Joint Long-Term Prediction of Human Motion Using a Planning-Based Social Force Approach

Andrey Rudenko, Luigi Palmieri and Kai O. Arras

***Abstract*— The ability to perceive and predict future positions of dynamic objects is essential for mobile robots and intelligent vehicles in dynamic environments. In this paper, we present a novel planning-based approach for long-term human motion prediction that accounts for local interactions and can accurately predict joint motion of multiple agents. Long-term predictions are handled using an MDP formulation that computes a set of stochastic motion policies. To obtain distributions over future motion trajectories, we sample the policies with a weighted random walk algorithm in which each person is locally influenced by social forces from other nearby agents. Unlike related work, the algorithm is environment-aware, can account for individual agent velocities, requires no training phase and makes joint predictions for multiple agents. Experiments in simulation and with real data show that our method makes more accurate predictions than two state-of-the-art methods in terms of probabilistic and geometrical performance measures.**

Fig. 1. Prediction results for four persons in the ATC shopping center dataset, obtained with our algorithm and shown in individual colors. The current position of each person is indicated by a **yellow** circle, ground truth trajectories are shown in **white**.

I. Introduction

Long-term prediction of human motion is an important task for applications such as robot navigation in crowded environments, autonomous driving, video surveillance or human-robot collaboration. Particularly for mobile robots operating among humans, predicting future trajectories of surrounding people over longer periods of time has the potential to significantly improve motion planning, tracking, behavior recognition, or human-robot interaction. Early predictions can make robots less reactive to irrelevant changes in their surrounding, can render robot motion more legible to humans, may improve the robot's efficiency in terms of time to goal, or allow for better planning of traffic maneuvers such as merging or overtaking.

The challenge of making accurate long-term predictions of human motion arises from the complexity of human behavior that may be influenced by other people, the variability of social relations to them, by the environment, its affordances and semantics, or by social rules and norms. Prior art has addressed this challenge using model-based, learning-based and planning-based approaches, considering the single-agent case, ignoring other agents, or the multi-agent case in which predictions are made jointly.

In this paper we present a novel planning-based approach that accounts for local social interactions to accurately predict motion of multiple agents jointly and in real time. Planning-based prediction methods make the assumption that humans essentially behave like planners by finding an (near-) optimal path through the environment. Our method extends the state-of-the-art by a novel MDP formulation of the joint motion prediction problem using *joint stochastic policy sampling* to produce such motion strategies. Those strategies are then adapted locally by social forces in order to account for the motion of other agents. We also propose a method for performing predictive robot motion planning under the obtained policies. Experiments on real-world datasets show that our method can accurately predict long-term trajectories of people involved in socially interactive tasks in real-time, outperforming two relevant state-of-the-art methods. A simpler, single-agent version of this method has been published in [1].

The paper is structured as follows: in Sec. II we review related work, followed by Sec. III in which we describe our approach. Experiments and results are given in Sec. IV and Sec. V, respectively, and Sec. VI concludes the paper.

All authors are with Bosch Corporate Research, Stuttgart, Germany. {andrey.rudenko, luigi.palmieri, kaioliver.arras}@de.bosch.com. This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732737 (ILIAD).

II. Related Work

Driven by increasing numbers of robots and intelligent vehicles in human environments, the problem of predicting human motion has received growing attention in recent years. Existing methods can be classified into physics- or model-based approaches, data-driven techniques to learn motion patterns, planning-based approaches, and methods that implicitly learn and solve a decision making problem.

Physics-based approaches forward-simulate a set of dynamics equations to compute pedestrian motion [2], [3], [4], [5]. Among them is the popular social force model [2] which is used for motion prediction by Elfring et al. [5] and in the

context of tracking by Luber et al. [6]. These social force based methods are local and provide valid short-term motion predictions that inherently account for interactions between multiple agents in the scene.

Learning motion patterns is a data-driven approach to long-term motion prediction taken in [7], [8], [9]. Trautman et al. and Vemula et al. [8], [9] consider interactions between multiple pedestrians. They make the assumption that people are involved in "joint collision avoidance" and jointly estimate most likely future trajectories with a non-parametric statistical model based on Gaussian processes. Techniques that learn motion patterns recognize the learned patterns from observed trajectories of people. However, such patterns are spatially grounded in the map and do not generalize to new or changed environments, furthermore they require a lot of data for training. Recently, supervised learning-based prediction techniques that use deep neural networks (e.g [15]) were introduced. They also consider joint predictions, learning complex human behaviors in social scenes.

Planning-based approaches are based on the assumption that humans follow paths through the environment in a goal-directed manner. Ziebart et al. [10] generate goal-directed motion predictions by solving a soft-maximum MDP using maximum entropy inverse optimal control. The similar approach by Kuderer et al. [11] extends the work of Ziebart et al. in the sense that they present a continuous formulation with reward features that better capture physical and topological properties of pedestrian trajectories. As in [8], their approach provides joint predictions. Kitani et al. [12] extend [10] to handle noisy tracker observations and include vision-based physical scene features. Karasev et al. [13] provide an interpretation of models from [10], [12] as jump-Markov processes with the goal represented by a hidden variable. Agent behavior is interpreted as switching nonlinear dynamical systems, with the latent goal variable governing the switches and the policy describing the nonlinear motion dynamics. Vasquez [14] extends the MDP-based approach by enabling a cost-to-go planner to represent the uncertainty of human motion rather than the MDP value function.

