

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

8-2022

Trajectory optimization for safe navigation in maritime traffic using historical data

Chaithanya BASRUR

Arambam James SINGH

Arunesh SINHA

Akshat KUMAR

Singapore Management University, akshatkumar@smu.edu.sg

T. K. Satish KUMAR

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Artificial Intelligence and Robotics Commons](#), and the [Databases and Information Systems Commons](#)

Citation

BASRUR, Chaithanya; SINGH, Arambam James; SINHA, Arunesh; KUMAR, Akshat; and KUMAR, T. K. Satish. Trajectory optimization for safe navigation in maritime traffic using historical data. (2022). *Proceedings of the 28th International Conference on Principles and Practice of Constraint Programming, Haifa, Israel, 2022 July 31 - August 5*.

Available at: https://ink.library.smu.edu.sg/sis_research/7718

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylids@smu.edu.sg.

Trajectory Optimization for Safe Navigation in Maritime Traffic Using Historical Data

Chaithanya Basrur ✉

Singapore Management University, Singapore

Arambam James Singh ✉

National University of Singapore, Singapore

Arunesh Sinha ✉

Singapore Management University, Singapore

Akshat Kumar ✉

Singapore Management University, Singapore

T. K. Satish Kumar ✉

University of Southern California, Los Angeles, CA, USA

Abstract

Increasing maritime trade often results in congestion in busy ports, thereby necessitating planning methods to avoid close quarter risky situations among vessels. Rapid digitization and automation of port operations and vessel navigation provide unique opportunities for significantly improving navigation safety. Our key contributions are as follows. *First*, given a set of future candidate trajectories for vessels in a traffic hotspot zone, we develop a multiagent trajectory optimization method to choose trajectories that result in the best overall close quarter risk reduction. Our novel MILP-based optimization method is more than an order-of-magnitude faster than a standard MILP for this problem, and runs in near real-time. *Second*, although automation has improved in maritime operations, current vessel traffic systems (in our case study of a busy Asian port) predict only a *single* future trajectory of a vessel based on linear extrapolation. Therefore, using historical data we learn a *generative model* that predicts *multiple* possible future trajectories of each vessel in a given traffic hotspot, reflecting past vessel movement patterns, and integrate it with our trajectory optimizer. *Third*, we validate our trajectory optimization and generative model extensively using a real world maritime traffic dataset containing 6 million Automated Identification System (AIS) data records detailing vessel movements over 1.5 years from one of the world's busiest ports, demonstrating effective risk reduction.

2012 ACM Subject Classification Computing methodologies → Multi-agent planning

Keywords and phrases Multi-Agent Path Coordination, Maritime Traffic Control

Digital Object Identifier 10.4230/LIPIcs.CP.2022.5

Supplementary Material

Audiovisual (Videos and Appendix): <https://github.com/rlr-smu/TrajOpt>

Funding This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-016).

1 Introduction

Increasing maritime vessel traffic in some of the busiest ports of world such as Tokyo bay and Singapore creates traffic hotspots and increases the risk of closer quarter near-miss situations [17]. Recently, disruptions in global supply chains, and adverse weather events have further endangered the navigational safety by causing unexpected traffic spikes in busy waterways such as Singapore's port [31]. Vessel collisions endanger not only human lives,



© Chaithanya Basrur, Arambam James Singh, Arunesh Sinha, Akshat Kumar, and T. K. Satish Kumar; licensed under Creative Commons License CC-BY 4.0

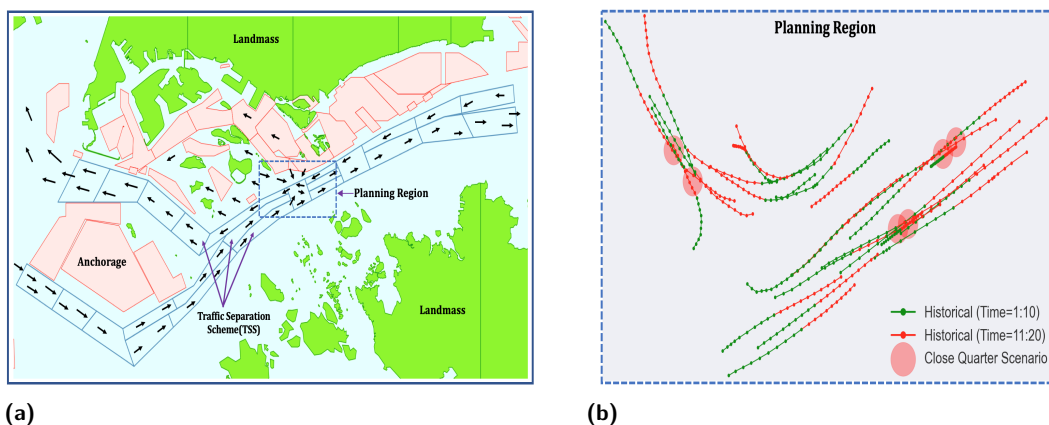
28th International Conference on Principles and Practice of Constraint Programming (CP 2022).

Editor: Christine Solnon; Article No. 5; pp. 5:1–5:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** (a) Electronic navigation chart(ENC) of Singapore strait. ENC is used by vessels for navigation through the strait. Traffic separation scheme (TSS) are the sea lanes through which vessels enter and leave the strait. (b) Enlarged view of planning region from ENC, an instance of a congested scenario in this region is shown. Green and red line denote first 10-step and next 10-step trajectories from historical data respectively. This is the most congested region in the whole strait because many vessels enter and come out of the port through this junction point.

but also endanger the environment by causing oil spills [20]. Therefore, our goal in this work is to study and develop maritime traffic coordination techniques to mitigate close quarter risky situations that may develop in near future, and improve safety of navigation.

Current model of operations, automation in maritime traffic. Most busy ports, such as Singapore’s, have a vessel traffic information system (VTIS) that is manned by port watch operators [19]. Operators keep a close watch over the vessel traffic 24x7 via radars and other sensors, and take action if a risky navigation situation is about to develop in the near future (e.g. in the next 10-15 mins). A key challenge is how to proactively advise involved vessels in a traffic hotspot to avoid the close quarter situation. Based on our discussions with Singapore’s port authority and our physical observation of their VTIS control center, watch operators’ traffic monitoring software predicts how vessels would move over the next 10-15 mins by *linearly* extrapolating their course. And if this linear prediction based trajectories suggest a close-quarter situation developing in the next 10-15 mins, it alerts the watch operator. However, it is left to the watch operator to decide how to advise vessels (e.g., to alter course) to avoid such a close quarter situation. This lack of automated trajectory optimization support creates high cognitive burden for watch operators, and is prone to human error. Therefore, our automated trajectory optimization tool can act as a decision support system for improved safety of maritime navigation.

