FOV-RVO: Velocity Obstacle-based pedestrian motion predictor

Dmytro Zabolotnii[1], Yar Muhammad[2], and Naveed Muhammad[1]

Abstract—**Predicting pedestrian motion is a crucial part of any safety-first autonomous driving system. We present FOV-RVO, a Velocity Obstacle-based motion prediction method that models pedestrian-to-pedestrian and pedestrian-to-scene interactions by integrating the gaze directions of the pedestrians and map information of the environment. The proposed solution is fast, robust, and does not require any prior data. Furthermore, we enhance the method by introducing an auxiliary pre-trained Deep Learning (DL) method and combining predictions for final evaluation to utilize the strengths of both knowledge-based and data-driven motion prediction methods. The combined model is implemented inside the autonomous driving framework — Autoware Mini and tested on data from trips in urban conditions in Tartu, Estonia. The proposed FOV-RVO method outperforms compared state-of-the-art DL methods at number of predicted candidate trajectories $K = 1$ in combined evaluation using minimal Average/Final Displacement Errors (minADE/minFDE), Miss Rate (MR), and non-Drivable Area Compliance (nonDAC). The combined solution at $K = 2$ performs equivalent or better than tested models that output significantly higher predictions (up to $K = 10$). The open-source code with instructions on accessing the dataset is available at https://github.com/dmytrozabolotnii/autoware_mini/tree/FOVRVO**

Index Terms—**Autonomous Vehicle Navigation, Computer Vision for Transportation, Datasets for Human Motion, Human Detection and Tracking, Intention Recognition**

I. Introduction

Autonomous driving vehicles are no longer in the realm of science fiction, with Waymo launching autonomous taxis across several cities of the United States, promising safer and less accident-prone traveling for passengers over traditionally human-driven vehicles [1]. Analogous projects of autonomously driven taxis have started in other countries, such as Switzerland [2], and China [3]. However, rapid integration of vehicles without a security driver present into commercial usage does not remove all human interactions from autonomous driving. Autonomous vehicles always have to consider interactions with other traffic agents, with one of the most prominent categories being pedestrians [4]. Interactions with pedestrians are especially crucial for vehicles operating in dense urban conditions, where a combined solution of complex engineering and behavioral science is needed to ensure that vehicles respond safely and appropriately to any possible actions and movements. As such, pedestrian motion prediction remains a key task for autonomous driving vehicles' current and future development.

Pedestrian motion prediction, however, is not a problem that belongs only to the autonomous driving field, but rather a research area with on the overlap of different robotic fields: autonomous driving, intelligent robots, and advanced surveillance systems [5]. As such, it is a well-established field with much scientific progress over the past decades. Over this time, two significantly different approaches to solving the pedestrian motion prediction problem were developed. Traditionally, knowledge-based (KB) methods were derived from the Newtonian motion model and customized to account for various social interactions between pedestrian agents and the surrounding environment [6]. In the last decade, developments in the machine learning field created immense interest in applying Deep Learning (DL) to many problems in robotics, including pedestrian motion prediction, and the trends strongly favor the DL approach [4]. However, despite the popularity of the DL methods and their excellent performance on the standard evaluation datasets such as Argoverse [7] and OpenWaymo [8], traditional KB approaches still have several advantages, such as interpretability, modifiability and simplicity [9]. Machine learning-based methods often struggle to generalize beyond standard datasets [10], while KB methods can be adapted quickly by explicitly changing the model's parameters. These factors are especially important in modular autonomous driving frameworks [11] that are expected to perform under variable environmental conditions that remain a popular and time-proven method for solving autonomous driving [12]. On the other hand, KB methods often underutilize available data and are rarely evaluated versus DL methods under a common framework.

In this paper, we present FOV-RVO (Field-of-View Reciprocal Velocity Obstacles), a simple and efficient pedestrian motion predictor built upon the standard Reciprocal Velocity Obstacle (RVO) model [13]. Under the RVO model, agents are assumed to mutually collaborate and share responsibility for avoiding each other Velocity Obstacles (VO), which is a collection of all relative velocities that will lead to collision. Each agent calculates the velocity of the obstacles of other agents and chooses the velocity outside of them, thus avoiding all potential collisions in a fully decentralized system. FOV-RVO utilizes additional auxiliary information obtained through autonomous driving system sensors, namely

[1] D. Zabolotnii and N. Muhammad are with the Institute of Computer Science at the University of Tartu, Narva mnt 18, Tartu 51009, Estonia. Email: dmytro.zabolotnii@ut.ee, naveed.muhammad@ut.ee

[2] Y. Muhammad is with the Department of Computer Science at the University of Hertfordshire, AL10 9AB Hatfield, U.K. Email: y.muhammad@herts.ac.uk

This work was supported in part by European Social Fund through the "ICT Programme" Measure and in part by Bolt Technologies through the Collaboration Project under Grant LLTAT21278.

pedestrian gaze and contextual map information, critical for integrating pedestrian-to-pedestrian and pedestrian-to-scene interactions [5]. Gaze information recently was used as an additional input to DL solutions for predicting pedestrian intention to cross the road [14] and general pedestrian motion prediction [15], but remains underutilized for both KB and DL methods, because of the absence of gaze direction data labeling in popular evaluation datasets. Such data can be implicitly obtained from analyzing first-person camera images from autonomous vehicle camera sensors but remains a non-trivial detection task even under best conditions [16]. FOV-RVO utilizes a state-of-the-art head pose detection algorithm [17], which extracts head pose representation from available camera data and then cross-references with pedestrian detections from Lidar data to estimate accurate gaze direction in BEV perspective. For the pedestrians not visible by camera data, either obscured by other objects or outside of camera range, FOV-RVO uses simplified gaze detection by assuming that the pedestrian's gaze direction equals their estimated velocity vector. The model also utilizes gaze information to calculate the share of responsibility pedestrians assume in avoiding obstacles.