In this paper, we combine the ability of planning-based approaches for goal-directed global motion prediction with a physics-based local interaction model using social forces. Unlike learning-based techniques [7], [8], [9], [15], our approach does not require a training phase, and contrary to physics-based approaches [4], [5], our predictions do not get stuck in local minima and are inherently environment-aware. Similarly to [8], [11], [9], [15], we perform prediction of multiple agents jointly. To this end, we introduce a novel random walk technique called *Joint Sampling MDP* (JS-MDP), which reactively adapts predictions to local interactions. Unlike these works, our approach offers obstacle-awareness as well as intent-awareness using a map of the environment. Extending our earlier work [1], in this paper, we add a motion inertia term, a more efficient implementation and the ability to jointly predict multiple agents.

III. Joint Sampling MDP for Motion Prediction

In this section, we present our approach. After introducing the MDP formulation in Sec. III-A, we describe the algorithm to generate predictions from the stochastic policies in Sec. III-B. In Sec. III-C we describe our approach for predictive motion planning under a given probabilistic occupancy map.

A. MDP Formulation

1) Notation: Markov Decision Processes provide a mathematical framework for modeling decision making problems for a discrete-time stochastic control process. Formally, a MDP is described by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ where $\mathcal{S}$ and $\mathcal{A}$ are finite sets of agent *states* and *actions*, respectively. The *transition function* $\mathcal{P}(s, s', a)$ defines the probability of getting to state s' from state s when executing action a. The *reward function* $\mathcal{R}(s, a)$ specifies the immediate reward gained for taking action a in state s. The discount factor γ controls the importance of future rewards relative to immediate rewards. The agent's *policy* $\pi : \mathcal{S} \to \mathcal{A}$ defines the action the agent should take in each state. The *optimal policy* π^*, which maximizes the cumulative expected future rewards (Eq. 2), is obtained alongside with the state and action values, $V^*(s)$ and $Q^*(s, a)$, by solving the recursive Bellman equations (Eq. 1), using e.g. value iteration.

$$\begin{cases} Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s,s',a) V^*(s') \\ V^*(s) = \max_a Q^*(s,a) \end{cases} \tag{1}$$

$$\pi^*(s) = \arg\max_a Q^*(s,a) \tag{2}$$

2) Representation: Agent states are represented by 2D (x, y) Cartesian coordinates, which essentially means that actions only depend on the current position. Goal-directed motion policies are obtained by assigning negative rewards to all states and actions except for the goal state g which is represented as an absorbing zero state, i.e. has only one self-transitioning action with zero reward. For modeling the action space, we assume that human motion is unconstrained in orientation and acceleration and describe actions as $\langle \theta, v \rangle$ orientation-velocity pairs, which reads as "making a move in direction θ with velocity v". We allow v to not exceed a maximum speed $v_{\max}$ (see also Sec. III-B.1 with details on handling speed). We assume deterministic action outcome, i.e. $\forall s \in \mathcal{S},\ \forall a \in \mathcal{A}\ \exists! s' : s \xrightarrow{a} s'$. This assumption, which is also made in [10] and [13], basically implies that humans know where they are going. For convenience we denote the transition function with deterministic action outcomes as $\mathcal{P}(s, a) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. For the start state $s = (s_x, s_y)$ and action $a = (\theta, v)$, transition $s \xrightarrow{a} s'$ is calculated as $s'_x = s_x + v\cos(\theta),\ s'_y = s_y + v\sin(\theta)$.

In a fashion similar to probabilistic localization, we frame the task of predicting a person's future location as estimating the probability $p(s|t)$ that the person will be in state s at time t, $\forall s \in \mathcal{S},\ t_0 < t < T$, where t_0 is the current time and T is the prediction horizon.

We use 2D grid maps of the environment to represent occupied and free space. Thus, $p(s|t = t_i)$ is a probability distribution over the map. A *static costmap* $C(s)$ carries the unitary cost of each state, which is set to 1 for occupied states and to a small value $\epsilon > 0$ for free states. We make the assumption, common to planning-based prediction methods, that goal states are known a-priori or can be learned off- or online. Using the set of goals, denoted as $\mathcal{G}$, we solve $|\mathcal{G}|$ MDP problems with the goal-oriented reward function $\mathcal{R}_g(s,a)$ and compute the optimal policy π_g^* for each goal, as well as the $V_g^*(s)$ and $Q_g^*(s,a)$ value functions. $\mathcal{R}_g(s,a)$ is constructed as follows:

$$\mathcal{R}_g(s,a) = \begin{cases} -w_1 C(s') - w_2 ||s - s'||, \text{ if } s \neq g \\ 0, \text{ otherwise,} \end{cases} \quad (3)$$

where $s' = \mathcal{P}(s,a)$ and $w_1, w_2 > 0$ control the relative importance of each component: the unitary cost of s' and the Euclidean distance $||\cdot||$ covered with the action a. Since the reward function is negative everywhere except the goal state, we solve the MDP with $\gamma = 1$ in Eq. 1. Thus, the $V_g^*(s)$ value of a state is actually the *cost-to-go* from s to g.

