# Learning Social Navigation from Demonstrations with Deep Neural Networks

Yigit YILDIRIM, Emre UGUR

Computer Engineering Department, Bogazici University, Istanbul, Turkey

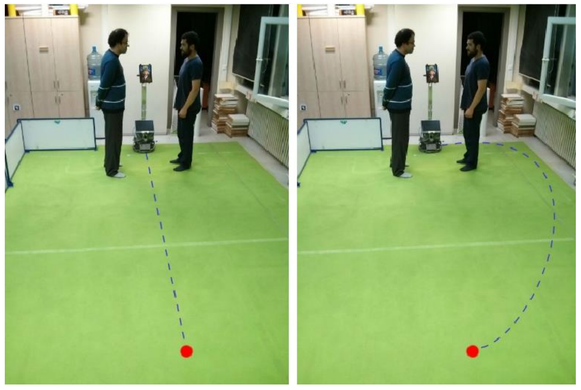
## Abstract

Traditional path planning techniques treat humans as obstacles. This has changed since robots started to enter human environments. On modern robots, social navigation has become an important aspect of navigation systems. To use learning-based techniques to achieve social navigation, a powerful framework that is capable of representing complex functions with as few data as possible is required. In this study, we benefited from recent advances in deep learning at both global and local planning levels to achieve human-aware navigation on a simulated robot. Two distinct deep models are trained with respective objectives: one for global planning and one for local planning. These models are then employed in the simulated robot. In the end, it has been shown that our model can successfully carry out both global and local planning tasks. We have shown that our system could generate paths that successfully reach targets while avoiding obstacles with better performance compared to feed-forward neural networks.

## Introduction

For decades, the **safety** and the **robustness** of the navigation have been the most important factors of navigation systems. **Human-aware** or **social** navigation systems also strive for the **comprehensibility** of the motions.

- Human-aware robot navigation aims to **comply with** the **social norms** of the people.
- Generated motion may be **suboptimal** in terms of **time** or **distance**.
- Objective is to reach to the destination while creating the **minimum possible disturbance** for the people around.



*Comparison between regular and social navigation.*

Efforts to create social navigation have been focusing on the **local planners**. Given the information about the positions of the people in the scene, we can create **human-aware global plans** to **increase** the **comfort** even further.

**Conditional Neural Processes (CNPs)** are used in this study to create a local and a global planner. CNPs

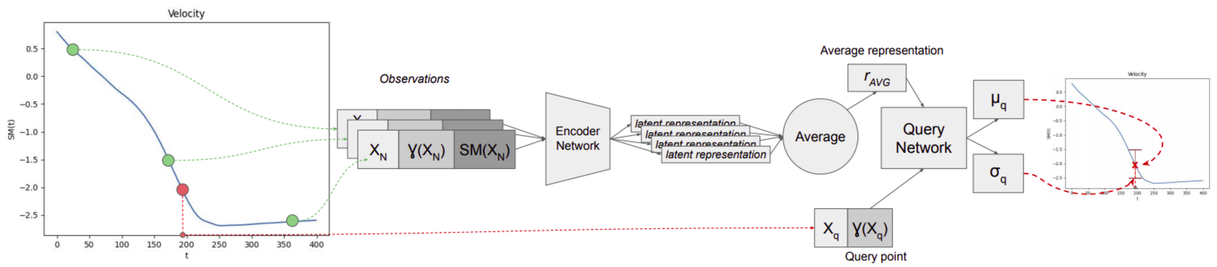
1. Extract the prior knowledge directly from the training data.
2. Learn complex temporal relations in connection with external parameters and goals.
3. Predict a conditional distribution over any other target points.

## Method

The standard hierarchical path planning pipeline is composed of 2 separate parts: global planning and local planning

- Global planner creates coarse paths from the source to the destination.
- Local planner generates motion commands to follow the created paths.

We suggest employing a variant of CNPs individually for the two parts. For each demonstration trajectory $D_i = (X_t, \gamma(X_t), SM(X_t))_t$, where $X$ is the state variable, $\gamma(X)$ is a function representing task parameters and $SM(X)$ is the sensorimotor function to be learned. The representation of the data is different for each CNP but the general layout is the same, as follows.




