

# Interaction-Aware Sampling-Based MPC with Learned Local Goal Predictions

Walter Jansma, Elia Trevisan, Álvaro Serra-Gómez and Javier Alonso-Mora

**Abstract**—Motion planning for autonomous robots in tight, interaction-rich, and mixed human-robot environments is challenging. State-of-the-art methods typically separate prediction and planning, predicting other agents’ trajectories first and then planning the ego agent’s motion in the remaining free space. However, agents’ lack of awareness of their influence on others can lead to the freezing robot problem. We build upon Interaction-Aware Model Predictive Path Integral (IA-MPPI) control and combine it with learning-based trajectory predictions, thereby relaxing its reliance on communicated short-term goals for other agents. We apply this framework to Autonomous Surface Vessels (ASVs) navigating urban canals. By generating an artificial dataset in real sections of Amsterdam’s canals, adapting and training a prediction model for our domain, and proposing heuristics to extract local goals, we enable effective cooperation in planning. Our approach improves autonomous robot navigation in complex, crowded environments, with potential implications for multi-agent systems and human-robot interaction.

**Dataset, Prediction Model, Video and Code available at:**  
[autonomousrobots.nl/pubpage/IA\\_MPPI\\_LBM.html](https://autonomousrobots.nl/pubpage/IA_MPPI_LBM.html)

## I. INTRODUCTION

Cities characterized by dense networks of urban canals, such as Amsterdam, could greatly benefit from deploying Autonomous Surface Vessels (ASVs) for various tasks including deliveries, transportation of people, and garbage collection [1]. However, navigating autonomously in urban canals amidst mixed human-robot crowds presents a significant challenge. Urban canals are typically narrow, frequently congested, and lack the structured nature of roads. While not as strictly enforced as on roads, navigation principles like right-of-way and right-hand conventions should still be considered. Thus, akin to autonomous ground robots among pedestrian crowds, successful navigation in urban canals relies on cooperation and awareness of interactions [2].

Recently, a sampling-based Model Predictive Control (MPC) called Interaction-Aware Model Predictive Path Integral (IA-MPPI) control has been developed for generating cooperative motion plans in urban canals among multiple non-communicating vessels while maintaining awareness of navigation rules [3]. This algorithm assumes rational and homogeneous agents, exact sensing of states, and knowledge of local goals. In real-time, the algorithm samples

This research is supported by the project “Sustainable Transportation and Logistics over Water: Electrification, Automation and Optimization (TRiLOGy)” of the Netherlands Organization for Scientific Research (NWO), domain Science (ENW), and the Amsterdam Institute for Advanced Metropolitan Solutions (AMS) in the Netherlands.

The authors are with the Cognitive Robotics Department, TU Delft, [walter.jansma@gmail.com](mailto:walter.jansma@gmail.com), [e.trevisan@tudelft.nl](mailto:e.trevisan@tudelft.nl), [a.serragomez@tudelft.nl](mailto:a.serragomez@tudelft.nl), [j.alonsomora@tudelft.nl](mailto:j.alonsomora@tudelft.nl)

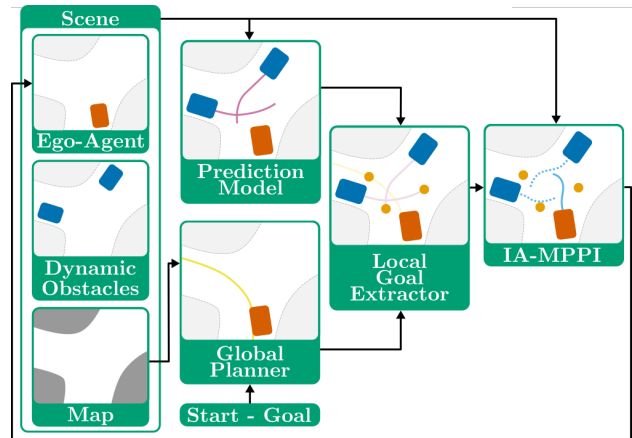


Fig. 1: Overview of the proposed framework. Firstly, the prediction model utilizes information from all elements in the scene to forecast trajectories for obstacle agents. Meanwhile, the global planner, equipped with the map, start, and goal positions, generates a path for the ego agent. Subsequently, the local goal extractor leverages this information to determine appropriate local goals for the motion planner. With inputs derived from the scene and the local goals, the Interaction-Aware Model Predictive Path Integral (IA-MPPI) algorithm simultaneously plans and predicts input sequences for all agents in the scene. The first input of the sequence is then assigned to the ego agent, and the algorithm iterates.

thousands of input sequences to approximate the optimal input sequence that enables all agents to progress toward their goals cooperatively. In scenarios where the local goals of other vessels are unavailable, such as in mixed human-robot environments or due to lack of communication, this previous approach has approximated these goals using a constant velocity model over a given horizon. However, in narrow and crowded environments, vessels often need to execute complex maneuvers to navigate tight intersections and avoid collisions while adhering to navigation rules. In such situations, relying solely on a constant velocity approximation can lead to inaccurate predictions, which can adversely affect the performance of the motion planner in terms of deadlocks, collisions, navigation rule violations, traveled distance, and travel time.