To predict also alternative paths to the goal and to allow deviations from the optimal policy, we relax the obtained π_g^* with the *stochastic Boltzmann policy* that assigns to each action a a probability to be executed in a particular state s proportional to its value $Q_g^*(s,a)$, see Eq. 5. Temperature α controls the level of stochasticity, i.e. the likelihood that sub-optimal actions are considered. We denote the stochastic policy as π_g and compute it as in Eq. 5. In order to encourage the agent to perform faster actions with larger v, we modify the action value function $Q_g^*(s,a)$ with an additional weight $w_a \in (0,1)$:

$$\hat{Q}_g^*(s,a) = w_a \mathcal{R}_g(s,a) + V_g^*(s') \quad (4)$$

$$a \sim \pi_g(s) \text{ with prob. } \propto \exp(\alpha(\hat{Q}_g^*(s,a) - V_g^*(s))). \quad (5)$$

B. Joint Human Motion Prediction Using Social Forces

Here we detail our algorithm for jointly predicting trajectories of all agents in the scene. We assume a tracking system that provides short sequences of estimated positions of people, also called tracklets $\mathcal{T}^i = \langle s_1^i, s_2^i, ..., s_{l(i)}^i \rangle$, where $s^i = (x,y)^i$, $i \in [1..N]$, $l(i)$ is the tracklet length, and N is the number of observed people in the scene. This is a weak assumption as advanced tracking systems (such as [16]) provide this information also across occlusions and misdetections using e.g. robust data association techniques. $s_{l(i)}^i = s^i(t_0)$ is the position of person i at the current time t_0 and $\boldsymbol{\mathcal{T}}$ is the set of all observed tracks.

From each tracklet we estimate the observed speed v_{obs}^i, the current orientation θ_{obs}^i and the probability distribution over goals. For each goal $g \in \mathcal{G}$, similarly to [10] and [14], we estimate the gradient of the value function $V_g^*(s)$ along $\mathcal{T}^i$ as the difference between values at s_1^i and $s_{l(i)}^i$ using a softmax function:

$$p(g) \propto \exp\left(\beta(V_g^*(s_{l(i)}^i) - V_g^*(s_1^i))\right). \quad (6)$$

Temperature β defines to what extent alternative, less likely goals are considered. Finally, we obtain a discrete distribution over goals for each person i denoted by $p^i(\mathcal{G})$.

1) Policy Cutting for Speed Selection: In Sec. III-A we train a policy that allows actions up to a given $v_{\max}$. In order to account for the fact that humans move with individual speeds, observed as v_{obs}^i, we could solve an individual MDP problem with $v_{\max} = v_{\text{obs}}^i$ which is, however, computationally costly and prevents motion predictions faster than v_{obs}^i. Thus, for handling individual velocities efficiently, we solve an MDP problem once for all people in the scene with a large $v_{\max}$ to obtain π_g and use a simple *policy cutting* technique to incorporate the information about v_{obs}^i into our prediction algorithm. For each person i, we redefine the action space as $\hat{\mathcal{A}}(v_{\text{obs}}^i) = \langle \theta, v \rangle$ with $\theta \in [0, 2\pi)$ and $v \in [0, 2\, v_{\text{obs}}^i]$. The policy $\hat{\pi}_g^i$ is then defined as

$$a = \langle \theta, v \rangle \sim \hat{\pi}_g^i(s) \text{ with prob. } \propto \begin{cases} \pi_g(\langle \theta, v \rangle), \text{ if } v \leq v_{\text{obs}}^i, \\ \pi_g(\langle \theta, 2\, v_{\text{obs}} - v \rangle), \text{ if } v > v_{\text{obs}}^i. \end{cases} \quad (7)$$

Basically, we assign the same probability to faster actions with $v > v_{\text{obs}}^i$ as to the symmetrically slower actions with $v < v_{\text{obs}}^i$. The original policy π_g is "cut" at the point of v_{obs}^i and "mirrored" backwards, hence the name *policy cutting*.

2) Local Interaction Modeling With Social Forces: The concept of social forces [2] describes how the intended motion of a person changes according to the influence of other people and the environment by superimposing repulsive forces from obstacles and other people with attractive forces to the goal. The approach, initially developed for crowd behavior analysis and egress research, performs well in modeling short-term local influences but performs poorly in making accurate long-term predictions as discussed in Sec. II. Here, the long-term aspects – attraction to the goal and repulsion from obstacles – are handled by the MDP formulation. We only utilize the local influence aspects of the social force model.

Formally, social force $\mathbf{f}_{i,k}^{\text{soc}}$, emitted by person k in the direction of person i is

$$\mathbf{f}_{i,k}^{\text{soc}} = a_k e^{\left(\frac{r_{i,k} - d_{i,k}}{b_k}\right)} \boldsymbol{n}_{i,k} \left(\lambda + (1-\lambda)\frac{1 + \cos(\varphi_{i,k})}{2}\right), \quad (8)$$

where $a_k \geq 0$ specifies the magnitude and $b_k > 0$ the range of the force, $d_{i,k}$ is the distance between people and $r_{i,k}$ is the sum of their radii. The term $\boldsymbol{n}_{i,k}$ is the normalized vector pointing from k to i, which describes the direction of the force. An anisotropic factor, controlled by $\lambda \in [0,1]$, scales the force in the person's direction of motion: the force reaches its full magnitude when the angle $\varphi_{i,k}$ between the intended motion direction of person i and $\mathbf{n}_{k,i}$ is zero, and has no effect when $\varphi_{i,k} = \pi$. The factor postulates that influences in the front of a person are stronger than those to the sides and weak in the back (see also Fig. 2). Social forces on person i are added for all k and used to change the motion direction $\mathbf{F}_i^{\text{pers}}$ which in our case is the action $a = \langle \theta, v \rangle$ sampled from

