The Role of Data-Driven Priors in
Multi-Agent Crowd Trajectory Estimation

Gang Qiao,[1] **Sejong Yoon,**[2] **Mubbasir Kapadia,**[1] **Vladimir Pavlovic**[1]

[1]Rutgers University, [2]The College of New Jersey

{gq19, mk1353, vladimir}@cs.rutgers.edu, yoons@tcnj.edu

Abstract

Trajectory interpolation, the process of filling-in the gaps and removing noise from observed agent trajectories, is an essential task for the motion inference in a multi-agent setting. A desired trajectory interpolation method should be robust to noise, changes in environments or agent densities, while also being able to yield realistic group movement behaviors. Such realistic behaviors are, however, challenging to model as they require avoidance of agent-agent or agent-environment collisions and, at the same time, demand computational efficiency. In this paper, we propose a novel framework composed of data-driven priors (local, global or combined) and an efficient optimization strategy for multi-agent trajectory interpolation. The data-driven priors implicitly encode the dependencies of movements of multiple agents and the collision-avoiding desiderata, enabling elimination of costly pairwise collision constraints, resulting in reduced computational complexity and often improved estimation. Various combinations of priors and optimization algorithms are evaluated in comprehensive simulated experiments. Our experimental results reveal important insights, including the significance of the global flow prior and the lesser-than-expected influence of data-driven collision priors.

Introduction

In a multi-agent scenario, a tracking system deployed on an observing, moving or stationary agent (camera), needs to estimate complete trajectories of other moving agents (people, robots, crowds) c.f., (Bera and Manocha 2014; Bera, Kim, and Manocha 2015). However, direct tracker output of the observing agent is often insufficient to reconstruct accurate trajectories because of factors such as noise, the environmental configuration, density of agents in a crowd, or hardware failures. For instance, during the movement of multiple agents in a complex scenario, there are inevitable inter-agent and agent-obstacle occlusions from the perspective of the observing agent, resulting in observed trajectories that are both noisy and incomplete.

To obtain high-quality complete trajectories, tracking systems typically adopt a multi-step strategy. First, the sensor deployed on the observing agent detects and tracks objects in its neighboring area to obtain *local tracklets*, continuous

but short trajectories, of other moving agents. The tracklets of specific agents are then linked together by applying tracklet similarity measurements and re-identification algorithms (e.g., min-cost flow). Next, a trajectory interpolation algorithm is applied to fill in the gaps between the linked tracklets, in a way that the filled portions present desired realistic properties such as containing few potential collisions and saving energy. Finally, trajectory extrapolation could be applied to the reconstructed trajectories so that the observing agent can further plan its own motion.

While several trajectory interpolation approaches have been proposed to-date (Rodriguez et al. 2011; Sharma, Huang, and Nevatia 2012; Bera and Manocha 2014), few have proven to work robustly in a general setting, often exhibiting slow performance and inability to tackle complex scenarios. In practice, many real-time tracking systems typically trade-off accuracy largely for speed (Bera and Manocha 2014). In addition, few approaches attempt to integrate realistic data-driven priors into the estimation framework except for few very recent attempts in simulation and content generation community (Bera, Kim, and Manocha 2016), resulting in interpolated trajectories that fail to match typical realistic crowd behaviors.

The main contribution of our work is a decentralized framework for multi-agent trajectory estimation that yields realistic trajectory estimates and exhibits reduced computational complexity. To that end, the proposed framework is composed of a data-driven prior (either local, global or the combination thereof) and an optimization algorithm. The data prior implicitly encodes the movement dependencies of multiple agents and thus eliminates the satisfaction of costly pairwise collision constraints and decouples individual agent trajectories, yielding reduced computational complexity while maintaining or improving the quality of estimation. We evaluate different combinations of prior representations in simulated experiments and demonstrate the essential role of these priors to accomplish low-complexity, high-accuracy multi-agent trajectory estimation.

Prior Work

Modeling, simulation and analysis of crowd motion have attracted significant interest in research community over the past years, e.g. (Jacques Junior, Raupp Musse, and Jung 2010; Ali et al. 2013; Kapadia, Pelechano,
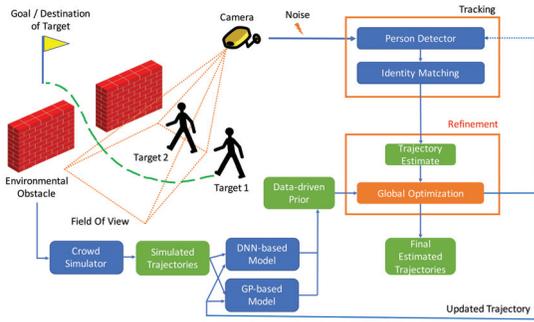
Figure 1: Overview of the proposed framework. A tracker generates noisy and missing trajectories due to occlusions or sensor failure. Our approach takes such observations and applies a global optimization-based trajectory interpolation framework, which incorporates data-driven priors from a model trained on trajectories generated by crowd simulator. Outcome of the optimization (updated trajectories) can be used as new observations to obtain new priors for iteratively improving the tracking performance.

and Allbeck 2015). We refer readers to survey papers for important works in this area (Zhan et al. 2008; Li et al. 2015). In the following, discuss related work on crowd motion modeling and estimation in context of the trajectory interpolation problem.

