Proceedings of the ASME 2024
International Design Engineering Technical Conferences
IDETC2024
August 25–28, 2024, Washington, DC

IDETC2024-143361

ADVANCING VEHICLE TRAJECTORY PREDICTION: A PROBABILISTIC APPROACH USING COMBINED SEQUENTIAL MODELS

Lichuan Ren[1] , Zhimin Xi[1]

[1]Rutgers University - Piscataway, NJ, 08854, USA

ABSTRACT

In the domain of autonomous vehicle control, the foundation often lies in vehicle dynamics models, which simplify the complex behavior of vehicles to design effective controllers. However, these simplified models can introduce biases, particularly in extreme driving scenarios. Additionally, environmental factors introduce uncertainty into vehicle trajectories. Even with identical control actions, vehicles can display varying behaviors in different scenarios. This paper proposes methods for predicting the probabilistic distribution of a vehicle's trajectory based on initial conditions and control action sequences. We leverage sequential learning and bias learning techniques to enhance learning efficiency. Our systematic study compares the Seq2Seq model, the particle-based model, and the hybrid model regarding prediction accuracy. The performance of these methods is evaluated and compared using both point error metrics and probabilistic prediction metrics. This research introduces novel approaches to probabilistic modeling of vehicle trajectories by combining physics-based and data-driven models, which can further contribute to the development of more robust and adaptable autonomous vehicle control systems that can navigate complex, uncertain, and extreme driving scenarios.

Keywords: autonomous vehicle, trajectory prediction, uncertainty propagation, bias learning, seq2seq model, sequential Monte Carlo, hybrid model

NOMENCLATURE

System States and Actions

$\mathbf{s}_t$	System state
x	Vehicle position on x-axis [m]
y	Vehicle position on y-axis [m]
θ	Vehicle yaw angle [rad]
v	Vehicle translational velocity [m s^{-1}]
r	Vehicle rotational velocity [rad s^{-1}]
$\mathbf{a}_t$	System action
δ	Steering angle [rad s^{-1}]
a	Throttle

Vehicle Model

β	vehicle slip angle [rad]
l_f	longitudinal distance from c.g. to front tires [m]
l_r	longitudinal distance from c.g. to rear tires [m]

1. INTRODUCTION

Autonomous vehicle navigation involves a complex pipeline of technologies and processes that allow a vehicle to perceive its environment, plan a path, and generate control signals to follow the planned path while avoiding obstacles. Trajectory prediction plays a vital role in autonomous vehicle control, and model predictive control (MPC) is a widely used model-based control technique. Model-based control methods such as MPC depend on a system dynamics model. This model allows them to predict and analyze the future trajectory of the vehicle, which in turn serves as the basis for optimizing control actions. By iteratively forecasting and optimizing the trajectory, these methods enable autonomous vehicles to make informed decisions, adapt to changing conditions, and navigate efficiently and safely.

The existing trajectory prediction methods usually provide deterministic predictions. However, in scenarios with extreme conditions such as sharp turns, low tire-road friction, or the need for rigorous obstacle avoidance, the actual vehicle trajectory can exhibit significant uncertainty [1]. Recognizing the significance of capturing this uncertainty, we have proposed methods in this paper to address the probabilistic modeling of vehicle trajectories. These methods aim to quantify uncertainty and predict trajectories as probability distributions based on the initial state and control actions.

The major contributions of this paper are as follows:

- Framework Proposals: We propose innovative frameworks for uncertainty quantification in ego-vehicle trajectories. These frameworks employ Seq2Seq modeling, the Sequential Monte Carlo method, and a hybrid combined model. Each approach offers unique advantages for probabilistic trajectory prediction.

- We present novel data processing methods for the multivariate time series probabilistic forecasting problem.
- Integration of Physics and Data-Driven Approaches: We extend our previous bias-learning method to probabilistic modeling. The fusion of the physics model and the data-driven model improves data efficiency and prediction accuracy.

2. RELATED WORK

2.1 Motion Prediction

Motion prediction, commonly referred to as trajectory prediction, is a research area that involves forecasting the future positions of ego vehicles or surrounding objects like other vehicles and pedestrians. Typically, this involves predicting vehicle trajectories based on historical data and/or recognizing driving maneuvers [1–4]. It's important to note that our work differs from traditional trajectory prediction in a fundamental way.

