeta(i))$ indicates the vector of features values for point i of path ζ, and $\|\zeta(i+1) - \zeta(i)\|$ is the Euclidean distance between the points i and $i + 1$ of the path ζ. Even though our FCN approach does not make use of such features, we also compute their counts for the resultant trajectories.

$$F(\zeta) = \frac{\sum_{i=1}^{N-1} f(\zeta(i)) + f(\zeta(i+1))}{2} \|\zeta(i+1) - \zeta(i)\| \quad (3)$$

D. Comparative results

We first train the RLT, RTIRL and FCN approaches with the learning set, and then compare the results using the testing set. We obtain the results for the three different data combinations described above.

First, the average values of the distance metric (2) between the planned paths and the 50 ground truth trajectories of each testing set are shown in Fig. 6. As can be seen, the performance of the three approaches is very similar. Even though no further information is provided to the FCN approach, it is able to learn an adequate representation of the task, and equals the results obtained by the other two algorithms that use a pre-defined set of features engineered for human-aware robot navigation.

Figure 7 shows a comparison of the cumulative density functions of the distance metric μ for the different approaches in the three sets of trajectories for evaluation.
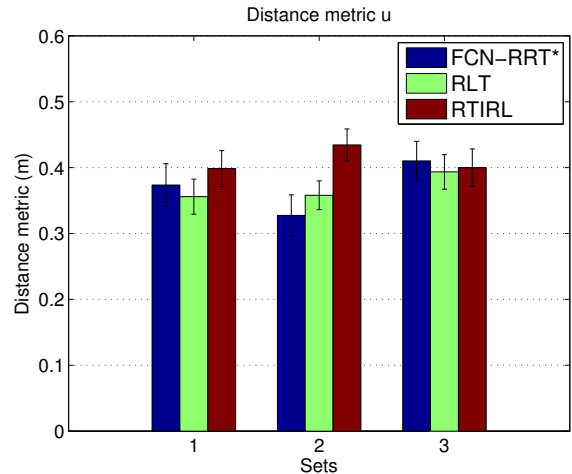
Fig. 6: Average distance metric μ and standard errors for the three approaches in the three sets of trajectories of evaluation.

Again, the performance of the different approaches is very similar to our approach.

Finally, we make a comparison in terms of the manually-designed features. We compare the differences in the feature counts of the planned paths with respect to the ground-truth paths of the three evaluation sets. Figure 8 shows the cumulative density functions of the average feature count difference. It is interesting to see how the RTIRL and RLT obtain similar results while our method under scores in two of the sets, but the resulting paths are very similar according to Fig. 7. This is an expected result given that the navigation features have not been specified to our approach, so the FCN could likely converge to another set of features that also leads to good imitation of the demonstration trajectories.

V. CONCLUSIONS AND FUTURE WORK

This paper presented an approach for learning human-aware path planning based on the integration of FCNs and RRT*. The introduction of FCNs to learn the path planning based on demonstration allows to address the problem without the need of hand-crafted social navigation features as many other IRL approaches in the state of the art. Additionally, the integration of the predicted path with RRT* guarantees an optimal feasible path no matter the situation.

The full approach has been tested with a set of real trajectories and benchmarked with state-of-the-art algorithms for human-aware navigation learning with good results. Our approach offers very similar metrics without defining the navigation features. In addition, the neural network prediction has been also successfully tested with a large dataset in order to validate the predicted paths.