Crowd Motion and Tracking. It is common to introduce a motion prior to improve pedestrian tracking accuracy (Hu, Ali, and Shah 2008; Ali and Shah 2008; Rodriguez, Ali, and Kanade 2009). The Social Force model (Helbing and Molnár 1995) is one particularly popular choice (Pellegrini et al. 2009; 2010), that represents individual dynamic behavior as a combination of different types of forces that characterize both attractive and repulsive effects between pairs of agents or between an agent and an obstacle (an obstacle could be viewed as a static agent with its own shape). Extensions include the multi-target tracking model (Pellegrini and Gool 2013) that incorporates several different priors such as appearance, physical constraints and social behavior of pedestrians. Bera and Manocha (Bera and Manocha 2014; Bera, Kim, and Manocha 2015) introduced a real-time algorithm for trajectory estimation, which works in medium density crowds using particle filtering. Their approach relies on a multi-agent motion model called velocity obstacle (van den Berg, Lin, and Manocha 2008) that delimits a space of relative velocity within which collision free behavior is guaranteed for a certain period of time. Similar to these model-based approaches (Pellegrini et al. 2009; Bera, Kim, and Manocha 2015), our approach also relies on a synthetic simulator to provide motion priors for crowd trajectory estimation, but it utilizes the prior as a term within a modified global optimization framework of (Yoon et al. 2016), which is different from the aforementioned methods. In this way, complex dependencies of movements are implicitly encoded in the prior to decouple trajectories

while the optimization provides flexibility of incorporating desired properties in the objective to capture more realistic behaviors.

Crowd Trajectory Estimation. Alahi et al. (Alahi, Ramanathan, and Fei-Fei 2014) proposed origin-destination priors to enhance trajectory estimation performance in the context of global optimization-based data-driven methods. They introduced a handcrafted feature called social affinity map (SAM) to capture relative adjoining positions of a number of people within a limited area, by computing the histograms of a number of agents in that region. In contrast, our global optimization framework allows to describe the motion patterns of crowds in a complex scenario within the visibility area of an agent that incorporates much richer information (e.g., the configuration of the environmental obstacles, the shapes and velocities of the agents, rather than a histogram of agent numbers in 2D space) than SAM feature. Obtaining high-quality simulation data of crowd motion is an important aspect in the context of data-driven crowd trajectory estimation, with many solutions proposed in the graphics community (Kapadia, Pelechano, and Allbeck 2015). Since crowd simulation, tracking and trajectory estimation are closely related, crowd simulators (Helbing and Molnár 1995; van den Berg, Lin, and Manocha 2008) may be utilized as the source of motion priors or training data for motion prior models necessary to improve the tracking accuracy (Lerner, Chrysanthou, and Lischinski 2007).

Neural Network-based Methods. The work in (Long, Liu, and Pan 2017) introduced an approach for data-driven collision-minimizing motion planning. In this approach a neural network (NN) is trained from examples of collision-free behaviors, aimed at cloning the Optimal Reciprocal Collision Avoidance (ORCA) policy (van den Berg et al. 2011), which provides a sufficient condition for avoiding collisions if the agents are not densely packed, otherwise ORCA has to select a reasonable velocity. However, this method can not mimic the movement behavior well without a large amount of training samples. Even with large training sets, emphasizing realistic behaviors such as collision minimization is still difficult, since behavior cloning generates only rough imitations while ignoring some detailed properties without environmental interactions to reinforce those properties. In contrast, our approach can embed various priors, including not only the neural network based local prior, but also globally informed probabilistic velocity priors (see Sec.) that are critical for proper behaviors, into a trajectory interpolation framework. This leads to an advantage over either ORCA or neural network-based behavior cloning, resulting in a framework that can handle large portions of missing trajectories.

Notation

We use the following notation to define the framework. Typical tracker output at a specific time point is represented with location and a timestamp. For 2D trackers, the triple $[x_t^i; y_t^i; s_t^i]$ specifies the x-y location and the corresponding

timestamp of agent i at video frame (time step) t. In this work, we assume a uniform temporal grid with a fixed temporal gap between each consecutive pair of frames of a trajectory. Under this assumption, we can omit the timestamp s_t^i from the triple.

Our goal is to estimate a set of trajectories for N agents $\{\mathbf{X}^i\}_{i=1}^N$, where $\mathbf{X}^i = \{\mathbf{x}_t^i\}_{t=1}^T$ is the desired but unobserved trajectory of agent i. We will be estimating $\{\mathbf{X}^i\}_{i=1}^N$ from the corresponding observed fragments of trajectories $\{\mathbf{O}^i\}_{i=1}^N$, $\mathbf{O}^i = \{\mathbf{o}_t^i\}_{t=1}^T$, with noisy and missing portions. We use the term *tracklet* to refer an observed continuous fragment of a trajectory, and we assume known identification (i.e., correspondences between measurements and estimates) of all tracklets. Let Δt denote the sampling period between two consecutive samples t and $t-1$, and $\mathbf{v}_t^i$ denote the true average velocity of agent i over time interval $[t-1, t]$, thus $\mathbf{x}_t^i = \mathbf{x}_{t-1}^i + \Delta t \cdot \mathbf{v}_t^i$. Index $t=0$ indicates known initial position of each agent, $\mathbf{x}_0^i$.

Data Driven Priors

We consider two types of data driven priors: local collision avoidance priors and global flow priors.

Local Collision Avoidance Prior

The goal of utilizing local collision avoidance priors is to effectively replace collision constraints in (Yoon et al. 2016). We will use the data driven regression approach that aims to estimate the velocity $\mathbf{v}^i$ of agent i at space-time point $\mathbf{x}^i$ as

$$\mathbf{v}_{t+1}^i = f_{NN}\left(\mathbf{o}_{t,NN}^i, \mathbf{v}_t^i \,\middle|\, \theta_{NN}\right), \qquad (1)$$

where $\mathbf{o}_{t,NN}^i$ is the local measurement which encodes the local visibility of the state space of agent i. f_{NN} indicates that this function is a neural network (NN) model that is trained with collision-free pairs from expert trajectories, to yield approximately collision-free agent velocities. θ_{NN} is the parameter of the model.