Our research focuses on trajectory prediction based on the current state of the vehicle and its planned future control signals. Unlike traditional motion prediction, which primarily aids high-level decision-making processes [1, 5, 6], our work is intended to quantify risk and optimize control actions. While traditional motion prediction focuses on understanding where objects might be in the future, our approach aims to predict how the ego vehicle itself will behave in response to specific control signals, which is crucial for real-time control and risk mitigation in autonomous systems.

There are existing research works that utilize physics-based models or data-driven models to predict the vehicle trajectory, such as [7]. However, our method explores a way to combine the physics-based and data-driven methods, and we emphasize quantifying uncertainties associated with the initial state and future actions.

2.2 Bias-Learning Vehicle Dynamics Model

In most applications of vehicle control and navigation, simplified vehicle dynamic models are used. However, the simplified dynamic models may induce modeling error, especially under some extreme environmental conditions. To improve the model fidelity, bias-learning methods [8] can be applied. The Bias-learning model combines a simplified baseline vehicle dynamic model and a data-driven bias compensation model. With the hybrid of the baseline physics model and the data-driven model, the compensated model has improved accuracy with a handful of training data. Our previous work [8] applied the compensated vehicle dynamic model to improve the performance of the model-based control system, but it only considered the deterministic data. In this paper, we extend our previous work to probabilistic modeling.

3. METHOD

3.1 Problem Formulation

In this study, our goal is to predict the future trajectory given the initial state and a sequence of actions. Our method goes beyond predicting the mean value of the trajectory; it especially focuses on the quantification of the trajectory uncertainty.

The system state $\mathbf{s}_t$ at time step t includes vehicle positions x_t, y_t, yaw angle θ_t, yaw velocity r_t, and linear velocity v_t,

$$\mathbf{s}_t = \begin{bmatrix} x_t & y_t & \theta_t & r_t & v_t \end{bmatrix}^\top \tag{1}$$

The control action $\mathbf{a}_t$ at time step t includes steering angle δ_t and throttle a_t,

$$\mathbf{a}_t = \begin{bmatrix} \delta_t & a_t \end{bmatrix}^\top. \tag{2}$$

The system model reflects how the vehicle state changes over time given noise and control inputs,

$$\mathbf{s}_t = f_t(\mathbf{s}_{t-1}, \mathbf{a}_{t-1}, \sigma_{t-1}). \tag{3}$$

We assume that the next state is only dependent on the current state $\mathbf{s}_{t-1}$, action $\mathbf{a}_{t-1}$, and noise σ_{t-1}. The noise σ_{t-1} accounts for the unknown environmental uncertainties, e.g., tire-road friction level. While the system model is unknown, we can approximate it using the baseline model, which is a physics-based vehicle kinematics model. We denote the baseline model as a deterministic time-invariant function $h(\cdot)$,

$$\mathbf{b}_t = h(\mathbf{s}_{t-1}, \mathbf{a}_{t-1}). \tag{4}$$

Since the baseline model is a simplified representation, it introduces notable modeling bias when the underlying assumptions are not met. Moreover, the bias will accumulate over the prediction horizon. Hence, we need to compensate for the modeling bias. Additionally, due to the presence of environmental uncertainty, the future trajectory exhibits growing uncertainty over the prediction horizon. To quantitatively assess and compensate for this mean bias and to accommodate the increasing uncertainty, our state of interest is a probabilistic distribution of the future state sequence,

$$\mathbf{S}_{1:T} = \{\mathbf{S}_t\}_{t=1}^T. \tag{5}$$

In summary, the problem is to predict the posterior state, which is the estimate of the state sequence given the initial state and all control actions on the prediction horizon,