In addition to improving current VTIS operations, developing automated trajectory optimization methods would also be highly impactful for the future of maritime traffic. E-Navigation [12] introduced by International Maritime Organisation (IMO) aims to improve maritime industry operations by digitizing both vessel navigation and port-based operations including digitizing communications between vessels and VTIS. Such digitization can further enable the usage of automated tools for improved safety of navigation. There are also recent advancements in the space of autonomous ships that have the potential to improve safety in navigation and also reduce costs to the industry [8, 24, 23]. Maritime Autonomous Surface Ships (MASS) [21] is an initiative by the IMO which provides regulations and guidelines on the advancement of technologies in this space. An example use-case of our tool would be

(autonomous) vessels, which are in a traffic hotspot, propose a set of candidate trajectories which they can take in the near future and transmit it to the port authority. Using our trajectory optimization tool, the port authority can then advise vessels to take the least-risky trajectory.

Electronic navigation chart and planning region. Figure 1a shows the electronic navigation chart of Singapore strait. The traffic separation scheme (TSS) are the sea lanes through which vessels enter and leave the strait. Each smaller polygon represents a sea zone. Other areas of interests are marked in different colors such as anchorages, landmass among others. Our area of focus (or *planning* region) highlighted in dotted square is a typical hotspot region. In this region, vessels enter port waters (in pink color) towards berths, outgoing vessels from berths enter TSS, and some vessels transit through the TSS. As a result, this planning region experiences heavy cross traffic with vessels often navigating across traffic separation areas to avoid hotspots.

Figure 1b shows a collection of real historical trajectories for different vessels (tankers and cargos). Each dot in a trajectory shows the corresponding vessel position 1 minute apart. Red circles highlight those locations where vessels are in a close quarter situation (distance between them is less than 500 meters).

Generative trajectory modeling using historical data. Our trajectory optimization tool requires a candidate set of possible future trajectory for each vessel involved in a hotspot region. In the future with increased automation, vessels themselves can digitally compute a number of feasible trajectories they can follow in the near future (next 10-15 mins) and transmit them to the port authority. There are existing tools such as ECDIS [13] for vessel route planning. However, the current vessel-port operations are unable to provide such information. Similarly, current VTIS (such as Singapore port's) predicts only a single trajectory based on linear extrapolation.

To address this, we develop a generative model takes as input the past 10 minute trajectory of each vessel in a specific hotspot area ('time=1:10'), and then predicts a number of possible trajectories for each vessel for the next 10 minutes ('time=11:20'). Our generative model is trained on a large historical dataset, which implies that generated future trajectories are feasible (that is, they do not involve unrealistic manoeuvres such as taking U-turns, or making vessels fully stop). Instead of vessels deciding independently their future trajectory ('time=11:20'), our trajectory optimization module can pick safest possible trajectories, among the ones generated by our generative model, to decrease the risk by maximizing the closest point of approach (CPA) between any two vessels. Note that all the information required for this system is available with the port authority as they monitor movements of all vessels, and based on our optimization tool, can advise vessels to follow a particular trajectory. As traffic is dynamic, another property our tool must have is to produce results in near real-time, and be able to run on a rolling horizon basis.

Contributions. Our key contributions are as follows. First, given a set of candidate future trajectories of vessels in a hotspot region, we develop a mixed-integer linear programming (MILP) based optimization method that can optimize over all possible combinations of future vessel trajectories to minimize the risk of close quarter situations developing. Second, using historical data of vessel movements in Singapore strait, one of busiest port in the world, we learn a deep conditional generative model based on LSTM [11] that can predict multiple possible future trajectories of vessels in a traffic hotspot region. Empirically, our

generative model is trained and evaluated on a real world historical data containing 6 million data points (AIS records detailing vessel traffic data) over 1.5 years. We extensively test several properties of our generative model, such as its ability to generate realistic and diverse trajectories, and show that our method is significantly superior to another generative model called social GAN [10]. We also show that our novel MILP-based trajectory optimization method is more than an order-of-magnitude faster than a standard MILP model for this problem, and can provide solutions in near real-time, a key requirement for the solver.

Generative modeling. The learning of probability distributions from data and ability to sample from them is a fundamental learning task known as generative modelling. Within generative models, a prominent sub-class are techniques that do not explicitly learn the probability distribution (or density) function but are able to generate samples from it. These include the classical Markov chain Monte Carlo methods [3] as well as modern Generative Adversarial Networks [9] (GANs). GAN-based models have also been used for generative trajectory modeling of pedestrians [10, 2, 15]. However, vessel traffic has movement characteristics which are unlike pedestrians' (such no sharp or uturns, prediction over longer time duration among others). Our proposed generative model is also computationally efficient than previous approaches such as SocialGAN [10] which take much longer to train, and often produce worse predictions as we show empirically.

Multi-agent path finding (MAPF). Given a set of agents with unique start and goal locations in a shared environment, the MAPF problem [28] is to find collision-free paths for all agents from their respective start locations to their respective goal locations. MAPF has many real-world applications, including in video games [25], automated warehousing [32], multi-drone delivery [5], and aircraft-towing vehicles [18]. Solving the MAPF problem optimally for either the minimum sum-of-costs or the minimum makespan is NP-hard [34, 16]. Although many MAPF solvers exist, they are not directly applicable in our problem domain for the following reasons. First, MAPF solvers require a discrete search space. They discretize continuous spaces even when they use motion primitives and thereby generate only piecewise smooth paths uncharacteristic of trajectories in our problem domain. Second, MAPF solvers don't constrain branching decisions at intermediate locations and therefore don't reason about historical data required for capturing the complex kinodynamic constraints on trajectories in our problem domain. Third, MAPF solvers are typically interested in avoiding collisions as hard constraints rather than optimizing close quarter risk reduction.

Maritime traffic optimization. Previous works [27, 26] have proposed a reinforcement learning based approach to address the maritime traffic control problem. Their main focus is to optimize the traffic for the entire Singapore Strait. Whereas, our focus is more on the *micro-level* traffic optimization, that is, minimizing the close quarter incidents in the near future. [1, 4, 35, 29] have also addressed the traffic control problem at the micro-level. However the solution methodologies in these approaches *do not* model the uncertainty in the maritime environment, which is an important real world feature. In this work, we capture the uncertainty in vessel movements using our proposed generative model.

2 Problem Formulation and Statement

We tackle the problem of recommending seafaring vessels a navigation route in and around congested ports (e.g., the planning region in figure 1a). The increasing number of vessels over the years has led to an increase in the frequency of collisions and close quarter situations.

Thus, the need of the hour is some intervention from a central port authority to encourage safe navigation around crowded ports. We envision a route recommendation system that suggest routes to vessels involved in a traffic hotspot. A number of competing considerations need to be taken into account for such a system: the recommendations should follow typical paths traversed historically, the path suggested must be easy to execute by the vessel, and achieve a global objective of minimizing the close-quarter risk among vessels.