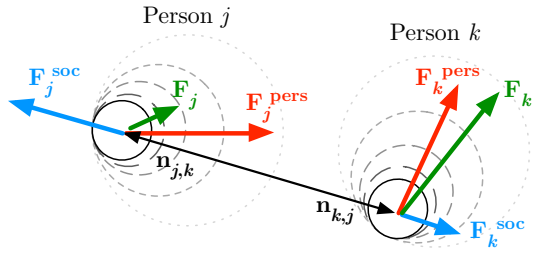

Fig. 2. Example of the anisotropic social force model with $\lambda = 0$. Person k is crossing in front of person j. Intended directions $\mathbf{F}^{\text{pers}}$ are shown with **red** arrows. Person j is influenced by a strong social force $\mathbf{F}^{\text{soc}}$, depicted in **blue**, while the effect on person k is limited due to the anisotropic factor. Resulting directions $\mathbf{F}$ are shown in **green**.

the stochastic policy:

$$\mathbf{F}_i \;=\; \mathbf{F}_i^{\text{pers}} + \mathbf{F}_i^{\text{soc}} \;=\; \mathbf{F}_i^{\text{pers}} + \sum_{k \neq i} \mathbf{f}_{i,k}^{\text{soc}} \,. \tag{9}$$

3) Stochastic Policy Sampling Using Random Walks: To make predictions using the stochastic policy π_g, we propose a random walk algorithm (Alg. 1) that samples K joint paths for all people in the scene, each path starting in the corresponding current state $s^i_{l(i)}$ of person i at time t_0. In each of the K samples we randomly draw a goal $g(i)$ for person i from the distribution $p^i(\mathcal{G})$ and randomly generate actions $a^i = (\theta^i, v^i)$ from the policy corresponding to $g(i)$. This is done by sampling the normalized discrete distribution $\hat{\pi}^i_{g(i)}(s^i_n)$ obtained from Eq. 5. During this random walk, we evaluate social interactions among the agents that affect each agent's instantaneous stochastic policy according to the social force model (see Fig. 3 for illustration). The position of each person at time t is then saved in the corresponding layer L^i_t of the probabilistic occupancy map L, that is shared among the K samples. After K iterations L is normalized $\sum_s L^i_t(s) = 1$ to properly represent the probability distribution $p(s|t = t_i)$ of the person's possible location at time t_i (see also Fig. 4).

To achieve smoother path predictions, we introduce a model-free motion inertia term, parametrized by $I = (I_\theta, I_v)$, that prevents sudden changes in speed and direction between $t-1$ and t. When sampling the stochastic policy π_g, we obtain action $a = (\theta_t, v_t)$ which is then shifted towards the current speed and direction as in Eq. 10. By varying I_θ and I_v, together with the temperature α, we get flexible control of angular and translational variability in a person's goal-directed motion behavior.

$$(\theta_t, v_t) := (1 - I) \cdot (\theta_t, v_t) + I \cdot (\theta_{t-1}, v_{t-1}) \tag{10}$$

Finally, we have Alg. 2 as our method for prediction. Its inputs are the obstacles map M of the environment, the set of goals $\mathcal{G}$, the set of tracklets $\mathcal{T}$, and the prediction horizon T. Its parameters are the cost of free space ϵ, temperatures α and β for the stochasticity level and goal uncertainty, inertia coefficients I_v and I_θ, social force parameters $SF_p = (a_k,\ b_k,\ \lambda)$ and K samples for the stochastic policy sampling. We keep the inertia and social force parameters constant for all people, however, online estimation of their values for individually observed persons is possible in future work.

Algorithm 1 Joint Random Walk Stochastic Policy Sampling

1: **function** JointStochPolicySampling($\mathcal{T}, V^*_g(s), \pi_g(s), K, T$)
2: Compute θ_{obs}, v_{obs}, $p(\mathcal{G})$ for each person using $\mathcal{T}$ and $V^*_g(s)$
3: Initialize $T \times N$ empty layers of the L occupancy map:
4: **for** $t = 1, \ldots, T,\ i = 1, \ldots, N$ **do**
5: $\quad L^i_t \leftarrow \texttt{zeros}(|\mathcal{S}|)$
6: Sample K joint paths for all people:
7: **for** $k = 1, \ldots, K$ **do**
8: $\quad$ For person i set initial state s^i_n, orientation θ^i_n and velocity v^i_n,
9: $\quad$ and sample the goal $g(i)$:
10: $\quad$ **for** $i = 1, \ldots, N$ **do**
11: $\quad\quad (s^i_n,\ \theta^i_n,\ v^i_n) \leftarrow (s^i_{l(i)},\ \theta^i_{\text{obs}},\ v^i_{\text{obs}})$
12: $\quad\quad g(i) \leftarrow \texttt{sample}(p^i(\mathcal{G}))$
13: $\quad$ Jointly predict for T steps ahead:
14: $\quad$ **for** $t = 1, \ldots, T$ **do**
15: $\quad\quad$ **for** $i = 1, \ldots, N$ **do**
16: $\quad\quad\quad$ **repeat**
17: $\quad\quad\quad\quad$ Sample random action a of person i according to $\hat{\pi}^i_{g(i)}$:
18: $\quad\quad\quad\quad (\theta^i_a,\ v^i_a) \leftarrow \texttt{sample}(\hat{\pi}^i_{g(i)}(s^i_n))$
19: $\quad\quad\quad\quad$ Apply inertia given current orientation θ^i_n and velocity v^i_n:
20: $\quad\quad\quad\quad (\theta^i_a,\ v^i_a) \leftarrow (1-I) \cdot (\theta^i_a,\ v^i_a) + I \cdot (\theta^i_n,\ v^i_n)$
21: $\quad\quad\quad\quad$ Calculate state transition of person i executing action a:
22: $\quad\quad\quad\quad s^i_{n+1} \leftarrow \mathcal{P}(s^i_n,\ a)$
23: $\quad\quad\quad\quad$ Social force on person i given current agents' positions s_n:
24: $\quad\quad\quad\quad F_s \leftarrow \texttt{socialForce}(s_n, i)$
25: $\quad\quad\quad\quad$ Modify transition of person i given the current social force:
26: $\quad\quad\quad\quad s^i_{n+1} \leftarrow s^i_{n+1} + F_s$
27: $\quad\quad\quad$ **until** $\texttt{lineOfSight}(s^i_n,\ s^i_{n+1})$
28: $\quad\quad\quad$ Add the next position s^i_{n+1} of person i to the occupancy map:
29: $\quad\quad\quad L^i_t(s^i_{n+1}) \leftarrow L^i_t(s^i_{n+1}) + 1$
30: $\quad\quad$ Update current positions, orientations and velocities of all people:
31: $\quad\quad$ **for** $i = 1, \ldots, N$ **do**
32: $\quad\quad\quad (s^i_n,\ \theta^i_n,\ v^i_n) \leftarrow (s^i_{n+1},\ \theta^i_a,\ v^i_a)$
33: **for** $t = 1, \ldots, T,\ i = 1, \ldots, N$ **do**
34: $\quad L^i_t \leftarrow \texttt{normalize}(L^i_t)$
35: **return** L