We implement and evaluate FOV-RVO inside a modular autonomous driving framework, Autoware Mini [18]. To complement our model, we also combine its prediction with the prediction from the DL method GATraj [19]. We test FOV-RVO and a combined solution against several state-of-the-art DL approaches on the sensor data recorded during past autonomous vehicle trips. The testing is performed in a fully online manner, concurrently running the rest of the framework to fully process Lidar and camera sensor data through all stages of the modular framework in real time to reproduce experimental conditions of running the prediction model in a deployed vehicle as closely as possible. From the evaluation results, our combined model produces more accurate and road-compliant predictions than compared DL methods, with a fixed amount of the predicted candidate trajectories to $k = 2$ for all models, while still performing comparably when compared to DL methods outputting up to $k = 10$ candidate trajectories. As such our contribution are two-fold:

1) Extension of the existing RVO methods to integrate pedestrian gaze direction information obtained from the head-pose detection and a generalized approach to integrating map data information.
2) Engineering adaptation and evaluation of FOV-RVO method inside an autonomous driving framework that allows real-time testing on both recorded raw sensor data and during deployed autonomous vehicle trips in Tartu, Estonia.

II. Related Works

As mentioned above, two main approaches to pedestrian motion prediction are Knowledge-based (KB) and Deep learning (DL) methods. As there is a wealth of literature covering DL methods in-depth [4], [5], [20], we will focus on reviewing KB methods, specifically those derived from the Velocity Obstacle (VO) paradigm and adjacent to our FOV-RVO method.

Velocity Obstacle (VO) model [21] is a continuation of previous efforts, such as Social Force [22] and Dynamic Window [23] to create a decentralized system for autonomous agents that could move in a defined environment without communication with each other. In the VO model, each agent has full responsibility to avoid velocity obstacles created by other agents that are assumed passive. This was improved in the Reciprocal Velocity Obstacle (RVO) model [13], where agents now have equally shared responsibility to avoid each other. RVO and its early variations, such as HRVO [24] or GVO [25], represent velocity obstacles as a cone and require solving a constraint optimization problem to find a suitable velocity to avoid all obstacles. ORCA [26] simplifies this definition, reducing each velocity obstacle cone in a relative velocity space between two agents to an equivalent ego-agent half-plane in an ego-agent velocity space. ORCA guarantees smooth trajectories in crowded areas and achieves excellent computational efficiency compared to previous VO variations.

One of the first applications of RVO to pedestrian motion prediction was Bayesian-RVO [27]. BRVO uses an Ensemble Kalman Filter to estimate and refine internal agent state values, such as position and velocity from noisy sensor data and uses these values in the RVO model to find the maximum likely next preferred speed but still assumes static responsibility values between different agents as per original RVO. PORCA [28] introduces variable responsibility for pedestrians that depends on their distance to the ego-agent used in their testing — a small robot scooter, and increases their responsibility factor to avoid scooter the closer they are to account for the robot's non-holonomic nature. However, the method lacks variable responsibility between pedestrians themselves. In GAMMA [29], a similar assumption is made, where pedestrians are assumed to strive to avoid collision with vehicles in all scenarios to avoid bodily harm.

Contrary to these works, we investigate pedestrian-to-pedestrian interaction more deeply, focusing on the information that gaze direction provides us that can direct them to decide if they want to avoid an obstacle or if they are not going to avoid it because they cannot see it. We deprioritize pedestrian-to-vehicle interactions, representing them via pedestrian compliance to general environmental constraints. What we presume here is that in the context of everyday traffic scenarios, pedestrians cannot be held responsible for avoiding large vehicles if they are otherwise compliant with traffic rules (i.e., a pedestrian crossing the street on a green light does not assume that the car moving toward them will not stop and slow down and that pedestrian will need to plan to avoid it).

III. Research Methods

A. Problem Formulation

Pedestrian motion prediction can be defined as the problem of future trajectory prediction based on past observations or, as they are sometimes referred to, stimuli [4]. With the

