



# A Bayesian Approach with Prior Mixed Strategy Nash Equilibrium for Vehicle Intention Prediction

Giovanni Lucente<sup>1,2</sup> · Reza Dariani<sup>1</sup> · Julian Schindler<sup>1</sup> · Michael Ortgiese<sup>1,2</sup>

Received: 15 August 2022 / Accepted: 22 April 2023 / Published online: 22 August 2023  
© The Author(s) 2023

## Abstract

The state-of-the-art technology in the field of vehicle automation will lead to a mixed traffic environment in the coming years, where connected and automated vehicles have to interact with human-driven vehicles. In this context, it is necessary to have intention prediction models with the capability of forecasting how the traffic scenario is going to evolve with respect to the physical state of vehicles, the possible maneuvers and the interactions between traffic participants within the seconds to come. This article presents a Bayesian approach for vehicle intention forecasting, utilizing a game-theoretic framework in the form of a Mixed Strategy Nash Equilibrium (MSNE) as a prior estimate to model the reciprocal influence between traffic participants. The likelihood is then computed based on the Kullback-Leibler divergence. The game is modeled as a static nonzero-sum polymatrix game with individual preferences, a well known strategic game. Finding the MSNE for these games is in the PPAD  $\cap$  PLS complexity class, with polynomial-time tractability. The approach shows good results in simulations in the long term horizon (10s), with its computational complexity allowing for online applications.

**Keywords** Vehicle intention prediction · Trajectory prediction · Bayesian approach · Mixed strategy · Nash equilibrium

## Abbreviations

GNI	Gradient-based nikaïdo isoda
minADE	Minimum average displacement error
minFDE	Minimum final displacement error
MR	Missing rate
MSNE	Mixed strategy nash equilibrium
NE	Nash equilibrium
pdf	Probability density function
PLS	Polynomial local search
PPAD	Polynomial parity arguments on directed graphs

## 1 Introduction

Risk identification in traffic is essential for guaranteeing safe driving in automated vehicles. Dangerous scenarios could arise from an inaccurate estimation of future trajectories of other traffic participants, given the uncertainty of human behavior. Prediction is therefore necessary in order to guarantee a safe decision making. Nevertheless, a reliable estimation can not be limited to a short-term prediction based on dynamic or kinematic models, but it has to take into account possible future interactions and influence between the traffic participants in the scenario.

Many trajectory predictors are data-driven-based and they suffer from exponentially increasing sample complexity when trying to predict trajectories across a joint space in a multi-agent environment, as the number of agents increases. Edge cases constitute another challenge, particularly when the learning data set is largely comprised of straightforward traffic scenarios.

This paper presents a methodology to predict the future trajectories of vehicles in traffic, considering the information coming from sensors in terms of actual state of the vehicles and interactions and mutual influence between them, that can come from possible future traffic outcomes. The proposed model is therefore multi-agent, not based on

---

Academic Editor: Hong Wang

---

✉ Giovanni Lucente  
giovanni.lucente@dlr.de

<sup>1</sup> Institute of Transportation Systems, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany

<sup>2</sup> Fakultät Verkehrs- und Maschinensysteme, TU Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany

data-driven learning and generally adaptable to any topology. The contributions of this study are the following:

1. Definition of an innovative and effective framework for vehicle intention prediction through the use of the Bayes' theorem. The approach considers both the rational prior outcome of the traffic scenario through a Mixed Strategy Nash Equilibrium (MSNE) and the current evidence from the vehicle data as the likelihood. The game-theoretic framework takes into account the interactions and reciprocal influence between the vehicles, while the likelihood corrects the prior estimate considering the actual short-term trajectories that the agents are going to take.
2. The model outputs probability distributions over a set of possible trajectories that vehicles can take. The finite action space of the agents allows the game to be modeled as a static nonzero-sum polymatrix game (network coordination game) with individual preferences. Polymatrix games with individual preferences belongs to the  $PPAD \cap PLS$  complexity class, with polynomial-time tractability [1–3]. The computational time satisfies the real-time application. The simplification of the game does not negatively impact performance, as the results are still acceptable even for long-term horizons of 10 seconds.

The organization of this paper are as below. In Sect. 2, some background related to the different models of trajectory prediction algorithm is provided. The formulation of the problem can be found in Sect. 3. In Sect. 4, the approach is presented. Section 5 shows the details of a case study. In Sect. 6, some results in simulation environment in terms of performances and computational time are provided. Finally, the results and potential future research directions are discussed in Sect. 7.

## 2 Related Works

Most motion prediction approaches can be categorized following the three-levels scheme proposed by Ref. [4]: (1) Physics-based motion prediction algorithms, which rely on dynamic and kinematic models of the system [5, 6]; (2) Maneuver-based motion models, which consider the maneuver intention of the traffic participant and incorporate a high-level-strategic layer [7–9]; (3) Interaction-aware prediction algorithms, which consider the possible interactions and inter-dependencies between the vehicles in the traffic scenarios [10, 11].

The physics-based motion prediction algorithms have an acceptable reliability only on a short-term horizon, since the possible maneuver and the long-term interactions between

the vehicles are ignored. Also the maneuver-based motion models fall short in the context of a complex traffic scenario. As stated by Ref. [10] indeed, many trajectory prediction approaches focus on estimating marginal prediction samples of possible future trajectories of single vehicles, failing in considering interactions and mutual influence between the traffic participants. The interaction-aware prediction algorithms try to provide a solution to this problem by modeling the multi-agent environment.

A second classification, transversal to the previous one proposed by Ref. [8], divides the approaches into: (1) Deterministic models, which predict for each agent a unique trajectory that is considered the most probable [12–14]; (2) Stochastic models, which define a non-deterministic framework for the estimation, for instance predicting the likelihood of a finite set of outcome-representative trajectories [8, 15–18], or defining a probabilistic state-space occupancy grid [19]. The predictive method proposed in this article can be classified as stochastic, maneuver-based and interaction-aware. It is stochastic because the output is not a single trajectory but a probability distribution over a set of trajectories, maneuver-based since maneuvers are predicted, interaction-aware because the forecast outcome includes the result of a MSNE, within a multi-agent interactive and game theoretic framework.

Game theory is a powerful approach that can be used in prediction and decision making. Recent examples can be found in Refs. [20–22]. In Ref. [22], Nash and Stackelberg equilibria are applied for human-like decision making, modeling the different driving styles and social interaction characteristics. In Ref. [20], the authors define an online method of predicting multiagent interactions estimating their SVO (Social Value Orientation). The interactions between agents are modeled as a best-response game and the control policy is found by solving the dynamic game and finding the Nash equilibrium. In Ref. [21], the other agents' cost function parameters are estimated online then used to find the Nash equilibrium in a discretized dynamic game. These works model the interaction as a multi-agent dynamic game, since the action space consists on the agents' vehicles inputs. However, the problem of finding a Nash equilibrium of a dynamic game can be computationally unfeasible in real-time applications, particularly under a long time horizon. For this reason, the current study considers the high-level decisions that vehicles can take, or the possible maneuvers in the scenario, as the action space. This allows to have a finite action space, a  $N$ -person static finite polymatrix game that can always be solved by finding a Nash equilibrium in mixed strategies [23].

The approach proposed in this article has been inspired particularly by the work of Refs. [11, 15] and [24]. In Ref. [15], the inference of a distribution of high-level, abstract driving maneuvers has been taken as reference for the

desired output. Moreover, the philosophy under the proposed Bayesian network in Ref. [15] reflects the Bayesian approach proposed in this article. In Ref. [11], a Nash equilibrium in mixed and behavioural strategies is used to predict the future vehicles' maneuvers. The mixed strategies for a player are a probability distribution over all of the player's pure strategies, therefore the possible maneuvers. Finally, the computation of the MSNE utilizes a merit function, known as the Gradient-based nikaido-isoda (GNI) function, as introduced in Ref. [24].

### 3 Problem Statement

For each traffic participant in the scenario, a set of representative trajectories are computed based on map information (Fig.1). These trajectories represent particular maneuvers and behaviours that the vehicle could take in the future. The definition of these trajectories has been inspired particularly by the work of Ref. [15]. The trajectory computation is performed in two stages: (1) Computation of the geometrical path through a Particle Swarm Optimization algorithm. (2) Definition of the acceleration profile along the path. Local uncertainty along the trajectory is introduced through a evolution kinematic model with Gaussian noise on the vehicle state and input variables.