Problem Statement. To specify the problem formally, we consider a planning horizon H and planning area \mathcal{Z} ; a polygon in sea space. The planning area is typically the port area prone to traffic hotspots, such as where vessels either enter the port waters for berthing and anchoring or pass through to open seas. Vessels enter and leave \mathcal{Z} during the planning period H . At a planning epoch t , we observe a snapshot of the whole planning region which includes information such as, total number of vessels M and their previous trajectories until time t . We use $[M]$ as a shorthand for $\{1, \dots, M\}$. $v \in [M]$ denotes a vessel, $\tau_v^{\text{prv}} = \langle l_t, l_{t-1}, \dots, l_{t-(n-1)} \rangle$ denotes the past n -step trajectory of vessel v at time t , where $l_t = (x_t, y_t)$ is the location on 2d plane. We also assume that time is discretized (e.g., 30 second intervals). The objective of a maritime traffic controller is to recommend future m -steps trajectories τ_v^{rec} for each vessel v so that it minimizes a risk function $\text{risk}(\tau_1^{\text{rec}}, \tau_2^{\text{rec}}, \dots, \tau_M^{\text{rec}})$ given by

$$\text{risk}(\tau_1^{\text{rec}}, \dots, \tau_M^{\text{rec}}) = - \min_{v, v' \in [M], v \neq v'} \left\{ \text{dist}(\tau_v^{\text{rec}}, \tau_{v'}^{\text{rec}}) \right\} \quad (1)$$

where the function dist provides the closest distance between the two input trajectories. Thus, in words, the risk measures the *negative* of the closest point of approach between any two vessels for the recommended trajectories. Minimizing the risk means maximizing the closest point of approach (CPA), which is a standard notion for maritime safety [6].

3 Approach

Our approach to solve the trajectory recommendation problem has two parts. First, given a set of candidate future trajectories for each vessel in a hotspot area, we develop an optimization method that selects trajectories to minimize the risk. Second, we develop a trajectory generation model that generates multiple plausible trajectories for each vessel that can be recommended.

3.1 Vessel Trajectory Optimization

We first formulate the path planning problem as a trajectory optimization problem. Suppose that $\{\tau_v^1, \tau_v^2, \dots, \tau_v^K\}$ are the K future plausible trajectories for a vessel v . There are multiple ways in which such future trajectories can be collected – vessels themselves send future possible trajectories they can follow (e.g., using route planning tools such as ECDIS as noted in Section 1), VTIS can use their own prediction methods, or as in our case, trajectory generation module can be used (described in next sub-section). Importantly, our trajectory optimization method is not dependent on the manner in which such future trajectories are collected.

Let x_v^k be a binary decision variable that denotes whether the trajectory k is selected as the recommendation. Thus, a natural constraint is $\sum_{k=1}^K x_v^k = 1, \forall v \in [M]$ which enforces that only one trajectory is selected per vessel.

■ **Table 1** RiskOPT: Mixed-integer non-linear program for trajectory optimization.

$$\begin{array}{c}
 \max_{\mathbf{x}} \min_{v \in [M], v' \in [M], v \neq v'} \left\{ \sum_{k \in [K], k' \in [K]} x_v^k x_{v'}^{k'} \text{dist}(\tau_v^k, \tau_{v'}^{k'}) \right\} \\
 \text{subject to } \sum_{k=1}^K x_v^k = 1, \quad \forall v \in [M] \\
 x_v^k \in \{0, 1\} \quad \forall v \in [M], k \in [K]
 \end{array}$$

We re-write the risk in terms of all the binary variables x_v^k 's. Let \mathbf{x} denote all the binary variables for all vessels. For defining $\text{risk}(\mathbf{x})$ only those trajectories must be considered that are selected, which we enforce by the bilinear term $x_v^k x_{v'}^{k'}$ below:

$$- \min_{v \in [M], v' \in [M], v \neq v'} \left\{ \sum_{k \in [K], k' \in [K]} x_v^k x_{v'}^{k'} \text{dist}(\tau_v^k, \tau_{v'}^{k'}) \right\} \quad (2)$$

We want to minimize risk, which given the negative sign in risk becomes the integer bilinear optimization problem RiskOPT in Table 1.

Naive Formulation. The problem RiskOPT is non-linear because of the bilinear terms. A naive and standard way of removing the bilinearity is to introduce additional continuous variables $z_{k,k',v,v'} = x_v^k x_{v'}^{k'}$ and constraints $z_{k,k',v,v'} \geq x_v^k + x_{v'}^{k'} - 1$ and $z_{k,k',v,v'} \leq x_v^k$ and $z_{k,k',v,v'} \leq x_{v'}^{k'}$. It can be readily checked that this re-formulation is equivalent to the original one. This reformulation uses $K^2 M^2$ extra variables and $3K^2 M^2$ extra constraints over the original bilinear formulation. However, our planning needs to be almost real time (solve within one minute) and as observed in experiments, this naive reformulation does not meet this requirement. Hence, we present a more compact reformulation that is orders of magnitude faster than the naive one.

Improved Formulation. Observe that the key part of expression (2) can be re-written as:

$$\sum_{k \in [K]} x_v^k \left(\sum_{k' \in [K]} x_{v'}^{k'} \text{dist}(\tau_v^k, \tau_{v'}^{k'}) \right) \quad (3)$$

We use the shorthand:

$$f_{k,v}(\mathbf{x}_{v'}) = \sum_{k' \in [K]} x_{v'}^{k'} \text{dist}(\tau_v^k, \tau_{v'}^{k'}).$$

Here $\mathbf{x}_{v'} = \langle x_{v'}^1, \dots, x_{v'}^K \rangle$ is the vector of variables for vessel v' . Note that $f_{k,v}(\mathbf{x}_{v'})$ is linear in $\mathbf{x}_{v'}$. The expression (3) now simplifies to:

$$\sum_{k \in [K]} x_v^k f_{k,v}(\mathbf{x}_{v'}) \quad (4)$$

We now replace $x_v^k f_{k,v}(\mathbf{x}_{v'})$ with a real valued variable $z_{k,v,v'}$ to get a reformulation of (4) as:

$$\sum_{k \in [K]} z_{k,v,v'} = \sum_{k \in [K]} x_v^k f_{k,v}(\mathbf{x}_{v'}).$$

■ **Table 2** CompactRiskOPT: Compact Mixed-integer linear program for trajectory optimization.

$$\begin{array}{l}
 \max_{\mathbf{x}, \mathbf{z}, y} \quad y \\
 \text{subject to} \quad \sum_{k=1}^K x_v^k = 1, \quad \forall v \in [M] \\
 \text{Constraint set from Eq. 7} \\
 \text{Constraint set from Eq. 5} \\
 \text{Constraint set from Eq. 6} \\
 x_v^k \in \{0, 1\} \quad \forall v \in [M], k \in [K]
 \end{array}$$

Additionally, we also show that $f_{k,v}(\mathbf{x}_{v'})$ can be easily lower and upper bounded so that the relationship between $z_{k,v,v'}$ and $x_v^k f_{k,v}(\mathbf{x}_{v'})$ can be expressed as linear constraints. Let lower bound:

$$L_{k,v,v'} = \min_{\mathbf{x}_{v'}} f_{k,v}(\mathbf{x}_{v'}) = \min_{k'} \text{dist}(\tau_v^k, \tau_{v'}^{k'}).$$

The second equality above follows from the definition of $f_{k,v}(\mathbf{x}_{v'})$ and the constraint that $\sum_{k' \in [K]} x_{v'}^{k'} = 1$. Similarly, let upper bound:

$$U_{k,v,v'} = \max_{\mathbf{x}_{v'}} f_{k,v}(\mathbf{x}_{v'}) = \max_{k'} \text{dist}(\tau_v^k, \tau_{v'}^{k'}).$$