Similar to (Long, Liu, and Pan 2017), the $\mathbf{o}_{t,NN}$ generally includes the following three components: (a) 2D desired velocity: we assume that each agent receives a global velocity guidance signal $\tilde{\mathbf{v}}_t^i$. This signal is typically a velocity vector pointing toward the goal position of each agent, while disregarding local environment geometry or other agents. (b) Local range/occupancy map: we assume that each agent is equipped with a 360-dimensional distance scanner to collect distances to surface points of other agents and obstacles within a certain range. Within the distance map relative positions and shapes of the environmental configurations and other agents are explicitly encoded as distances along a 360 degree circularly sampled grid. (c) 360×2 dimensional velocity map: in addition to aforementioned distances to other agents or obstacles, we also assume the local velocity measurements of neighboring agents or obstacles.

Global Flow Prior

In a multi-agent setting, individual agent movement typically follows a flow pattern that depends on the environment and obstacles, other agents and their density, as well as the global movement goal. This pattern can be encoded in a global flow-field. While one could obtain it from a path planning algorithm, we instead use a data driven approach to capture the global flow field patterns. Specifically, we model the flow field with a Gaussian Process (GP) prior:

$$\mathbf{x}_{t+\Delta t}^i = \mathbf{x}_t^i + \delta^i\left(\mathbf{x}_t^i\right), \qquad (2)$$

where

$$\delta_x^i\left(\mathbf{x}_t^i\right) \sim GP\left(\delta | \mathbf{x}_t^i, \mathbf{X}_{train}, \theta_{GP}\right). \qquad (3)$$

$\delta_x^i(\mathbf{x}_t^i)$ is the "global" velocity of agent i at frame t. The GP will now model the data-prior velocity field in (x, y, s) space, similar to (Kim, Lee, and Essa 2011). $\mathbf{X}_{train}$ denotes the trajectory data used to train the GP model and θ_{GP} the model hyper-parameter.

Embedding Priors into Multi-agent Optimization Framework

The essence of the multi-agent optimization framework is to estimate (interpolate) the trajectories of a set of N agents $\mathbf{X}$ from some set of observations $\mathbf{O}$, given a partially observable environment $\mathbf{Z}$, $Pr(\mathbf{X}|\mathbf{O}, \mathbf{Z})$. In (Yoon et al. 2016), this problem is formulated as a MAP estimation problem by minimizing the following objective

$$\sum_i E_u^i(\mathbf{x}^i, \mathbf{v}^i | \mathbf{o}^i) + \sum_i \sum_{i \neq j} E_p^{i,j}(\mathbf{x}^i, \mathbf{v}^i, \mathbf{x}^j, \mathbf{v}^j), \qquad (4)$$

where E_u^i is the energy term related to an individual agent i (i.e., unary) and $E_p^{i,j}$ denotes the pairwise energy term describing dependencies of a pair of agents (i, j). The unary term includes energies that model kinematic constraint, maximum velocity constraint as well as the compatibility between the estimated trajectories and the measurements $\mathbf{o}^i$. The pairwise term is responsible for avoiding collisions between agents. However, joint optimization of $\mathbf{X}$ is a challenging task, in part because of the existence of the pairwise terms as well as the lack of strong motion priors. We next describe some of the weak but frequently used priors and then suggest a way to combine our data-driven priors while keeping the computational complexity under control.

Existing Unary Priors

The following three unary energy terms are defined in (Yoon et al. 2016) to model individual behavior of an agent: tracker output, kinetic energy, and maximum velocity constraint. The Tracker Output term seeks to keep the estimated trajectory close to the measured trajectories while taking into account the amount of observation uncertainty u_t^i:

$$E_{gt}^i(\mathbf{x}^i | \mathbf{o}^i) = \sum_t u_t^i \|\mathbf{x}_t^i - \mathbf{o}_t^i\|^2. \qquad (5)$$

For instance, $u_t^i = 0$ indicates that the measurement is missing. The Kinetic Energy term ensures that the total traveled distance is minimized:

$$E_{kn}^i(\mathbf{x}^i) = C_{kn} \sum_t \|\mathbf{x}_t^i - \mathbf{x}_{t-1}^i\|^2. \qquad (6)$$

Here parameter C_{kn} can be interpreted as the mass of an agent. Finally, the Maximum Velocity Constraint certifies

Algorithm 1: Proposed Optimization Framework

> **Input** : $\mathbf{O}$, $(\theta_{NN}, \sigma_{NN})$, $(\theta_{GP}, \mu_{GP}, \sigma_{GP})$,
> C_{kn}, C_{mv}
> **Output:** $\hat{\mathbf{X}}$
> 1   Initialize $\mathbf{X}$;
> 2   **repeat**
> 3   Compute NN prior velocities of $\mathbf{X}$ using Eq. 1;
> 4   Compute GP prior velocities of $\mathbf{X}$ using Eq. 3;
> 5   Find $\mathbf{X}$ by minimizing Eq. 9;
> 6   **until** $\mathbf{X}$ *converges*;
> 7 $\hat{\mathbf{X}} = \mathbf{X}$;

that each agent's speed not exceed a physically feasible velocity, given as C_{mv}:

$$E_{mv}^i(\mathbf{x}^i) = \begin{cases} 0 & \text{if } \|\mathbf{v}_t^i\| \leq C_{mv} \\ \infty & \text{otherwise} \end{cases}, \forall t = 1..T. \quad (7)$$