The trajectories defined for each vehicle are:

- Acceleration trajectory: trajectory in which  $a \sim N(a | \mu_a, \sigma_a^2)$ , where  $\mu_a = 1.5 \text{ m/s}^2$  and  $\sigma_a^2 = \sigma_0^2 + \alpha t$ ,  $\sigma_0^2 = 0.5 \text{ m}^2/\text{s}^4$ ,  $\alpha = 1e - 3$

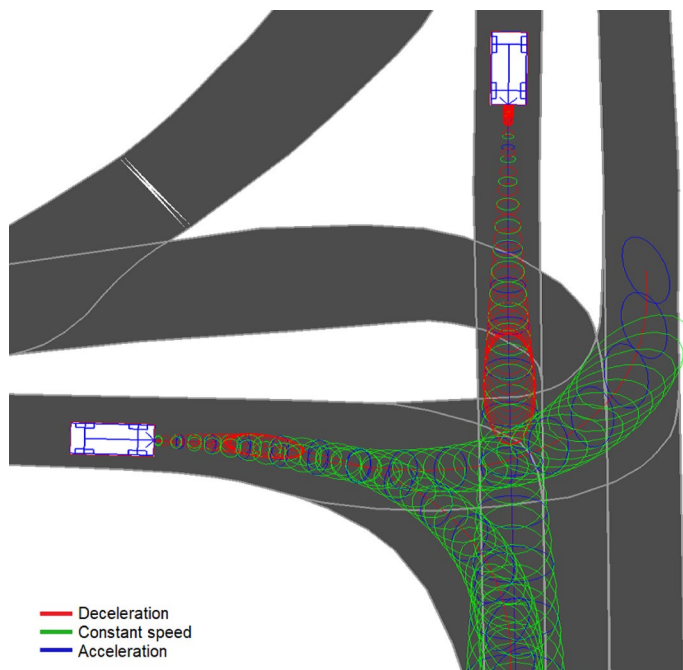
- Constant speed trajectory: trajectory in which  $a \sim N(a | \mu_a, \sigma_a^2)$ , where  $\mu_a = 0 \text{ m/s}^2$  and  $\sigma_a^2 = \sigma_0^2 + \alpha t$ ,  $\sigma_0^2 = 0.5 \text{ m}^2/\text{s}^4$ ,  $\alpha = 1e - 3$
- Braking trajectory: trajectory in which the acceleration is the output of an *Intelligent driver model* (IDM) when it is overlapping with trajectories of other vehicles,  $a \sim N(a | \mu_a, \sigma_a^2)$ , where  $\mu_a = \text{IDM}$  or  $-0.5 \text{ m/s}^2$  (if there is no overlapping) and  $\sigma_a^2 = \sigma_0^2 + \alpha t$ ,  $\sigma_0^2 = 0.5 \text{ m}^2/\text{s}^4$ ,  $\alpha = 1e - 3$
- Harsh braking trajectory: trajectory in which  $a \sim N(a | \mu_a, \sigma_a^2)$ , where  $\mu_a = -3.0 \text{ m/s}^2$  and  $\sigma_a^2 = \sigma_0^2 + \alpha t$ ,  $\sigma_0^2 = 0.5 \text{ m}^2/\text{s}^4$ ,  $\alpha = 1e - 3$

Considering the state vector  $\mathbf{x} = \{x, y, \psi, v\}^T$ , where  $x$  and  $y$  are the Cartesian coordinates of the position,  $\psi$  is the heading and  $v$  is the speed, the time step  $t$  in the trajectory can be represented by the following normal distribution:

$$p(\mathbf{x}_t | \tau) = N(\mathbf{x}_t | \boldsymbol{\mu}_t^\tau, \boldsymbol{\Sigma}_t^\tau) \tag{1}$$

Where  $p(\mathbf{x}_t | \tau)$  is the probability density function (pdf) of the state  $\mathbf{x}$  conditioned by the choice of the trajectory  $\tau$  at time step  $t$ ,  $\boldsymbol{\mu}_t^\tau$  comes from the  $\tau$  trajectory computation and  $\boldsymbol{\Sigma}_t^\tau$  is the covariance matrix on the state space that gives local uncertainty. Starting from a simple bicycle model of the system  $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ , where  $\mathbf{u}(t) = \{a(t), \delta(t)\}^T$  is the vector of the inputs (acceleration  $a(t)$  and steering angle  $\delta(t)$ ), through inverse kinematics the steering angle  $\delta(t)$  required is computed. Considering the linearized time-discrete version  $\mathbf{x}_{t+1} = \mathbf{A}_t \mathbf{x}_t + \mathbf{B}_t \mathbf{u}_t$  the dynamic model of the covariance matrix is defined:

**Fig. 1** Example of trajectories. The ellipses represent the multivariate normal distributions at each time step



$$\Sigma_{t+1} = A_t^T \Sigma_t A_t + B_t^T \Gamma_t B_t \tag{2}$$

Where  $\Gamma_t = \text{diag}(\sigma_a^2(t), \sigma_\delta^2(t))$  is the diagonal covariance matrix of the inputs, with variances linearly increasing over time.

The policy of the  $i$ -th traffic participant  $\pi(\tau_j | \mathbf{x}_0)$  is a probability distribution over the finite number of trajectories available  $\tau_j, j = 1, \dots, N_i$  and represents the future intention of the vehicle, conditioned by the actual vehicle state  $\mathbf{x}_0$ . Marginalizing on the agent policy, the pdf of the state vector  $\mathbf{x} = \{x, y, \psi, v\}^T$  at time  $t$  conditioned by the initial state  $\mathbf{x}_0$  is:

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{x}_0) &= \sum_{j=1}^{j=N_i} \pi(\tau_j | \mathbf{x}_0) p(\mathbf{x}_t | \tau_j) \\ &= \sum_{j=1}^{j=N_i} \pi(\tau_j | \mathbf{x}_0) N(\mathbf{x}_t | \boldsymbol{\mu}_t^{\tau_j}, \Sigma_t^{\tau_j}) \end{aligned} \tag{3}$$

The result is a Gaussian Mixture Model distribution, weighted by the agent policy  $\pi(\tau_j | \mathbf{x}_0)$ . This policy represents the strategic uncertainty over the intentions of the vehicle and it is the object of interest of the prediction approach presented in this article.

### 4 Application of the Bayes' Theorem

The Bayes' theorem is applied:

$$\pi(\tau_j | \mathbf{x}_0) = \frac{p(\tau_j) p(\mathbf{x}_0 | \tau_j)}{p(\mathbf{x}_0)} = \frac{p(\tau_j) p(\mathbf{x}_0 | \tau_j)}{\sum_{k=1}^{k=N_i} p(\tau_k) p(\mathbf{x}_0 | \tau_k)} \tag{4}$$

Where  $p(\tau_j)$  is the prior probability of the trajectory  $\tau_j$  and  $p(\mathbf{x}_0 | \tau_j)$  is the likelihood, which gives a measure of compatibility between the current evidence coming from data  $\mathbf{x}_0$  and the pre-computed trajectory  $\tau_j$ . The  $p(\tau_j)$  is the result of a MSNE,  $p(\mathbf{x}_0 | \tau_j)$  is found defining a likelihood function.

#### 4.1 Prior Probability Through Mixed Strategy Nash Equilibrium

The MSNE is the solution to a non-cooperative game involving two or more players, considering mixed strategies (probability distributions over the action space) instead of pure strategies. A mixed strategy profile is considered an MSNE if each player's strategy is the best response to the strategies of all other players. This decision to use the prior distribution as an MSNE comes from the hypothesis that every traffic participant, as a rational player, is prone to adopt his optimal strategy in the multi-agent traffic scenario.

Each vehicle (player)  $i = 1, \dots, M$  in the scenario is supposed to choose among a finite set of  $N_i$  trajectories, defined

by the notation  $\mathcal{T}^i = \{\tau_1^i, \dots, \tau_{N_i}^i\}$ .  $N$  indicates the total number of trajectories that the vehicles can take, i.e.  $N = \sum_{j=1}^M N_j$ .

Let the notation

$$\boldsymbol{\theta}_i = [p(\tau_1^i), \dots, p(\tau_{N_i}^i)]^T \in \Theta_i \tag{5}$$

where  $\Theta_i = \left\{ \boldsymbol{\theta} \in \mathbb{R}^{N_i} \mid \sum_{k=1}^{N_i} \theta_k = 1 \right\}$

denote the mixed-strategy vector of the  $i$  player, i.e. the probability distribution over the available trajectories, while

$$\boldsymbol{\gamma} = [\boldsymbol{\theta}_1^T, \dots, \boldsymbol{\theta}_M^T]^T \in \Gamma$$

where  $\Gamma = \left\{ \boldsymbol{\gamma} \in \mathbb{R}^N \mid \boldsymbol{\gamma} = [\boldsymbol{\theta}_j^T]_{j=1, \dots, M}, \boldsymbol{\theta}_j \in \Theta_j \right\}$  (6)

denotes the vector of the mixed-strategies of all the  $M$  players.