Algorithm 2 Motion Prediction

1: Parameters: $\alpha, \beta, K, \epsilon, I, SF_p$
2: Inputs: $M, \mathcal{T}, T, \mathcal{G}$
3: **for all** $g \in \mathcal{G}$ **do**
4: $\quad$ compute $\mathcal{R}_g(s, a)$ as in Eq. 3
5: $\quad V^*_g,\ Q^*_g,\ \pi^*_g \leftarrow \texttt{ValueIteration}(\mathcal{R}_g(s, a))$ as in Eq. 1, 2
6: $\quad$ compute π_g as in Eq. 5
7: $L \leftarrow \texttt{JointStochPolicySampling}(\mathcal{T}, V^*_g(s), \pi_g(s), K, T)$
8: **return** L

Lines 3-6 prepare and solve the MDP. This part of the algorithm can be precomputed offline or updated online at lower frequency since the stochastic policy remains valid as long as the map stays relatively static. Line 7 calls the joint stochastic policy sampling method that computes predictions for all people and returns the occupancy map L. See Fig. 1 for example predictions obtained with our method.

C. Robot Motion Planning Using Predictions

The key idea for incorporating predictions into motion planning is to penalize robot locations that will probably be occupied by other agents at the same time. To this end, we overlay the predicted regions of occupancy L with the gridmap of static obstacles M.

In our previous work [1], we have compared three different predictive planning approaches: spatio-temporal discrete search in a time-augmented state space, used e.g. in [7], *costmap inflation* suggested by Bai et al. [17] and the *infer-*

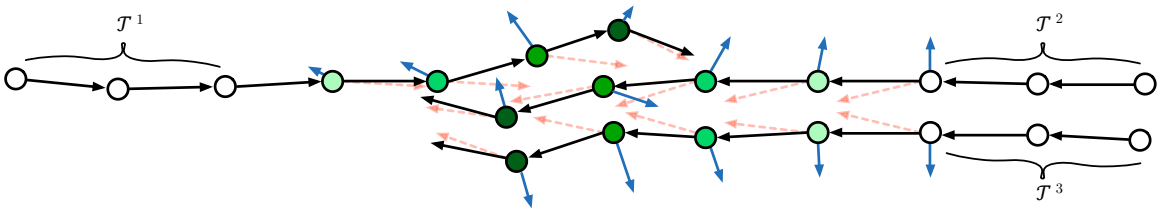

Fig. 3. Social interaction example with our random walk algorithm using social forces. There are two persons side-by-side and a third person moving in opposite direction. Observed tracklets $\mathcal{T}^{1,2,3}$ are shown by **white** circles, jointly predicted future positions are shown by the same **shade of green**. **Red** arrows indicate the sampled action of each person, **blue** arrows show the social force and **black** arrows show the resulting action the person executes.

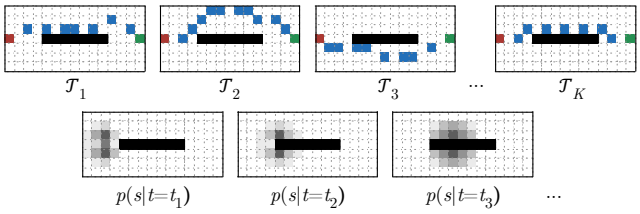

Fig. 4. Illustration of the random walk stochastic policy sampling in a scenario with one person. **Top row:** For each of the K sampled paths $\mathcal{T}_1 \ldots \mathcal{T}_K$, locations at time t_j are added to the corresponding time layer $p(s|t=t_j)$, which is then properly normalized. The path is depicted in **blue**, with **red** being the start s_0 and **green** the goal. **Bottom row:** Probability distributions $p(s|t=t_j)$ are shown by **shades of gray**, darker areas mean higher probabilities.