Lower bounds L and U can be easily computed for each tuple $\langle k, v, v' \rangle$ before we setup the optimization problem. To replace $x_v^k f_{k,v}(\mathbf{x}_{v'})$ with a real valued $z_{k,v,v'}$, we first add the constraints:

$$L_{k,v,v'} x_v^k \leq z_{k,v,v'} \leq U_{k,v,v'} x_v^k \quad \forall k \in [K], v \in [M], v' \in [M], v \neq v' \quad (5)$$

This constraint ensures that $z_{k,v,v'} = 0$ if $x_v^k = 0$. We still need to ensure that if $x_v^k = 1$ then $z_{k,v,v'} = f_{k,v}(\mathbf{x}_{v'})$. Towards this end, we add the constraints:

$$f_{k,v}(\mathbf{x}_{v'}) - U_{k,v,v'}(1 - x_v^k) \leq z_{k,v,v'} \leq f_{k,v}(\mathbf{x}_{v'}) - L_{k,v,v'}(1 - x_v^k) \quad \forall k \in [K], v \in [M], v' \in [M], v \neq v' \quad (6)$$

In the above constraint, if $x_v^k = 1$ then $z_{k,v,v'} = f_{k,v}(\mathbf{x}_{v'})$ and this value of $z_{k,v,v'}$ is also feasible for the previous constraint. Also, when $x_v^k = 0$ then the previous constraint gives $z_{k,v,v'} = 0$ which is still feasible for the above constraint.

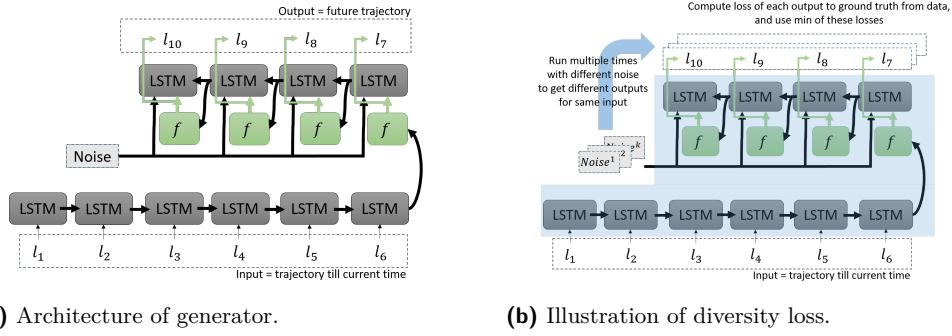
This adds KM^2 continuous variables and $4KM^2$ inequalities; note the reduction in the number of these additional variables and constraints as compared to the naive approach. Also, note that the new objective $\sum_{k \in [K]} z_{k,v,v'}$ is now completely continuous. By introducing an additional variable y which is to stand for $\min_{v \in [M], v' \in [M], v \neq v'} \left\{ \sum_{k \in [K]} z_{k,v,v'} \right\}$ and the constraints

$$\sum_{k \in [K]} z_{k,v,v'} \geq y \quad \forall k \in [K], v \in [M], v' \in [M], v \neq v' \quad (7)$$

the max-min optimization becomes the Mixed Integer Linear Program (MILP) CompactRiskOPT in Table 2.

The arguments presented till now directly leads to the following formal claim of correctness

► **Proposition 1.** *Optimization CompactRiskOPT is equivalent to RiskOPT.*



■ **Figure 2** A LSTM based generative model (left) with diversity loss computation (right). This instance of the generative model is shown with $n = 6$ and $m = 4$. The exact diversity loss formula is in text in Equation 8.

3.2 Trajectory Generation

Our aim is to use historical data to generate the K plausible trajectories. In practice, such generation would be performed by port authority using human expertise, their own prediction methods or vessels themselves can compute it using routing tools such as ECDIS, as mentioned in the introduction. However, these possible trajectories are not recorded in the maritime traffic dataset that are commercially available¹, which only record historical movement of vessels. Clearly, randomly generating trajectories produces very unrealistic trajectories. Instead, we use a generative adversarial networks (GAN) [9] like set-up to learn from historic data and generate multiple future trajectories; the GAN-like learning ensures realism by generating trajectories close to observed ground truth trajectories in the data. We generate a set of K trajectories $\{\tau_v^1, \tau_v^2, \dots, \tau_v^K\}$ for each vessel v . The selection of one trajectory among this set for each vessel is done in the path planning part as described in the previous sub-section. We take inspiration from a GAN to build a simpler architecture for trajectory generation that is easier to train and achieves better results in experiments.

More formally, the goal is to output multiple future trajectories $\{\tau_v^1, \tau_v^2, \dots, \tau_v^K\}$ for each of the M vessels starting from current time step t , where each trajectory τ_v^i is of m time steps. Each time step is 1 minute in wall-clock time. The input to this task is the previous n -step trajectory τ_v^{prev} . During training the future trajectory (ground truth) is known and specified as $\tau_v^{\text{true}} = \langle l_{t+1}^{\text{true}}, \dots, l_{t+m}^{\text{true}} \rangle$, where $l_t^{\text{true}} = (x_t^{\text{true}}, y_t^{\text{true}})$ is the location of the vessel on the 2D plane (this is available from the historical data).

LSTM architecture. Our architecture for this generative task is shown in Figure 2a. It is essentially a LSTM layer (we call this the generator g_θ). LSTMs are a special kind of recurrent neural networks, capable of learning long-term dependencies [11]. All recurrent neural networks have the form of a chain of repeating modules of neural network. A LSTM layer also has this chain like structure where each repeating structure is called a LSTM cell. Each LSTM cell takes in an input (one element of a sequence which is a location l_i in our case) and outputs a hidden value h_i that is fed to the next LSTM cell. The chain structure ensures that h_i captures the information about all the inputs l_j with $j \leq i$. The last cells in a LSTM layer output the predicted future elements of the sequence.

¹ <https://www.marinetraffic.com>

In our architecture, the first n LSTM cells take in as input the past n locations given by τ_v^{PV} . The next m cells output the future prediction with a sequential structure where the prediction at a time-step forms the input for the LSTM cell that predicts the location for the next time step. The predicted location output is formed by transforming the h_i value from the output LSTM cells by passing h_i through a fully connected layer represented by f in Figure 2a.

Stochastic predictions. Importantly, all the future predictor cells also take in Gaussian noise z_1 of dimension D as input, which enables stochastic predictions that provide the multiple future trajectories we need. Multiple trajectories provide flexibility for the optimizer to reach better solutions.