Let  $\boldsymbol{\gamma}_{-i} = [\boldsymbol{\theta}_1^T, \dots, \boldsymbol{\theta}_{i-1}^T, \boldsymbol{\theta}_{i+1}^T, \dots, \boldsymbol{\theta}_M^T]^T$  denote the set of mixed-strategies except  $i$ . The notation  $\boldsymbol{\tau} = \{\tau^1, \dots, \tau^M\}$  indicates the joint pure strategy among all the players ( $i$  player chooses trajectory  $\tau^i$ ), while  $f_i$  denotes the payoff function for the  $i$  player and it takes into account safety, efficiency and comfort.

The following problem is considered:

Find  $\boldsymbol{\gamma}^* = [\boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_M^*]^T$   
 s.t.  $\boldsymbol{\theta}_i^* = \arg \min_{\boldsymbol{\gamma}: \boldsymbol{\gamma}_{-i} = \boldsymbol{\gamma}_{-i}^*} \mathbb{E}_{\boldsymbol{\tau} \sim \boldsymbol{\gamma}} [f_i(\boldsymbol{\tau})]$  (7)

A point  $\boldsymbol{\gamma}^*$  that satisfies Eq.(7) is called a *Nash equilibrium* (NE). Every N-person static finite game in normal form admits a noncooperative NE solution in mixed strategies [23].

#### 4.1.1 Payoff Function

The payoff function  $f_i$  for each traffic participant takes into account safety ( $f_i^S$ ), comfort ( $f_i^C$ ) and efficiency ( $f_i^E$ ) with weighting coefficients ( $\alpha^S, \alpha^C, \alpha^E$ ) as shown in Eq.(8). The definition of payoff function considering pure strategies is presented here, which means that player  $i$  chooses trajectory  $\tau^i$  with probability 1.

$$f_i(\boldsymbol{\tau}) = \alpha^S f_i^S(\boldsymbol{\tau}) + \alpha^C f_i^C(\tau^i) + \alpha^E f_i^E(\tau^i) \tag{8}$$

The payoff function for safety  $f_i^S$  depends on the joint pure strategy of all the players  $\boldsymbol{\tau}$ , while both the ones for efficiency  $f_i^E$  and comfort  $f_i^C$  depend only on the trajectory chosen by player  $i$ , that is  $\tau^i$ . The weighting coefficients have been tuned to optimize the performance in the simulation environment used in this article [25].

The safety payoff is computed following Eq. (9):

$$f_i^S(\tau) = w^S \sum_{j \neq i} f_{ij}^S(\tau^i, \tau^j)$$

$$f_{ij}^S(\tau) = w^S \sum_{j \neq i} \sum_{t=0}^{T-1} \gamma^t \exp\left(-(\mu_t^j - \mu_t^i)^T \hat{\Sigma}^{-1} (\mu_t^j - \mu_t^i)\right) \quad (9)$$

Where  $f_{ij}^S(\tau^i, \tau^j)$  is the safety payoff for player  $i$  (and player  $j$  for symmetry) considering trajectory  $\tau^i$  for player  $i$  and trajectory  $\tau^j$  for payer  $j$ ,  $w^S$  is the penalty for crash,  $\gamma$  is the discount factor,  $\mu_t^i$  is the mean vector  $(x, y)$  of the multivariate normal distribution  $N(x_t | \mu_t^i, \Sigma_t^i)$ , which gives the pdf of the position of vehicle  $j$  on its trajectory  $\tau^j$  at time step  $t$ .

The definition of the safety payoff, in particular the exponent in Eq. (9), is inspired by the Mahalanobis distance [26], with a slightly modification. Indeed, the Mahalanobis distance defines a distance measure between a point and a distribution, while here a distance between two distributions is needed. Therefore it is necessary to modify the covariance matrix, that is  $\hat{\Sigma} = 0.5(\Sigma_t^i + \Sigma_t^j) + \beta I$ . The covariance matrix is therefore the average between the covariance matrices of the two distributions, with the additional term  $\beta I$ . This term has two scopes: (1) to guarantee in any case a payoff when the two distributions are close, even with small covariance matrices, (2) to avoid that  $\hat{\Sigma}$  is badly conditioned for the inversion. A further description is given in Fig.2.

The comfort payoff is the following:

$$f_i^C(\tau^i) = \sum_{t=0}^{T-1} \gamma^t \left( w_{\text{long}}^C \|a_t^{\text{long}}\| + w_{\text{lat}}^C \|a_t^{\text{lat}}\| \right) \quad (10)$$

Where  $a_t^{\text{long}}$  and  $a_t^{\text{lat}}$  are the longitudinal and lateral acceleration at time step  $t$  of the trajectory  $\tau^i$ , while  $w_{\text{long}}^C$  and  $w_{\text{lat}}^C$  are the respective penalties.

The efficiency payoff is the following:

$$f_i^E(\tau^i) = w^E \sum_{t=0}^{T-1} \gamma^t (v_{\text{lim}} - v_t)^2 \quad (11)$$

**Fig. 2** The computation of the safety payoff in Eq. (9) is based on a distance measure between the multivariate normal distributions that describe the uncertainty about the future steps of the trajectories

In Eq.(11),  $v_{\text{lim}}$  is the speed limit and  $v_t$  the speed at time step  $t$  of the trajectory  $\tau^i$ .

Considering the previous definitions, it is possible now to extend it in the general case of mixed strategies, in which the action of player  $i$  is a probability distribution over the possible trajectories:  $\theta_i = [p(\tau_1^i), \dots, p(\tau_{N_i}^i)]^T$ .  $\tau_k$ , with  $k = 1, \dots, N_i$ , are the available trajectories of player  $i$ .

Reminding that  $\gamma = [\theta_1^T, \dots, \theta_M^T]^T$  denotes the set of the mixed-strategies of all the  $M$  players, the objective is to find an expression for the expected payoff in case of mixed strategies, i.e.  $\mathbb{E}_{\tau \sim \gamma} [f_i(\tau)]$ .

### 4.1.2 Polymatrix Coordination Game with Individual Preferences

The definition of the players' payoff allows to model the game as a polymatrix coordination game with individual preferences, a well known strategic game. Let's define the matrix  $P^{ij} \in \mathbb{R}^{N_i \times N_j}$  as the matrix of safety payoffs between vehicle  $i$  and vehicle  $j$ . The  $p_{n,m}^{ij}$  element of the matrix is:

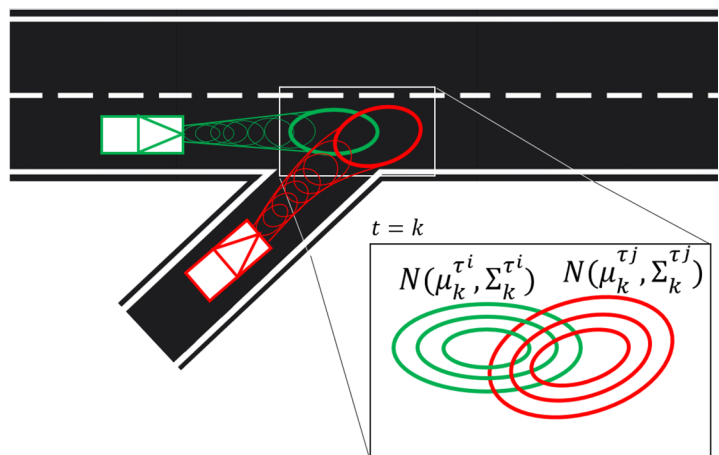
$$p_{n,m}^{ij} = f_{ij}^S(\tau_n^i, \tau_m^j) \quad \tau_n^i \in \mathcal{T}^i, \tau_m^j \in \mathcal{T}^j \quad (12)$$

Where  $\mathcal{T}^i$  and  $\mathcal{T}^j$  indicate the sets of trajectories available to the  $i$  and  $j$  vehicle. Therefore the  $(n, m)$  element of the matrix  $P^{ij}$  is the safety payoff for vehicle  $i$  and  $j$  given that vehicle  $i$  chooses the trajectory  $\tau_n^i$  and vehicle  $j$  the trajectory  $\tau_m^j$ .

