Probabilistic Motion Planning and Prediction via Partitioned Scenario Replay

Oscar de Groot, Anish Sridharan, Javier Alonso-Mora and Laura Ferranti

***Abstract*— Autonomous mobile robots require predictions of human motion to plan a safe trajectory that avoids them. Because human motion cannot be predicted exactly, future trajectories are typically inferred from real-world data via learning-based approximations. These approximations provide useful information on the pedestrian's behavior, but may deviate from the data, which can lead to collisions during planning. In this work, we introduce a joint prediction and planning framework, Partitioned Scenario Replay (PSR), that stores and *partitions* previously observed human trajectories, referred to as *scenarios*. During planning, scenarios observed in similar situations are reintroduced (or *replayed*) as motion predictions. By sampling real data and by building on scenario optimization and predictive control, the planner provides probabilistic collision avoidance guarantees in the real-world. Relying on this guarantee to remain safe, PSR can incrementally improve its prediction and planning performance online. We demonstrate our approach on a mobile robot navigating around pedestrians.**

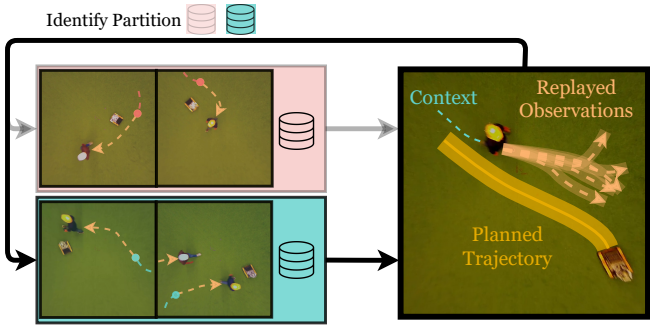


Fig. 1: Illustration of PSR. Human trajectories are collected in a database and partitioned based on the associated context (highlighted in red and cyan). During planning, the current context identifies one of the partitions and trajectories in this partition (e.g. cyan) are replayed as motion predictions that the planned trajectory (in yellow) must avoid.

I. Introduction

AUTONOMOUS navigation among humans typically builds on a pipeline that perceives humans (**perception**), predicts their future motion (**prediction**) and plans a safe motion around them (**planning**). Predicting human motion is subject to significant uncertainty because the intentions of humans cannot directly be observed. Therefore, several distinct future trajectories may be plausible at any time. Human behavior additionally varies per person (e.g., age) and depends on the environment (e.g., the layout of the space or position of other humans).

The uncertainty of human motion is typically predicted by learning a probability distribution of future motion conditioned on contextual information (e.g., velocity or the position of other humans). Although the learned distribution approximates the real distribution, it can deviate from the data points to improve the overall fit. This makes it hard to provide guarantees on inferred trajectories, since not all data points are respected. Additionally, prediction methods are typically designed in isolation, without considering the planner that relies on its outputs.

In this work, we propose Partitioned Scenario Replay (PSR), for data-driven prediction and planning, illustrated in Fig. 1. Instead of learning a probability distribution from previously observed human trajectories and the associated contextual information, we first partition human trajectories offline based on the context in which they were observed. In each online iteration of the planner, we use the context to decide from which partition trajectories are reintroduced (or *replayed*) to predict the human's motion. We use scenario-based trajectory optimization [1] to avoid all of the replayed trajectories, which inherently provides a probabilistic safety guarantee on the planned trajectory in the real-world.

The authors are with the Dept. of Cognitive Robotics, TU Delft, 2628 CD Delft, The Netherlands. Email: o.m.degroot@tudelft.nl

This work received support from the Dutch Science Foundation NWO-TTW, within the Veni project HARMONIA (18165).

A. Related Work - Human Motion Prediction

Human motion prediction methods can be categorized by how they incorporate models and data.

1) Model-Based Human Motion Prediction methods such as constant velocity or physics based rules predict human trajectories based on model-based approximations and are popular for their simplicity [2], [3]. They fail, however, to capture contextual information.

2) Planning-Based Human Motion Prediction methods apply motion planning methods from the perspective of the human to predict the human's future motion. In [4] pedestrian predictions are modeled by a relaxed maximum value Markov Decision Process (MDP) that infers pedestrian goal locations from previously observed trajectories. These predictions are used to avoid humans using a graph-based planner. Static obstacles are considered in [5], where predictions are informed with their distance to goals. Nonlinear optimization towards goals is used in [6]. Planning-based methods capture human intent more accurately, but due to the planning step suffer from long inference times. Additionally, wrongly inferred goals result in incorrect predictions and can lead to collisions.

3) Learning-Based Human Motion Prediction methods learn a distribution of probable human trajectories from a dataset (e.g., [7]). The temporal dependencies of human motion can be modeled by Recurrent Neural Networks [8]

such as Long Short-Term Memory (LSTM). We distinguish between *uni-modal* and *multi-modal* prediction models.

Uni-modal methods, such as Social LSTM [9] and Social-STGCNN [10], predict a single trajectory (or mode) for each human. Hence, when multiple distinct trajectories (modes of the probability distribution) are possible, they tend to average the modes without representing any of the modes accurately [11], [12].