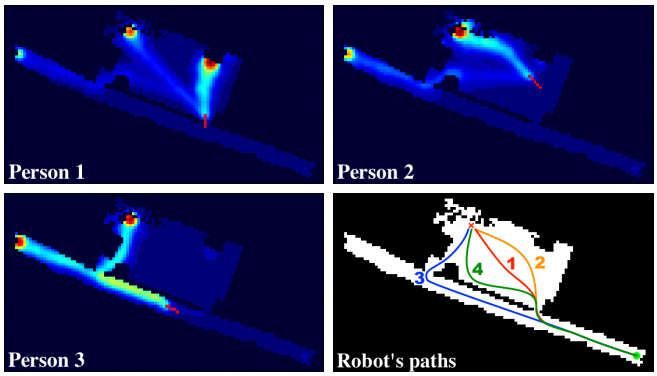

Fig. 5. Predictive planning results. Predictions for three people are shown in **top row** and **bottom left**. Observed tracklets are depicted in **red**, predicted trajectories are represented with **heatmaps** (warmer colors correspond to higher occupancy probability). **Bottom right:** the robot is located in the top left corner of the room, its goal is in the bottom right corridor. Using the *inferring collision points* algorithm, the robot iteratively plans three paths (in **red**, **orange** and **blue**), before it finds the **green** path with no predicted collisions.

ring collision points (ICP) technique by Ziebart et al. [10]. We found the ICP method to be the best compromise between performance and efficiency. The method iteratively shapes a time-independent navigational cost function to remove known points of hindrance. Initialized with the costmap $C = M$, at each iteration, ICP finds the A* solution in C, simulates it forward in time and compares the position of the robot to the corresponding time layer $L_t = \sum_i L_t^i$, inflating the cost of collision regions. See Fig. 5 for examples of predictive planning with the ICP algorithm.

D. Implementation Details

When solving the MDP problem for a goal g, we need to make sure that g is reachable from every free state, i.e. there are no isolated states which may prevent convergence of the value iteration algorithm in the absorbing zero setting. We use a wavefront algorithm starting from the goal state to determine the subset of approachable states, and solve the MDP problem only for those states. To speed-up convergence, we process the states in the order of increasing Manhattan distance to the goal. Moreover, since in our MDP $\forall s, a, a' : \mathcal{P}(s,a) = \mathcal{P}(s,a') = s' \Rightarrow Q^*(s,a) = Q^*(s,a')$, i.e. actions a and a' have the same effect, we iterate directly over target states s' instead of every pair $\langle \theta, v \rangle \in \mathcal{A}$. Starting with undefined state values, we run value iteration until all states are assigned with some value, thus obtaining approximate costs-to-go. In our experiments, value iteration converges to the approximate cost-to-go after only one iteration.

For a fine discretization of action space $\mathcal{A} = \langle \theta, v \rangle$, storing the stochastic policy $\pi_g(s)$ for every state implies significant storage burden. We store the policy in a sparse form, saving only actions with probability higher than the factor of $\frac{1}{|\mathcal{A}|}$. This yields no visible effect on the random walk predictions, but saves up to 10x storage space, depending on the level of stochasticity in the original policy. To reduce the number of samples K, we smooth the layers of L with three iterations of a separable box filter. We found that this leads to very similar distributions compared to those obtained with 10 to 20 times more samples.

IV. Experiments

In this section we evaluate our *Joint Sampling MDP* (JS-MDP) approach both qualitatively and quantitatively and compare it to several baselines. All algorithms are implemented in C++, running on a laptop with a 2.8 GHz Xeon processor and 32 GB RAM. Values of w_1, w_2, w_a in Eq. 4 are estimated to match the expected behavior of the pedestrian to the best of our knowledge: $w_1 = 1$, $w_2 = 1$, $w_a = 0.5$ and the cost of the free space is $\epsilon = 10^{-10}$. Action space parameters are set as follows: angular discretization of θ is $\pi/20$; translational discretization of v is 0.1 m/s, $v \in [0,3]$ m/s. Cell sizes of our grid maps are 0.1 m in Experiment 1 and 0.15 m in Experiment 2. The frequency of predictions is 4 Hz, the number of random walk samples $K = 100$.

A. Experiment 1: Predicting Social Interactions

The first experiment aims to qualitatively evaluate the local predictive ability of JS-MDP in predicting future trajectories of humans involved in cooperate collision avoidance.

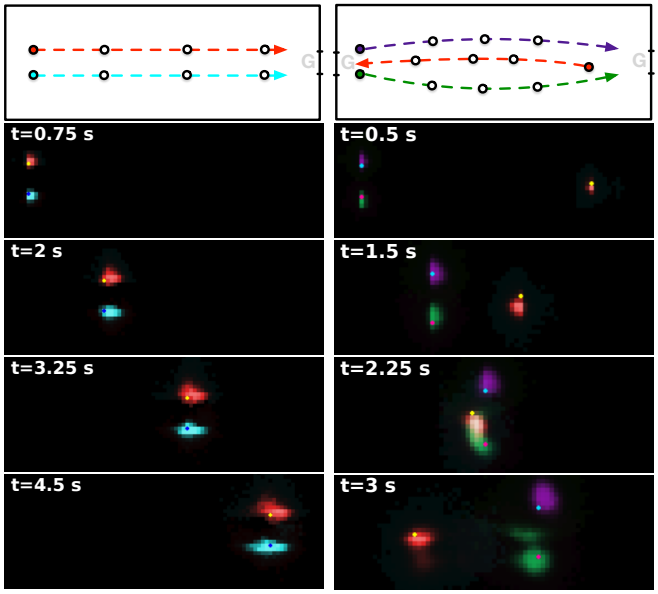

Fig. 6. Prediction results in simulated interactive scenarios. Predicted distributions are color-coded, augmented with the ground truth position shown as a dot in contrasting color. **Top row:** schematic depiction of the situation, dashed lines show the path of each person. **Left:** two people walking together side-by-side, sharing a common goal ahead of them. **Right:** people walking in opposite flows, two of them make room for the third person walking in between.