$$p(\mathbf{S}_{1:T} | \mathbf{s}_0, \mathbf{a}_{0:T-1}). \tag{6}$$

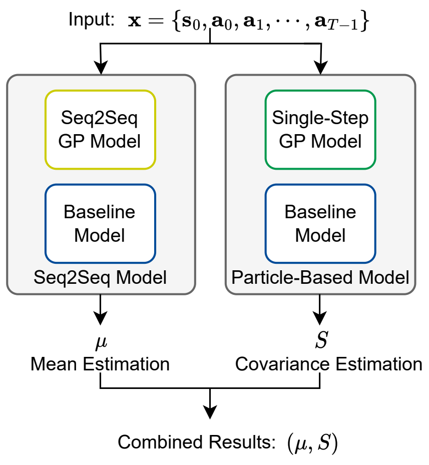
3.2 System Design

We assume the random variable $\mathbf{S}_{1:T}$ can be decomposed into two parts: the baseline value $\mathbf{b}_{1:T}$, which comes from the deterministic function $h(\cdot)$, and the modeling bias $\mathbf{y}_{1:T}$, which is a random variable that we aim to estimate. The state of interest, $\mathbf{S}_{1:T}$, can be represented by the PDF of $\mathbf{y}_{1:T}$ with the shift on the input axis $\mathbf{b}_{1:T}$.

The input of the system is a vector that encompasses the initial state and a sequence of control actions,

$$\mathbf{x} = \{\mathbf{s}_0, \mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{T-1}\}. \tag{7}$$

The baseline model is applied in both the Seq2Seq framework and the particle-based framework. To estimate the probabilistic distribution of the vehicle trajectory, we adopt Gaussian Process (GP) within both of these frameworks. In the Seq2Seq approach, the GP model predicts sequential data representing vehicle state residuals, and then the GP output is integrated with the



Input: $\mathbf{x} = \{\mathbf{s}_0, \mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{T-1}\}$
Seq2Seq GP Model
Baseline Model
Seq2Seq Model
Single-Step GP Model
Baseline Model
Particle-Based Model
μ Mean Estimation
S Covariance Estimation
Combined Results: (μ, S)

FIGURE 1: System diagram

baseline reference trajectory, resulting in the final results. In the particle-based approach, we construct the GP model as a single-step prediction model and then propagate the particles using the hybrid of this single-step GP model and the baseline model. The prediction is further refined by combining the strengths of two approaches: the Seq2Seq model is utilized as a mean estimator, and the particle-based model serves as a covariance estimator.

3.3 Baseline Physics Model

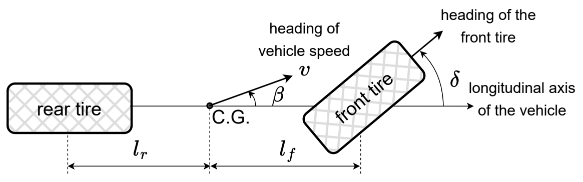


FIGURE 2: Baseline vehicle model

A physics-based vehicle model is employed as the baseline model to fasten the training process. In this study, we apply a kinematic model based on a bicycle model of the vehicle. The baseline model describes the motion of the vehicle in terms of three coordinates x, y, θ. (x, y) is the global position of the c.g. of the vehicle, and θ describes the orientation of the vehicle. The vehicle motion can be described by,

$$\dot{x} = v \cdot \cos(\theta + \beta), \tag{8}$$

$$\dot{y} = v \cdot \sin(\theta + \beta), \tag{9}$$

$$\dot{\theta} = \frac{v \cdot \cos\beta}{l_r + l_f} \tan\delta, \tag{10}$$

where l_f, l_r are longitudinal distances from c.g. to front and rear tires, respectively. The slip angle β can be obtained by,

$$\beta = \arctan\frac{l_r \tan\delta}{l_f + l_r}. \tag{11}$$

The velocity v is an external variable or can be obtained from a longitudinal vehicle model. As the baseline model exclusively incorporates the vehicle's lateral kinematics, we assume the velocity to be constant in the longitudinal direction for the sake of simplicity. The details of the baseline model can be found in [9].

3.4 Seq2Seq Model

In this section, we present a framework for probabilistic vehicle trajectory prediction using Seq2Seq models based on GP regression. Seq2Seq models are designed to handle sequence learning. They have shown strong capabilities in capturing patterns, trends, and dependencies within the data over time. In the context of vehicle trajectory prediction, the input and output data are inherently sequential. Thus, we can apply Seq2Seq models to the trajectory prediction problem. Moreover, we apply GP regression to predict the probabilistic distribution of the states and quantify the uncertainty of the future vehicle trajectory. Regarding model selection, GP is employed not only for its strong capability of probabilistic prediction but also for its significant strengths in capturing temporal dependencies. Despite these strengths, one limitation of GP is the computational complexity. To solve the scalability issue, variational GP is applied in this work.