In this paper, we present a framework (see Fig. 1) that utilizes a learning-based trajectory prediction method to improve the estimation of agents’ intended destinations. We introduce heuristics to extract local goals from the predicted trajectories and provide the motion planner with the flexibility to influence the behavior of other agents while expecting cooperation in collision avoidance.

## A. Related Work

Robot motion planning in dynamic environments is a challenging problem for which a series of classical and

heuristic-based approaches have been developed [4], such as the Dynamic Window Approach [5] or Reciprocal Velocity Obstacles [6], [7]. Despite their successful applications, e.g. to non-holonomic robots [8] or vessels in open waters [9], the motions planned by this class of methods are often reactive. This, especially in crowded environments, can lead to unsafe and unpredictable behaviors.

Model Predictive Control (MPC) has become a popular approach to trajectory planning for autonomous vehicles [10] because of its ability to optimize accounting for the system’s dynamics and constraints. Moreover, by planning over a sufficiently large horizon, MPC can anticipate dynamic obstacles resulting in trajectories that are less reactive. To anticipate other agents, however, the free space over the entire planning horizon needs to be computed [11], which requires knowledge about other agents’ positions in the future. If all the agents in the environment are autonomous, communication and distributed optimization can be used to plan trajectories in multi-agent environments [12]. In mixed human-robot environments, however, such communication is not possible and predictions of the future motion of the other agents have to be employed. For instance, recent work on MPC for rule-aware navigation in urban canals uses constant velocity to model the future behavior of other vessels [13].

In interaction-rich scenarios, however, constant velocity can be an inaccurate approximation which may lead to unsafe motion plans [14]. Therefore, several works rely on learning-based models to predict the future motion of other agents [15] and can include prediction confidence [16] and multimodality [17]. These methods, however, decouple prediction and planning which, in high-interaction environments, may lead the ego agent to wrongly assume that no collision-free path exists [18]. To avoid the so-called freezing robot problem the robot has to expect cooperation in collision avoidance from the other agents [19]. Coupled prediction and planning can be done with MPC by modeling the interacting agents as a system, but it quickly becomes expensive to solve via constrained optimization leading to long computation times and short planning horizons [20], [21].

Building upon a novel sampling-based Model Predictive Control (MPC) framework [22], Interaction-Aware Model Predictive Path Integral (IA-MPPI) control [3] has successfully demonstrated decentralized coupled predictions and planning in real-time, accommodating long prediction horizons, nonlinear dynamics, and discontinuous cost functions in multi-agent environments. While IA-MPPI has exhibited superior performance compared to optimization-based MPC approaches that rely on fixed predictions of other agents’ motion, it necessitates knowledge of their near-term local goals, which can either be communicated or estimated.

### B. Contribution

This paper presents a novel framework for interaction-aware decentralized motion planning in urban canals without relying on communication. Our framework encompasses the following contributions:

- **Realistic Dataset:** We generate and publish a realistic dataset of simulated rule-abiding vessel trajectories in real sections of Amsterdam’s urban canals.
- **Learning-Based Trajectory Prediction:** We adapt a pedestrian prediction model [23] to vessels and train it specifically for urban canals. This approach enables us to generate trajectory predictions for other agents.
- **Local Goal Extraction:** We propose heuristics to extract local goals from the predicted trajectories, thereby providing the motion planner with information about where agents intend to go.
- **Communication-Free Coupled Prediction and Planning:** By combining the local goal extraction with the IA-MPPI control [3], we achieve coupled prediction and planning without the need for communication. This approach ensures that the ego agent can influence the behavior of other agents while anticipating cooperation in collision avoidance.

We validate our planning framework through extensive simulated experiments, comparing it against baseline approaches and providing insights into the benefits of coupled prediction and planning over decoupled methods. The framework can be adapted to other robot types beyond vessels.

## II. INTERACTION-AWARE MPPI

In this section, we introduce the main ideas of IA-MPPI [3], upon which our proposed framework is built. For details on the method, models used and cost function please refer to the original paper. For insights on the underlying sampling-based MPC, one can refer to the work on Information-Theoretic MPC [22]. In short, IA-MPPI assumes that all the agents are homogenous and rational, i.e. have the same model and cost function. Under this assumption, we can create a large multi-agent system and plan input sequences resulting in cooperative trajectories for the ego agent as well as all the obstacle agents. This being a decentralized planning framework, we then apply the first input of the sequence to our ego agent, observe the environment and plan again. In more detail, IA-MPPI models the ego-agent  $i$  as a discrete-time dynamical system,

$$\mathbf{q}_{i,t+1} = \mathcal{F}(\mathbf{q}_{i,t}, \mathbf{u}_{i,t}) \quad (1)$$