Multi-modal methods do account for multiple distinct trajectories. *Variational* methods such as the Conditional Variational AutoEncoder (CVAE) [13] (based on the Variational AutoEncoder (VAE) [14]), model latent variables to represent the data as a lower dimensional distribution. Trajectron++ [15] is a CVAE that incorporates dynamics and scene context to improve predictions. Variational Recurrent Neural Networks (VRNN) [16] are extended VAEs that model high dimensional sequences. This network was applied for Social-VRNN [17] where scene-aware multi-modal trajectory predictions are represented by a Gaussian Mixture Model (GMM). Y-Net [18] predicts several trajectories per endpoint using waypoints. NSP-SFM [19] incorporates a physics model and CVAE to learn realistic physical behavior.

Generative Adversarial Networks (GANs) [20] train a generator together with a discriminator network. The discriminator enforces the generator to produce realistic predictions, which can be queried after training. Examples are Social-GAN [21] that encodes pedestrian interactions and MG-GAN [22] that produces modes through multiple generators.

State-of-the-art learning-based prediction algorithms still have severe limitations. The accuracy of the learned distribution is limited because finite data is available and may be insufficient for guaranteeing safety [23]. In addition, the model distribution has a predefined structure, assuming for example a static number of modes [17], [24], which may not accurately capture the real distribution and auxiliary uncertainties such as tracking and sensing errors. While these inaccuracies are always present, the planner is typically not aware of their magnitude, which may lead to collisions in practice. Finally, learning-based models are computationally expensive and resource intensive to train and deploy.

B. Related Work - Motion Planning under Uncertainty

The planning problem can be solved via an optimization problem, where constraints impose collision avoidance (e.g., [25]). Under uncertainty, constraints can be reformulated as chance constraints, i.e., constraints that must hold with a probability. Chance constraints cannot be evaluated online due to their computational complexity, but several works have formulated approximations, such as [26]-[27] for Gaussian uncertainty and [28] for non-Gaussian distributions. Similarly for non-Gaussian distributions, [1] and [29] use scenario optimization to reformulate the chance constraints as a large number of sampled deterministic constraints. All of these planners rely on a learned distribution of human trajectories to evaluate the chance constraint and associated probabilistic safety guarantees are subject to the accuracy of the predicted distribution.

Because of potential errors in learning the distribution, *distributionally robust* planners (e.g., [30]) account for the mismatch between the predicted and real distributions. These methods currently rely on strong assumptions and are orders of magnitudes too slow for online control.

C. Contribution

Under the previous considerations, we propose a data-driven framework for prediction and planning that does not learn the distribution, but rather reintroduces previously observed human trajectories during online planning. In contrast with learning-based methods, we do not need to assume any structure on the distribution and we can provide a real-world safety guarantee on the planned trajectory. We achieve this via scenario optimization [31]. Scenario optimization is a data-based decision framework under uncertainty that leverages recorded samples of a distribution to satisfy chance constraints. Scenario-based optimization was previously used in [1] and [29]. Compared to these works where a model distribution generated the scenarios for optimization, this work replays observed human trajectories to achieve collision-free motion planning. By using real data, the probabilistic safety guarantee in this work applies to the real recorded data rather than an estimated model. Our contributions are:

1) A joint data-driven method (PSR) for prediction and planning under arbitrary uncertainty that provides a real-world safety guarantee for collision avoidance (see Theorem 1), while optimizing planning task performance. Our framework divides the training data into *partitions*. During planning, the current sensor data is mapped to one of the partitions and data in this partition is replayed to represent the uncertainty in the scenario-based planner [1]. The approach is fast to train and query and supports diverse inputs such as contextual information (e.g., road-layout, tracking information, social cues) or pre-processed contextual data (e.g., classifier or autoencoder outputs).
2) We propose a scenario optimization for non-stationary (i.e., context dependent) probability distributions.

The prediction component of PSR attains close to state-of-the-art performance on the ETH/UCY data-set [7], in terms of the Average Displacement Error (ADE) and the Final Displacement Error (FDE). In comparison, our method is simpler, computationally more efficient and includes a planner with a real-world safety guarantee. We demonstrate on a mobile robot in the real-world that the joint prediction and planning framework evades pedestrians, without an initial model of pedestrian motion.

II. Problem Formulation

A. Human Motion Prediction

Accurate prediction of human motion relies on contextual information. We associate with each human a context state $\boldsymbol{x}_{obs}^{\text{full}} \in \mathbb{X}_{obs}^{\text{full}}$, that is partially unobservable from the robot's perspective (e.g., human intentions). We denote the observable subset of the context state by

$$\boldsymbol{x}_{obs} \in \mathbb{X}_{obs} \subseteq \mathbb{X}_{obs}^{\text{full}}, \tag{1}$$