We develop a GP regression model for sequential prediction. Training a GP model involves several key steps that collectively allow the model to learn patterns and make predictions from data. First, we gather a dataset containing input-output pairs. Since the model takes sequential input and predicts sequential output, in the data preprocessing phase, we transform the collected data into sequences and flatten the sequences of multivariate variables into vectors. The input vector includes the initial state and an action sequence,

$$\mathbf{x}_{seq} = \{\mathbf{s}_0, \mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\}. \tag{12}$$

Instead of learning the sequential vehicle dynamics completely from the data, we utilize the bias-learning (BL) method [8], where a physics-based model and a data-driven model are combined. Using the baseline vehicle dynamic model in Sec. 3.3, we can obtain a baseline reference trajectory,

$$\mathbf{y}_{base,seq} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_T\}. \tag{13}$$

To do this, we apply the actions one by one, updating the system's state prediction at each step based on the physics-based model. This results in a sequence of predicted states. The first baseline prediction is obtained from the known initial state and action,

$$\mathbf{b}_1 = h(\mathbf{s}_0, \mathbf{a}_0), \tag{14}$$

and others are obtained from the previous predicted state and the corresponding action,

$$\mathbf{b}_t = h(\mathbf{b}_{t-1}, \mathbf{a}_{t-1}), \quad t = 2, 3, \ldots, T. \tag{15}$$

Note that the baseline physics model provides a single-value prediction rather than a probabilistic prediction.

The GP output is a sequence of residual states, wherein a residual state represents the difference between baseline predicted values $\mathbf{b}_t$ and observed values of $\mathbf{s}_t$.

$$\mathbf{y}_{seq} = \{\mathbf{s}_1 - \mathbf{b}_1, \mathbf{s}_2 - \mathbf{b}_2, \ldots, \mathbf{s}_T - \mathbf{b}_2\}. \tag{16}$$

Figure 3 shows the designed input and output sequences for GP. The radial basis function (RBF) kernel is chosen as the kernel function. Once the hyperparameters of the kernel function are optimized, we can use the GP model in combination with the baseline physics model to predict the future trajectory.

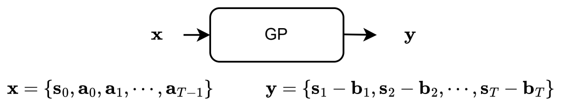

x
GP
y
$\mathbf{x} = \{\mathbf{s}_0, \mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{T-1}\}$
$\mathbf{y} = \{\mathbf{s}_1 - \mathbf{b}_1, \mathbf{s}_2 - \mathbf{b}_2, \cdots, \mathbf{s}_T - \mathbf{b}_T\}$

FIGURE 3: Seq2Seq trajectory prediction based on GP

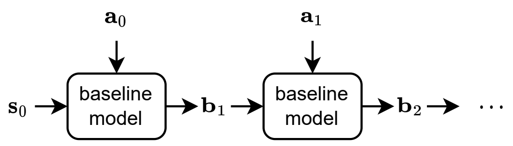


FIGURE 4: Baseline calculation in Seq2Seq model

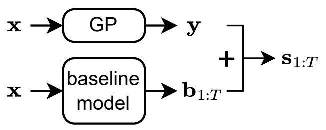


FIGURE 5: System diagram for Seq2Seq model

The final prediction is obtained by combining the baseline reference trajectory with the output from the GP model,

$$\hat{p}(\mathbf{S}_{1:T}|\mathbf{s}_0, \mathbf{a}_{0:T-1}) = \mathbf{y}_{base,seq} + GP(\mathbf{x}_{seq}), \tag{17}$$

where $\mathbf{y}_{base,seq}$ is a deterministic component, which the baseline physics model implies, and $GP(\mathbf{x}_{seq})$ is a stochastic component which not only compensates for the modeling bias of the baseline model but also enriches the prediction by incorporating probabilistic insights.