Next, we describe the loss function. During training, for any given predicted sequence output $\hat{\tau}_v^j = (\hat{l}_{t+1}^j, \dots, \hat{l}_{t+m}^j)$ we define a loss $L(\hat{\tau}_v^j, \tau_v^{\text{true}}) = \sum_{i=1}^m \|l_{t+i}^{\text{true}} - \hat{l}_{t+i}^j\|_2$. However, instead of using just one predicted sequence we invoke the generator S times with different noise samples (and same past location input) to obtain S distinct predicted sequences and form the overall loss \mathcal{L} as follows:

$$\mathcal{L}(\{\hat{\tau}_v^1, \dots, \hat{\tau}_v^S\}, \tau_v^{\text{true}}) = \min_{j \in \{1, \dots, S\}} L(\hat{\tau}_v^j, \tau_v^{\text{true}}) \quad (8)$$

This loss is illustrated in Figure 2b. The above loss function is known as Minimum over N (MoN) loss [7] in prior literature and has been used as an additional loss term for diverse samples in SocialGAN [10] for pedestrian trajectory prediction. To understand this loss, note that replacing the min with average or max will force all generated trajectories to collapse to that single trajectory that provides the lowest loss, thereby producing a deterministic prediction instead of the desired stochastic prediction. The min allows for diverse samples while still ensuring that the distribution that generates these samples is able to generate samples close to the ground truth.

Observe that our stochastic prediction is history dependent, which implicitly takes into account the speed of the vessel (which in turn depends on external but unknown factors such as weather and vessel type). In particular, we use only the information that the current VTIS system uses in Singapore port, which simply linearly extrapolates the vessel's current trajectory for prediction.

Also note the distinct aspect that unlike a GAN (e.g., SocialGAN) there is no discriminator network in our architecture, but the loss function of the generator g_θ uses the diversity loss to generate required trajectories. The absence of a discriminator removes the need for adversarial training process of typical GANs, making our training process much more stable and computationally faster, which is critical for us given the large data size. Moreover, we demonstrate experimentally that our approach outperforms SocialGAN as well as a simple linear extrapolation which is the current approach followed by Singapore port's VTIS. In particular, we use three prior proposed metrics to demonstrate the superiority of our approach; these include two common metrics in trajectory prediction, namely Average Displacement Error (ADE) and Final Displacement Error (FDE), and a metric named discriminative score proposed in time-series generation [33]. The ADE and FDE compare the generated trajectories with the actual historical trajectory, and also showcase the diversity in our stochastic predictions. The discriminative score metric ensures that our generated trajectories are realistic (i.e., similar to trajectories in the historical dataset). These metrics are explained in the experiment section.

4 Experiments

We evaluate our proposed learning and planning based system on real world maritime dataset. We use 1.5 years of historical Automatic Identification System (AIS) data (spanning the months between January 2018 -June 2019) of vessels voyaging in the Singapore Strait purchased from the company MarineTraffic. Each AIS record contains information such as timestamp, vessel unique id, lat-long (GPS) positions, course over ground (COG), speed over ground (SOG) and navigation status (e.g anchored/sailing etc). The vital vessel navigation information such as lat-long positions are logged every few seconds interval resulting in total of around 6 million records. Our evaluation is mainly for tankers and cargo vessels because majority of traffic involved in hotspot formation belong to these two types. They are also generally considered as *high-risk* category vessels due to type and size of cargo they carry.

We further process the data to get about 1.6 million individual vessel trajectories for our proposed method in the planning region (shown in figure 1a) . Each vessel trajectory includes 20 latitude-longitude reported at intervals of one minute. These trajectories are used to train our generative model as explained later. Additional experimental details are in the supplemental material.

4.1 Trajectory Generation

In addition to the maritime data we also evaluate our trajectory generation model on three publicly available human pedestrian trajectory datasets (ETH, Hotel, Zara1) [14, 22]. The data includes 2200 trajectories of human movement behaviour in congested environments. The results are in the supplement and are provided mainly to showcase that our proposed approach is competitive with socialGAN even on datasets socialGAN is optimized for.

Evaluation metrics. We use commonly adopted metrics – ADE and FDE [15, 2] and discriminative score [33] for evaluating generated trajectories:

- **Average displacement error (ADE):** Average L2 distance between the ground truth τ_v^{true} and the k^{th} predicted trajectory $\hat{\tau}_v^k$ over all predicted locations in $\hat{\tau}_v^k$.

$$\text{ADE}(\hat{\tau}_v^k, \tau_v^{\text{true}}) = \frac{\sum_{i=1}^m \|l_{t+i}^k - l_{t+i}^{\text{true}}\|_2}{m} \quad (9)$$

For a given trajectory τ_v^{prv} , we sample K future trajectories from the generator. The best and mean ADEs are given by:

$$(\text{best}) \text{ ADE} = \min_{k \in [K]} \text{ADE}(\hat{\tau}_v^k, \tau_v^{\text{true}}) \quad (10)$$

$$(\text{mean}) \text{ ADE} = \frac{\sum_{k=1}^K \text{ADE}(\hat{\tau}_v^k, \tau_v^{\text{true}})}{K} \quad (11)$$

We compare our approach with Social GAN on both best and mean ADE.

- **Final displacement error (FDE):** It is the L2 distance between the ground truth and the k^{th} prediction at the final predicted location for this trajectory.

$$\text{FDE}(\hat{\tau}_v^k, \tau_v^{\text{true}}) = \|l_{t+m}^k - l_{t+m}^{\text{true}}\|_2 \quad (12)$$

The calculations for best and mean FDE is similar to the ones of the ADE in equation (10) and (11).

The ADE and FDE metric show both the quality of predictions and diversity of trajectory generation. If the best ADE (and FDE) is low, then it implies, there is at least one

■ **Table 3** ADE and FDE comparison between our approach and SocialGAN for the Maritime navigational data (lower is better).

Metric (in meters)	SocialGAN	Ours
(best) ADE	491.6	281.2
(best) FDE	975.5	496.3
(avg) ADE	698.3	463.7
(avg) FDE	1340.6	814.6

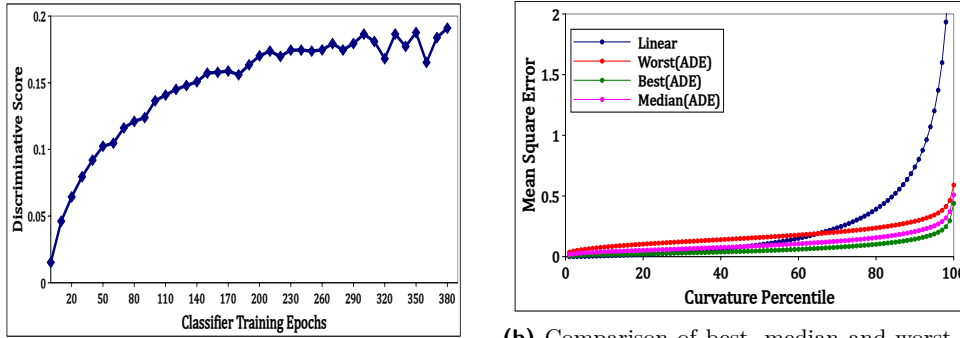
trajectory that is close to the actual ground truth trajectory. We also observe empirically that average ADE and FDE are different than the best ADE and FDE. This implies that there is diversity in predictions, which is incorporated by the MoN loss in (8).