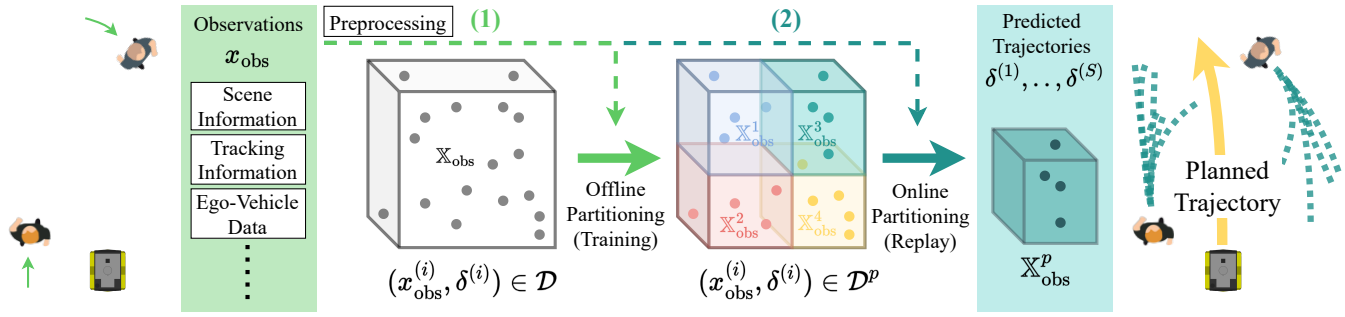


Fig. 2: Schematic overview of PSR. Raw or processed observations are recorded into a dataset $\mathcal{D}$ containing contextual information and associated trajectories. (1) The dataset is partitioned into P partitions. (2) The current contextual information identifies a single partition. Trajectories in this partition are replayed to predict motion of nearby humans.

which is available to the prediction and planning pipeline in each iteration. These observations can contain *deterministic* mappings of sensor data such as classification or autoencoder outputs. The uncertain position of a single human at future time k is denoted by $\boldsymbol{\delta}_k \in \mathbb{R}^2$ and its future N positions by $\boldsymbol{\delta} = \{\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_N\}$. We assume that there exists a probability distribution $\mathbb{P}$ that describes human motion and denote with $\mathbb{P}_x = \mathbb{P}[\boldsymbol{\delta} \mid \boldsymbol{x}_{obs}]$ the probability distribution conditioned on the observed context. Finally, we assume that a dataset,

$$\mathcal{D} = \{(\boldsymbol{x}^{(1)}_{obs}, \boldsymbol{\delta}^{(1)}), \dots, (\boldsymbol{x}^{(D)}_{obs}, \boldsymbol{\delta}^{(D)})\}, \tag{2}$$

containing observed contextual information and the associated realization of the uncertainty (i.e., trajectories) is available. This dataset can be recorded at runtime, accumulating over time, or can be a public dataset recorded in a similar setting (e.g., [7]).

B. Scenario-Based Robot Motion Planning

We control a robot with nonlinear discrete-time dynamics

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{3}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ and $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$ denote the states and inputs, respectively and n_x and n_u are the number of states and inputs, respectively. The vehicle state is assumed to contain its x-y position $\boldsymbol{p} = [x, y] \in \mathbb{R}^2 \subseteq \mathbb{R}^{n_x}$. For simplicity, consider collision avoidance with a single human. A collision avoidance constraint $g(\boldsymbol{x}_k, \boldsymbol{\delta}_k) \leq 0$ imposes that the vehicle does not collide with the human at time k. For example, $g(\boldsymbol{x}_k, \boldsymbol{\delta}_k) = r - ||\boldsymbol{p}_k - \boldsymbol{\delta}_k||_2$, with r the summed radius of robot and human. With the human's future motion uncertain, we may formulate the following Chance Constrained Problem (CCP):

$$\min_{\boldsymbol{u}\in\mathbb{U}, \boldsymbol{x}\in\mathbb{X}} \sum_{k=0}^{N} J(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{4a}$$

$$\text{s.t.} \quad \boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}} \tag{4b}$$

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k), \ k = 0, \dots, N-1 \tag{4c}$$

$$\mathbb{P}\left[\bigwedge_{k=1}^{N} (g(\boldsymbol{x}_k, \boldsymbol{\delta}_k) \leq 0)\right] \geq 1 - \epsilon, \boldsymbol{\delta} \in \Delta, \tag{4d}$$

where formally the joint uncertainty $\boldsymbol{\delta}$ belongs to a probability space $\Delta = \mathbb{R}^{2N}$, associated with a σ-algebra $\mathcal{F}$ and probability measure[1] $\mathbb{P}$, and we use $\bigwedge$ to denote the "and" operation. Our goal is to solve the CCP in Eq. 4 to find robot control inputs $\boldsymbol{u}$ that avoid collisions with humans with a probability of at least ϵ in the real-world.

III. Preliminary - Scenario Program

The CCP in Eq. 4 can be solved via a sampling-based reformulation, known as a Scenario Program (SP) [1]. In the reformulation, one collects an independent set of samples $\{\boldsymbol{\delta}^{(1)}, \dots, \boldsymbol{\delta}^{(S)}\}$ from $\mathbb{P}$ referred to as *scenarios* and formulates a deterministic variant of the constraint (4d) for each sample. This gives the following SP

$$\min_{\boldsymbol{u}\in\mathbb{U}, \boldsymbol{x}\in\mathbb{X}} \sum_{k=0}^{N} J(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{5a}$$

$$\text{s.t.} \quad \boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}} \tag{5b}$$

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k), \ k = 0, \dots, N-1 \tag{5c}$$

$$\bigwedge_{k=1}^{N} \left(g(\boldsymbol{x}_k, \boldsymbol{\delta}^{(i)}_k) \leq 0\right), \ i = 1, \dots, S. \tag{5d}$$

The SP is deterministic and can therefore be solved using a NonLinear Program (NLP) solver. Independent of the probability distribution $\mathbb{P}$, the required number of scenarios S can be computed from a desired risk ϵ (collision avoidance probability), confidence $1 - \beta$ (probability that (4d) is satisfied by the SP) and support n (number of scenarios that affect the solution). A Jupyter notebook to perform this computation is provided with this paper [33].