3.5 Sequential Monte Carlo Model

An alternative way to model the future trajectory is to build a single-step system dynamics model and perform prediction on it multiple times over the prediction horizon. Inspired by sequential Monte Carlo methods, we present a framework for probabilistic trajectory prediction using sequential Monte Carlo simulation. Similar to particle filters, we use particles, i.e., samples, to represent the distribution.

Regarding cause-and-effect relationships between actions and states, the actions beyond time step t are of no consequence to the state $\mathbf{s}_t$. Hence, from a causality perspective, the actions beyond time step t should not participate in predicting $\mathbf{s}_t$. Compared with the seq2seq model, the Monte Carlo model has a more accurate understanding of causal relationships. Moreover, by using the particles, the Monte Carlo model can be used for uncertainty propagation.

Algorithm 1 Seq2Seq Trajectory Prediction

```
1: Require:
       a trained seq2seq Gaussian Process $GP(\cdot)$
       a baseline vehicle dynamic model $h(\cdot)$
2: Inputs:
       initial state $\mathbf{s}_0$
       action sequence $\{\mathbf{a}_t\}_{t=0}^{T-1}$
3: Initialize:
       $x_{seq} \leftarrow \mathbf{s}_0, \mathbf{a}_0, ..., \mathbf{a}_{T-1}$
       $x_0 \leftarrow s_0, a_0$
4: $S \leftarrow GP(x_{seq})$
5: draw $N$ samples $\{s_q\}_{q=1}^N$ from $S$, where $s_q = (s_1, ..., s_T)$
6: for t = 1 to T do
7:     $b_t \leftarrow h(x_{t-1})$
8:     use samples $\{s_{t,q} + b_t\}_{q=1}^N$ to approximate the predicted distribution on time step $t$
9: end for
```

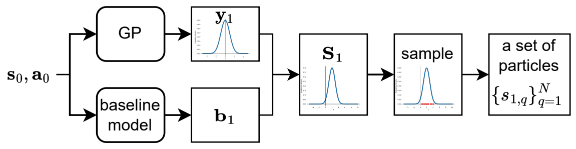


FIGURE 6: Particle-based model initialization

Compared to the seq2seq model, the Monte Carlo method needs multiple GP predictions. Hence, the computation complexity is increased.

Like the seq2seq model, the Monte Carlo method relies on a bias-learning vehicle dynamics model as its foundation. However, it introduces a distinct input-output structure. The input vector comprises a single-step state and single-step action,

$$\mathbf{x}_{MC,t} = \{\mathbf{s}_t, \mathbf{a}_t\}, \quad t = 0, 1, \ldots, T-1. \tag{18}$$

The baseline predicted state is obtained as,

$$\mathbf{b}_{MC,t} = h(\mathbf{s}_{t-1}, \mathbf{a}_{t-1}), \quad t = 1, 2, \ldots, T. \tag{19}$$

It's important to note that the baseline calculation method in this framework differs from the approach detailed in Section 3.4. This is because our aim here is to construct a GP model that specifically captures the single-step vehicle dynamics. Consequently, we avoid modeling bias over a long prediction horizon, focusing instead on the single-step dynamics. The GP output is the residual state, which is the difference between the baseline predicted state and the ground truth state on the next time step,

$$\mathbf{y}_{MC,t} = \{\mathbf{s}_t - \mathbf{b}_{MC,t}\}, \quad t = 1, 2, \ldots, T. \tag{20}$$

The training dataset is a collection of input-output pairs gathered from each single time step across multiple training trajectories.

Algorithm 2 describes how to predict the future trajectory using the trained GP model and the baseline physics model. Initially, the prediction starts by running the GP with the input of the initial state $\mathbf{s}_0$ and action $\mathbf{a}_0$ to obtain a distribution for the state of