where  $\mathbf{q}_{i,t}$  and  $\mathbf{u}_{i,t}$  are, respectively, the state and the input of the ego-agent at timestep  $t$ . The state  $\mathbf{q}_{i,t} = [\mathbf{p}_{i,t}, \mathbf{v}_{i,t}]$  contains the position and velocity of the agent. IA-MPPI assumes that all agents in the environment are homogenous. The state and the input of the multi-agent system consisting of the ego-agent and the obstacle agents can therefore be stacked, resulting in,

$$\begin{aligned} \mathbf{q} &= [\mathbf{q}_i^\top \quad \mathbf{q}_j^\top]^\top, \quad \forall j \in \mathcal{M} \setminus i, \\ \mathbf{u} &= [\mathbf{u}_i^\top \quad \mathbf{u}_j^\top]^\top, \end{aligned} \quad (2)$$

where  $(\cdot)_j$  is a variable that the ego-agent  $i$  estimates of agent  $j$  and  $\mathcal{M} = \{0, 1, \dots, m\}$  is the set of all agents in the scene. By also stacking the state transition functions  $\mathcal{F}$  over all agents, we obtain a model for the multi-agent system

$\mathbf{q}_{t+1} = \mathcal{G}(\mathbf{q}_t, \mathbf{u}_t)$ . Given a planning horizon  $T$  and a prior input sequence  $\mathbf{U} = [\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{T-1}]$ , IA-MPPI samples  $K$  input sequences for the entire multi-agent system,

$$\tilde{\mathbf{U}}_k = [\tilde{\mathbf{u}}_{0,k}, \tilde{\mathbf{u}}_{1,k}, \dots, \tilde{\mathbf{u}}_{T-1,k}], \quad \tilde{\mathbf{u}}_{t,k} = \mathcal{N}(\mathbf{u}_t, \nu \Sigma) \quad (3)$$

with  $k = 1, \dots, K$ , variance  $\Sigma$  and scaling parameter  $\nu$ . At the first iteration, the prior input sequence  $\mathbf{U}$  is initialized at zero. By the end of this section, it will become clear how this prior input sequence is updated in subsequent iterations. Having a model for the multi-agent system, we can forward simulate the  $K$  input sequences into  $K$  state trajectories  $\mathbf{Q}_k$  for the multi-agent system,

$$\mathbf{Q}_k = [\mathbf{q}_0, \mathcal{G}(\mathbf{q}_0, \tilde{\mathbf{u}}_{k,0}), \dots, \mathcal{G}(\mathbf{q}_{k,T-1}, \tilde{\mathbf{u}}_{k,T-1})]. \quad (4)$$

Each of the resulting state trajectories is evaluated with respect to both an agent-centric cost as well as a system-wide cost, resulting in a total sample cost  $S_k$ . The reader can refer to the original publication for details on the cost function [3]. For the scope of our paper, it is important to know that the agent-centric cost includes a tracking cost to encourage progress towards a local goal  $p_g$  computed as,

$$C_{\text{tracking}} = k_{\text{tracking}} \frac{\|\mathbf{p}_g - \mathbf{p}_t\|_2}{\|\mathbf{p}_g - \mathbf{p}_{t_0}\|_2}, \quad (5)$$

where  $p_t$  is the position of the agent at timestep  $t$ ,  $p_{t_0}$  is the position of the agent at the beginning of the planning horizon and  $k_{\text{tracking}}$  is a tuning parameter. Notice that we need to know the position of the local goal of each agent. For the ego agent, the local goal is extracted from a global plan. For all the other agents, the local goal has to be either communicated or estimated. We propose in the following section how this goal can be estimated. Once  $S_k$ ,  $\forall k \in [1, \dots, K]$  has been computed, importance sampling weights  $w_k$  can be calculated as,

$$w_k = \frac{1}{\eta} \exp\left(\frac{-1}{\lambda} (S_k - S_{\min})\right), \quad \sum_{k=0}^{K-1} w_k = 1, \quad (6)$$

where  $S_{\min}$  is the minimum sampled cost,  $\eta$  a normalization factor and  $\lambda$  a tuning parameter. We then compute an approximation of the optimal control sequence through a weighted average of the sampled control sequences,

$$\mathbf{U}^* = \sum_{k=0}^{K-1} w_k \tilde{\mathbf{U}}_k \quad (7)$$

and apply the first input  $\mathbf{u}_{i,0}^*$  to the ego-agent. We can now use a time-shifted version of  $\mathbf{U}^*$  as the prior input sequence  $\mathbf{U}$  to warm-start the sampling strategy at the next iteration.

### III. PREDICTING GOAL POSITIONS

In Fig. 1 we provide an overview of the proposed framework. In Section III-A, we outline the prediction model. In Section III-B, we describe the dataset we have collected to train a prediction model that is interaction and rule-aware. In Section III-C, we present the steps taken to port the prediction model to urban vessel environments. In Section III-D, we propose a heuristic to extract a local goal suitable for IA-MPPI using the predicted trajectories.