Let's recall that  $N$  indicates the total number of all the available trajectories of the vehicles in the scenario. Considering the matrix  $\hat{Q}^i \in \mathbb{R}^{N_i \times N}$ , defined as:

$$\hat{Q}^i = [P^{i,1} \dots P^{i,i-1} \mathbf{0}^{N_i \times N_i} P^{i,i+1} \dots P^{i,M}] \quad (13)$$

It is possible to define the matrix  $Q^i \in \mathbb{R}^{N \times N}$  for each traffic participant  $i$ :



$$Q^i = \left[ \mathbf{0}^{N \times \sum_{j=1}^{i-1} N_j} \hat{Q}^i \mathbf{0}^{N \times \sum_{j=i+1}^M N_j} \right]^T \tag{14}$$

Considering the vector  $\hat{r}^i \in \mathbb{R}^{N_i}$  for each vehicle  $i$ , whose element  $\hat{r}_j^i$  is:

$$\hat{r}_j^i = f_i^C(\tau_j^i) + f_i^E(\tau_j^i) \tag{15}$$

The vector  $r^i \in \mathbb{R}^N$  for each vehicle  $i$  is defined:

$$r^i = \left[ \mathbf{0}^{1 \times \sum_{j=1}^{i-1} N_j} \hat{r}^i \mathbf{0}^{1 \times \sum_{j=i+1}^M N_j} \right] \tag{16}$$

The expected payoff in the mixed strategy game is:

$$\begin{aligned} \mathbb{E}_{\tau \sim \gamma} [f_i(\tau)] &= \sum_{k=1}^{N_i} p(\tau_k^i) [f_i^C(\tau_k^i) + f_i^E(\tau_k^i)] \\ &+ \sum_{j=1}^M \sum_{n=1}^{N_j} f_{ij}^S(\tau_k^i, \tau_n^j) p(\tau_n^j) \end{aligned} \tag{17}$$

That, considering the definitions Eqs. (12) and (15) are:

$$\mathbb{E}_{\tau \sim \gamma} [f_i(\tau)] = \theta_i^T [\hat{r}^i + \sum_{j=1}^M P^{i,j} \theta_j] \tag{18}$$

The expression can be made quadratic with respect to  $\gamma \in \mathbb{R}^N$  by means of the definitions Eqs.(14) and (16):

$$\mathbb{E}_{\tau \sim \gamma} [f_i(\tau)] = \gamma^T r^i + \gamma^T Q^i \gamma \tag{19}$$

The component  $\gamma^T Q^i \gamma$  gives the payoff given by the interaction with other drivers, so linked with safety and trajectories' overlapping. The component  $\gamma^T r^i$  is the payoff exclusively dependent by the choice of the player, therefore linked with comfort and efficiency.

Note that the Eq. (18) is the definition of a static nonzerosum polymatrix coordination game (network coordination game) with individual preferences. The matrix  $P^{i,j} \in \mathbb{R}^{N_i \times N_j}$  is the symmetric payoff of the bimatrix game  $\{i, j\}$ , the vector  $\hat{r}^i \in \mathbb{R}^{N_i}$  represents the individual preference function of player  $i$ . Polymatrix games with individual preferences belongs to the PPAD  $\cap$  PLS complexity class, with polynomial-time tractability [1, 2].

The Gradient-based Nikaido-Isoda (GNI) function [24] is used to find the NE. With the payoff function defined in Eq. (19), the gradient can be computed analytically with evident gain in terms of computational efficiency.

### 4.1.3 Optimization via Gradient-Based Nikaido-Isoda Function

The NE of the N-player game considered in this article Eq. (7) is found using the GNI Function, introduced in Ref.

[24]. Let's indicate the expected payoff for player  $i$  with the function  $g_i$  for simplicity of notation:

$$g_i(\gamma) = \mathbb{E}_{\tau \sim \gamma} [f_i(\tau)] = \gamma^T r^i + \gamma^T Q^i \gamma \tag{20}$$

The GNI function  $V(\gamma, \eta)$  is the following:

$$\begin{aligned} V(\gamma, \eta) &= \sum_{i=1}^M g_i(\gamma) - g_i(\tilde{\gamma}(i, \eta)) \\ \text{where } \tilde{\gamma}_j(i, \eta) &= \tilde{\theta}_j = \begin{cases} \theta_j - \eta \nabla_j g_j(\gamma), & \text{if } j = i \\ \theta_j, & \text{otherwise} \end{cases} \end{aligned} \tag{21}$$

Reminding that  $M$  is the number of players (traffic participants) in the scenario,  $\gamma = [\theta_1^T, \dots, \theta_M^T]^T \in \mathbb{R}^N$  with  $N = \sum_{j=1}^M N_j$  is the vector of the mixed strategies of the players and  $\theta_j \in \mathbb{R}^{N_j}$  is the mixed strategy vector of the  $j$  player.  $\nabla_j g_j(\gamma)$  denotes the gradient of  $g_j$  with respect to  $\theta_j$  that indicates the direction of maximum increase of the payoff for player  $i$  in its action space  $\Theta_i$ .

The idea under this merit function is that every player can locally improve their objectives using the steepest descent direction, instead of computing a globally optimal solution [24].

The gradient of  $V_i(\gamma, \eta) := g_i(\gamma) - g_i(\tilde{\gamma}(i, \eta))$  is the following:

$$\nabla V_i(\gamma, \eta) = \nabla g_i(\gamma) - (\mathbf{I} - E_i H_{g_i}(\gamma)) \nabla g_i(\tilde{\gamma}(i, \eta)) \tag{22}$$

Where  $E_i := F_i F_i^T \in \mathbb{R}^{N \times N}$ ,  $F_i = [\mathbf{0}^{N_i \times \sum_{j=1}^{i-1} N_j} \mathbf{I}_i \mathbf{0}^{N_i \times \sum_{j=i+1}^M N_j}]^T \in \mathbb{R}^{N \times N_i}$ ,  $H_{g_i}(\gamma) := \nabla(\nabla g_i(\gamma))$  is the hessian matrix,  $\mathbf{I} \in \mathbb{R}^{N \times N}$  and  $\mathbf{I}_i \in \mathbb{R}^{N_i \times N_i}$  are identity matrices.

Considering the function  $g_i(\gamma)$  in Eq. (20), the gradients are:

$$\begin{aligned} \nabla g_i(\gamma) &= (Q^i + Q^{i \ T}) \gamma + r^i \in \mathbb{R}^N \\ \nabla_i g_i(\gamma) &= F_i^T \nabla g_i(\gamma) \in \mathbb{R}^{N_i} \end{aligned} \tag{23}$$

The modified vector of mixed strategies  $\tilde{\gamma}(i, \eta) \in \mathbb{R}^N$  is:

$$\begin{aligned} \tilde{\gamma}(i, \eta) &= \gamma - \eta F_i \nabla_i g_i(\gamma) \\ \tilde{\gamma}(i, \eta) &= \gamma - \eta E_i \nabla g_i(\gamma) \end{aligned} \tag{24}$$

The Hessian  $H_{g_i}(\gamma) \in \mathbb{R}^{N \times N}$  is:

$$H_{g_i}(\gamma) := \nabla(\nabla g_i(\gamma)) = (Q^i + Q^{i \ T}) \tag{25}$$

As it is evident,  $\nabla V_i(\gamma, \eta)$  can be computed analytically by using Eqs. (22), (23), (24) and (25). Finally, the gradient of  $V(\gamma, \eta)$  is:

$$\nabla V(\gamma, \eta) = \sum_{i=1}^M \nabla V_i(\gamma, \eta) \tag{26}$$

That is used for the descent iteration:

$$\boldsymbol{\gamma}^{(k+1)} = \boldsymbol{\gamma}^{(k)} - \rho \nabla V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta}) \tag{27}$$

The fact that  $\boldsymbol{\gamma}^{(k+1)}$  belongs to the allowed space  $\Gamma$  (defined in Eq. (6)) is not ensured by considering only Eq. (27) because the gradient  $\nabla V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta})$  does not belong in general to  $\Gamma$ . This is the reason for which it is necessary to project  $\nabla V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta})$  into  $\Gamma$  before computing the descent in Eq. (27) and this theme is faced in Sect. 4.1.4.