Combining Local and Global Priors

The key challenge in the aforementioned multi-agent optimization framework arises from the existence of the pairwise terms, implying coupling trajectories between agents, the highly nonlinear nature of the coupling (collision) constraints and the expensive computations due to this coupling. If such coupling were to be eliminated, the optimization of trajectories for each agent could be solved independently from other agents. However, their elimination would result in infeasible motion with possibly many collisions. We therefore propose to replace the computationally costly pairwise terms with stronger global motion and local data-driven collision priors. Specifically, we propose to modify the unary objectives by augmenting them as follows:

$$E_{dg}^i(\mathbf{v}^i|\mathbf{o}^i) = \frac{1}{\sigma_{GP}^2(\mathbf{o}_t^i, \theta_{GP})} \left\| \mathbf{v}_{t+\Delta t}^i - \mu_{GP}\left(\mathbf{o}_t^i, \theta_{GP}\right) \right\|^2$$
$$+ \frac{1}{\sigma_{NN}^2} \left\| \mathbf{v}_{t+\Delta t}^i - f_{NN}\left(\mathbf{o}_{NN}^i|\theta_{NN}\right) \right\|^2 \quad (8)$$

Here, μ_{GP} is the predictive mean and σ_{GP} is the predictive standard deviation given by the learned GP and σ_{NN} is the standard deviation of the NN regression model. Integrating all unary terms above, we obtain the final global objective, which we seek to minimize in the estimation process:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \sum_i E_u^i(\mathbf{x}^i, \mathbf{v}^i|\mathbf{o}^i) = \arg\min_{\mathbf{X}} \sum_i E_{gt}^i(\mathbf{x}^i|\mathbf{o}^i)$$
$$+ E_{kn}^i(\mathbf{x}^i) + E_{mv}^i(\mathbf{x}^i) + E_{dg}^i(\mathbf{x}^i|\mathbf{o}^i) \quad (9)$$

Therefore, the method proposed in (Yoon et al. 2016) must optimize all trajectories jointly due to the existence of the paired term. In contrast, in our framework, each trajectory could be optimized independently because the complex spatial dependencies among trajectories have been encoded in the priors. This leads to a practical disjoint (but iterative) optimization paradigm where the decoupled agent trajectories could be optimized in parallel. Consequently, our method exhibits faster convergence.

Optimization of the Global Objective

We introduce three optimization approaches to solve the optimization problem in Eq. 9. In general, our optimization framework is iterative and outlined in Alg. 1. The iterative nature of our algorithm stems from the coupling between the essential collision-avoiding NN term in Eq. 1 and the solution to the optimization problem. Namely, the NN term's range observations require the knowledge of the agents' locations and velocities, which are the variable we are solving for. To mitigate this, we propose the alternating optimization scheme where the agent's trajectories from a previous iteration are used as the proxies for measurements in Eq. 1.

The choice of the minimizer in Step-5 of this algorithm is important but less essential. We consider three methods for optimizing Eq. 9: a message-passing algorithm (MPA) of (Bento et al. 2013; Yoon et al. 2016), a general interior point method (IPM), and an unscented Kalman smoother (UKS) that exploits the sequential nature of each (independent) agent's trajectory optimization task while applying a nonlinear Rauch-Tung-Striebel smoother (Rauch, Striebel, and Tung 1965). Further details of some of the selected approaches are provided in the Supplement.

Experiments

To evaluate the proposed framework, we consider six experimental settings similar to those in (Yoon et al. 2016): three different settings of bottlenecks (each contains a challenging egress in evacuation-like scenarios), concentric circle (agents are symmetrically placed along a circle and aim to reach their antipodal positions), two-way and four-way hallways (the environment is divided by two or four building blocks and agents move along the regulated ways). The configuration of each scenario and the details of training can be found in the supplementary.

We considered five different prior velocity predictors: Gaussian Process (GP), Neural Network (NN), a linear combination of the NN and GP (LinComb GP+NN), GP-driven NN (GP-fed-NN), and GP-driven ORCA (GP-fed-ORCA). For GP, NN, and LinComb GP+NN, we used the training split of the trajectories to train the data-driven priors we described in Eq. 1, Eq. 3, and Eq. 8, respectively. For GP-fed-NN, we used the outputs of the trained GP (the velocity (mean) and the variance) as two additional input branches to a neural network model, besides the local observation. For GP-fed-ORCA, we used the trained GP's velocity (mean) as the preferred velocity for ORCA (van den Berg et al. 2011), a common local collision avoiding framework. When training the models, GP was trained within each scenario while the NN was trained across all six scenarios to obtain a more general velocity predictor.

Experimental Setup. The ground truth trajectories are obtained by running SteerSuite (Singh et al. 2009) library with social force AI (Helbing and Molnár 1995), and are split into a training and testing sets. In testing set, each trajectory contains a challenging missing segment around 30% points, which are initially inferred with linear interpolation (again, the overall framework is not a forward prediction procedure. Instead, it iteratively estimates the missing portion of a trajectory (interpolation)). See illustration in Fig.2.