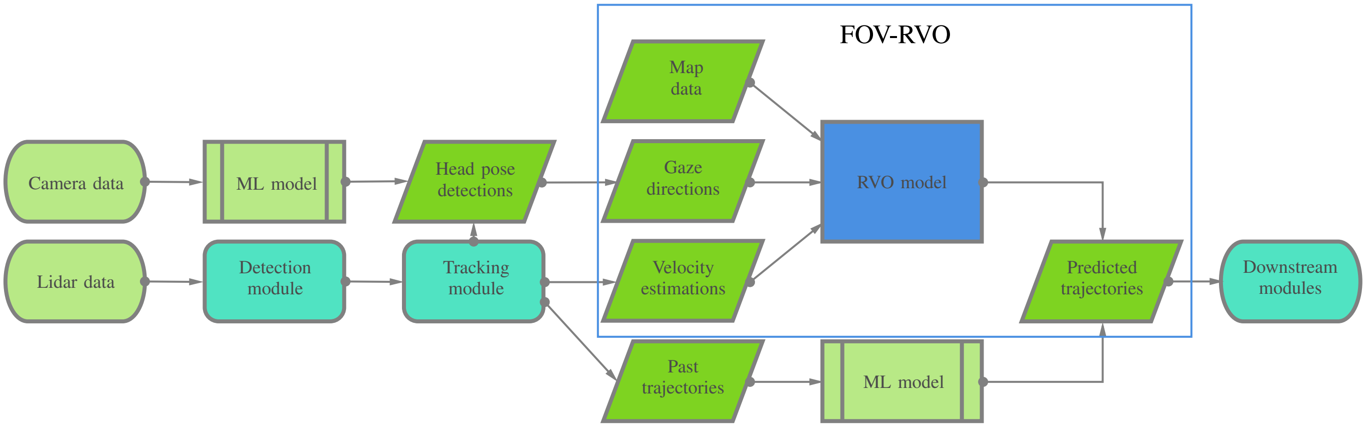


Fig. 1. Overview of the FOV-RVO architecture inside the autonomous driving framework. Velocity estimations, gaze directions, and map data inputs for FOV-RVO are received from upstream tracking and detection modules that process raw sensor data. Gaze direction is obtained using head pose detected in camera images cross-referenced with pedestrian detections in Lidar data. In parallel, the history of pedestrian trajectories is fed into the auxiliary DL model, and predictions output from both models are merged before forwarding to downstream modules.

increasing usage and development of DL models, the most popular input used is the previous history of the pedestrian locations represented in 2D from a Birds-Eye-View (BEV) perspective [20]. However, recent literature has emphasized the importance of incorporating other stimuli types to represent the wide variety of pedestrian interactions with surrounding objects (divided into pedestrian-to-pedestrian and pedestrian-to-vehicle interactions) and environment (pedestrian-to-scene interactions) [5].

FOV-RVO integrates gaze direction in order to better model pedestrian-to-pedestrian interactions and map information to model pedestrian-to-scene interaction. In order to generalize available map information from potential variable representations of environment like semantic maps, HD maps, and underlying traffic rules, we define it contextually as a collection of locally adjacent polygons $M_i = m_i^1, m_i^2, ..., m_i^L$, where every polygon must be within a threshold distance from current pedestrian i location X_i^0. Polygons can be flagged as being explicitly walkable (for example, sidewalks, uncontrolled/green light crossings) or non-walkable (drivable roads, highways, crossings during red light). Following that, in the following way: given the historical trajectories of N pedestrians over H previous states $X_i = x_i^1, x_i^2, ..., x_i^H$, gaze direction vector G_i and local map representation $M_i = m_i^1, m_i^2, ..., m_i^L$, find the future trajectories of the pedestrians over F future states $Y_i = y_i^{H+1}, y_i^{H+2}, y_i^{H+F}$, or in another form, find function f that satisfies $Y_i = f(X_i, G_i, M_i)$ [30].

B. FOV-RVO

FOV-RVO is built upon a VO paradigm [21]. We extend the RVO approach by integrating gaze and map information as additional inputs, creating additional constraints to which RVO-modified velocity should conform. Furthermore, to narrow the gap between KB and DL approaches [9], we employ pre-trained DL model GATraj [19] as an auxiliary model and combine the most likely single prediction of GATraj with a prediction obtained from FOV-RVO for the final prediction result with the number of candidate trajectories $K = 2$. The combined model is implemented in Autoware Mini [18], which is a Python-based, open-source, autonomous driving software stack. The overview of the architecture is available in Figure 1.

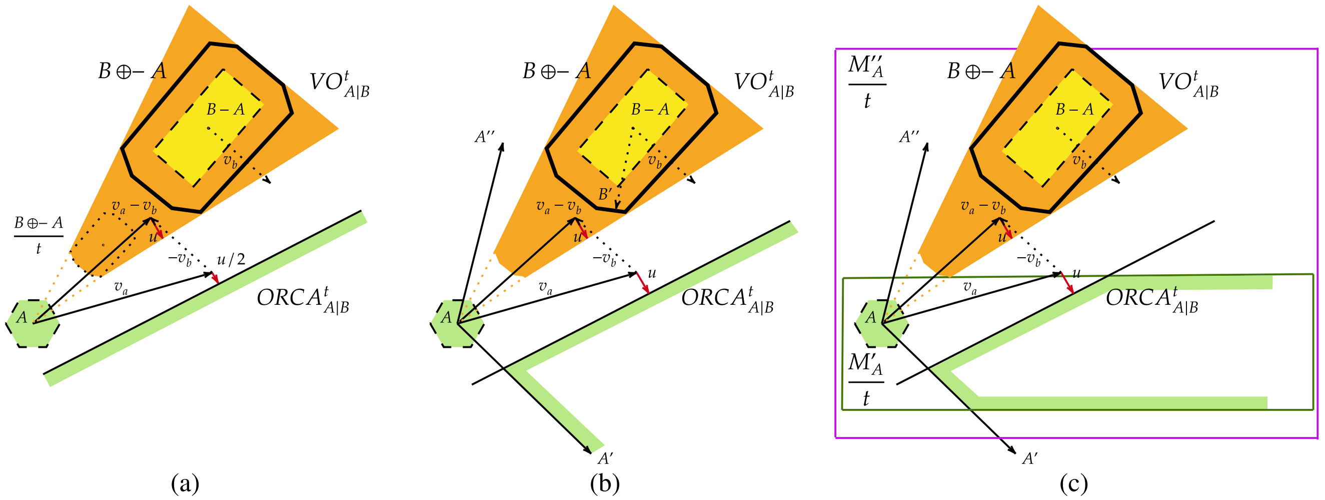
FOV-RVO implementation is based on open-source RVO2 code [31], which implements a computationally effective method for avoiding constructed velocity obstacles named Optimal Reciprocal Collision Avoidance (ORCA). In ORCA, the velocity obstacle of agent B from the point of view of agent A that will cause the collision in time t $VO^t_{A|B}$ is constructed under the assumption that agents A and B are represented by either a circular or square shape for simplicity of calculation. We remove this restriction and extend ORCA method to accommodate any valid convex polygon. As such:

$$VO^t_{A|B} = \{v | v \cap \frac{1}{t}(B \oplus -A) \neq \varnothing\},$$

where $\oplus$ denotes Minkowski sum of the polygons centered on A. After obtaining velocity obstacles of all agents within a threshold distance from A, ORCA constructs a collection of half-planes of admissible velocity vector values where each half-plane $ORCA^t_{A|B}$ is generated from corresponding velocity obstacle $VO^t_{A|B}$ [32]

$$ORCA^t_{A|B} = \{v | (v_a + \alpha u) \cdot u \geq 0\},$$

where u denotes minimal deviation vector necessary to exit velocity obstacle for agent A calculated from relative speed vector $v_a - v_b$ (Figure 2a). In ORCA, agent A and B assume equal responsibility to deviate from collision, so their responsibility factor is set to $\alpha = 0.5$. FOV-RVO considers only velocity obstacles produced by pedestrian and light vehicle drivers such as bicycle/scooter drivers within a specific fixed area $Area(A)$ around every agent A detected by ego-agent. As such, given all neighbors of A $B \in Area(A), B = B_1, ..., B_n$ the valid velocities that



(a) (b) (c)

Fig. 2. Green shade represents the viable velocity set for agent A that avoids orange velocity obstacle $VO^t_{A|B}$ produced by different stages of FOV-RVO: (a) Extended ORCA implementation with any valid convex shape of agents A and B (b) Integrating gaze direction information creates 120-degree FOV cone that restricts viable velocity set. Additionally, as agent A is outside of agent's B FOV, agent A assumes full responsibility $\alpha = 1$ to avoid VO (c) Integrating map information, viable velocity set is further decreased by un-walkable map polygon M''_A that also overlaps explicitly walkable polygon M'_A which takes precedence.

avoid all velocity obstacles are defined by the intersection of all ORCA half-planes $ORCA^t_A = \bigcap_{i=1}^{n} ORCA^t_{A|B_i}$.

FOV-RVO expands on ORCA by introducing additional constraints to a set of potentially viable velocities, with one constraint responsible for integrating agent A gaze direction information G_A and another for integrating local map information M_A. To extract gaze direction information of the pedestrians, we first process synchronized camera images using the Yolo11 model [33] to extract precise bounding boxes containing pedestrians' views — one for each visible pedestrian. Next, we employ the Real-time 6DoF Full-Range Markerless Head Pose Estimation method [17] that processes every bounding box and extracts head pose prediction in the three degrees of freedom - roll, yaw, and pitch, represented as the rotation matrix R^C_A. This information is obtained from the camera coordinate system. We need to convert it to the gaze direction in BEV representation. We first project 3D bounding boxes representing pedestrians obtained from the tracker module (that processes Lidar data) to associate each head pose prediction with the specific tracked pedestrian coordinates in the BEV representation using IoU matching between projected 3D boxes and original corresponding 2D bounding boxes. Finally, we obtain the gaze direction vector using the rotation matrix and the transform between the camera frame and the 3D frame at time t T_t.:

$$G_A = \begin{vmatrix} 0 \\ 0 \\ 1 \end{vmatrix} R^C_A T_t$$

The first additional constraint is obtained from generating a 120-degree angle $A'AA''$ centered on G_A projected to 2D BEV. (Figure 2b) representing the binocular field of view of pedestrian [34]. Binocular vision is crucial for depth perception and, thus, for accurate estimation of other pedestrian positions and velocities. As such, any agents $B_x \notin A'AA''$ are deemed non-perceivable by A, and we relax the assumed responsibility of vehicle obstacle avoidance $ORCA^t_{A|B_x}$ to $\alpha = 0$. Vice versa, for agent B_x if agent A is in their 120 degrees FOV, their responsibility factor in $ORCA^t_{B_x|A}$ rises to $\alpha = 0$ fulfilling the primary consideration of RVO algorithm that the sum of responsibilities should be equal to one. Furthermore, we assume that the v_{new} pedestrian chooses also belongs to their binocular field of view, and as such, add an additional geometric constraint to the set of valid velocities with the new set denoted as $FOV^t_A = ORCA^t_A \bigcap A'AA''$. To accommodate the possible side-stepping or back-gliding behavior of pedestrians, the FOV angle constraint is union with a small circle centered on A in the implementation.