#### 4.1.4 Gradient Projection

In the optimization problem described in Sect. 4.1.3, the gradient  $\nabla V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta})$  is projected into the feasible space  $\Gamma$ . This is necessary in order to allow that  $\boldsymbol{\gamma}^{(k)} \in \Gamma \forall k = 1, 2, \dots$ . The projection procedure is based on the consideration that, from the definition of  $\Gamma$  in Eq. (6), every mixed strategy  $\boldsymbol{\theta}_i$  must belong to the space  $\Theta$ , defined in Eq. (5). This definition ensures that  $\boldsymbol{\theta}_i$  is actually a probability distribution over the possible trajectories of player  $i$ , i.e.  $\sum_{j=1}^{N_i} \theta_{i,j} = 1$ . Considering that  $\nabla V = [\nabla_1 V, \dots, \nabla_i V, \dots, \nabla_M V]^T$ , where  $\nabla_i V$  is the gradient with respect to  $\boldsymbol{\theta}_i$ , for each  $\nabla_i V$  the following steps are executed:

1. Projection properly defined:

$$\nabla_i^{\parallel} V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta}) = \nabla_i V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta}) - \langle \nabla_i V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta}), \hat{\boldsymbol{n}} \rangle \hat{\boldsymbol{n}} \tag{28}$$

Where  $\hat{\boldsymbol{n}}$  is the normal vector to the hyperplane  $\sum_{j=1}^{N_i} \theta_{i,j} = 1$ , that is  $\hat{\boldsymbol{n}} = \mathbf{1}$ , while with the notation  $\langle \cdot, \cdot \rangle$  is indicated the scalar product.

2. Tuning of the module:

- (a) Computation of the step:

$$\boldsymbol{\theta}_i^{(k+1)} = \boldsymbol{\theta}_i^{(k)} - \rho \nabla_i^{\parallel} V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta}) \tag{29}$$

- (b) While  $\exists \theta_{i,j} < 0$ :

$$\begin{aligned} \nabla_i^{\parallel} V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta}) &= \alpha \nabla_i^{\parallel} V(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\eta}) & \alpha &= 0.9 \\ \text{repeat (a)} & & & \end{aligned} \tag{30}$$

#### 4.2 Likelihood Function

The evaluation of the likelihood term  $p(\mathbf{x}_0 | \tau_j)$ , which gives a measure of how likely is the current state  $\mathbf{x}_0$ , coming from sensor data, with respect to trajectory  $\tau_j$ , is based on the following steps:

1. A short-term trajectory  $\hat{\tau}$  is computed through a simple bicycle model  $f(\cdot)$ , starting from current state  $\mathbf{x}_0 \sim N(\mathbf{x} | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$  and considering the input variables

$\mathbf{u}_t = \{a_t, \delta_t\} \sim N(\mathbf{u} | \mathbf{u}_0, \boldsymbol{\Sigma}_t^u)$ , therefore considering acceleration and steering angle normally distributed around the initial estimated input  $\mathbf{u}_0 = \{a_0, \delta_0\}$ .

2. The Kullback-Leibler divergence is measured between the pdf of the short-term trajectory  $p(\mathbf{x}_t | \hat{\tau})$  and the pdf of the reference trajectory  $p(\mathbf{x}_t | \tau_j)$  at each time step  $t$ :

$$\begin{aligned} D_{\text{KL}}(p(\mathbf{x}_t | \hat{\tau}) || p(\mathbf{x}_t | \tau_j)) \\ = \int_{\mathbb{R}^n} N(\mathbf{x}_t | \boldsymbol{\mu}_t^{\hat{\tau}}, \boldsymbol{\Sigma}_t^{\hat{\tau}}) \ln \frac{N(\mathbf{x}_t | \boldsymbol{\mu}_t^{\hat{\tau}}, \boldsymbol{\Sigma}_t^{\hat{\tau}})}{N(\mathbf{x}_t | \boldsymbol{\mu}_t^{\tau_j}, \boldsymbol{\Sigma}_t^{\tau_j})} d\mathbf{x} \end{aligned} \tag{31}$$

The Kullback-Leibler divergence  $D_{\text{KL}}(P||Q)$  can be seen as the information lost when the distribution  $Q$  is used to approximate the distribution  $P$ , in this case, considering the whole horizon, how much the trajectory  $\tau_j$  ( $Q$ ) is failing in representing the "true" trajectory  $\hat{\tau}$  ( $P$ ).

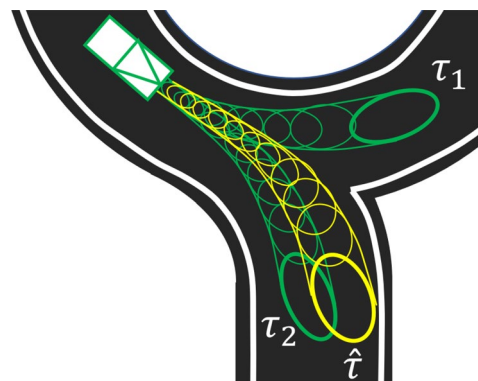
3.  $p(\mathbf{x}_0 | \tau_j)$  is computed through a soft-max function that takes as a parameter the sum over the horizon of the Kullback-Leibler divergence between the time steps of the two trajectories:

$$p(\mathbf{x}_0 | \tau_j) = \frac{\exp(-\beta \sum_{t=1}^T D_{\text{KL}}(p(\mathbf{x}_t | \hat{\tau}) || p(\mathbf{x}_t | \tau_j)))}{\sum_{k=1}^{N_i} \exp(-\beta \sum_{t=1}^T D_{\text{KL}}(p(\mathbf{x}_t | \hat{\tau}) || p(\mathbf{x}_t | \tau_k)))} \tag{32}$$

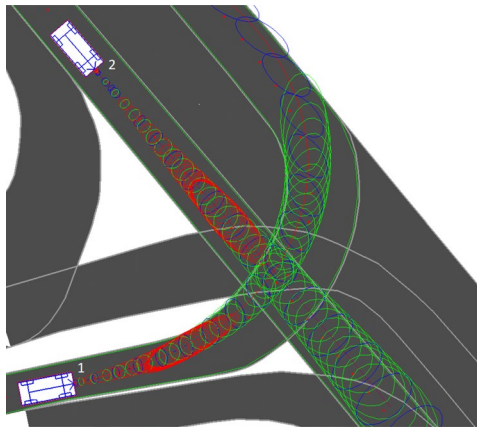
The philosophy under the computation of  $p(\mathbf{x}_0 | \tau_j)$  is to measure how much the current state of the vehicle  $\mathbf{x}_0$ , considering the initial input  $\mathbf{u}_0$ , is coherent with respect to the strategic choice of trajectory  $\tau_j$ . A graphical representation is given in Fig. 3.

#### 5 Case Study

In this section, an example of application is reported, showing briefly how the predictor performs.



**Fig. 3** The likelihood Eq. (4) defines a distance between the trajectories coming from the map topology ( $\tau_1$  and  $\tau_2$ ) and a trajectory obtained considering constant acceleration and steering angle ( $\hat{\tau}$ ). In this case,  $\tau_2$  is expected to be more likely with respect to  $\tau_1$ , considering initial state and input  $(\mathbf{x}_0, \mathbf{u}_0)$



**Fig. 4** The intersection scenario in the simulation

**Table 1** Estimation at  $t = 6$  s, corresponding to the situation in Fig.4

Trajectories	Prior	Likelihood	Posterior
V. 1 acceleration	0.0196	0.0173	0.0007
V. 1 constant speed	0.1693	0.0990	0.0338
V. 1 stop at intersection	0.8663	0.9433	0.9562
V. 1 harsh braking	0.26434	0.0173	0.0092
V. 2 acceleration	0.6733	0.4800	0.8705
V. 2 constant speed	0.0193	0.3688	0.0191
V. 2 stop at intersection	0.2881	0.1415	0.1098
V. 2 harsh braking	0.0193	0.0096	0.0004

The simulation environment is Automated Driving Open Research (ADORE) [25], an open source modular software library and toolkit for decision making, planning, control and simulation of automated vehicles, developed by the Institute of Transportation Systems of the German Aerospace Center (DLR).

The trajectory predictor is applied in the intersection scenario showed in Fig. 4. In the simulation, the vehicle 2 accelerates till the speed of 10 m/s while vehicle 1 decelerate smoothly and stops at the cross. The prediction for this example has a horizon of 5 s.

In Table 1, a punctual estimate is given at time  $t = 6$  s circa, corresponding to the situation shown in Fig. 4. The table provides an example on how the correction mechanism works: the Prior predicts with a high probability that vehicle 1 stops at the intersection and vehicle 2 accelerates, but it also gives space to the opposite possibility, which is strongly reduced by the Likelihood. Note that the Prior does not predict a crash situation, indeed even if vehicle 1 proceeds with constant speed (16%), vehicle 2 accelerates or stops at the intersection, avoiding the collision. This is also evident in the  $P^{ij}$  matrix, defined in Eq. (12) and shown in Table 2.