We consider two evaluation strategies. In the first *Fixed-Density* strategy, we evaluate our trained models on the test sets from the same agent density setting. Namely, both the
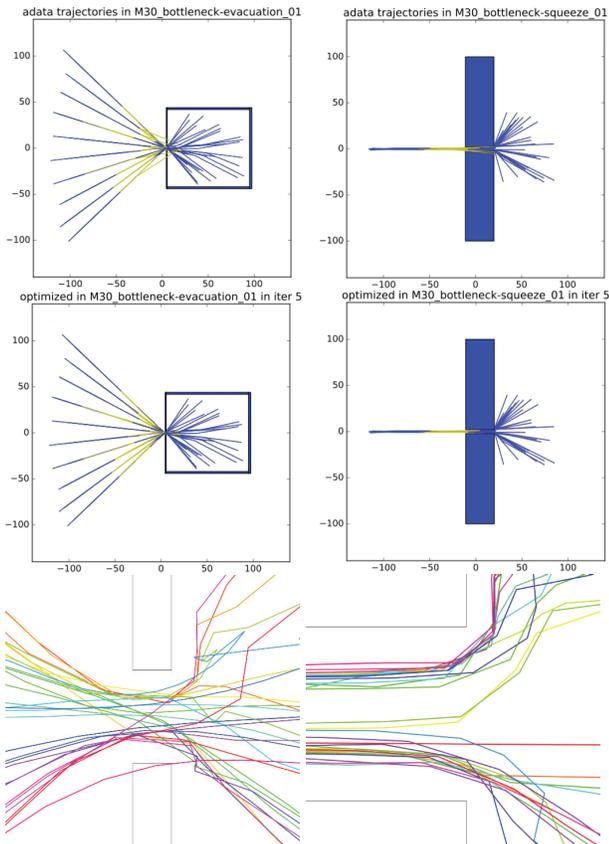
Figure 2: Visualization: the two columns show bottleneck evacuation and bottleneck squeeze scenario respectively. In each column, the top figure illustrates the initial linear interpolation and the middle figure shows the optimized trajectories in the yellow portion, which is further enlarged in the bottom figure. Best viewed by zooming in.
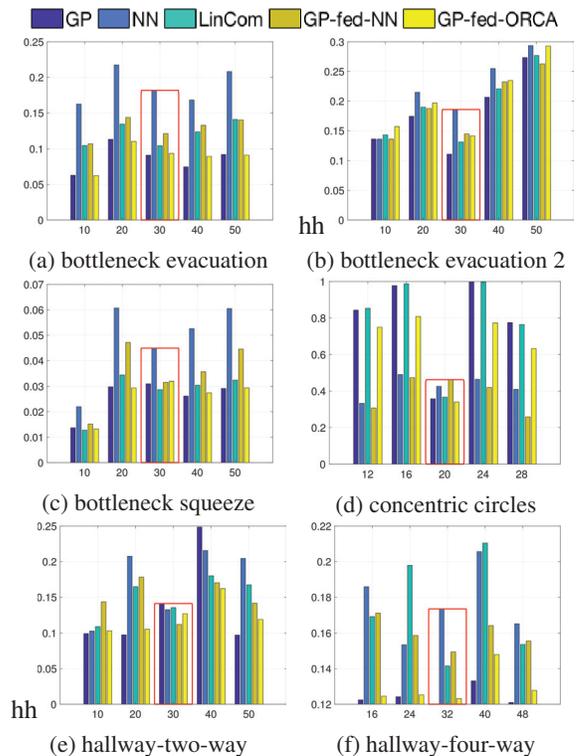

Figure 3: DTW over 6 scenarios using UKS optimization for Varying-Density evaluation. Horizontal axis denotes the number of agents (density). Bounding box indicates the case where training and test densities are the same.

training and the test scenarios contain the matching (identical) number of agents. In the second *Varying-Density* strategy, we consider test scenarios where the agent density varies compared to training setting to evaluate the generalization ability of our approaches. Details of these evaluations are described below.

The following methods are evaluated: explicit collision avoidance local optimization using message-passing ADMM (MPA) (Yoon et al. 2016) that includes pair-wise constraints, our framework with various prior velocity predictor settings (GP, NN, LinComb GP+NN, GP-fed-NN, and GP-fed-ORCA), while the optimizer is either IPM or UKS. We set the parameter as: $u_t = 1$ if the point is actually observed, otherwise $u_t = 0$; $C_{kn} = 1$ by assuming homogeneous crowd, $C_{mv} = 2.6m/s$, $\Delta t = 1.5s$ and $\lambda = 1/(\sigma_{NN}^2 \Delta t^2) \approx 108.0$.

We employ four evaluation scores. The deviation in spatial locations of the reconstructed trajectory w.r.t. the ground truth trajectory is measured with dynamic time warping distance (DTW). In order to incorporate velocity (temporal information) in the score, average mean squared error (MSE)

is also reported. We also measure the number of agent-agent collisions and agent-obstacle collisions. A collision occurs when the distance between the centers of two agents is strictly less than the sum of their radii during their continuous movements, and it could be checked by solving a quadratic equation provided with locations of two agents at consecutive time points. The number of collisions is accumulated by counting all collisions along every time step, which means this metric is strict. Note that for simplicity a collision does not change the velocities of involved agents. We also measure time-to-completion as a proxy for the computational complexity of each approach. Indices are measured after the 5th optimization loop, when the change of a trajectory decreases are sufficiently small.

Experimental Results. Results of experiments in the first, matching-agent-density setting, are summarized in Table 1, 2, 3, 4. The average rankings of different methods can be used to ascertain relative performance and are presented in Table 5. Table 6 shows the computational time of different evaluated approaches. Finally, results of evaluations across different train-test agent densities are shown in Figures 3, 4 and 5. We only show evaluations for one of the optimization approaches, the UKS, for brevity and because other approaches follow similar trends.

Table 1: Missing 30% of frames, Average Relative DTW Distance in Percentage for Fixed-Density evaluation. Red color indicates the best method in each scenario, followed by the second best in blue. Best viewed in color.