- **Discriminative score** [33]: It is a well adopted measure to validate the quality of generated samples from a generator. Given a generator, we use a test trajectory dataset (which is not used in training the generator) with N trajectories, each of length $n + m$: $\{(\tau_{v_1}^{\text{prv}}, \tau_{v_1}^{\text{true}}), \dots, (\tau_{v_N}^{\text{prv}}, \tau_{v_N}^{\text{true}})\}$. We generate (using our already trained generator) N future trajectories corresponding to each $\tau_{v_i}^{\text{prv}}$ for $i \in [N]$ to obtain $\{\hat{\tau}_{v_i}, \dots, \hat{\tau}_{v_N}\}$. Then, we have a dataset of $2N$ trajectories, half of which are true trajectories $\{\tau_{v_1}^{\text{true}}, \dots, \tau_{v_N}^{\text{true}}\}$ (labelled 1) and the other half generated using our generator $\{\hat{\tau}_{v_i}, \dots, \hat{\tau}_{v_N}\}$ (labelled 0). We train a classifier on this dataset and measure its accuracy. A *perfect* generator would generate data indistinguishable from real ones and hence the classifier would have 50% accuracy. Any deviation from this 50% is a measure of how *inaccurate* the generator is. The discriminative score measures this deviation and is defined as $\text{abs}(0.5 - \text{accuracy})$. A lower discriminative score quantitatively indicates a better generator. Empirically, our generator achieves a low discriminative score, which implies that our generator generates trajectories that are representative of the typical vessel movement patterns found in the historical dataset.

Maritime data results. We divide the whole vessel trajectories data into training and testing set in a 80/20 ratio. Each vessel trajectory consists of 20 locations, first 10 locations (i.e., $n = 10$) are used as input to the model and next 10 locations (i.e., $m = 10$) as the labels. This corresponds to using last 10 mins of trajectory to generate trajectories for next 10 mins. We use the same number of model parameters for both SocialGAN and our approach, additional details on hyper-parameter settings are provided in supplementary material. Table 3 shows the ADE and FDE measures of both approaches. We observe empirically that in all four metrics, our approach is able to achieve better solution quality than SocialGAN. This result shows effectiveness of our proposed generative model on the maritime data.

Note that tanker and cargo vessels are about 200-300 meter in length. Therefore, ADE and FDE achieved by our approach are small relative to the size of tankers. Furthermore, best ADE/FDE in our case are quite different than the average ADE/FDE. This demonstrates that there is diversity in the generated trajectories.

Discriminative score. Our generator achieves a discriminative score of 0.19 as shown in Figure 3a. As the classifier is trained the score steadily increases but hits a plateau of 0.19. As reported in past work [33], 0.19 is competitive (better in some cases) with the scores obtained for other time series generation tasks. Having a low discriminative score ensures that the trajectories generated are realistic, and reflect typical movement patterns observed in the historical dataset. However, this doesn't necessarily imply that our samples are not



(a) Discriminative score to distinguish between real or fake trajectory sample. A score between (0.0 – 0.2) is reasonable.

(b) Comparison of best, median and worst of the generated trajectories against linear extrapolation as a function of increasing curvature in the real trajectories.

■ Figure 3

diverse. The FDE values for the generated trajectories differ by a significant amount as shown in Table 3; the average FDE is significantly higher than the best FDE. The same argument can be made for the metric ADE as well.

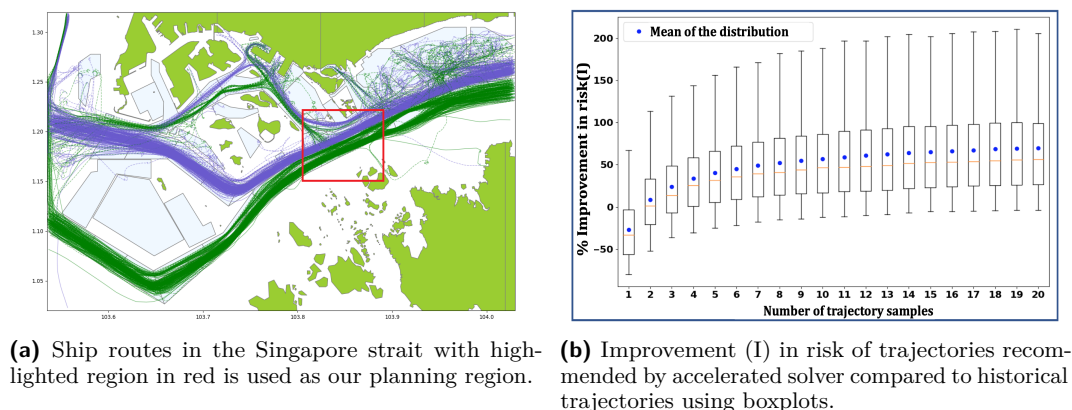
Varying curvatures. To demonstrate the robustness of our trajectory generator, we test it on trajectories with varying curvatures. We assign a curvature percentile to a trajectory where having a higher percentile implies that the vessel trajectory is more curved. In figure 3b, x-axis is curvature percentile, and y-axis is the average error of all trajectories in that curvature bucket. Results in figure 3b show that while the generator’s performance is comparable to linear extrapolation in the case of low curvature percentiles (vessels almost moving in a straight line). It does much better with vessels that are changing their direction. Even the worst of the trajectory samples start doing better than the linear extrapolation as the curvature increases. This shows that our generator is a much better predictor in challenging scenarios when vessels are turning, than the current linear extrapolation method used by Singapore port’s VTIS. We also emphasize that linear extrapolation is a very good metric in most cases, as large vessels typically are unable to turn sharply. Therefore, these results show that our generator has learned much better movements patterns found in the historical data than the linear prediction.

4.2 Path Planning

Here we present our experimental results for path planning module on the maritime data. The path planning part requires a generative model for generating trajectory samples. As empirically observed, SocialGAN performs worse on the maritime data. Therefore, we use our proposed model as the generative model for trajectory generation.

The evaluation of path planning part is mainly for close quarter scenarios where two or more vessels come very close to each other (less than 500 meters). This number (500 meters) was set after our discussions with maritime domain experts; however, it is configurable and does not affect our algorithmic methods. We use the objective in Table 2 as our evaluation criterion which essentially measures the minimum CPA between any any two vessels.

We use the naive formulation in Section 3.1 as baseline and refer to it as the naive solver. We refer to the improved formulation as the accelerated solver. This is our main proposed solver. We also test against the linearly extrapolated trajectories, which shows what would be the risk if vessels moved over this trajectory (linear extrapolation is the current prediction method Singapore port’s VTIS uses).