*General layout of the training phase of our model.*

Global Planner:

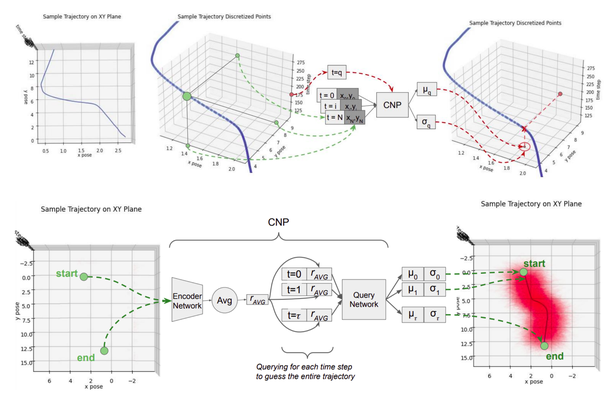
To show the path planning capability of our method, the model is fed with the entire trajectories of positions of the robot and trained on these demonstrations. The representation of the data is as follows:

$X = \text{time\_step}$
$\gamma(X) = (\text{start}_x, \text{start}_y, \text{goal}_x, \text{goal}_y, \text{obstacle\_pose}_x, \text{obstacle\_pose}_y)$
$SM(X) = (\text{position}_x, \text{position}_y)$

Local Planner:

Conditioned on the starting and destination poses, the use of the task parameter $\gamma(X)$ gave the model the ability to reactively change the velocity commands with respect to changing obstacle positions.

$X = (\text{distance\_to\_goal}_x, \text{distance\_to\_goal}_y)$
$\gamma(X) = (\text{distance\_to\_obs}_x, \text{distance\_to\_obs}_y)$
$SM(X) = (\text{velocity}_x, \text{velocity}_y)$



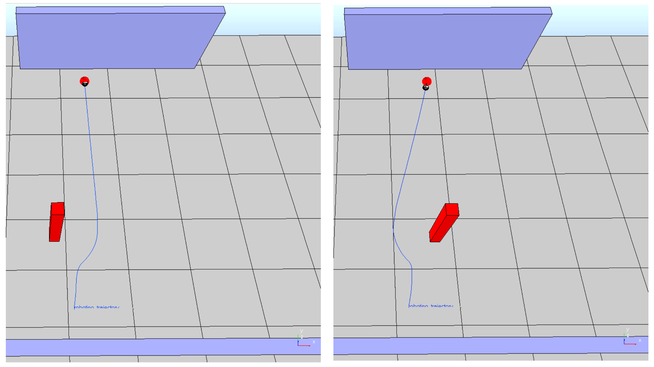

*CNP as the global planner.*

## Experiments

Data Collection:

Our system was verified in CoppeliaSim simulation environment that includes an omnidirectional robot platform (Robotino).

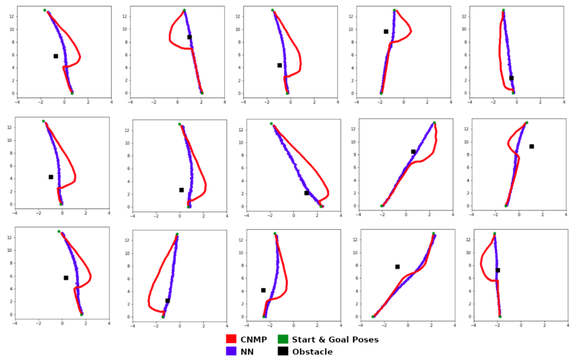
- The Social Force Model is implemented as the local controller of the robot to gather demonstration trajectories.
- 1000 trajectories with randomly different starting, goal and obstacle poses are recorded.
- Single, multiple, stationary and dynamic objects are placed at random positions in each trial.
- These trajectories are processed differently for each task to create the data sets explained above.



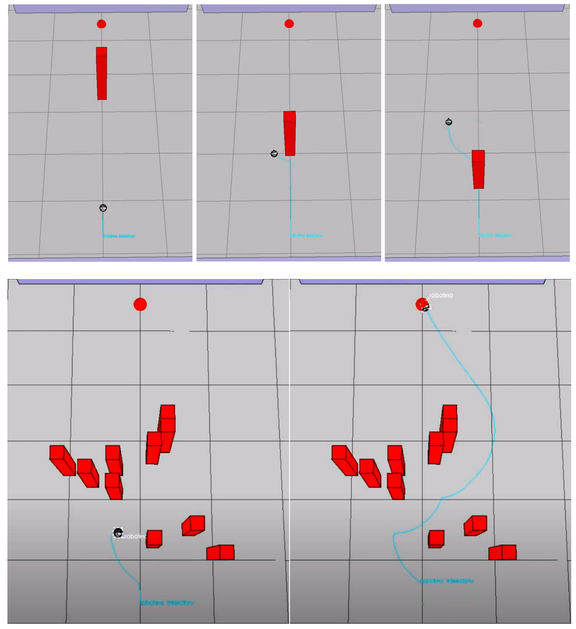
*Data collection on the simulated environment*

## Results

Obstacle avoidance is an important skill for global planners. To show the strength of CNPs over standard neural networks, we compare their performance on the trajectory planning task.




To show that CNP can generate motion commands that can successfully steer the robot among moving or stationary obstacles, we conducted tests on different scenarios.



*Performance of CNP as the local planner.*

## Conclusion

In this study, the preliminary results of our framework which is a hierarchical framework that is built on top of CNPs is presented. We showed that our model can generate reasonable paths at both global and local levels while avoiding obstacles.

We plan to extend this study with actual human data and test the resulting planners on a real robot.