**Table 2** Matrix  $P^{ij}$  at  $t = 6$  s of the collision payoffs for the vehicles 1 and 2, the definition of this matrix can be found in Eq.12

Trajectories	V.1 acc	V.1 const sp	V.1 stop	V.1 braking
V.2 acc	25641.8	0.0	0.0	0.0
V.2 const sp	947.8	33034.3	0.0	0.0
V.2 stop	0.0	0.6	0.0	0.0
V.2 braking	0.0	0.0	0.0	0.0

**Table 3** minADE and minFDE of the episode shown in Fig. 4. The length of the episode is around 5 s

	Vehicle 1	Vehicle 2	Total
minADE (m)	2.79	1.21	2.00
minFDE (m)	3.01	3.06	3.04
MR	–	–	0.34

The performance of the approach is measured using metrics defined in Ref. [27], in particular the minimum average displacement error (minADE):

$$\text{minADE} = \frac{1}{MT} \sum_{i=1}^M \sum_{t=0}^T \|x_t^i - \mu_t^{\tau_j^i}\|_{L_2} \tag{33}$$

the minimum final displacement error (minFDE):

$$\text{minFDE} = \frac{1}{M} \sum_{i=1}^M \|x_T^i - \mu_T^{\tau_j^i}\|_{L_2} \tag{34}$$

and the Missing rate (MR):

$$\text{MR} = \frac{1}{MT} \sum_{i=1}^M \sum_{t=0}^T \mathbb{1}_{\chi_{0.95}^2} \left( (x_t^i - \mu_t^{\tau_j^i})^T \Sigma_t^{\tau_j^i} (x_t^i - \mu_t^{\tau_j^i}) \right) \tag{35}$$

The trajectory  $\tau_j^i$ , considered for the computation, is the one with the highest posterior probability  $\pi(\tau_j^i | x_0^i)$  of player  $i$ . The expression  $\mathbb{1}_{\chi_{0.95}^2}(\cdot)$  in the MR definition indicates the indicator function that is equal to one if the null hypothesis of the chi-squared goodness of fit test is not rejected, otherwise zero. In particular, the argument of  $\mathbb{1}_{\chi_{0.95}^2}(\cdot)$  is the test statistic, the reference distribution is a chi-squared distribution  $\chi_2^2$  with 2 degree of freedom and 0.95 confidence interval. The MR is basically testing at each time step if the actual point of the trajectory belongs to the distribution of the predicted trajectory.

Regarding the episode in Fig. 4, the results are shown in Table 3. The Table illustrates that both vehicles, vehicle 1 and vehicle 2, have an average displacement error along the episode of 2.79 m and 1.21 m, respectively, with a combined average of 2.0 m. During the last step of the episode, vehicle 1 exhibits a final displacement error of 3.01 m, while vehicle



2 records a final displacement error of 3.06 m, with 3.04 m considering both. This reaffirms the reliability of the predictions even after 5 seconds have elapsed. The missing rate considering both vehicles is 0.34, meaning that for the 66% of the episode length, the vehicles belong to the predicted distributions.

In Fig. 5, the minADE for vehicles 1, 2 and for both vehicles is shown. For the vehicle that stops (vehicle 1, Fig. 4), the minADE is higher because of the more dynamic behavior of the vehicle but it converges around 2.7 m when the vehicle starts to decelerate. For vehicle 2 the minADE is lower, the vehicle has indeed a more constant behavior, but it grows rapidly, this is due to the higher uncertainty on the position for a vehicle that is in movement.

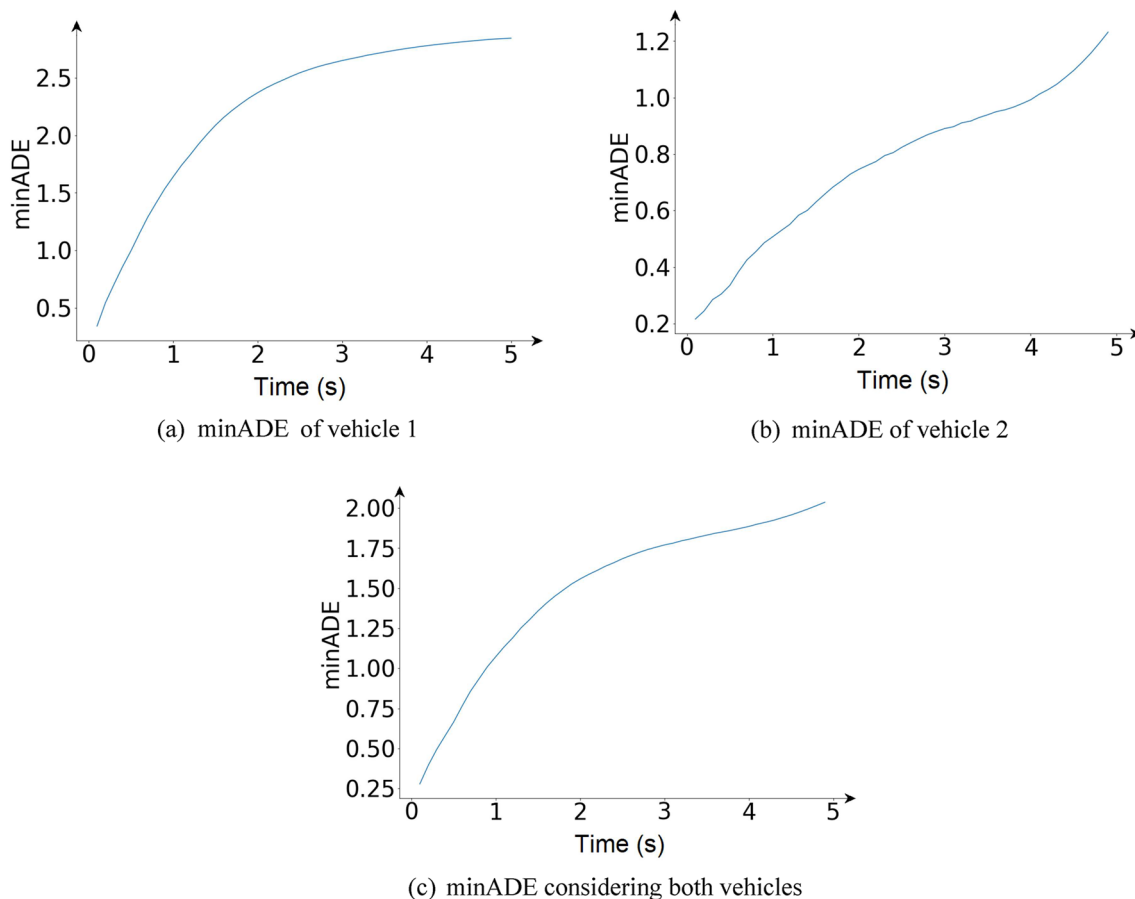
In Fig. 6, the real trajectories are represented in red. In blue are shown the predicted trajectories and the probability is given by the color intensity. A slight shift is applied to each trajectory to allow a better readability. The observation of the graph reveals that trajectories with the highest posterior probability (indicated by the dark blue lines) align more accurately with the actual trajectories (depicted in red), affirming the prediction's high quality. The graph provides a

clearer comprehension of the episode, showcasing the potential trajectories considered and the actual ones followed by the vehicles.

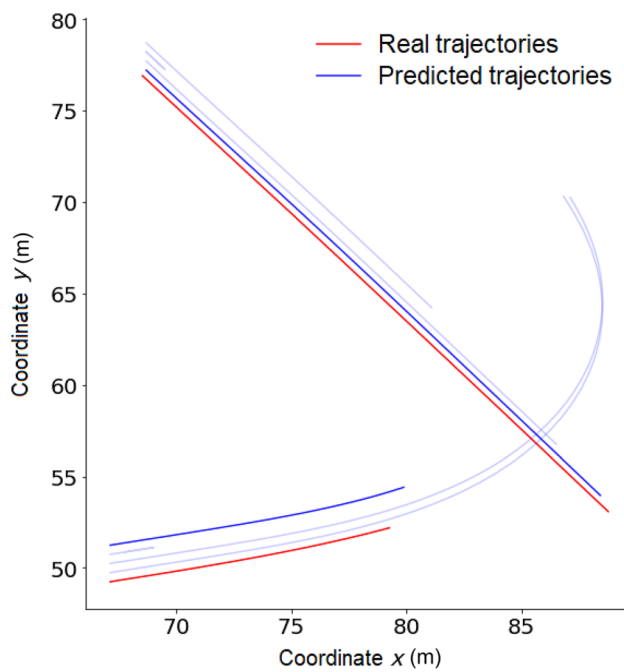
## 6 Results and Discussion

This section presents the results of some simulations in ADORe in terms of minADE, minFDE and MR. Figure 7 shows the scenarios simulated. Here is a brief description of the scenarios:

1. Scenario 1: this scenario represents a case study from the preceding chapter, wherein the car turning left halts at the intersection, affording the car with high priority to proceed.
2. Scenario 2: in this case the car turning left holds precedence over the car proceeding straight, which stops at the intersection.
3. Scenario 3: in this scenario, the merging car pauses at the intersection, granting passage to the car with high priority.