The second additional constraint directly integrates map data information in the FOV-RVO model. Given the polygon collection M_A of local map representation for agent A, we define M'_A as a collection of only explicitly walkable polygons and M''_A as a collection of only un-walkable polygons. Primarily, we want to avoid predicted velocity from intersecting un-walkable polygons in a time window of t. Then, the constraint generated by map data on the set of valid velocities can be represented as $FOVMAP_A = FOV^t_A \setminus \frac{1}{t} M''_A$. As map representation can be imperfect, and polygons from explicitly walkable subset likely intersect un-walkable polygons, we assign the pedestrians right-of-way and assume if they are on the explicitly walkable polygon they can also be on the un-walkable polygon at the same time (for example during crossing on the regulated crosswalk that is also denoted as a driving area). As such, the final set of all viable velocities of the FOV-RVO model is:

$$FOVMAP^t_A = (FOV^t_A \bigcap \frac{1}{t} M'_A) \bigcup (FOV^t_A \setminus \frac{1}{t} M''_A)$$

Finally FOV-RVO chooses new velocity for agent A v_a^{new}

that is closest to old velocity v_a such as

$$v_a = \underset{v \in FOVMAP_A^t}{argmin} ||v - v_a||,$$

such as also $v_a \leq v_{max}$, (where v_{max} is assumed maximum speed of the pedestrian) from which we can construct a future pedestrian's trajectory.

C. Experimental Setup

1) Data collection: Our goal is to evaluate FOV-RVO and compare state-of-the-art models in the environment as close as possible to an actual autonomous vehicle. As such, we closely follow our own previous work procedure [35] and base our setup on the Autoware Mini framework [18], which is based on Robot Operating System (ROS) [36]. This framework has been developed for research and pedagogical purposes and deployed on Lexus RX450h vehicle on the streets of Tartu, Estonia. The vehicle is equipped with Velodyne VLP-32C Lidar as the primary sensor for object detection and two Mako frontal cameras setup for auxiliary tasks such as traffic light status detection that we reuse for pedestrian gaze direction detection. Data collected by these sensors is recorded during regular autonomous- and manual-driving trips and stored inside a .bag file format. This raw sensor data collection provides a primary dataset for our evaluation purposes. However, the entire data collection is 3400+ different .bag files, totaling over 43 TB in size, so excessive filtering was necessary to produce a valid dataset. The primary filter was the average length of pedestrian presence as we cannot viably evaluate pedestrians' motion prediction if they are only detected for a short period of time. As such, we have selected the .bag files with average pedestrian presence time $\frac{1}{N}\sum_{i=1}^{N} t_{presence}^{i} \geq 1.6sec$, high amount of total pedestrians detected $N \geq 100$, availability of the synchronized camera data (as not all trips are done with the full sensor suite) and restricted the vehicle route to the pedestrian dense center of Tartu. Additionally, we included a few .bag files recorded during non-typical weather conditions: rain and light snow. The result is a .bag dataset with 17 .bag files, where every .bag file represents a continuous scene, with a total runtime of two hours of raw sensor footage.

2) Implementation Details: Recorded .bag files can be "replayed" inside the Autoware Mini framework, simulating the stream of data that the autonomous vehicle received during the trip and allowing the entire framework to process the data through autonomous driving architecture. Bag files output raw Lidar point cloud data to the detector module and camera data to the gaze direction detection module. For general detection, our framework uses an SFA3D detector based on [37] to process raw point cloud data and output labeled vector representation of the objects. Next, the simplistic exponential moving averages-based tracker estimates pedestrians' locations and velocities and keeps the objects' permanence that allows the construction of historic pedestrian trajectories in real-time. Camera data is forwarded to the gaze detection module, which processes it through two pre-trained neural networks and outputs gaze direction according to the algorithm described in the previous subsection. Estimated location allows the extraction of locally adjacent map data from a global map associated with the trip. The global map was created manually, aiming for the decimeter-level precision, which is equal to the precision of the employed localization GNSS system. Finally, the auxiliary DL predictor module receives historical trajectories. After combining prediction outputs from FOV-RVO and the auxiliary model, the candidate trajectories are forwarded to the rule-based planner module of Autoware Mini. While the planner module reacts to the predicted trajectories and directs controls of the vehicle, this does not affect the behavior of the exo-agents during the replay of the recorded sensor data.

For the auxiliary model, we use GATraj model pre-trained on ETH/UCY dataset. [38], [39]. The framework implementation prefers pre-trained models because the errors introduced by upstream modules (detection/perception) makes raw dataset unsuitable for prediction models training or finetuning. As we rely on FOV-RVO to integrate map data and provide half of the final predictions, we strongly lessen the effect of the out-of-distribution problem of the pre-trained model and can employ fast and reliable DL model that relies only on minimum input data of pedestrian past trajectory histories. As the ETH/UCY dataset follows the input format of 8-point historical trajectories to predict a 12-point candidate trajectory, where points are 0.4 seconds apart, we adopt the same prediction horizon for the FOV-RVO model. As a result, both models predict the next 4.8 seconds of motion. Consequently, models were adapted to run with a constant stream of incoming data, where every 0.4 seconds, they receive updated estimated locations/velocities/map data/trajectories of all visible pedestrians. Despite this interval limiting the effective performance of the predictor module to 2.5 Hz, it does not slow down the performance of the framework that runs at 10 Hz at minimum to ensure real-time vehicle reactions.

To simulate the limited resources of an embedded system of the real vehicle, the full evaluation framework was performed on a consumer-grade laptop equipped with one NVIDIA GeForce 5070 GPU and Intel Core Ultra 7 255HX CPU.