IV. Partitioned Scenario Replay

The SP in Eq. 5 offers a data-driven way to compute a solution to the CCP in Eq. 4 in the real-world, using recorded observations of human motion. However, it does not consider the context in which these trajectories were recorded. This leads in practice to poor predictions since human motion is strongly context dependent. In the following, we extend the SP to a context dependent distribution that leads to a provably safe prediction and planning framework.

[1]For more details, see [32].

A. Partitioning the dataset

By leveraging on the properties of scenario-based motion planning [1], probabilistic collision avoidance can be ensured in the real-world in two steps: an offline *training* phase and an online *replay* phase. In the training phase, realizations of an uncertainty distribution $\mathbb{P}$ are collected in a dataset $\mathcal{D}$. In the online replay phase, S samples from the dataset are reintroduced in the planner as scenarios $\boldsymbol{\delta}^{(1)}, \ldots, \boldsymbol{\delta}^{(S)}$ for the SP in Eq. 5.

We need to carefully consider the distribution $\mathbb{P}$ and dataset $\mathcal{D}$. Ignoring context, any sample in $\mathcal{D}$ is an independent sample of $\mathbb{P}$. Although this means that we can solve the SP in Eq. 5 by drawing scenarios from $\mathcal{D}$, its solution will be conservative as any sample may be replayed at any time. To incorporate contextual information, the CCP in Eq. 4 should consider the conditioned distribution $\mathbb{P}_x$. The associated SP in Eq. 5 is then constructed by accumulating and replaying samples from $\mathbb{P}_x$. However, as the domain of the observed information $\mathbb{X}_{obs}$ is generally continuous, the probability of observing any particular case, $\boldsymbol{x}_{obs} \in \mathbb{X}_{obs}$ is zero. We therefore cannot accumulate samples from $\mathbb{P}_x$.

We propose instead to partition the space of observed information $\mathbb{X}_{obs}$ into a finite number of subsets or *partitions*. Each partition is constructed such that the probability of observing data belonging to the partition is non-zero. To formalize this idea, we construct $P > 0$ partitions $\mathbb{X}^p_{obs} \subseteq \mathbb{X}_{obs}$ as follows:

$$\bigcup_{p=0}^{P} \mathbb{X}^p_{obs} = \mathbb{X}_{obs} \qquad \text{(Partitioning)} \tag{6}$$

$$\bigcap_{p=0}^{P} \mathbb{X}^p_{obs} = \emptyset \qquad \text{(No Overlap)} \tag{7}$$

$$\mathbb{P}\left[\boldsymbol{x}_{obs} \in \mathbb{X}^p_{obs}\right] > 0. \quad \text{(Density)} \tag{8}$$

Considering the partitioned contextual information, we can formulate the following chance constraint,

$$\mathbb{P}\left[\bigwedge_{k=1}^{N} \left(g(\boldsymbol{x}_k, \boldsymbol{\delta}_k) \leq 0\right) \mid \boldsymbol{x}_{obs} \in \mathbb{X}^p_{obs}\right] \geq 1 - \epsilon, \tag{9}$$

for which the associated SP is given by (5), but where samples of $\boldsymbol{\delta}$ come from a subset of the dataset.

B. Prediction and Planning Algorithm

We propose the following prediction and planning framework, outlined in Algorithm 1. Offline, we take the dataset $\mathcal{D}$ and assign each data point to the associated partition (see (1) in Fig. 2 and lines 4-7 in Algorithm 1), resulting in a partitioned dataset $\mathcal{D} = \bigcup_{p=0}^{P} \mathcal{D}^p$, where

$$\boldsymbol{x}^{(i)}_{obs} \in \mathbb{X}^p_{obs} \;\rightarrow\; (\boldsymbol{x}^{(i)}_{obs}, \boldsymbol{\delta}^{(i)}) \in \mathcal{D}^p. \tag{10}$$

Online, given a currently active partition $\mathbb{X}^p_{obs}$, we satisfy chance constraint (9) by solving the SP in Eq. 5 with samples from $\mathcal{D}^p$ (see (2) in Fig. 2 and lines $8-13$ in Algorithm 1). The risk of the planning and prediction pipeline is then certified in the sense that (9) is satisfied in the real-world.

Algorithm 1: Partitioned Scenario Replay

Input: Dataset $\mathcal{D}$, Observed features $\boldsymbol{x}_{\text{obs}}$, Trajectories $\boldsymbol{\delta}$, Sample size S, Partition threshold S_{th}