Scenario	MPA	NN		GP-fed-NN		LinComb GP+NN		GP		GP-fed-ORCA	
		IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS
bottleneck-evacuation	22.2	11.4	18.1	9.2	12.0	10.2	10.4	**9.0**	9.1	**9.0**	9.3
bottleneck-evacuation-2	20.9	16.3	18.5	12.7	14.4	11.4	13.1	**9.3**	**11.1**	12.3	14.1
bottleneck-squeeze	19.2	3.7	4.4	**2.3**	3.1	**2.6**	2.8	3.1	3.1	3.3	3.1
concentric-circles	47.2	**14.8**	42.5	**11.2**	46.2	17.2	36.6	17.1	35.8	37.3	34.0
hallway-two-way	16.9	17.4	13.2	13.7	**11.2**	14.3	13.5	15.3	14.1	14.8	**12.6**
hallway-four-way	18.0	25.0	17.3	15.5	14.9	15.0	14.1	12.6	**12.0**	12.9	**12.3**

Table 2: Missing 30% of frames, Average MSE Distance for Fixed-Density evaluation. Red color indicates the best method in each scenario, followed by the second best in blue. Best viewed in color.

Scenario	MPA	NN		GP-fed-NN		LinComb GP+NN		GP		GP-fed-ORCA	
		IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS
bottleneck-evacuation	11.0	1.8	2.8	1.6	1.9	1.1	1.1	1.1	1.1	**1.0**	**1.0**
bottleneck-evacuation-2	0.7	0.5	0.9	0.2	0.5	0.3	0.2	**0.2**	**0.2**	0.3	0.3
bottleneck-squeeze	6.1	1.0	0.9	**0.5**	**0.3**	0.6	0.5	0.8	0.5	0.8	0.5
concentric-circles	0.4	0.6	0.6	0.4	0.5	**0.1**	0.2	**0.1**	0.2	0.5	0.3
hallway-two-way	3.1	1.8	1.8	**0.6**	**0.5**	1.8	1.1	5.3	4.0	4.2	2.4
hallway-four-way	4.5	1.8	1.9	**0.7**	**0.5**	1.5	1.4	2.9	1.4	2.5	1.4

Table 3: Missing 30% of frames, Agent-Agent Collisions for Fixed-Density evaluation. Red color indicates the best method in each scenario, followed by the second best in blue. Best viewed in color.

Scenario	MPA	NN		GP-fed-NN		LinComb GP+NN		GP		GP-fed-ORCA	
		IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS
bottleneck-evacuation	**10.6**	16.2	15.4	17.2	15.0	14.6	**12.6**	14.2	13.8	16.0	14.2
bottleneck-evacuation-2	147.0	159.8	140.4	157.0	**132.4**	**139.4**	140.8	144.8	139.6	158.6	143.4
bottleneck-squeeze	132.0	49.0	**46.6**	47.4	47.0	**46.6**	47.0	47.0	47.0	47.4	47.4
concentric-circles	63.4	**18.4**	52.4	**19.6**	71.0	26.0	44.0	26.0	44.0	67.0	54.4
hallway-two-way	53.6	30.4	29.6	**27.8**	29.0	29.8	**27.6**	29.8	29.6	29.0	30.0
hallway-four-way	35.8	13.6	13.6	**12.2**	13.0	**10.6**	12.4	13.6	13.4	15.0	15.4

Table 4: Missing 30% of frames, Average-Obstacle Collisions for Fixed-Density evaluation. Red color indicates the best method in each scenario, followed by the second best in blue. Best viewed in color.

Scenario	MPA	NN		GP-fed-NN		LinComb GP+NN		GP		GP-fed-ORCA	
		IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS
bottleneck-evacuation	**39.2**	52.4	61	45.4	60.2	46.6	54.6	**37.2**	46.2	39.4	49.4
bottleneck-evacuation-2	**46.2**	75.4	83.2	76.4	88.0	88.8	87.6	**70.4**	73.8	89.4	81.4
bottleneck-squeeze	28.0	30.8	32.6	**7.8**	28.8	8.2	18.8	**6.6**	18.6	13.2	16.8
hallway-two-way	0.0	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
hallway-four-way	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.8	0.0	3.2	0.4

Table 5: Average ranks of different combinations over scenarios for Fixed-Density evaluation. Lower rank is better. Red color indicates the best method for each evaluation approach, followed by the second best in blue. Best viewed in color.

Evaluation	MPA	NN		GP-fed-NN		LinComb GP+NN		GP		GP-fed-ORCA	
		IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS
DTW	10.67	8.33	8.50	4.00	6.67	4.83	5.33	**3.67**	**3.75**	5.58	4.67
MSE	9.67	8.17	8.83	**3.67**	4.83	5.33	**3.83**	5.83	4.17	7.00	4.67
Agent-Agent	8.50	8.17	5.50	5.67	5.17	**3.58**	**3.50**	5.67	4.41	8.25	7.58
Agent-Obstacle	**3.30**	6.60	8.00	4.40	7.80	6.00	7.00	**4.10**	5.00	7.10	6.70

Table 6: Computational time for Fixed-Density evaluation (in second). Note that the time of IPM and UKS is sequentially accumulated over trajectories, while in practice, trajectories could be optimized in parallel when using IPM and UKS.