**Fig. 5** The figure shows how does the minADE evolve in time



**Fig. 6** Here the real trajectories (red) and the predicted ones (blue) are shown

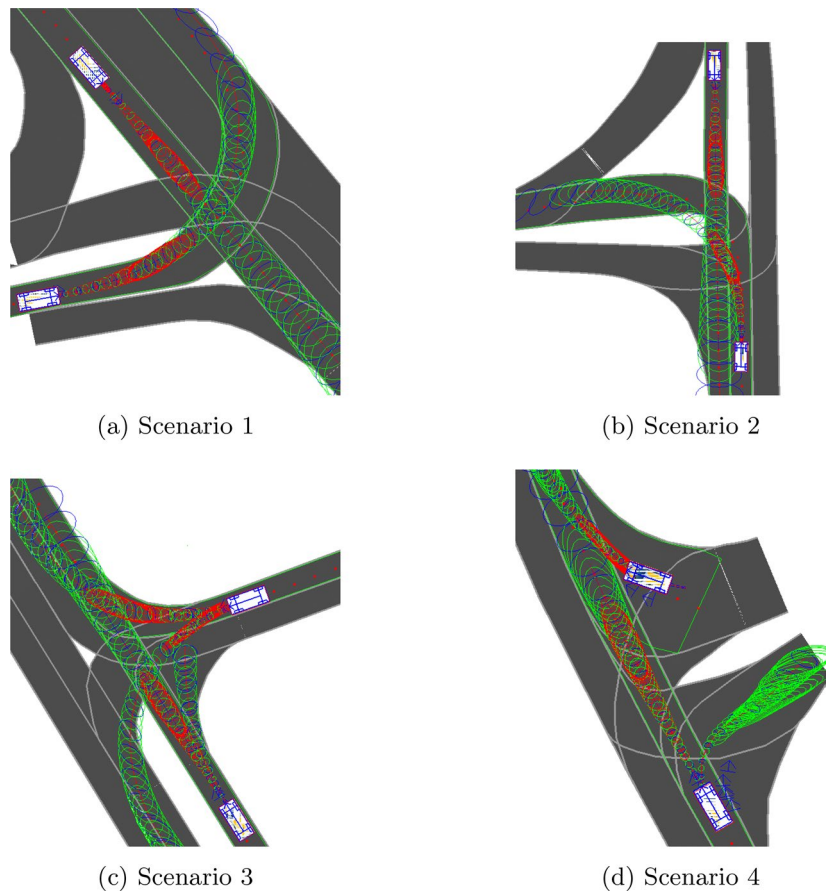
4. Scenario 4: the car with high priority reduces its speed, permitting the other car to merge smoothly.

Each scenario has been repeated different times. The prediction horizon is 10s. For the first 8s of the simulation, predictions are collected. The simulation ends at  $t = 18$ s.

Table 4 shows the results of the simulation scenarios in terms of minADE, minFDE and MR, while Table 5 presents the results of the 1st, 12th, 24th, 50th places in the WAYMO competition in 2021. The two tables cannot be compared, since the first shows the results of predictions in a simulation environment, the second presents results of prediction on a real road dataset. However, the simulator used (ADORE) is not only a simulation environment but also a tool for decision making and control of AVs in traffic, currently used by the Institute of Transportation Systems of the German Aerospace Center. This suggests a high level of realism in the trajectories taken by the vehicles. Figure 8 showcases the time required by the algorithm to compute the Nash equilibrium and to perform the complete prediction. From these data some observations can be drawn:

- The approach shows good performances in simulation in the long term horizon (10 s) and outperforms the

**Fig. 7** Test scenarios



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

**Table 4** minADE, minFDE and MR of the scenarios

	minADE (m)	minFDE (m)	MR	Horizon (s)
Scenario 1	4.91	11.59	0.44	10.0
Scenario 2	4.80	11.48	0.49	10.0
Scenario 3	4.27	5.62	0.33	10.0
Scenario 4	4.86	9.95	0.36	10.0

**Table 5** Results of the 1st, 12th, 24th, and 50th places in the WAYMO competition in 2021

	minADE (m)	minFDE (m)	MR	Horizon (s)
1. place	1.12	2.31	0.15	8.0
12. place	1.40	3.17	0.27	8.0
24. place	1.56	3.50	0.34	8.0
50. place	16.02	35.49	0.98	8.0

acceptable standards of the WAYMO interactive data set results. Nevertheless, a fair comparison can only be established once the model is tested using real-road data sets.

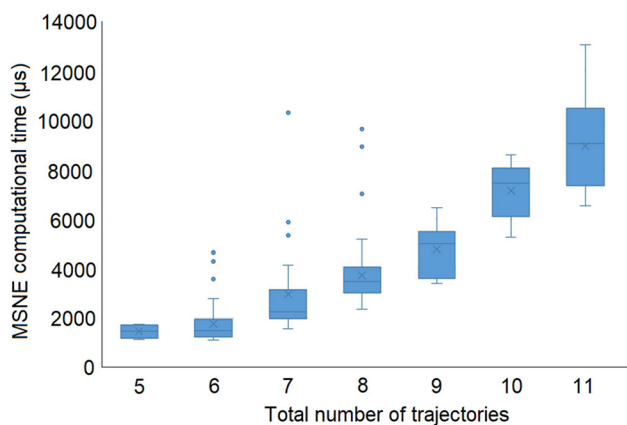
- The computation of the equilibrium always takes less than 12 ms and the whole prediction is performed in less than 120 ms, considering 11 trajectories in the scenario. This allows a online application of the predictor. Other algorithms, based on the computation of Nash equilibria in multi-agent dynamic games, often require computational time on the order of seconds. The simulations have been carried out with a processor Intel(R) Core(TM) i7-10850H CPU @ 2.70GHz 2.71 GHz.
- The weak point of the algorithm is the trajectories computation, as the Particle Swarm Optimization approach

is not computationally efficient. This is reflected in the significant difference between the time required from the Nash equilibrium computation and the total time required. The difference is basically required by the definition of the trajectories. Nonetheless, the total time required still allows a online application of the algorithm.

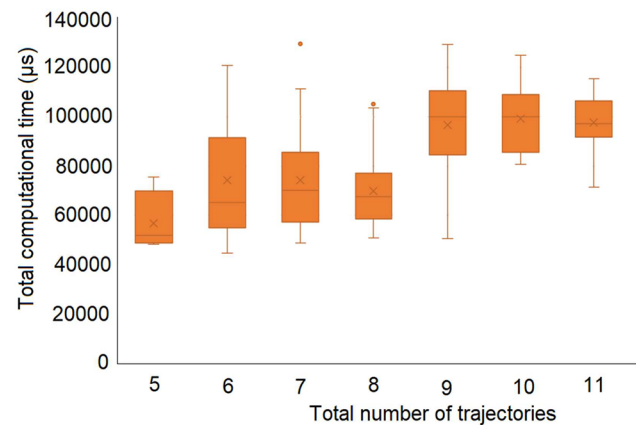
### 7 Conclusions

In this paper, an innovative approach for predicting trajectories in traffic is proposed. The approach combines an interaction-aware motion model with a physics-based and maneuver-based model. The Bayes' theorem is applied: the prior estimate takes into account the rational evolution of the traffic scenario by computing the Mixed strategy nash equilibrium (MSNE) among the participating vehicles. The likelihood adjusts the prior estimate by incorporating data coming from the vehicles in terms of position, heading, speed, acceleration and steering angle. This allows for consideration of the possibility of irrational decisions by the participating vehicles that may have been discarded in the prior estimate. The output of the approach is a probability distribution over a set of representative trajectories for each vehicle.

This innovative framework, which combines a priori game-theoretic considerations with the a posteriori data from the road, constitutes a crucial contribution of this study. Another important contribution is the modeling of the interactive scenario as a polymatrix coordination game with individual preferences, a well known strategic game with desirable computational complexity. The preliminary and indicative experiments to test the approach show good results and good computational efficiency. Future areas of



(a) Time required to compute the Nash equilibrium



(b) Total time required for the prediction

**Fig. 8** Computational time required considering the number of trajectories in the scenario. The simulations have been carried out with a processor Intel(R) Core(TM) i7-10850H CPU @ 2.70GHz 2.71 GHz

development include incorporating a model for online estimation of drivers' behavior and implementing a decision-making algorithm based on this predictor.

**Acknowledgements** The work was supported by the Institute of Transportation Systems of the German Aerospace Center (DLR).