1 `// Process new data`
2 **for** *each pedestrian* **do**
3 Append $(\boldsymbol{x}_{\text{obs}}, \boldsymbol{\delta})$ to dataset $\mathcal{D}$
4 `// Partition the dataset (training phase)`
5 **if** $|\mathcal{D}| > S_{th}$ **then**
6 $\mathbb{X}^p_{\text{obs}}\ \forall p \leftarrow$ Size Constrained K-Means($\mathcal{D}$, S)
7 $\mathcal{D}^p \leftarrow$ Partition($\mathcal{D}$, $\mathbb{X}^p_{\text{obs}}$) $\forall p$ (Eq. (10))
8 `// Retrieve scenarios (replay phase)`
9 **for** *each pedestrian* **do**
10 Find p for which $\boldsymbol{x}_{\text{obs}} \in \mathbb{X}^p_{\text{obs}}$ (Assign partition)
11 $(\boldsymbol{\delta}^{(0)}, \ldots, \boldsymbol{\delta}^{(S)}) \leftarrow \mathcal{D}^p$ (Retrieve scenarios)
12 `// Trajectory optimization`
13 $\boldsymbol{x}, \boldsymbol{u} \leftarrow$ Solve SP in Eq. 5 for retrieved scenarios
14

Output: $\boldsymbol{u}_0$

The risk ϵ that can be guaranteed depends on the number of data points in the smallest partition $\operatorname{argmin}_p |\mathcal{D}^p|$, since the sample size cannot be larger than any of the datasets. In practice, we set a desired risk ϵ and compute a sample size S for a given confidence β and support n using Notebook [33]. Then we ensure that each partition is large enough, that is,

$$|\mathcal{D}^p| \geq S, \ \forall p. \quad \text{(Data Requirement)} \tag{11}$$

We obtain the following main result.

Theorem 1: Consider the CCP in Eq. 4 and the associated SP in Eq. 5. Assume that dataset $\mathcal{D}$ containing Independent and Identically Distributed (IID) trajectories and observations is partitioned into P partitions according to partitioning rules (6)-(8) and that for the desired risk ϵ, data requirement (11) is satisfied. Then, the trajectory computed by the SP in Eq. 5 where scenarios are sampled from the current partition p (i.e., $\left\{\boldsymbol{\delta}^{(1)}, \ldots, \boldsymbol{\delta}^{(S)}\right\} \subseteq \mathcal{D}^p$) is collision free in the real-world, in the sense that (9) is satisfied.

Proof: First note that (7) ensures that each observation $\boldsymbol{x}_{\text{obs}} \in \mathbb{X}_{\text{obs}}$ identifies a single partition, while together with (6) it is guaranteed that there is always a single partition for each observation. Hence, one may consider the motion planning problem of the CCP in Eq. 4 as P different motion planning problems (each with its own dataset $\mathcal{D}^p$, collected for the same problem) where one problem is active at each time instance. For each problem, the scenario approach certifies the risk ϵ based on S (and β, n) [31, Theorem 1]. Finally, (11) ensures that S samples can be sampled from each partition, proving the result. ■

In practice, Theorem 1 provides a safety guarantee that helps to understand how safe the predictions are based on the size of the collected dataset. Several observations are in order. First, note that safety and performance are traded-off through the size of the partitions. A partition is safe when (11) holds. With more data available, the partition volumes shrink, leading to more accurate predictions and faster motion plans. More insight can come from the two extreme applications of Theorem 1. If the dataset has S samples, it fits in a single

partition. The planner evades all previously seen human trajectories and will be overly conservative in practice, but safe by Theorem 1. In the other extreme, many low variance partitions with at least S samples exist. It may happen that a new data point lands far away from each partition in which case it is likely that the low variance predictions do not capture the true future motion. Theorem 1 captures this in the risk ϵ. That is, given that we observed S samples in this partition without observing the new sample, the probability of seeing the current case is in the ϵ tail of the distribution.

C. Partitioning Algorithm

Deciding how the observation space $\mathbb{X}_{obs}$ is divided into P partitions, according to partition rules (6)-(8), can be seen as an unsupervised learning problem. In this work, prioritizing simplicity and computational efficiency, we normalize the observations and apply K-means clustering [34] to the resulting data points. K-means partitions the dataset in K clusters, where each data point belongs to the cluster with the nearest mean. To satisfy the data requirement in Eq. 11, we run a size constrained K-means clustering [35] with a minimum cluster size of S (see lines 4-7 in Algorithm 1). Online, we retrieve the active partition $\mathbb{X}^p_{\text{obs}}$ by classifying the current context $\boldsymbol{x}_{\text{obs}}$ (see line 10 of Algorithm 1). We note that Theorem 1 does not assume the partitioning to be static. When a partition has more than S samples, we replay the S samples with the most similar current velocity magnitude.

D. Observations

The observations need to be deterministic, but each observation can be either continuous or discrete. Examples of observations are sensor data (e.g., relative position, velocity, etc.), categorical data (e.g., obstacle is pedestrian) or Boolean data (e.g., has/has-not seen robot). Any deterministic algorithm that pre-processes the data is also admissible. An autoencoder could, for example, encode scene information into a lower dimensional latent space that is used for partitioning.

E. Continual Application

Theorem 1 guarantees probabilistic safety in the real-world for a given risk. While safety is guaranteed, more collected data leads to smaller partitions which, when partitioned effectively, leads to less conservative predictions of human motion. Because of this ability to improve safely, PSR can be deployed without prior data in a real-world environment. In this setting, new observations are continuously recorded and the partition algorithm is repeated at regular intervals (e.g., when dataset is larger than a threshold S_{th}, see line 5 in Algorithm 1). This makes the proposed approach well suited for practical applications where no model is available.