D. Evaluation Metrics

The standardized metrics for evaluating trajectory prediction are Average Displacement Error (ADE) and Final Displacement Error (FDE) [4]. ADE metric is commonly defined as the average L2 distance between all points of predicted trajectory and ground truth future trajectory:

$$ADE = \frac{1}{N}\frac{1}{F}\sum_{i=1}^{N}\sum_{j=1}^{F}||y_i^{H+j} - \hat{y}_i^{H+j}||_2,$$

where $\hat{Y}_i = \hat{y}_i^{H+1}, \hat{y}_i^{H+2}, ..., \hat{y}_i^{H+F}$ is the ground truth future trajectory. FDE is similar, but considers only L2 distance between final points of predicted and ground-truth trajectories.

$$FDE = \frac{1}{N}\sum_{i=1}^{N}||y_i^{H+F} - \hat{y}_i^{H+F}||_2$$

TABLE I

QUANTITATIVE RESULTS WITH VARYING AMOUNT OF CANDIDATE TRAJECTORIES K, **BOLD** IS THE BEST ENTRY AND UNDERLINED IS THE SECOND BEST ENTRY

Model	FOV-RVO	FOV-RVO+GATraj	PECNet			SGNet			GATraj		
K	1	2	1	5	10	1	5	10	1	5	10
.bag dataset											
minDynADE (m)	0.973	0.867	1.497	1.357	1.311	1.541	0.982	0.883	1.716	0.979	**0.828**
minDynFDE (m)	1.783	1.566	2.791	2.436	2.346	2.885	1.844	1.680	3.321	1.691	**1.371**
MR_2	0.322	0.278	0.488	0.436	0.421	0.624	0.348	0.310	0.622	0.267	**0.193**
nonDAC	**0.976**	0.944	0.964	0.964	0.963	0.934	0.954	0.949	0.922	0.919	0.915
ETH/UCY (k=20)											
minADE (m)	-	-	0.29			0.18			**0.17**		
minFDE (m)	-	-	0.48			0.35			**0.29**		

It is typical for a prediction model to output multiple candidate trajectories. To resolve this, the modification of ADE/FDE was introduced called minADE/minFDE, which considers only the best metric score. This variant is used for benchmarking on the most popular pedestrian motion prediction datasets, such as SDD, ETH-UCY, Nuscenes, and Argoverse. However, minADE/minFDE only evaluates the mean error of the prediction. To measure the variability of error distribution, we implement Miss Rate (MR) metric [5], which is defined as the percentage of predictions where none of the candidate trajectories FDE values are lower than as static threshold (the standard value for Argoverse evaluation is 2 meters, which we replicate and denote as MR_2).

Finally, to evaluate the integration of map information, we adapt Drivable Area Compliance [5] to pedestrian motion prediction. Typically, Drivable Area Compliance is calculated for vehicle motion prediction using formula $DAC = \frac{n-m}{n}$, where n denotes the total amount of prediction that the model makes, and m denotes the number of trajectories that go outside of the drivable area at some point. We apply the same logic as the map information constraint in the FOV-RVO model to adopt it for pedestrian motion prediction. We calculate n similarly, while we increase m value if the trajectory fulfills the following two conditions: a) intersects un-walkable area M''_a at any point and b) is NOT entirely contained within explicitly walkable are M'_a. We call this variant non-Drivable Area Compliance (nonDAC) and report it accordingly. Intuitively, nonDAC metric was employed to evaluate the behavior of prediction methods in critical scenarios, such as when pedestrians walk alongside the road/sidewalk border.

The methodology of using only best candidate trajectory to calculate minADE/minFDE has been recently called into question [40]–[44]. We follow [42] and implement Dynamic Average/Final Displacement Error (DynADE/DynFDE) metrics to resolve this partially. The crucial difference is that in standard ADE/FDE calculation, we calculate metric once per given pre-defined trajectory in the dataset. In contrast, in our setup, as we receive the updated trajectory of each agent in the online stream of data, we calculate the new metric value with every newly added trajectory point, averaging these L2 distance values for every agent and then averaging over all agents at the end of each scene. Formally:

$$DynADE = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\sum_{j=1}^{L_i}\frac{1}{F}\sum_{k=1}^{F}||y_i^{j+k} - \hat{y}_i^{j+k}||_2$$

$$DynFDE = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\sum_{j=1}^{L_i}||y_i^{j+F} - \hat{y}_i^{j+F}||_2,$$

where L_i is the agent i trajectory length in the observed scene. We apply analogous modifications to reported MR_2 and nonDAC metrics. Finally, we apply weightened average results from each .bag file scene for our dataset to obtain the final result with weights equal to total N_{ped} present in each .bag file. The new metrics are similar to standard ADE/FDE but correlate more with driving performance when integrated into the autonomous driving framework [42]. Correspondingly, for models that output multiple candidate trajectories, the minADE/minFDE approach is still used by choosing the candidate trajectory with the best metric score to allow comparison between performance on our dataset and standard datasets such as Nuscenes, denoted as minDynADE/minDynFDE.

IV. FINDINGS

A. Quantitative results

We evaluate the FOV-RVO and combined FOV-RVO + GATraj methods on the collected .bag dataset. For the baseline, we have chosen three DL models for pedestrian motion prediction from the last 4 years, one of them being GATraj itself [19], and others being PecNet [45] and SGNet [46]. The models were chosen for their strong performance on conventional datasets such as ETH/UCY [38], [39] and Nuscenes [47], inference real-time performance on the chosen limited hardware, and provided pre-trained models. Crucially, these models were trained without relying on other inputs except past pedestrian trajectories to minimize possible out-of-distribution problems. We evaluate the pre-trained models with a variable amount of the generated candidate trajectories $k = 1, 5, 10$, where the most likely predicted candidate trajectories were selected.