#### A. Interaction-aware trajectory prediction method

Our approach leverages interaction-aware trajectory prediction for goal estimation. We employ an adapted version of *Social-VRNN* [23], which was originally designed for pedestrians, to obtain trajectory predictions. However, we remark that our framework is agnostic to the choice of trajectory predictor as long as it accounts for obstacles and interactions between agents in the environment.

Social-VRNN [23] is an interaction-aware trajectory prediction method that leverages a generative model based on Variational Recurrent Neural Networks (VRNNs) [24]. The model combines three types of contextual cues to define a joint representation of an agent's current state: information on the past trajectory of the agent of interest, environment context, and agent-agent interactions. The input to predict the trajectory of agent  $i$  is denoted as:

$$\mathbf{x} = \{\mathbf{v}_{-T_0:0}^i, \mathbf{O}_{env}^i, \mathbf{O}_{int}^{-i}\}, \quad (8)$$

where  $\mathbf{v}_{-T_0:0}^i$  corresponds to the sequence of velocity states over the previous observed horizon  $T_O$  of the agent of interest  $i$ . The environment information  $\mathbf{O}_{env}^i$  is represented in the form of a grid map extracted around the agent of interest. Then,  $\mathbf{O}_{int}^{-i}$  represents the information on agent-agent interactions. It is a vector with the relative positions and velocities of all other agents from agent  $i$ 's perspective, listed in ascending order based on the absolute distance to it. The output of the model is a sequence of velocity probability distributions represented by  $T_H$  diagonal gaussian distributions  $\mathcal{N}(\mu_{\mathbf{v},k}, \text{diag}(\sigma_{\mathbf{v},k}^2))$ . For details on the method and its architecture, please refer to the original paper [23].

#### B. Artificial Dataset

In the absence of a publicly available dataset for short-term vessel trajectory prediction, an artificial dataset of vessel interactions is collected in a simulation environment. In order to obtain trajectories that resemble those of real vessels in urban canals, four real canal section maps in Amsterdam: the Herengracht (HG), the Prinsengracht (PG) and the Bloemgracht (BG) are used to collect data. The Open Crossing (OC) environment is created to collect vessel interactions in open water. Data on an additional environment, the Amstel (AM), is included only for testing our framework's generalization to environments not seen during training. Figure 2 depicts two of these canal sections. The yellow rectangles correspond to the areas in which start and goal locations are randomly initialized. These areas are placed around the entire map and in each canal section to improve the diversity of the trajectories and interactions.

To collect the data, more than four thousand experiments are conducted by initializing up to four vessels simultaneously in the mentioned environments. Each vessel is assigned a randomized start and goal location in one of the predefined areas. All sampled locations are ensured to be collision-free. The vessels run a centralized IA-MPPI to sail toward their respective goals while accounting for navigation rules. This ensures that the recorded trajectories are safe, interaction-aware, and mostly rule-abiding.

TABLE I: Specifications of the artificial dataset. *Exp.* refers to the number of experiments done in each scenario. *Frames* and *Vessels* refer to the total number of frames and vessels present in the data set, respectively. All data is recorded at a rate of 10 Hz.

Scenario	Exp.	Frames	Vessels
Herengracht	1000	406229	2499
Prinsengracht	1247	420285	4122
Bloemgracht	1188	372173	3564
Open Crossing	1182	417515	3544
Amstel	79	23468	316
<b>Total</b>	<b>4696</b>	<b>1639670</b>	<b>14045</b>

For each experiment and vessel in the environment we record the current timestamp, the vessel ID, its position and velocity in the global frame. Each timestamp is unique across timesteps and experiments, which enables to identify vessels belonging to the same scene. The specifications of the artificial vessel dataset can be found in Table I. In order to evaluate the prediction model, 10% of the dataset is used as the test set. The remaining data is used for training and is split into a training set (72%) and a validation set (18%). The distribution of data from each scenario is ensured to be equal in all splits.

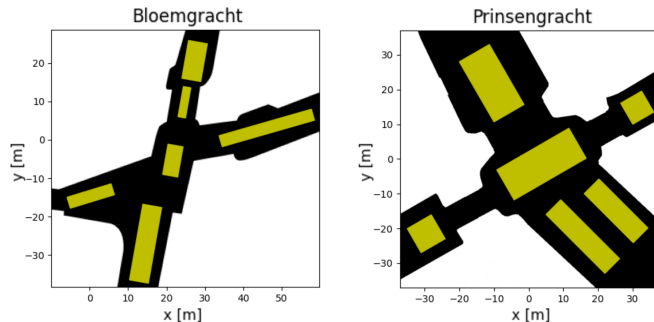


Fig. 2: Canal sections of the Bloemgracht and Prinsengracht. The black areas are the canals. The yellow rectangles correspond to the initialization areas in which goals and starting locations were randomly initialized for each agent during the simulations.

### C. Model Training and Adaptation

We adapt the variational inference architecture presented in [23] to generate unimodal trajectory probability predictions of vessels. In contrast to humans, vessels are slower and have lower-order dynamics, which results in less reactive behaviors and smoother trajectories. To take this into account and avoid overfitting to the dataset, we reduce the dimensionality of the method’s latent space. We also add an L2-regularization term to the loss function and weight it with a hyperparameter we define as  $\gamma$ .