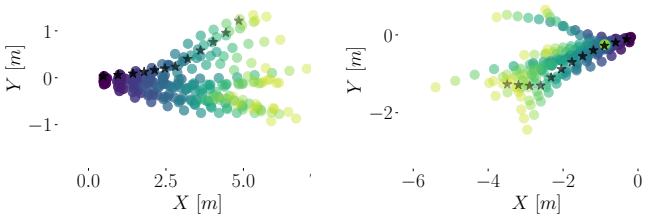
V. Results

A. Comparison with Learning-Based Prediction

We compare PSR prediction against state-of-the-art prediction methods on the ETH dataset for pedestrian motion prediction [7]. The dataset contains 35k+ pedestrian tracks each with 3.2s motion history and 4.8s ground-truth trajectories.

TABLE I: Comparison in ADE and FDE of state-of-the-art prediction models and PSR on the ETH/UCY data set [7].

Method	Metrics	ETH	Hotel	UNIV	ZARA1	ZARA2	AVG
[21] Social GAN	ADE	0.81	0.72	0.60	0.34	0.42	0.58
	FDE	1.52	1.61	1.26	0.69	0.84	1.18
[10] Social-STGCNN	ADE	0.64	0.49	0.44	0.34	0.30	0.44
	FDE	1.11	0.85	0.79	0.53	0.48	0.75
[15] Trajectron++	ADE	0.39	0.12	**0.20**	**0.15**	**0.11**	0.19
	FDE	0.83	0.21	0.44	0.33	0.25	0.41
[18] Y-Net	ADE	0.28	0.10	0.24	0.17	0.13	0.18
	FDE	0.33	0.14	0.41	**0.27**	0.22	0.27
[19] NSP-SFM	ADE	**0.25**	**0.09**	0.21	0.16	0.12	**0.17**
	FDE	**0.24**	**0.12**	**0.38**	**0.27**	**0.20**	**0.24**
PSR	ADE	0.60	0.22	0.41	0.24	0.18	0.33
	FDE	0.94	0.40	0.79	0.41	0.33	0.57
PSR (No History)	ADE	0.68	0.25	0.37	0.24	0.18	0.34
	FDE	1.12	0.44	0.74	0.43	0.34	0.61



(a) Variance captured in regular walking.

(b) Multi-modal trajectory predictions.

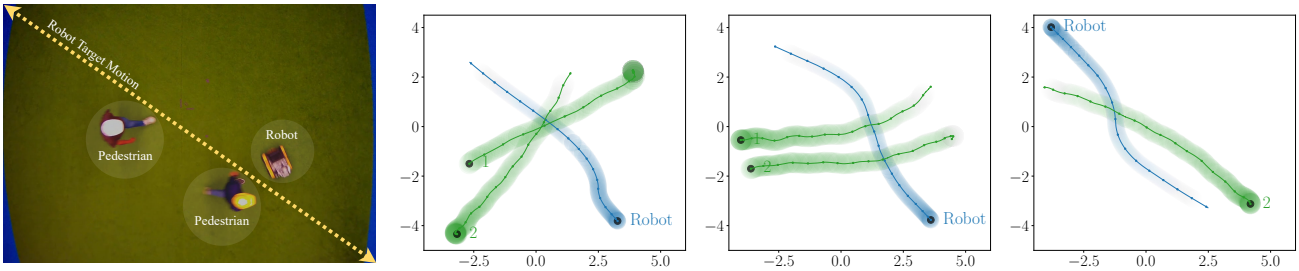
Fig. 3: Examples of PSR predictions on the ETH dataset, showing 20 samples. Predicted trajectories are drawn from the first step ahead (purple) to the final time step (yellow). The ground-truth is depicted by black stars. Positions are drawn with increasing transparency along the time horizon.

Each trajectory contains 20 steps with a 0.4s time step. The data is split in 5 smaller datasets where one dataset is used as test set and the others are available for training. We incorporate the validation set in the training set, since the validation set is not required for PSR.

Evaluation Metrics: We evaluate motion predictions with two metrics, similarly to prior works [15], [18], [19], [21]:

1) *Average Displacement Error (ADE)*: the average ℓ_2 distance between the ground truth and predicted trajectories.
2) *Final Displacement Error (FDE)*: the ℓ_2 distance between the ground truth and predicted trajectories at the prediction horizon T.

To make the motion uncertainty invariant to absolute position and orientation, we represent velocities v_k in the frame positioned at the pedestrian center and oriented forward. The features we use for comparison are the velocity x and y component over the past 3 steps (we found that more steps do not improve performance). Additionally, we run an ablation study with the last velocity only. Example predictions of PSR are depicted in Fig. 3 and quantitative results are listed in Table I. PSR achieves close to state-of-the-art performance. The gap to the state-of-the-art is partially due to the inherent safety guarantee that accounts for less likely outcomes, degrading average performance. PSR could be further improved by using encoded scene information (e.g., an autoencoder) on top of the agent's velocity information.



(a) Experimental setup. (b) Evasion in between. (c) Evasion in front. (d) Pedestrians turns.

Fig. 4: Experimental setup and three observed trajectories of the robot (blue) and pedestrians (green) towards the end of the experiment. Trajectories are depicted with increased transparency over time and start positions are indicated by a black dot.