We simulated four scenarios: two people walking together side-by-side, one person overtaking another person, people walking in opposite flows, and a situation in which a person blocks a narrow passage (see Fig. 6 top row, Fig. 7 top row).

B. Experiment 2: Prediction Evaluation

In this experiment we quantitatively evaluate the predictive performance of JS-MDP using the ATC dataset[1] of real-world trajectories recorded in a shopping center. The map of the environment, covering an area of 900 m^2, is shown in Fig. 1. Using the large selection of trajectories, we identify 15 common goal points in the area with trajectory endpoint clustering. From the dataset we select 25 scenarios, each having several interacting pedestrians (i.e. between 2 and 10). In each scenario, people are following various paths to their intended destinations with different velocities, adjusting paths to comply with other agents nearby. The presence of high level motion stochasticity, observation noise and close proximity to other people makes this dataset a challenging one, especially for longer prediction horizons.

Since JS-MDP combines a planning-based and a social force-based prediction approach, we choose as baselines the planning-based approach by Karasev et al. [13] and the social force-based approach by Elfring et al. [5]. For the sake of a fair comparison, we apply our own goal estimation technique (that requires no learning data, see Eq. 6) to the baselines. We also consider our previous method [1] in the comparison, called *Independent* or *Individual Sampling MDP*

[1]http://www.irc.atr.jp/crest2010_HRI/ATC_dataset/

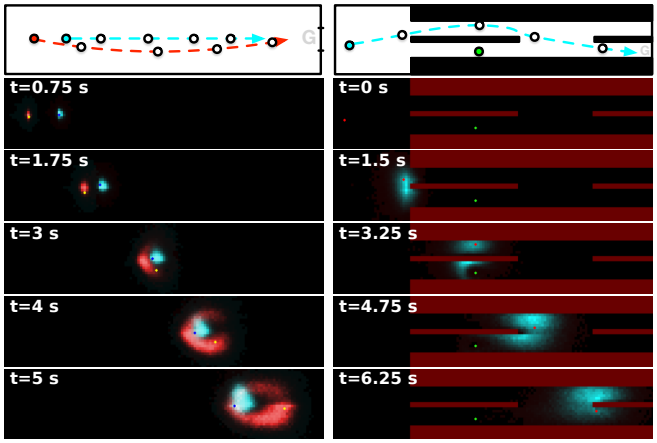

Fig. 7. Prediction results in simulated interactive scenarios. **Left:** a fast walking person is overtaking a slow walker, both of them are heading towards the same goal. **Right:** a person causes a hindrance by blocking a narrow passage. In all cases, the algorithm makes goal-oriented predictions that correctly represent the local ambiguity caused by the other agent or the environment.

(IS-MDP). For each algorithm we perform hyperparameter optimization using the SMAC3 toolbox [18] with results: $\alpha = 15.95$, $\beta = 5.44$ for IS-MDP; $\alpha = 5.03$, $\beta = 13$, $I = (0.6873,\ 0.7249)$, $(a_k, b_k, \lambda) = (0.2708,\ 0.2207,\ 0)$ for JS-MDP; $(w_{g,t}, w_{s,t}) = (0.0312,\ 0.1404)$, $\alpha = 21.31$, $\beta = 18.68$ for [13]; $(q_w, f_w, c_w) = (1.436,\ 0.23,\ 3.097)$, $\zeta_\rho = 83.74$ for [5].

We evaluate the predictive performance of all algorithms using the following metrics: *Negative Log-Probability* (NLP) is a direct measure of ground truth path $\mathcal{T}$ probability, measured at each point of path $\mathcal{T}_i$ according to predictions for that time instance t_i: $\text{NLP}(\mathcal{T}) = -\frac{1}{T}\sum_{i=1}^{T} \log\ p(\mathcal{T}_i|t_i)$. *Modified Hausdorff Distance* (MHD) [12] is a geometric measure of distance between the ground truth path and the most probable path in the predicted probability distribution. For both metrics, lower values corresponds to better prediction accuracy or smaller geometric deviation, respectively. Metric values are calculated for each trajectory in the 25 interactive scenarios and averaged across 50 experiments for each scenario. We use 2 seconds as observation period, and predictions are obtained for $T = 2.5 - 15$ seconds ahead. We also give the average times to compute predictions using our algorithm.

V. Results

Fig. 6 and Fig. 7 show the qualitative results of the first experiment. Our method correctly predicts the development of each scenario, handling typical cooperative actions that people carry out in social spaces: the approach is able to predict overtaking and avoidance maneuvers (see Fig. 7), and to infer usual social interactions such as walking side-by-side or offering the way to a pedestrian moving in opposite direction (Fig. 6), without discarding the goal intentionality of the pedestrians.