1) *Hyperparameters*: The model is trained using back-propagation through time and the RMSProp [25] optimizer. With a time step of  $\Delta T = 0.4$  seconds, the prediction horizon is set to  $T_H = 24$  steps (9.6 seconds) and the previous horizon to  $T_O = 14$  steps (5.6 seconds). Furthermore, we employ learning rate starting at  $\alpha = 1e-4$  that decays by a factor of 0.9 after every gradient step. The regularization weight is kept at  $\gamma = 0.0001$ . Finally, the model is trained for  $4e4$  training steps, using early stopping.

### D. Local Goal Extraction

In eq. (5) we show that the IA-MPPI needs to know the local goal  $p_g$  of each agent. There are two requirements for a goal to be suitable: it has to lie within a radius  $r_{p_g}$  from the agent it corresponds to and cannot be in space occupied by static obstacles. Therefore, we first search the predicted trajectory backward until we obtain a position  $p_{\leq r_{p_g}}$  within the desired radius. If  $p_{\leq r_{p_g}}$  is in collision with a static obstacle, we construct a circle centered on the agent’s position  $p_a$  with radius  $p_a - p_{\leq r_{p_g}}$  and find the point on the circle closest to  $p_{\leq r_{p_g}}$  which is not in collision with static obstacles. This goal extraction method is illustrated in Fig. 3. Once the goals for all agents are predicted, IA-MPPI can plan interaction-aware trajectories in a decentralized fashion.

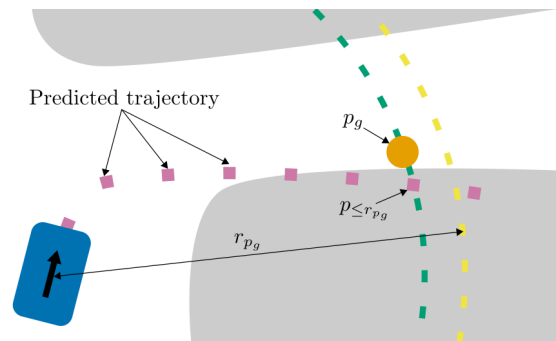


Fig. 3: A visual illustration of how the local goal is extracted from a colliding trajectory prediction.

## IV. EXPERIMENTS

The experiments are conducted in real maps of Amsterdam’s canals, namely the Herengracht (HG), Bloemgracht (BG), Prinsengracht (PG), and the Amstel (AM). In addition, experiments are conducted in an Open Crossing (OC) map without static obstacles. In Section IV-A we evaluate the prediction model, in Section IV-B we show the performances of the proposed framework for motion planning, and in Section IV-C we highlight the benefits of coupled prediction and planning with respect to a decoupled approach.

### A. Prediction Accuracy

In Fig. 4 we compare the proposed Learning-Based Model (LBM) to a Constant Velocity Model (CVM) on test data. We evaluate the methods against the displacement error at each prediction step, which is defined as the Euclidean distance between a prediction and the ground truth. In all maps the LBM outperforms the CVM, showing a lower average displacement error and a smaller standard deviation. Note that the Amstel map was previously unseen during training, demonstrating generalization capabilities.

### B. Interaction-Aware Motion Planning with Predictions

In this study, we evaluate the performance of the proposed decentralized framework that uses a Learning-Based prediction Model to extract local goals (IA-MPPI-LBM), by comparing it against a decentralized approach that extracts

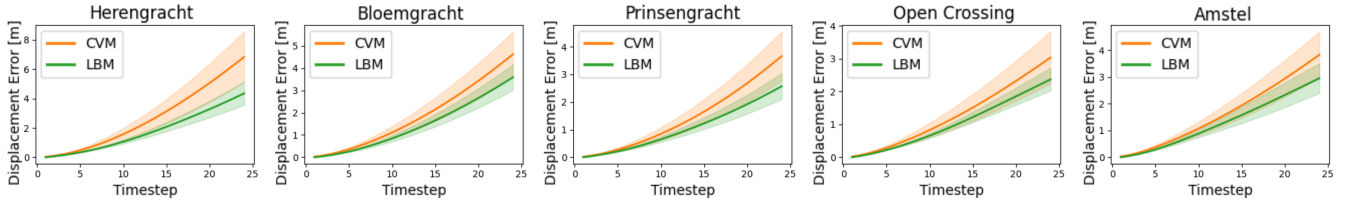


Fig. 4: The displacement error of the predictions from CVM and the LBM over the prediction horizon for each canal section. The solid line represents the mean error and the shaded area represents 30% of the standard deviation.