A key feature of PSR is its computational efficiency given that PSR simply loads samples in its partition for each pedestrian. For 8 obstacles and $S = 528$, PSR takes on average 0.94 ms to partition and sample trajectories. This is significantly faster than other methods (e.g., [15] takes roughly 100 ms just to fit the distribution). Existing methods are typically limited to the order of 10 samples. The planner therefore cannot use the predicted distribution in detail even if the distribution is learned accurately.

B. Real-world evaluation

We demonstrate PSR (see Algorithm 1) on a mobile robot (Clearpath Jackal) navigating among pedestrians using the continual PSR approach described in Sec. IV-E. The experimental setup is depicted in Fig. 4a. We mimic an open environment by asking the pedestrians to move from one side to another, standing still for a brief time afterwards. We do not save data when a pedestrian is standing still. The robot's task is to drive from corner to corner while avoiding collisions with the pedestrians. To implement this behavior, the cost $J(\boldsymbol{x}_k, \boldsymbol{u}_k)$ of the SP in Eq. 5 includes contouring and lag terms (see [25]) that track a diagonal path, a term $||v - v_{\text{ref}}||_2^2$ to track a velocity of 2 m/s, and penalties $||a||_2^2$ and $||\omega||_2^2$ on the acceleration and rotational velocity, respectively.

We detect the robot and pedestrian positions with a marker-based tracking system. The experiment ran for 45 minutes. Pedestrian trajectories are continually collected and the dataset is partitioned whenever its size increased by 10%. We save pedestrian trajectories every 10 steps to ensure that samples are independent. As observations, we use for simplicity the x, y components of the previous three velocities and the current velocity magnitude. We solve the SP in Eq. 5 online for linearized constraints (see [1]) with Forces Pro [36]. Experimental settings are listed in Table II.

Fig. 4 depicts three experiments. Figs. 4b and 4c show that the robot avoids collisions, while we did not encode any model for the pedestrians. In Fig. 4d the pedestrian turns towards the robot, but the planner still evades the pedestrian smoothly. This indicates that the predicted distribution captures the deviation in behavior. Partitions at two time instances are depicted in Fig. 5. After 45 minutes the partitions capture *regular walking at different speeds* (nrs. $1, 2, 6$), *fast walking at different speeds* (nrs. $3, 7$) and *moving from stand still* (nrs. 4 and 8). This shows that partitions reduce in variance over time.

On the planner side, planning performance can be further improved by guiding the robot into the most suitable local optimum (e.g., using global dynamic guidance [37]), but this is outside the scope of this paper.

TABLE II: Experimental settings with T_s the control time step and T_{int} the timestep of predictions.

ϵ	β	$\bar{n}$	S	T_s	T_{int}	N
0.25	0.01	5	101	0.05s	0.1s	30

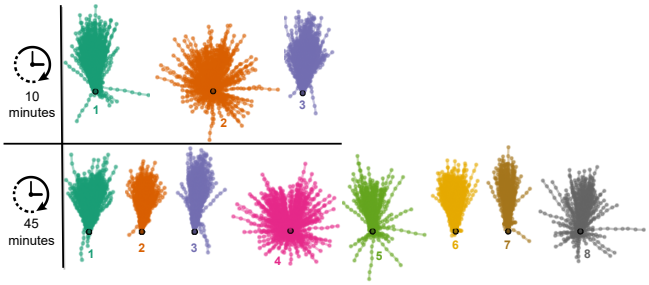


Fig. 5: Trajectories in the partitions at two time instances.

VI. Conclusion

We presented a data-driven framework for human motion prediction and planning where we collected and categorized observed trajectories into several partitioned datasets. During planning, trajectories from one partition were replayed for each human to predict their future motion. This allowed us to provide probabilistic safety guarantees on collision avoidance in the real-world. We showed that PSR attained close to state-of-the-art prediction performance, while providing a safety guarantee. We then deployed PSR on a mobile robot and navigated successfully around pedestrians in the real-world even when starting the framework without prior data.

Our future work will combine learning-based context processing with PSR to generalize its applicability.

REFERENCES

[1] O. de Groot, L. Ferranti, D. Gavrila, and J. Alonso-Mora, "Scenario-Based Motion Planning with Bounded Probability of Collision," July 2023, arXiv:2307.01070 [cs]. [Online]. Available: http://arxiv.org/abs/2307.01070

[2] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. Dietmayer, "IMM object tracking for high dynamic driving maneuvers," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 825–830, 2004.

[3] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang, "Vehicle Trajectory Prediction by Integrating Physics- and Maneuver-Based Approaches Using Interactive Multiple Models," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5999–6008, 7 2018.

[4] 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 *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 3931–3936, iSSN: 2153-0866.

[5] G. Best and R. Fitch, "Bayesian intention inference for trajectory prediction with an unknown goal destination," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2015, pp. 5817–5823.

[6] M. N. Finean, L. Petrović, W. Merkt, I. Marković, and I. Havoutis, "Motion planning in dynamic environments using context-aware human trajectory prediction," *Robotics and Autonomous Systems*, vol. 166, p. 104450, Aug. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889023000891

[7] A. Ess, B. Leibe, and L. Van Gool, "Depth and Appearance for Mobile Scene Analysis," in *2007 IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8, iSSN: 2380-7504.

[8] L. Medsker and L. Jain, *Recurrent Neural Networks: Design and Applications*. CRC Press, 2001.