Fig. 8 presents the results of the quantitative evaluation, conducted in Exp. 2, showing the mean of the NLP and MHD

metrics over the prediction horizon of 2.5–15 sec. The NLP results show that our method outperforms the other three approaches, assigning higher probability to the future ground truth location of people, which is essential e.g. for predictive motion planning as in Sec. III-C. The two planning-based methods [13], [1] accumulate errors with growing prediction horizon from non-predicted social interactions, while the social force-based method of Elfring et al. [5] gives worse results as predictions do not account for the global environment structure. The results for the probabilistic MHD metric show that our approach is on par with the others methods for short prediction horizons but outperforms them for the more relevant longer horizons.

We also give the runtime resuls for JS-MDP in Fig. 9. Our approach takes in average 0.4 seconds to predict for $T = 7.5$ seconds ahead in a scenario with 5 people. Short-term predictions for $T = 2.5$ seconds can be quickly obtained in less than 0.2 seconds. Note that these measurements exclude time for value iteration (Alg. 2 line 5) and stochastic policy computation (Alg. 2 line 6), which take 0.4 s and 2.35 s for each goal respectively, but can be computed offline for a known environment or updated in a low-frequency cycle.

VI. Conclusions

In this paper, we present a novel method for long-term prediction of human motion. The approach is environment-aware, intent-aware, efficient and able to jointly predict multiple agents. We formulate the task as an MDP planning problem with stochastic motion policies and introduce a random walk policy sampling algorithm that accounts for local interactions with other nearby agents using a social-force model. The experiments in simulation and on publicly available data from a shopping center demonstrate that the method is more accurate in terms of probabilistic and geometric metrics than an individual, single-agent prediction baseline and two state-of-the-art methods.

Future work will aim at more advanced predictive planning under kinodynamic constraints and alternative ways for automatic goal inference in a priori unknown environments.

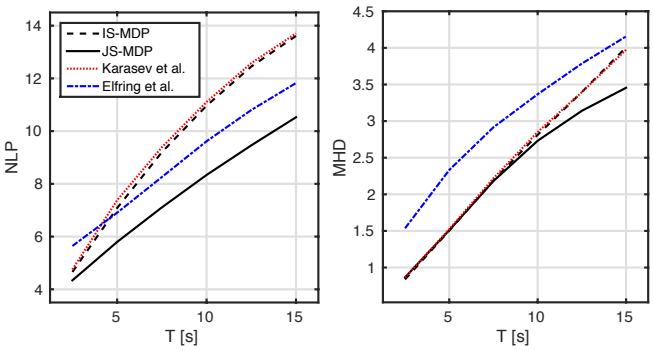

Fig. 8. **Left:** Mean of the Negative Log-Probability (NLP) metric in the ATC dataset. Our approach outperforms the baselines along the entire prediction horizon of up to 15 seconds. **Right:** Mean of the Modified Hausdorff Distance (MHD) metric. Our approach is on par with the baselines for short-term predictions and outperforms them for longer horizons.

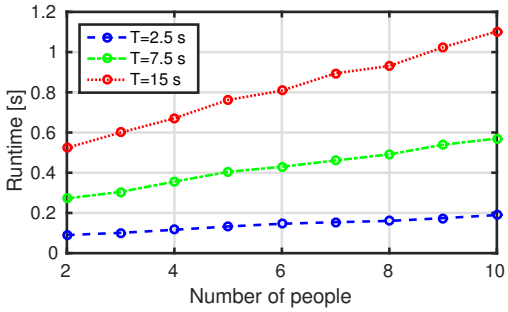

Fig. 9. Average runtime of our algorithm for prediction horizons $T = 2.5$, 7.5, 15 seconds ahead in the ATC scenario with various numbers of people.

References

[1] A. Rudenko, L. Palmieri, and K. O. Arras, "Predictive planning for a mobile robot in human environments," in *IEEE Int. Conf. on Robotics and Automation (ICRA), Workshop on PlanRob*, 2017.

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

[3] S. Pellegrini, A. Ess, M. Tanaskovic, and L. V. Gool, "Wrong turn - no dead end: A stochastic pedestrian motion model," in *IEEE Conf. on Computer Vision and Pattern Recognition - Workshops*, June 2010.

[4] G. Ferrer and A. Sanfeliu, "Behavior estimation for a complete framework for human motion prediction in crowded environments," in *2014 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[5] J. Elfring, R. Van De Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 591–602, 2014.

[6] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[7] 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, 2005.

[8] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *IEEE/RSJ Int. Conf. IROS*, Oct 2010.

[9] A. Vemula, K. Mülling, and J. Oh, "Modeling cooperative navigation in dense human crowds," in *IEEE Int. Conf. ICRA*, 2017.

[10] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Piscataway, NJ, USA, 2009.

[11] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation," in *Proc. of Robotics: Science and Systems (RSS)*, Sydney, Australia, 2012.

[12] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *European Conf. on Computer Vision (ECCV)*, 2012.

[13] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[14] D. Vasquez, "Novel planning-based algorithms for human motion prediction," in *2016 IEEE Int. Conf. ICRA*, May 2016.

[15] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments," in *IEEE Int. Conf. ICRA*, 2016.

[17] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online POMDP planning for autonomous driving in a crowd," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2015.

[18] M. Lindauer, K. Eggensperger, M. Feurer, S. Falkner, A. Biedenkapp, and F. Hutter, "Smac v3: Algorithm configuration in python," https://github.com/automl/SMAC3, 2017.