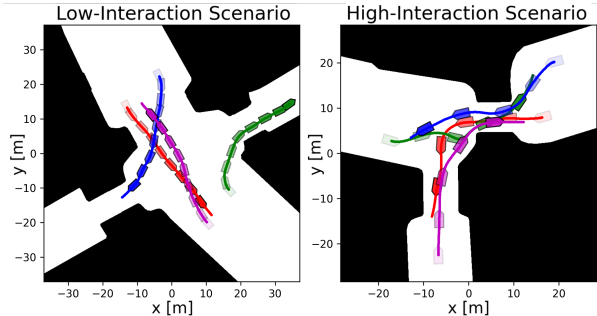


Fig. 5: Examples of experiments in the low-interaction scenario (left) and high-interaction scenario (right).

local goals from a Constant Velocity Model (IA-MPPI-CVM) and decentralized with communication (IA-MPPI-w/comm.), which assumes perfect knowledge of other agents' local goals. It is important to stress that, in similar experiments, the IA-MPPI-CVM which serves as the communication-free baseline in our comparisons has already been demonstrated to outperform an optimization-based Model Predictive Control (MPC) approach that relies on fixed predictions [3].

In the simulated experiments taking place in real sections of the canals of Amsterdam, we randomize the initial positions and goals of four interacting agents, all running the same algorithm. To challenge each method, we design regions within which each agent's start and goal position are randomly initialized in a way that forces all four agents to interact in a narrow section of the map. These *high-interaction* scenarios are discussed in Section IV-B.1.

For completeness, we also design experiments where agents' starting and goal positions are randomized across much larger spaces. In these experiments, however, vessels don't often interact and usually have larger free spaces to avoid each other. These *low-interaction* scenarios are discussed in Section IV-B.2.

An example of experiments in low- and high-interaction scenarios is shown in Fig. 5. The IA-MPPI plans with a time horizon  $T$  of 100 time steps with step size  $\delta T = 0.1s$  and  $K = 4500$  samples. Each method is evaluated on the same set of randomly initialized experiments. For fairness, metrics such as rule violations, goal displacement error, total traveled distance, and time are only displayed for experiments that ended successfully with all methods.

1) *High-Interaction Scenario*: The experiments in high-interaction scenarios are conducted in narrow intersections in the Bloemgracht, Herengracht, and Prinsengracht. Since the Amstel canal is very wide and the Open Crossing has no static map constraints, it is difficult to generate experiments

TABLE II: Successes (Succ.), Deadlocks (Deadl.), Collisions (Coll.), Rule Violations (Rule Viol.) and Goal Displacement Error (Goal DE) for all methods in high-interaction scenarios per canal sections.

	Method	Succ. / Deadl. / Coll.	Rule Viol.	Goal DE
HG	IA-MPPI-CVM	18 / 0 / 2	16	5.82 m
	IA-MPPI-LBM (ours)	19 / 1 / 0	16	5.26 m
	IA-MPPI-w/comm.	20 / 0 / 0	16	
PG	IA-MPPI-CVM	19 / 0 / 1	11	7.30 m
	IA-MPPI-LBM (ours)	19 / 1 / 0	5	4.11 m
	IA-MPPI-w/comm.	20 / 0 / 0	5	
BG	IA-MPPI-CVM	17 / 0 / 3	7	4.98 m
	IA-MPPI-LBM (ours)	20 / 0 / 0	4	4.13 m
	IA-MPPI-w/comm.	20 / 0 / 0	3	

with high-interactions, and thus these two maps are excluded from this experiment section. The results of the experiments are summarized in Table II and Figure 6. It can be seen that in these high-interaction scenarios, the LBM consistently outperforms the CVM in terms of the goal displacement error (Goal DE). As a consequence, the motion planning framework that estimates other agents' local goals using predictions from the LBM outperforms the framework that uses the CVM on all the metrics. Moreover, we demonstrate our framework with the LBM has negligible performance losses compared to the method with perfect communication.

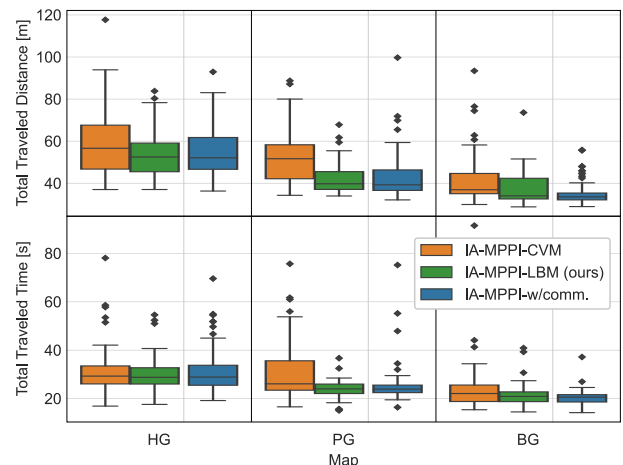


Fig. 6: This figure displays the distribution of the total traveled distance and total traveled time of the vessels during the experiments in the high-interaction scenarios. The results are displayed per map and for each method.