Future work will consider comparing the results with different network architectures in order to benchmark our network with respect existing solutions. Also, learning in dynamic scenarios will be considered, so that the approach can exploit people trajectories to improve the quality of the path planning in such scenarios.
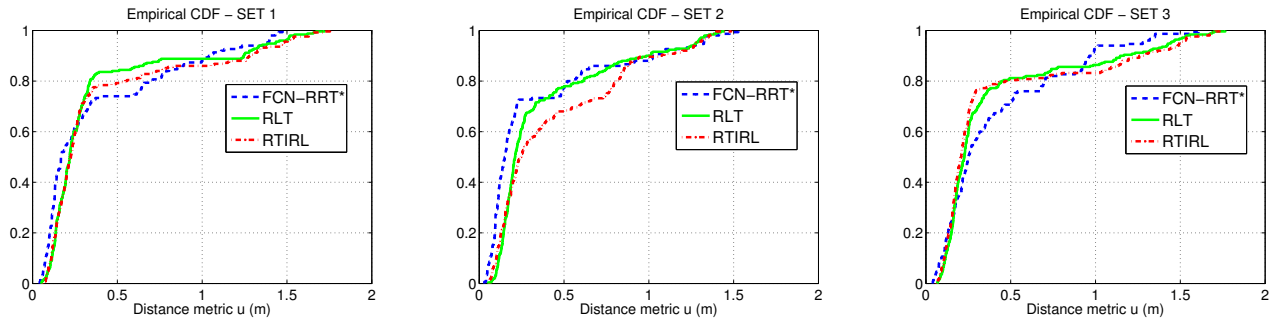
Fig. 7: Cumulative error in the distance metric μ for the different approaches in the three sets of trajectories for evaluation.
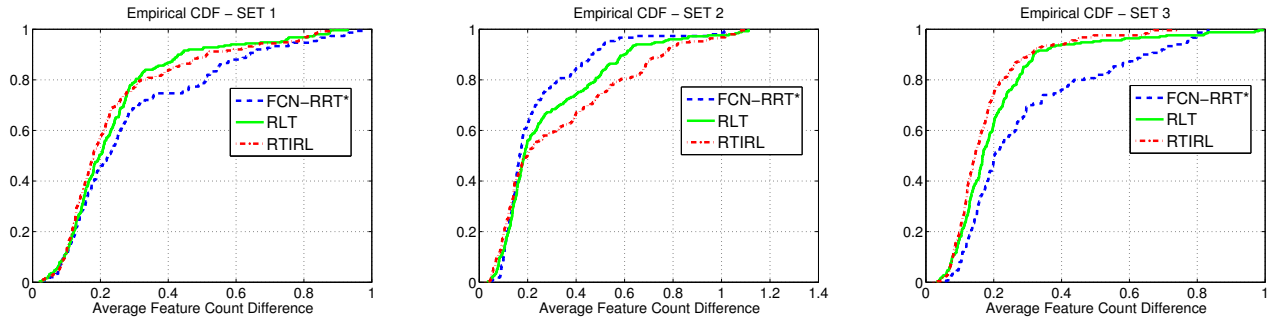

Fig. 8: Cumulative error in the average feature count difference for the 3 approaches in the 3 sets of trajectories for evaluation.

REFERENCES

[1] 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.

[2] R. Kirby, R. G. Simmons, and J. Forlizzi, "Companion: A constraint-optimizing method for person-acceptable navigation." in *RO-MAN*. IEEE, 2009, pp. 607–612.

[3] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Siméon, "A Human Aware Mobile Robot Motion Planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[4] M. Luber, L. Spinello, J. Silva, and K. O. Arras, "Socially-aware robot navigation: A learning approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 902–907.

[5] B. Argali, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstrations," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, 2009.

[6] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 663–670.

[7] D. Vasquez, B. Okal, and K. O. Arras, "Inverse Reinforcement Learning Algorithms and Features for Robot Navigation in Crowds: an experimental comparison," *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS), 2014*, pp. 1341–1346, 2014.

[8] R. Ramón-Vigo, N. Pérez-Higueras, F. Caballero, and L. Merino, "Transferring human navigation behaviors into a robot local planner," in *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*, 2014.

[9] B. Kim and J. Pineau, "Socially Adaptive Path Planning in Human Environments Using Inverse Reinforcement Learning," *International Journal of Social Robotics*, vol. 8, no. 1, pp. 51–66, 2016.

[10] B. Okal and K. O. Arras, "Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning," in *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, 2016, pp. 2889–2895.

[11] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA), Seattle, USA*, vol. 134, 2015.

[12] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, 2016.

[13] N. Pérez-Higueras, F. Caballero, and L. Merino, "Learning robot navigation behaviors by demonstration using a rrt* planner," in *International Conference on Social Robotics*. Springer International Publishing, 2016, pp. 1–10.

[14] B. Ziebart, A. Maas, J. Bagnell, and A. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2008.

[15] K. Shiarlis, J. Messias, and S. Whiteson, "Rapidly exploring learning trees," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore: IEEE, May 2017.

[16] N. D. Ratliff, J. A. Bagnell, and M. a. Zinkevich, "Maximum margin planning," *International conference on Machine learning - ICML '06*, no. 23, pp. 729–736, 2006.

[17] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," *CoRR*, vol. abs/1703.08862, 2017.

[18] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to Drive using Inverse Reinforcement Learning and Deep Q-Networks," in *NIPS workshop on Deep Learning for Action and Interaction*, 2016.

[19] M. Wulfmeier, P. Ondruska, and I. Posner, "Deep inverse reinforcement learning," *CoRR*, vol. abs/1507.04888, 2015.

[20] M. Wulfmeier, D. Z. Wang, and I. Posner, " Watch This: Scalable Cost-Function Learning for Path Planning in Urban Environments ," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[21] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2366–2374.