**Funding** Open Access funding enabled and organized by Projekt DEAL.

## Declarations

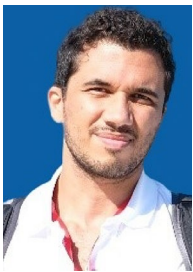
**Conflict of interest** The authors have no competing interests to declare that are relevant to the content of this article.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Cai, Y., Daskalakis, C.: On minmax theorems for multiplayer games. In: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms. **10**, 217–234 (2011). <https://doi.org/10.1137/1.9781611973082.20>
- Rahn, M., Schäfer, G.: Efficient equilibria in polymatrix coordination games. *Math Found. Comput. Sci.* 529–541, (2015)
- Fearnley, J., Goldberg, P.W., Hollender, A., Savani, R.: The complexity of gradient descent:  $CLS = PPAD \cap PLS$ . *J. ACM* **70**(1), 1–74 (2022). <https://doi.org/10.1145/3568163>
- Lefevre, S., Vasquez, D., Laugier, C.: A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH J.* **1**, 1 (2014). <https://doi.org/10.1186/s40648-014-0001-z>
- Brännström, M., Coelingh, E., Sjöberg, J.: Model-based threat assessment for avoiding arbitrary vehicle collisions. *IEEE Trans. Intell. Transp. Syst.* **11**, 658–669 (2010)
- Kaempchen, N., Schiele, B., Dietmayer, K.C.J.: Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios. *IEEE Trans. Intell. Transp. Syst.* **10**, 678–687 (2009)
- Althoff, M., Stursberg, O., Buss, M.: Model-based probabilistic collision detection in autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **10**, 299–310 (2009)
- Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: MultiPath: multiple probabilistic anchor trajectory hypotheses for behavior prediction. In: Proceedings of the Conference on Robot Learning, PMLR. **100**, 86–99 (2020)
- Xie, G., Gao, H., Qian, L., Huang, B., Li, K., Wang, J.: Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models. *IEEE Trans. Ind. Electron.* **65**(7), 5999–6008 (2018). <https://doi.org/10.1109/TIE.2017.2782236>
- Sun, Q., Huang, X., Gu, J., Williams, B.C., Zhao, H.: M2I: from factored marginal trajectory prediction to interactive prediction. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 18–24 June, New Orleans, LA, USA. (2022) <https://doi.org/10.48550/ARXIV.2202.11884>
- Hu, M., Xie, G., Gao, H., Cao, D., Li, K.: Manoeuvre prediction and planning for automated and connected vehicles based on interaction and gaming awareness under uncertainty. *IET Intell. Transp. Syst.* **13**(6), 933–941 (2019). <https://doi.org/10.1049/iet-its.2018.5353>
- Luo, W., Yang, B., Urtasun, R.: Fast and furious: real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18–23 June, Salt Lake City, UT, USA. 3569–3577 (2018)
- Sadeghian, A., Legros, F., Voisin, M., Vesel, R., Alahi, A., Savarese, S.: CAR-net: clairvoyant attentive recurrent network. In: ECCV: European Conference on Computer Vision, 8–14 September, Munich, Germany. 162–180 (2018)
- Barrios, C., Motai, Y.: Improving estimation of vehicle's trajectory using the latest global positioning system with Kalman filtering. *IEEE Trans. Instrum. Meas.* **60**(12), 3747–3755 (2011). <https://doi.org/10.1109/TIM.2011.2147670>
- Schreier, M., Willert, V., Adamy, J.: Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 8–11 October, Qingdao, China. 334–341 (2014) <https://doi.org/10.1109/ITSC.2014.6957713>
- Cui, H., Radosavljevic, V., Chou, F., Lin, T., Nguyen, T., Huang, T., et al.: Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In: 2019 International Conference on Robotics and Automation (ICRA), 20–24 May, Montreal, QC, Canada. 2090–2096 (2018) <https://doi.org/10.1109/ICRA.2019.8793868>
- Hubmann, C., Becker, M., Althoff, D., Lenz, D., Stiller, C.: Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles. In: IEEE Intelligent Vehicles Symposium (IV), 11–14 June. Redondo Beach, CA, USA **2017**, 1671–1678 (2017). <https://doi.org/10.1109/IVS.2017.7995949>
- Gu, J., Sun, C., Zhao, H.: DenseTNT: end-to-end trajectory prediction from dense goal sets. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 10–17 October, Montreal, QC, Canada. 15283–15292 (2021)
- Hong, J., Sapp, B., Philbin, J.: Rules of the road: predicting driving behavior with a convolutional model of semantic interactions. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15–20 June, Long Beach, CA, USA. 8446–8454 (2019)
- Schwartz, W., Pierson, A., Alonso-Mora, J., Karaman, S., Rus, D.: Social behavior for autonomous vehicles. *Proc. Natl. Acad. Sci.* **116**(50), 24972–24978 (2019). <https://doi.org/10.1073/pnas.1820676116>
- Le Cleac'h, S., Schwager, M., Manchester, Z.: LUCIDGames: online unscented inverse dynamic games for adaptive trajectory prediction and planning. *IEEE Robot. Autom. Lett.* **6**(3), 5485–5492 (2021). <https://doi.org/10.1109/LRA.2021.3074880>
- Hang, P., Lv, C., Xing, Y., Huang, C., Hu, Z.: Human-like decision making for autonomous driving: a noncooperative game theoretic approach. *IEEE Trans. Intell. Transp. Syst.* **22**(4), 2076–2087 (2021). <https://doi.org/10.1109/TITS.2020.3036984>
- Basar, T., Olsder, G.: Dynamic noncooperative game theory. 2nd ed. (Classics in applied mathematics 23). United States: Society for Industrial and Applied Mathematics (1999)

24. Raghunathan, A., Cherian, A., Jha, D.: Game Theoretic optimization via gradient-based Nikaido-Isoda function. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, 9–15 June, Long Beach, CA, USA. vol. 97 of Proceedings of Machine Learning Research. PMLR. pp. 5291–5300 (2019) Available from: <https://proceedings.mlr.press/v97/raghunathan19a.html>
25. Heß, D., Lapoehn, S., Lobig, T., Nichting, M., Markowski, R., Laueremann, J., et al.: Automated Driving Open Research (ADORE). DLR (German Aerospace Center), Institute of Transportation Systems. <https://github.com/eclipse/adore#readme>
26. Mahalanobis, P.C.: On the generalized distance in statistics. *Natl. Inst. Sci. India* **2**, 49–55 (1936)
27. Ettinger, S.M., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., et al.: Large scale interactive motion forecasting for autonomous driving: the Waymo Open Motion Dataset. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 10–17 October, Montreal, QC, Canada. 9690–9699 (2021)



**Giovanni Lucente** completed his Master's degree in Mechanical Engineering from Politecnico di Milano in 2020. With previous working experience in automotive companies, he is currently a Ph.D. student at the Institute of Transportation Systems of the German Aerospace Center (DLR) in Braunschweig. His research interests revolve around decision-making processes and cooperation between automated and connected vehicles in urban traffic environments.



**Reza Dariani** received his bachelor's degree in Power Electrical Engineering in 2008 from the University of Saveh-Iran. In 2010, he obtained a Master's degree in Electronic-Electrical and Automatic from the University of Reims-France, followed by another Master's degree in Mechatronics from the University of Strasbourg- France in 2011. He completed his Ph.D. in Trajectory Planning for Autonomous Vehicles at Otto-von-Guericke Universität of Magdeburg-Germany. Since 2016, he has been working as a

researcher at the Institute of Transportation Systems of the German Aerospace Center in Braunschweig-Germany.



**Julian Schindler** has been working at the Institute of Transportation Systems at the German Aerospace Center (DLR-ITS) in Braunschweig, Germany since 2006. As a computer scientist, he initially focused on the software architecture of various driving simulators. His work then shifted towards the ergonomic design of vehicle automation functions and the cooperation between drivers and vehicles. He has participated in several national and international projects, including Citymobil (FP6), interactIVe,

HAVEit, and ISiPADAS (FP7). With a growing emphasis on cooperation with other entities such as infrastructure or VRUs, Julian has coordinated projects such as H2020-MAVEN and H2020-TransAID. Since 2017, he has been leading the “System-Automation & Integration” group at DLR-ITS.



**Prof. Dr. Michael Ortgiese** has been a Professor of Traffic and Mobility Management at the Einstein Center Digital Future (ECDF) since 2019. This joint professorship between Technische Universität Berlin and the German Aerospace Center (DLR) is affiliated with the head of the Department of Traffic Management and Public Transport at the DLR Institute of Transport Systems Technology.