2) *Low-Interaction Scenarios*: Table III and Fig. 7 summarize the results in low-interaction scenarios. Note that we here also test on the Open Crossing maps and the Amstel,

which the LBM has not previously seen in training. The results show that also when the start and goal positions of all agents are randomly initialized over large areas, our proposed communication-free framework with the LBM performs just as well as the baseline with full communication, even in a map unseen in training. However, perhaps unsurprisingly, the framework that approximates the local goals with a CVM can also achieve the same performance as the framework with full communication. Intuitively, in low-interaction scenarios where agents mostly navigate straight to their goal, CVM is a reasonably good approximator.

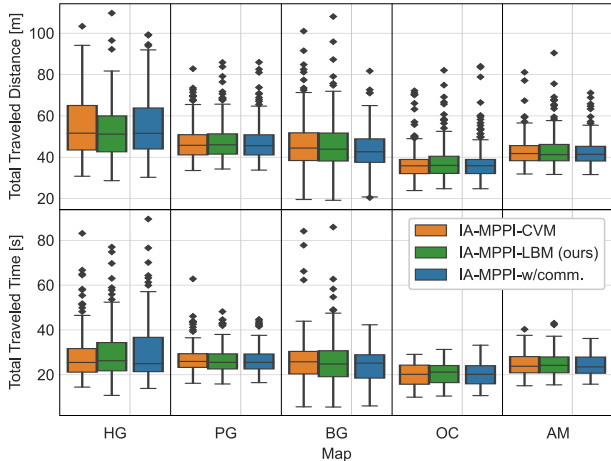


Fig. 7: This figure displays the distribution of the total traveled distance and total traveled time by vessels during the experiments in the low-interaction scenarios. The results are displayed per map and for each method.

### C. Decoupled Prediction and Planning

The framework we proposed utilizes a Learning-Based Model (LBM) to predict trajectories for obstacle agents and extract local goals while employing Interaction-Aware Model Predictive Path Integral (IA-MPPI) for coupled predictions and planning. To assess the advantages of this framework, we compare it to a planner without interaction awareness (MPPI-LBM), which decouples prediction and planning. Like other state-of-the-art methods, MPPI-LBM treats the predicted future trajectories of obstacle agents as occupied space and plans the ego agent’s motion without considering interaction awareness. This approach reduces the system size and computational burden by minimizing the space to be sampled. However, apart from this difference, MPPI-LBM shares the same sampling strategy and cost function as the proposed IA-MPPI-LBM. We conducted 100 low-interaction experiments across the Amstel, Bloemgracht, Herengracht, Open Crossing, and Prinsengracht, comparing different methods. Table IV presents the outcomes, including total successes, deadlocks, collisions, and rule violations. Again, the proposed IA-MPPI-LBM shows similar performances to the method with communication (IA-MPPI-w/comm).

However, MPPI-LBM exhibited a significantly lower success rate and a higher number of rule violations. The LBM, while trained to be somewhat rule- and interaction-aware in its predictions, occasionally struggles to capture complex

TABLE III: Successes (Succ.), Deadlocks (Deadl.), Collisions (Coll.), Rule Violations (Rule Viol.), and Goal Displacement Error (Goal DE) for all methods in the various canal sections.

	Method	Succ. / Deadl. / Coll.	Rule Viol.	Goal DE
HG	IA-MPPI-CVM	38 / 0 / 2	22	5.31 m
	IA-MPPI-LBM (ours)	37 / 0 / 3	26	5.42 m
	IA-MPPI-w/comm.	37 / 2 / 1	26	
PG	IA-MPPI-CVM	38 / 0 / 2	14	3.12 m
	IA-MPPI-LBM (ours)	38 / 0 / 2	13	2.94 m
	IA-MPPI-w/comm.	38 / 0 / 2	14	
BG	IA-MPPI-CVM	38 / 0 / 2	17	5.36 m
	IA-MPPI-LBM (ours)	39 / 0 / 1	20	4.93 m
	IA-MPPI-w/comm.	40 / 0 / 0	20	
OC	IA-MPPI-CVM	40 / 0 / 0	27	3.98 m
	IA-MPPI-LBM (ours)	40 / 0 / 0	29	4.04 m
	IA-MPPI-w/comm.	40 / 0 / 0	27	
AM	IA-MPPI-CVM	40 / 0 / 0	22	3.12 m
	IA-MPPI-LBM (ours)	39 / 1 / 0	18	3.49 m
	IA-MPPI-w/comm.	40 / 0 / 0	21	

reciprocal collision avoidance maneuvers when agents are in close proximity. This, combined with the motion planner’s unawareness of the ego agent’s influence on other agents’ motion and their cooperation in collision avoidance, often led the MPPI-LBM to wrongly assume that no feasible solution existed. Consequently, this resulted in agents drifting into collisions due to their large inertia.

TABLE IV: Successes (Succ.), Deadlocks (Deadl.), Collisions (Coll.), and Rule Violations (Rule Viol.) for the non-interactive MPPI and the IA-MPPI baseline in low-interaction scenarios.