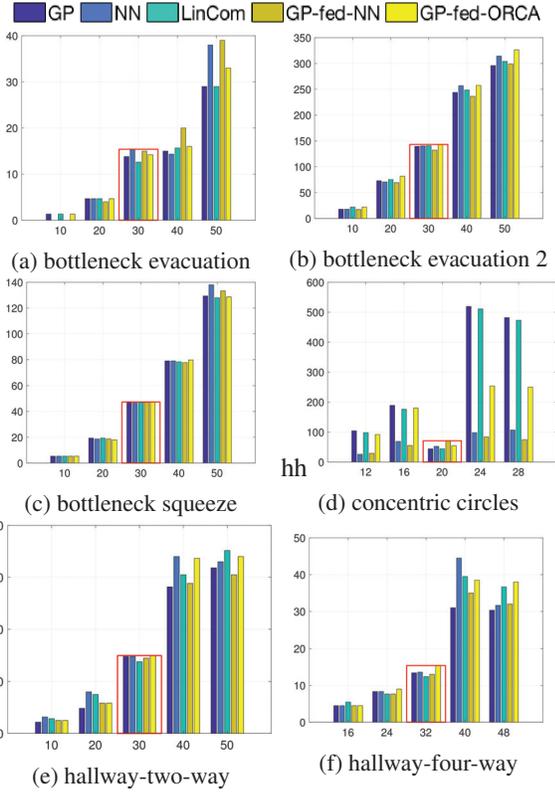| Scenario	MPA	NN		GP-fed-NN		LinComb GP+NN		GP		GP-fed-ORCA	
		IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS	IPM	UKS
bottleneck-evacuation	1343.6	887.9	304.4	887.5	352.7	922.2	337.9	881.1	353.4	681.1	144.4
bottleneck-evacuation-2	521.1	290.1	201.4	321.2	233.8	308.7	222.4	320.0	234.7	174.7	87.7
bottleneck-squeeze	738.8	746.8	283.9	769.3	329.6	637.5	313.4	615.7	331.2	475.5	136.4
concentric-circles	29.5	173.0	143.8	191.6	162.0	178.0	148.5	192.1	162.2	85.2	56.3
hallway-two-way	876.3	841.5	293.9	876.4	342.6	885.8	324.9	880.7	343.7	694.7	143.2
hallway-four-way	1135.9	901.9	319.1	941.4	369.5	944.4	350.6	944.4	368.8	729.7	147.2

Figure 4: Number of agent-agent collisions over 6 scenarios using UKS optimization for Varying-Density evaluation. See Fig. 3 for additional caption details.
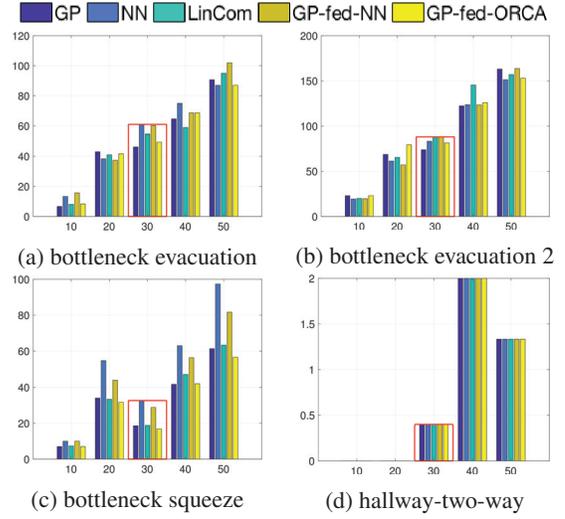

Figure 5: Number of agent-obstacle collisions over 4 scenarios using UKS optimization for Varying-Density evaluation. See Fig. 3 for additional caption details.

Discussion

For the Fixed-Density evaluations where the agent density of the test set matches that of the training set, Table 1, 2, 3 and 4 show that (i) for DTW score, GP-fed-NN, GP and GP-fed-ORCA perform better than other priors. This illustrates that GP is important for regulating the trajectory to roughly follow the movement (flow) pattern. Furthermore, MSE measurement follows a similar trend: GP-fed-NN, Lin-Comb GP+NN and GP perform better than other priors. (ii) In terms of the number of agent-agent collisions, GP-fed-NN, LinComb GP+NN perform best, followed by NN. This demonstrates that pure GP alone is not sufficient to guar-

antee the desired collision avoidance, and has to be augmented by another policy learner like NN to achieve the least number of agent-agent collisions. Surprisingly, for this metric GP-fed-ORCA does not give satisfactory results, which might stem from our complex training/testing datasets, containing densely packed agents (especially for bottleneck-evacuation-2 scenario) such that the permitted set of ORCA is prone to be empty under such agent density. (iii) For the number of agent-obstacle collisions, MPA and GP perform better than others, indicating the importance of global flow patterns to avoid stationary environmental obstacles.

Tab. 5 suggests that, as the optimization approach, IPM is slightly better than UKS and MPA for DTW; on the other hand, UKS achieves the fewest agent-agent collisions, lower than IPM and MPA. MPA yields the fewer agent-obstacle collisions than IPM and UKS. However, in terms of computational complexity, Tab. 6 shows that MPA is the most expensive approach since it includes pair-wise constraint. In addition, UKS is cheaper than IPM, in that UKS exploits fast forward-backward computations specific to the trajectory domain. The lack of conformability in the above observations implies that there might not exist a dominant opti-

mization algorithm and one might need to trade off when choosing the optimizations.

Evaluation across different agent densities above and below the training density, shown in Fig 3, 4 and 5, provides an insight into the important generalization ability and robustness of proposed approaches. In these figures, the model is trained with density in the third group, marked with a red bounding box. We can see that (i) for DTW, GP-fed-ORCA performs best except for concentric-circle scenario, where NN presents least DTW distance w.r.t. the ground truth test trajectories. The reason might be that in concentric-circle scenario, agents are symmetrically placed along a circle, leading to observation patterns similar across all agents and, hence, local NN policy patterns that can be learned by reusing (sharing) the data across agents. On the other hand, global GP fails to reuse the data, leading to a weaker generalization model. (ii) There is a general increasing monotonic trend in the number of collisions across densities. This is an expected outcome, however without a clear winner in terms of collisions except for concentric-circle scenario. For this scenario, again, the ability to reuse data across agents, due to symmetry, may lead to better generalization of the local NN approach. (iii) Models in the matching density settings outperform those in the mismatched density settings in terms of the agent-agent collisions only for the concentric-circle scenario, while in terms of the agent-obstacle collisions this happens for the bottleneck-squeeze scenario. These results suggest that varying density impact the performance of different models in a reasonable and predictable manner.