■ **Figure 4**

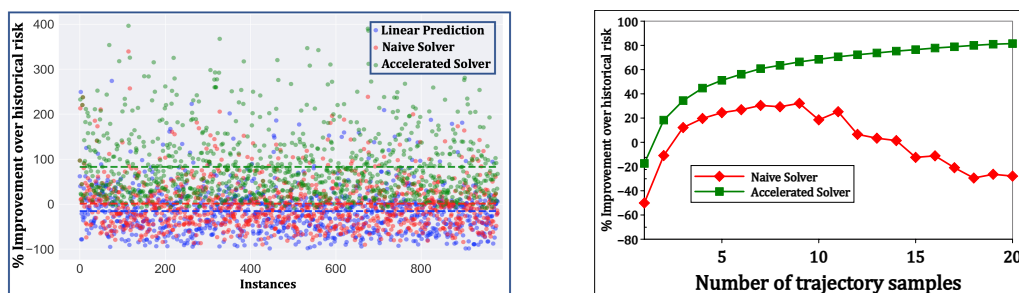
Planning instance generation. For each day we generate instances from peak hour period (7 AM - 9 AM). Majority of close quarter incidents occur during this period. We first select a planning region near the port waters that has high traffic activity based on historical data as shown in Figure 4a. We also select an instance window of 20 minutes because our complete trajectory is of 20 locations at one minute intervals. So a planning instance includes a set of vessel trajectories that have at least one location present within the given planning region and the time window. A snapshot of a planning instance is shown in Figure 1b. For each instance we compute a risk value based on historical data as defined in Section 2. We evaluate our path planning approach on 1000 different instances with the highest risk.

Distributional result. In Figure 4b, we show distributional information about improvement of risk values in 1000 different instances. The x-axis denotes the number of samples (K) and y-axis shows percentage improvement of risk using accelerated solver. For a given instance the percentage improvement of risk (I) is given by

$$I = 100 \cdot \frac{\text{risk}(\tau_1^{\text{true}}, \dots, \tau_M^{\text{true}}) - [\text{risk}(\tau_1^{\text{rec}}, \dots, \tau_M^{\text{rec}})]}{\text{abs}(\text{risk}(\tau_1^{\text{true}}, \dots, \tau_M^{\text{true}}))} \quad (13)$$

Note that risk, as defined in Equation 1, is always negative. Thus, the absolute value in the denominator is needed to show the percentage improvement [30]. We set optimization time limit to one minute to test the near real-time performance of the trajectory optimization module. We observe that the mean (in blue circle) values are higher than medians (orange) thus indicating a positively skewed distribution with long tail. The boxes cover the data range from 25th to 75th percentile. And the fences around the boxes cover the whole range of data. There exist some rare outliers where recommended trajectories are slightly worse off than historical trajectories. Based on our investigation, it was because a ship captain performed an atypical maneuver (such as taking sharp turns) which is rarely observed in the dataset. We also observe an overall good improvement (around 50%) of solution quality across 1000 instances starting from 7 samples. This result shows robustness of our proposed accelerated solver across different instances in near real-time.

Different instances. In Figure 5a we demonstrate the performance of linear prediction, naive and accelerated solvers on 1000 different instances. The x-axis denotes instance id and y-axis denotes percentage improvement of risk compared to historical trajectories. Green color



(a) Comparison of linear prediction, naive and accelerated solver on improvement of risk compared to historical trajectories.

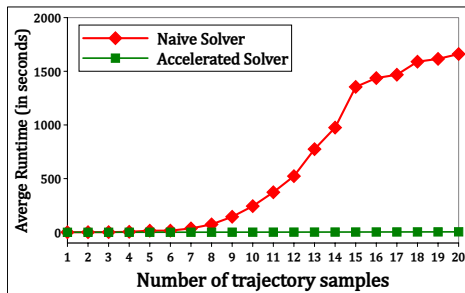
(b) Comparison of naive and accelerated solver on improvement of risk over historical trajectories with varying number of samples.

■ Figure 5

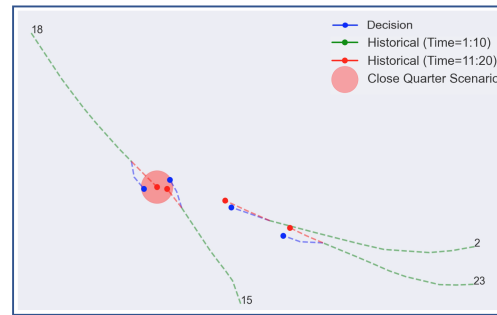
points denote the improvement using the accelerated solver, computed as per equation 13. Similarly, blue and red color denotes the improvement using the linear prediction and naive solver. We set an optimization time limit of one minute for solving each instance. We use a sample size of $K = 20$ for each trajectory. For all the instances, the accelerated solver achieves equal or better solution quality than both the linear prediction baseline and naive solver. This is because optimization time limit of 1 minute is limited for the naive solver to achieve good quality solution. On average the accelerated solver (green dotted line) achieves around 80% improvement of risk. Linear prediction in blue color perform poorly than both the solvers as it is just the linear extrapolation of data from previous time steps.

Varying samples. In our path planning optimization sample size is an important parameter. Therefore, in this experiment we test both the solvers with varying number of sample sizes. Results in Figure 5b show the comparison of naive and accelerated solver on solution quality with varying number of samples K . The x-axis denotes number of samples used in the optimization and y-axis denotes average percentage improvement of risk over historical data. The results shown are averaged over 15 instances. For this experiment also we set the optimization time limit to 1 minute. We observe that for accelerated solver solution quality improve with increasing number of samples, and quality does not change much after 15 samples. This is an expected result because at low number of samples the solution space is small. As the number of samples increase the solution space also increase which leads to better solution quality. But beyond a certain point there are upper limits to maximum possible distance between ships in a region with finite space. Thus the risk plateaus out with increasing number of samples. In this experiment also we observe that the accelerated solver is able to provide better solution quality than both historical data and the naive solver. In case of naive solver after about 7-8 samples the effect of optimization time limit kicks in. More number of samples would require longer optimization time to get the same solution quality, and thus we see a drop in solution quality. This experiment provides vital information about how to choose the sample size parameter in our approach.

Runtime comparison. Results in Figure 6a shows comparison of naive and accelerated solver on optimization time with varying number of samples K . The x-axis denotes number of samples and y-axis denotes average optimization runtime. The results shown are averaged over 20 instances. For this experiment, we set the optimality gap of the solver to 10%. We observe that runtime of naive solver rise almost exponentially with increasing number of



(a) Comparison of naive and accelerated solver on runtime with varying sample count.



(b) Close quarter scenario.

Figure 6

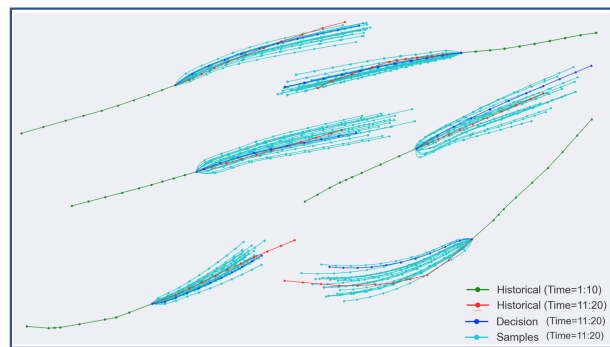
samples. However, accelerated solver is able to maintain a constant runtime irrespective of sample size. The solver has a runtime of around 4 seconds at 20 sample size. We also observed that accelerated solver is able to achieve a runtime of around 5 seconds at 20 sample size even for 5% optimality gap (not shown in the figure). For any system to be used in real-time scenario, a decision time of few seconds is very crucial. The empirical result shows our proposed system is well adapted for a real-time safe trajectory recommendation system.