Method	Succ. / Deadl. / Coll.	Rule Viol.
MPPI-LBM.	72 / 1 / 27	40
IA-MPPI-LBM (ours)	97 / 1 / 2	30
IA-MPPI-w/comm.	98 / 0 / 2	28

## V. CONCLUSIONS

In this paper, we introduced a framework that combines a learning-based trajectory prediction model with Interaction Aware MPPI, enabling decentralized and communication-free coupled prediction and planning. Our experimental results demonstrated the superiority of our Learning-Based Model (LBM) over the Constant Velocity Model (CVM) in accurately predicting the trajectories of interacting vessels, even in unseen maps. Through simulated experiments in Amsterdam’s canals, we showed that our motion planning framework achieved comparable performance to a method with ground truth knowledge of local goals, which was shown to outperform classical optimization-based MPC approaches with decoupled prediction and planning in previous work [3]. Additionally, we highlighted the limitations of the CVM in tight environments with multiple interacting agents. Finally, by comparing our approach with a non-interactive planner, we emphasized the advantages of coupled planning and predictions.

## REFERENCES

- [1] W. Wang, T. Shan, P. Leoni, D. Fernández-Gutiérrez, D. Meyers, C. Ratti, and D. Rus, "Roboat II: A Novel Autonomous Surface Vessel for Urban Environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 1740–1747.
- [2] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 797–803.
- [3] L. Streichenberg, E. Trevisan, J. J. Chung, R. Siegwart, and J. Alonso-Mora, "Multi-Agent Path Integral Control for Interaction-Aware Motion Planning in Urban Canals," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 1379–1385.
- [4] M. G. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, Feb. 2018.
- [5] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar. 1997.
- [6] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 1928–1935.
- [7] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Springer, 2011, pp. 3–19.
- [8] J. Alonso-Mora, A. Breitenmoser, M. Ruffi, P. Beardsley, and R. Siegwart, "Optimal Reciprocal Collision Avoidance for Multiple Non-Holonomic Robots," in *Distributed Autonomous Robotic Systems*, ser. Springer Tracts in Advanced Robotics, A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M. A. Hsieh, L. E. Parker, and K. Støy, Eds. Berlin, Heidelberg: Springer, 2013, vol. 83, pp. 203–216.
- [9] Y. Kuwata, M. T. Wolf, D. Zargitsky, and T. L. Huntsberger, "Safe Maritime Autonomous Navigation With COLREGS, Using Velocity Obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, Jan. 2014, conference Name: IEEE Journal of Oceanic Engineering.
- [10] M. Zanon, J. V. Frasch, M. Vukov, S. Sager, and M. Diehl, "Model Predictive Control of Autonomous Vehicles," in *Optimization and Optimal Control in Automotive Systems*, ser. Lecture Notes in Control and Information Sciences, H. Waschl, I. Kolmanovsky, M. Steinbuch, and L. del Re, Eds. Cham: Springer International Publishing, 2014, pp. 41–57.
- [11] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, Apr. 2016, publisher: Institute of Electrical and Electronics Engineers Inc.
- [12] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online Trajectory Generation With Distributed Model Predictive Control for Multi-Robot Motion Planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 604–611, Apr. 2020.
- [13] J. de Vries, E. Trevisan, J. van der Toorn, T. Das, B. Brito, and J. Alonso-Mora, "Regulations Aware Motion Planning for Autonomous Surface Vessels in Urban Canals," in *2022 IEEE International Conference on Robotics and Automation*, May 2022, pp. 3291–3297.
- [14] H. Zhu and J. Alonso-Mora, "Chance-Constrained Collision Avoidance for MAVs in Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.
- [15] H. Zhu, F. M. Claramunt, B. Brito, and J. Alonso-Mora, "Learning Interaction-Aware Trajectory Predictions for Decentralized Multi-Robot Motion Planning in Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2256–2263, Apr. 2021.
- [16] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin, "Confidence-aware motion prediction for real-time collision avoidance1," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 250–265, Mar. 2020, publisher: SAGE Publications Ltd STM.
- [17] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal Probabilistic Model-Based Planning for Human-Robot Interaction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 3399–3406, iSSN: 2577-087X.
- [18] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core Challenges of Social Robot Navigation: A Survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 36:1–36:39, Apr. 2023.
- [19] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, Mar. 2015, publisher: SAGE Publications Ltd STM.
- [20] K. Liu, N. Li, H. E. Tseng, I. Kolmanovsky, and A. Girard, "Interaction-Aware Trajectory Prediction and Planning for Autonomous Vehicles in Forced Merge Scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 474–488, Jan. 2023, conference Name: IEEE Transactions on Intelligent Transportation Systems.
- [21] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for Autonomous Cars that Leverage Effects on Human Actions," in *Proceedings of Robotics: Science and Systems*, Jun. 2016.
- [22] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and I. A. Theodorou, "Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, Dec. 2018.
- [23] B. F. d. Brito, H. Zhu, W. Pan, and J. Alonso-Mora, "Social-VRNN: One-Shot Multi-modal Trajectory Prediction for Interacting Pedestrians," in *Proceedings of the 2020 Conference on Robot Learning*. PMLR, Oct. 2021, pp. 862–872.
- [24] 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.
- [25] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.