The results are presented in Table I. We also include the performance of the chosen baseline models on the ETH/UCY dataset for comparison. Both the ETH/UCY and .bag datasets assume that the predictions are made for the next 4.8 seconds using 3.2 seconds of the past data. Our FOV-RVO + GATraj model achieves the second-best performance in MinADE and minFDE metrics behind GATraj itself with $k = 10$ while only outputting $k = 2$ candidate trajectories without a need for any training process before deployment. Limiting the number of candidate trajectories that our predictor produces has a positive impact on the computational performance of both the Autoware mini planner module and, in general, on the performance of popular planners such as DESPOT [48], which is a valuable result that leads to the improvement of the overall autonomous driving framework at the cost of small trade-off on the level of the prediction module.

The FOV-RVO + GATraj combined output also significantly improves nonDAC metrics in comparison to GATraj, incorporating contextual map information in motion planning. Pure FOV-RVO model achieves the best result at the nonDAC metric; however, even forcing strict compliance with map constraints does not achieve a perfect nonDAC score, as often pedestrians themselves are detected on non-walkable areas of the map. Another result is that while limiting $k = 1$, the pure FOV-RVO model outperforms all tested DL models despite not requiring the entire trajectory history, but rather only the final velocity estimation, which is significant, as during real-world autonomous vehicle driving, we rarely have the entire observed previous 3.2-second trajectory, but the predictor module still has to produce accurate and reliable predictions immediately.

TABLE II
ABLATION TESTING OF RVO-MODEL REMOVING ADDITIONAL INTRODUCED CONSTRAINTS

Model	RVO	RVO+FOV	RVO+MAP	RVO+FOV+ MAP
minDynADE	0.991	**0.963**	0.998	0.973
minDynFDE	1.790	1.786	1.811	**1.783**
MR_2	0.336	0.337	0.337	**0.322**
nonDAC	0.958	0.956	**0.984**	0.976

B. Ablation testing

We perform an ablation study to evaluate the effect of additional constraints on top of the pure ORCA-expanded model. In Table II, we refer to the RVO as the base ORCA model where pedestrians consider only the velocity obstacles of other pedestrians and assign equal responsibility of $\alpha = 0.5$ to avoid each other. RVO + FOV adds gaze direction considerations and variable responsibilities calculations depending on FOV, while RVO + MAP explores the addition of only map information without FOV constraint. Finally, RVO + FOV + MAP includes both FOV and map constraints and is the same model used for quantitative evaluation. RVO + FOV and RVO + FOV + MAP were also evaluated with the gaze detection module working, while the other two methods didn't process camera information during runtime. It should be noted that out of the total 8600 pedestrians recorded in the .bag dataset, only 800 had corresponding camera-based detections. However, as these 800 pedestrians are in front of the vehicle's front-facing cameras, their movement is the most important to predict. For the rest, the simplified heuristic model of estimating the field-of-view from the velocity vector direction was used. Every variant outputs $k = 1$ predictions, where prediction is obtained from velocity modified by the chosen method. The results show that the combinatorial addition of constraints improves overall performance, leading to optimal model, while adding MAP constraint significantly improves non-DAC metric as expected.

V. CONCLUSION

This work presents the FOV-RVO, a Velocity Obstacle-based model for pedestrian motion prediction. FOV-RVO produces fast and accurate predictions without requiring a complete observed history of agents' trajectories and incorporates auxiliary data like local map representation and pedestrians' gaze directions to better model relationships between pedestrians and the surrounding environment. We evaluate the model inside a modular autonomous driving framework and compare its performance online with selected similarly integrated Deep Learning methods. Our model performs better at a limited amount of candidate trajectories K that is optimal for the downstream planner module.

The limitation of our model and primary future work direction is the need to expand and refine the current method of pedestrian gaze direction detection to achieve better and more accurate coverage of all detected pedestrians by Lidar sensor. This requires expansion of the current frontal-facing camera hardware setup to cover 360-degree camera suit coverage. Another point of future work direction would be to use the RVO model as a direct differentiable layer within a novel DL method to achieve a fusion KB/DL model rather than the current simplified prediction combining. Finally, while our map data input model is general enough to be derived from most underlying map representations in potential application areas, it still requires manual analysis of every specific representation and designing a corresponding converter module, limiting the generalization of the entire FOV-RVO model.

REFERENCES

[1] L. Di Lillo, T. Gode, X. Zhou, M. Atzei, R. Chen, and T. Victor, "Comparative safety performance of autonomous-and human drivers: A real-world case study of the waymo driver," *Heliyon*, vol. 10, no. 14, 2024.

[2] Radio SRF 1, "Neues zeitalter: Die schweiz rüstet sich für den selbstfahrenden verkehr," 2025, accessed: 2025-02-26. [Online]. Available: https://www.srf.ch/radio-srf-1/neues-zeitalter-die-schweiz-ruestet-sich-fuer-den-selbstfahrenden-verkehr

[3] Y. Zhou and M. Xu, "Robotaxi service: The transition and governance investigation in china," *Research in Transportation Economics*, vol. 100, p. 101326, 2023.