[9] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces." Las Vegas, NV, USA: IEEE, June 2016, pp. 961–971. [Online]. Available: http://ieeexplore.ieee.org/document/7780479/

[10] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction," Mar. 2020, arXiv:2002.11927 [cs]. [Online]. Available: http://arxiv.org/abs/2002.11927

[11] B. Ivanovic and M. Pavone, "Injecting Planning-Awareness into Prediction and Detection Evaluation," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, June 2022, pp. 821–828.

[12] F. Fang, P. Zhang, B. Zhou, K. Qian, and Y. Gan, "Atten-GAN: Pedestrian Trajectory Prediction with GAN Based on Attention Mechanism," *Cognitive Computation 2022*, vol. 1, pp. 1–10, 6 2022. [Online]. Available: https://link.springer.com/article/10.1007/s12559-022-10029-z

[13] K. Sohn, H. Lee, and X. Yan, "Learning Structured Output Representation using Deep Conditional Generative Models," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[14] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 12 2013. [Online]. Available: https://arxiv.org/abs/1312.6114v10

[15] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data," *arXiv:2001.03093 [cs]*, Jan. 2021, arXiv: 2001.03093. [Online]. Available: http://arxiv.org/abs/2001.03093

[16] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A Recurrent Latent Variable Model for Sequential Data," in *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/b618c3210e934362ac261db280128c22-Abstract.html

[17] B. Brito, H. Zhu, W. Pan, and J. Alonso-mora, "Social-vrnn: one-shot multi-modal trajectory prediction for interacting pedestrians," in *2020 Conference on Robot Learning (CoRL)*, 2020.

[18] K. Mangalam, Y. An, H. Girase, and J. Malik, "From Goals, Waypoints & Paths To Long Term Human Trajectory Forecasting," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 15 213–15 222. [Online]. Available: https://ieeexplore.ieee.org/document/9709992/

[19] J. Yue, D. Manocha, and H. Wang, "Human Trajectory Prediction via Neural Social Physics," Mar. 2023, arXiv:2207.10435 [cs]. [Online]. Available: http://arxiv.org/abs/2207.10435

[20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," June 2014, arXiv:1406.2661 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1406.2661

[21] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, June 2018, pp. 2255–2264. [Online]. Available: https://ieeexplore.ieee.org/document/8578338/

[22] P. Dendorfer, S. Elflein, and L. Leal-Taixe, "MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 13 138–13 147. [Online]. Available: https://ieeexplore.ieee.org/document/9711160/

[23] R. Cheng, R. M. Murray, and J. W. Burdick, "Limits of Probabilistic Safety Guarantees when Considering Human Uncertainty," *arXiv:2103.03388 [cs, eess]*, Mar. 2021, arXiv: 2103.03388. [Online]. Available: http://arxiv.org/abs/2103.03388

[24] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It Is Not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction," July 2020, arXiv:2004.02025 [cs]. [Online]. Available: http://arxiv.org/abs/2004.02025

[25] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4459–4466, Oct. 2019.

[26] L. Blackmore, Hui Li, and B. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *Proc. Amer. Control Conf.*, June 2006, pp. 2831–2837.

[27] H. Zhu and J. Alonso-Mora, "Chance-Constrained Collision Avoidance for MAVs in Dynamic Environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 776–783, Apr. 2019.

[28] A. Wang, X. Huang, A. Jasour, and B. Williams, "Fast Risk Assessment for Autonomous Vehicles Using Learned Models of Agent Futures," *arXiv:2005.13458 [cs, stat]*, June 2020.

[29] O. de Groot, B. Brito, L. Ferranti, D. Gavrila, and J. Alonso-Mora, "Scenario-Based Trajectory Optimization in Uncertain Dynamic Environments," *IEEE Robot. Autom. Lett.*, pp. 5389 – 5396, 2021.

[30] K. P. Wabersich and M. N. Zeilinger, "Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning," *arXiv:1812.05506 [cs]*, Apr. 2019, arXiv: 1812.05506. [Online]. Available: http://arxiv.org/abs/1812.05506

[31] M. C. Campi, S. Garatti, and F. A. Ramponi, "A General Scenario Theory for Nonconvex Optimization and Decision Making," *IEEE Trans. Automat. Contr.*, vol. 63, no. 12, pp. 4067–4078, Dec. 2018.

[32] P. Billingsley, *Probability and Measure*, 3rd ed., 1995.

[33] O. de Groot, "Jupyter Notebook for Scenario Optimization," Aug. 2023. [Online]. Available: https://colab.research.google.com/drive/1Ur20Oqx4WzUT3rNU-Gm1g0QoZ5ChufR_?usp=sharing

[34] X. Jin and J. Han, "K-Means Clustering," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2010, pp. 563–564. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_425

[35] J. Levy-Kramer, "k-means-constrained," Aug. 2023. [Online]. Available: https://github.com/joshlk/k-means-constrained

[36] A. Domahidi and J. Jerez, *FORCES Professional*. Embotech AG, July 2014. [Online]. Available: https://embotech.com/FORCES-Pro

[37] O. de Groot, L. Ferranti, D. Gavrila, and J. Alonso–Mora, "Globally Guided Trajectory Planning in Dynamic Environments," in *IEEE Int. Conf. Robot. Autom.*, May 2023, pp. 10 118–10 124.