Close quarter scenario. Figure 6b shows an instance of close quarter situation. Green and red dotted line denote previous and next 10-step trajectories from historical data respectively. Blue dotted line is the recommended trajectory from our learning and planning based system. It is one of the predicted samples from the generative model. In the figure vessels with id 15 and 18 are heading in opposite direction. They come very close to each other (less than 500 meters) which is a close quarter situation as highlighted in the big red circle. We observe that our recommended trajectories (in blue) are able to maintain a safe distance and thus avoid the close quarter incident. We provide four video files for such instances of close quarter situations using our maritime traffic simulator on our GitHub repo.

Close quarter scenario. Figure 7 shows qualitative results of some of the generated trajectories from our generative model. Trajectories in green and red are complete historical trajectories with time=1:10 and time=11:20 respectively. Trajectories in cyan color are generated sample future trajectories (time=11:20). Trajectories in blue color are the selected trajectories for time=11:20 from the path planning solver. Here we observe that the generated trajectories in cyan are a good representative sample of the historical trajectories in red.

5 Conclusion

We have presented a multiagent path planning approach to the problem of alleviating close quarter incidents in a highly congested maritime traffic environment. We proposed a data-driven based optimization methodology to the problem. We first learn a generative model of vessel movement behaviors from historical data. Empirically, we have shown the superior quality of our generative model over the baseline model. The trajectory samples generated from our model are then used in our proposed novel and efficient MILP solver to reduce close quarter incidents. Empirically, we have shown that our solver is able to provide high quality safe trajectory recommendations in near real-time in a variety of real-world close quarter situations mined from past data.



■ **Figure 7** Qualitative result for generative model.

References

- 1 Lucas Agussurja, Akshat Kumar, and Hoong Chuin Lau. Resource-constrained scheduling for maritime traffic management. In *AAAI Conference*, 2018.
- 2 Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *IEEE conference on CVPR*, pages 961–971, 2016.
- 3 Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43, 2003.
- 4 Saumya Bhatnagar, Akshat Kumar, and Hoong Chuin Lau. Decision making for improving maritime traffic safety using constraint programming. In *Proceedings of the 28th IJCAI*, 2019.
- 5 S. Choudhury, K. Solovey, M. J. Kochenderfer, and M. Pavone. Efficient large-scale multi-drone delivery using transit networks. In *IEEE ICRA*, pages 4543–4550, 2020.
- 6 Lei Du, Floris Goerlandt, and Pentti Kujala. Review and analysis of methods for assessing maritime waterway risk based on non-accident critical events detected from ais data. *Reliability Engineering and System Safety*, 200, 2020.
- 7 Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *IEEE conference on CVPR*, pages 605–613, 2017.
- 8 Futureautics. Autonomous ships | white paper. <https://www.sipotra.it/old/wp-content/uploads/2017/05/Autonomous-Ships.pdf>, 2016.
- 9 Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- 10 Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *IEEE Conference on CVPR*, pages 2255–2264, 2018.
- 11 Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- 12 IMO. E-navigation. <https://www.imo.org/en/OurWork/Safety/Pages/eNavigation.aspx>, 2019.
- 13 International Maritime Organization. Electronic Nautical Charts (ENC) and Electronic Chart Display and Information Systems (ECDIS). <https://www.imo.org/en/OurWork/Safety/Pages/ElectronicCharts.aspx>, 2022.
- 14 Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. In *IEEE Conference on CVPR*, pages 3542–3549, 2014.
- 15 Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *IEEE Conference on CVPR*, pages 336–345, 2017.

- 16 H. Ma, C. Tovey, G. Sharon, T. K. S. Kumar, and S. Koenig. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In *AAAI Conference*, pages 3166–3173, 2016.
- 17 Faris Mokhtar. Busy shipping lane’s narrow passageway hard for vessels to navigate. <https://www.todayonline.com/singapore/busy-shipping-lanes-narrow-passageway-hard-vessels-navigate>, 2017.
- 18 Robert Morris, Corina S. Pasareanu, Kasper S e Luckow, Waqar Malik, Hang Ma, T. K. Satish Kumar, and Sven Koenig. Planning, scheduling and monitoring for airport surface operations. In *AAAI Workshop on Planning for Hybrid Systems*, 2016.
- 19 MPA. Vessel Traffic Information System. <https://www.mpa.gov.sg/web/portal/home/port-of-singapore/operations/vessel-traffic-information-system-vtis>, 2021.
- 20 MPA Singapore. Over 250 participate in Joint Oil Spill Exercise to Test Responsiveness to Oil Spills at Sea. <https://www.mpa.gov.sg/web/portal/home/media-centre/news-releases/mpa-news-releases/detail/091cd124-ca60-4f34-bdb6-a0967f82defd>, 2018.
- 21 International Maritime Organization. Autonomous shipping. <https://www.imo.org/en/MediaCentre/HotTopics/Pages/Autonomous-shipping.aspx>.
- 22 Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European conference on computer vision*, pages 452–465. Springer, 2010.
- 23 Henrik Ringbom. Regulating autonomous ships – concepts, challenges and precedents. *Ocean Development & International Law*, 50(2-3):141–169, 2019.
- 24 Rolls-Royce. Remote and autonomous ship – The next steps. <https://www.rolls-royce.com/~media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/aawa-whitepaper-210616.pdf>, 2016.
- 25 David Silver. Cooperative pathfinding. In *AIIDE*, pages 117–122, 2005.
- 26 Arambam James Singh, Akshat Kumar, and Hoong Chuin Lau. Hierarchical multiagent reinforcement learning for maritime traffic management. In *Proceedings of the 19th AAMAS*, 2020.
- 27 Arambam James Singh, Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. Multiagent decision making for maritime traffic management. In *AAAI Conference*, 2019.
- 28 Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Roman Bart ak, and Eli Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *SoCS*, pages 151–159, 2019.
- 29 Teck-Hou Teng, Hoong Chuin Lau, and Akshat Kumar. Coordinating vessel traffic to improve safety and efficiency. In *Proceedings of the 16th AAMAS*, pages 141–149. ACM, 2017.
- 30 Leo T ornqvist, Pentti Vartia, and Yrj  O Vartia. How should relative changes be measured? *The American Statistician*, 39(1):43–46, 1985.
- 31 Kevin Varley. Ships Queues Worsen Port Delays From Singapore to Piraeus. <https://www.bloomberg.com/news/articles/2021-11-02/ships-queues-worsen-port-delays-from-singapore-to-piraeus>, 2021.
- 32 Peter R. Wurman, Raffaello D’Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29(1):9–20, 2008.
- 33 Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32:5508–5518, 2019.
- 34 J. Yu and S. M. LaValle. Structure and intractability of optimal multi-robot path planning on graphs. In *AAAI Conference*, pages 1443–1449, 2013.
- 35 Jinfen Zhang, Tiago A Santos, C Guedes Soares, and Xinping Yan. Sequential ship traffic scheduling model for restricted two-way waterway transportation. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 231(1):86–97, 2017.