[4] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[5] Z. Fu, K. Jiang, C. Xie, Y. Xu, J. Huang, and D. Yang, "Summary and reflections on pedestrian trajectory prediction in the field of autonomous driving," *IEEE Transactions on Intelligent Vehicles*, 2024.

[6] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.

[7] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8748–8757.

[8] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.

[9] R. Korbmacher and A. Tordeux, "Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 126–24 144, 2022.

[10] Y. Yao, S. Yan, D. Goehring, W. Burgard, and J. Reichardt, "Improving out-of-distribution generalization of trajectory prediction for autonomous driving via polynomial representations," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 488–495.

[11] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1364–1384, 2020.

[12] R. Trauth, K. Moller, G. Würsching, and J. Betz, "Frenetix: A high-performance and modular motion planning framework for autonomous driving," *IEEE Access*, 2024.

[13] 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*. Ieee, 2008, pp. 1928–1935.

[14] K. M. Abughalieh and S. G. Alawneh, "Predicting pedestrian intention to cross the road," *IEEE Access*, vol. 8, pp. 72 558–72 569, 2020.

[15] Y. Su, J. Du, Y. Li, X. Li, R. Liang, Z. Hua, and J. Zhou, "Trajectory forecasting based on prior-aware directed graph convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16 773–16 785, 2022.

[16] X. Wang, J. Zhang, H. Zhang, S. Zhao, and H. Liu, "Vision-based gaze estimation: a review," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 2, pp. 316–332, 2021.

[17] R. Algabri, H. Shin, and S. Lee, "Real-time 6dof full-range markerless head pose estimation," *Expert Systems with Applications*, vol. 239, p. 122293, 2024.

[18] T. Matiisen, "Ut-adl/autoware_mini: Autoware mini is a minimalistic python-based autonomy software." 2023. [Online]. Available: https://github.com/UT-ADL/autoware_mini/

[19] H. Cheng, M. Liu, L. Chen, H. Broszio, M. Sester, and M. Y. Yang, "Gatraj: A graph-and attention-based multi-agent trajectory prediction model," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 205, pp. 163–175, 2023.

[20] E. Schuetz and F. B. Flohr, "A review of trajectory prediction methods for the vulnerable road user," *Robotics*, vol. 13, no. 1, p. 1, 2023.

[21] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The international journal of robotics research*, vol. 17, no. 7, pp. 760–772, 1998.

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

[23] 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, 1997.

[24] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 5917–5922.

[25] D. Wilkie, J. Van Den Berg, and D. Manocha, "Generalized velocity obstacles," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 5573–5578.

[26] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research: The 14th International Symposium ISRR*. Springer, 2011, pp. 3–19.

[27] S. Kim, S. J. Guy, W. Liu, D. Wilkie, R. W. Lau, M. C. Lin, and D. Manocha, "Brvo: Predicting pedestrian trajectories using velocity-space reasoning," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 201–217, 2015.

[28] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "Porca: Modeling and planning for autonomous driving among many pedestrians," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3418–3425, 2018.

[29] Y. Luo, P. Cai, Y. Lee, and D. Hsu, "Gamma: A general agent motion model for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3499–3506, 2022.

[30] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[31] J. van den Berg, S. J. Guy, J. Snape, M. C. Lin, and D. Manocha, "Rvo2 library: Reciprocal collision avoidance for real-time multi-agent simulation," *See https://gamma. cs. unc. edu/RVO2*, 2011.

[32] K. Guo, D. Wang, T. Fan, and J. Pan, "Vr-orca: Variable responsibility optimal reciprocal collision avoidance," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4520–4527, 2021.

[33] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[34] I. P. Howard and B. J. Rogers, *Binocular vision and stereopsis*. Oxford University Press, USA, 1995.

[35] D. Zabolotnii, Y. Muhammad, and N. Muhammad, "Pedestrian motion prediction evaluation for urban autonomous driving," *arXiv preprint arXiv:2410.16864*, 2024.

[36] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[37] P. Li, H. Zhao, P. Liu, and F. Cao, "Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving," in *European Conference on Computer Vision*. Springer, 2020, pp. 644–660.

[38] S. Pellegrini, A. Ess, and L. Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*. Springer, 2010, pp. 452–465.

[39] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.

[40] L. A. Thiede and P. P. Brahma, "Analyzing the variety loss in the context of probabilistic trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9954–9963.

[41] B. Ivanovic and M. Pavone, "Rethinking trajectory forecasting evaluation," *arXiv preprint arXiv:2107.10297*, 2021.

[42] H. Wu, T. Phong, C. Yu, P. Cai, S. Zheng, and D. Hsu, "What truly matters in trajectory prediction for autonomous driving?" *arXiv preprint arXiv:2306.15136*, 2023.

[43] A. Mohamed, D. Zhu, W. Vu, M. Elhoseiny, and C. Claudel, "Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation," in *European Conference on Computer Vision*. Springer, 2022, pp. 463–479.

[44] N. Shahroudi, M. Lepson, and M. Kull, "Evaluation of trajectory distribution predictions with energy score," in *Forty-first International Conference on Machine Learning*, 2024.

[45] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 759–776.

[46] C. Wang, Y. Wang, M. Xu, and D. J. Crandall, "Stepwise goal-driven networks for trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2716–2723, 2022.

[47] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[48] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "Despot: Online pomdp planning with regularization," *Advances in neural information processing systems*, vol. 26, 2013.