Overall, the above observations emphasize the importance of the global flow priors, embodied in the GP model, in our multi-agent trajectory optimization framework. Local collision avoidance priors, manifested through the NN model, have lesser than expected yet still measurable impact. We plan to further investigate these factors as well as the generalization across environments and considerations of exploiting advanced trajectory learning algorithms where the dynamics of the trajectory and the cost of making an inference could be implicitly learned, in our future work.

Acknowledgments

References

Alahi, A.; Ramanathan, V.; and Fei-Fei, L. 2014. Socially-aware large-scale crowd forecasting. In *CVPR*, 2211–2218.

Ali, S., and Shah, M. 2008. Floor Fields for Tracking in High Density Crowd Scenes. In *ECCV*.

Ali, S.; Nishino, K.; Manocha, D.; and Shah, M. 2013. *Modeling, Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective*. Springer Publishing Company, Incorporated.

Bento, J.; Derbinsky, N.; Alonso-Mora, J.; and Yedidia, J. S. 2013. A message-passing algorithm for multi-agent trajectory planning. In *NIPS*. Curran Associates, Inc. 521–529.

Bera, A., and Manocha, D. 2014. Realtime Multilevel Crowd Tracking using Reciprocal Velocity Obstacles. In *ICPR*.

Bera, A.; Kim, S.; and Manocha, D. 2015. Efficient Trajectory Extraction and Parameter Learning for Data-Driven Crowd Simulation. In *Graphics Interface*.

Bera, A.; Kim, S.; and Manocha, D. 2016. Online parameter learning for data-driven crowd simulation and content generation. *Computers and Graphics*.

Helbing, D., and Molnár, P. 1995. Social force model for pedestrian dynamics. *Phys. Rev. E* 51:4282–4286.

Hu, M.; Ali, S.; and Shah, M. 2008. Learning Motion Patterns in Crowded Scenes Using Motion Flow Field. In *ICPR*.

Jacques Junior, J.; Raupp Musse, S.; and Jung, C. 2010. Crowd analysis using computer vision techniques. *Signal Processing Magazine, IEEE* 27(5):66–77.

Kapadia, M.; Pelechano, N.; and Allbeck, J. 2015. *Virtual Crowds: Steps Toward Behavioral Realism*. MORGAN & CLAYPOOL.

Kim, K.; Lee, D.; and Essa, I. 2011. Gaussian Process Regression Flow for Analysis of Motion Trajectories. In *ICCV*.

Lerner, A.; Chrysanthou, Y.; and Lischinski, D. 2007. Crowds by example. *Comput. Graph. Forum* 26(3):655–664.

Li, T.; Chang, H.; Wang, M.; Ni, B.; Hong, R.; and Yan, S. 2015. Crowded scene analysis: A survey. *IEEE Transactions on Circuits and Systems for Video Technology* 25(3):367–386.

Long, P.; Liu, W.; and Pan, J. 2017. Deep-learned collision avoidance policy for distributed multiagent navigation. *IEEE Robotics and Automation Letters* 2(2):656–663.

Pellegrini, S., and Gool, L. V. 2013. Tracking with a mixed continuous-discrete conditional random field. *CVIU*.

Pellegrini, S.; Ess, A.; Schindler, K.; and van Gool, L. 2009. You'll Never Walk Alone: Modeling Social Behavior for Multitarget Tracking. In *ICCV*.

Pellegrini, S.; Ess, A.; Tanaskovic, M.; and Gool, L. V. 2010. Wrong turn - no dead end: a stochastic pedestrian motion model. In *SISM*.

Rauch, H.; Striebel, C.; and Tung, F. 1965. Maximum likelihood estimates of linear dynamic systems. *AIAA Jrnl.* 3(8):1445–1450.

Rodriguez, M.; Ali, S.; and Kanade, T. 2009. Tracking in Unstructured Crowded Scenes. In *ICCV*.

Rodriguez, M.; Sivic, J.; Laptev, I.; and Audibert, J.-Y. 2011. Data-driven Crowd Analysis in Videos. In *ICCV*.

Sharma, P.; Huang, C.; and Nevatia, R. 2012. Unsupervised incremental learning for improved object detection in a video. In *CVPR*, 3298–3305.

Singh, S.; Kapadia, M.; Faloutsos, P.; and Reinman, G. 2009. An Open Framework for Developing, Evaluating, and Sharing Steering Algorithms. In *MIG*, 158–169.

van den Berg, J.; Guy, S. J.; Lin, M. C.; and Manocha, D. 2011. Reciprocal n-Body Collision Avoidance. In *ISRR*.

van den Berg, J.; Lin, M. C.; and Manocha, D. 2008. Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation. In *ICRA*.

Yoon, S.; Kapadia, M.; Sahu, P.; and Pavlovic, V. 2016. Filling in the blanks: Reconstructing microscopic crowd motion from multiple disparate noisy sensors. In *WACV Workshop*, 1–9.

Zhan, B.; Monekosso, D.; Remagnino, P.; Velastin, S.; and Xu, L.-Q. 2008. Crowd analysis: a survey. *MVA* 19:345–357.