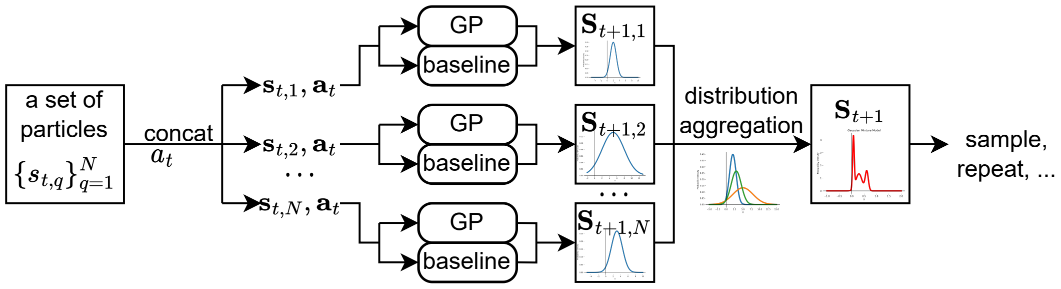


FIGURE 7: Particle-based propagation process

```
Algorithm 2 Sequential Monte Carlo Trajectory Prediction
 1: Require:
        a trained single-step Gaussian Process $GP(\cdot)$
        a baseline vehicle dynamics model $h(\cdot)$
 2: Inputs:
        initial state $\mathbf{s}_0$
        action sequence $\{\mathbf{a}_t\}_{t=0}^{T-1}$
 3: Initialize:
        $x_0 \leftarrow \mathbf{s}_0, \mathbf{a}_0$
        $S_1 \leftarrow GP(x_0)$
 4: for t = 1 to T do
        Sampling Step:
 5:     draw $N$ samples $\{s_{t,q}\}_{q=1}^N$ from $S_t$
 6:     $b_t \leftarrow h(x_t)$
 7:     use samples $\{s_{t,q} + b_t\}_{q=1}^N$ to approximate the predicted
    distribution on time step $t$
        Prediction Step:
 8:     for $s_{t,q} \in \{s_{t,q}\}_{q=1}^N$ do
 9:         $x_{t,q} \leftarrow s_{t,q}, \mathbf{a}_t$
10:         $S_{t+1,q} \leftarrow GP(x_{t,q})$
11:     end for
        Aggregation Step:
12:     combine distributions $\{S_{t+1,q}\}_{q=1}^N$ with equal weights
13:     $S_{t+1} \leftarrow$ mixture distribution
14: end for
```

the first time step, which we denote as $\mathbf{S}_1$. From this distribution, N independent samples are drawn. Then, the uncertainty is propagated over the prediction horizon using Monte Carlo Simulation. In the sequential Monte Carlo Simulation, the algorithm generates a probabilistic trajectory prediction by sampling, predicting, and aggregating distributions at each time step.

During the aggregation step, we combine the outcome estimated distributions generated by each individual particle into a mixture distribution, resulting in a Gaussian Mixture Model (GMM).

This approach leverages the Gaussian Process for modeling uncertainties and the baseline dynamics model to refine the predictions. The algorithm iteratively refines its trajectory prediction as it progresses through time steps, leading to improved accuracy in trajectory uncertainty quantification.

3.6 Combined Model

To leverage the advantages of both approaches, we introduce a combined model that employs the Seq2Seq model as the mean estimator and the Sequential Monte Carlo model as the covariance estimator. This synergistic combination harnesses the Seq2Seq model's ability to generate accurate point estimates, while simultaneously benefiting from the Sequential Monte Carlo model's capacity to provide reasonable estimates of uncertainty and covariance.

4. EXPERIMENTS AND RESULTS

4.1 Simulation Experiment Setup

Our experiment includes three stages, shown in Figure 8. The first two phases are for data collection. With the collected data, we then train and evaluate our proposed models. A video of the experiment process can be found at https://youtu.be/H6WIXxX9nDw.

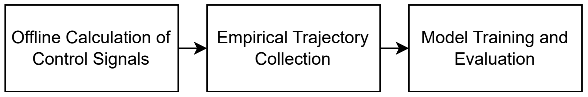

Offline Calculation of Control Signals
Empirical Trajectory Collection
Model Training and Evaluation

FIGURE 8: Experiment process

During the data collection phase, we first calculate a set of control signals offline, which are subsequently divided into a train set and a test set. These control signals are then applied to the vehicle. Under the environmental uncertainties, we observe the vehicle trajectory uncertainty and record the resulting trajectories for model training and evaluation. CARLA simulator [10] is employed as the simulation environment. CARLA uses NVIDIA PhysX vehicle model, which is a high-fidelity vehicle dynamics model.

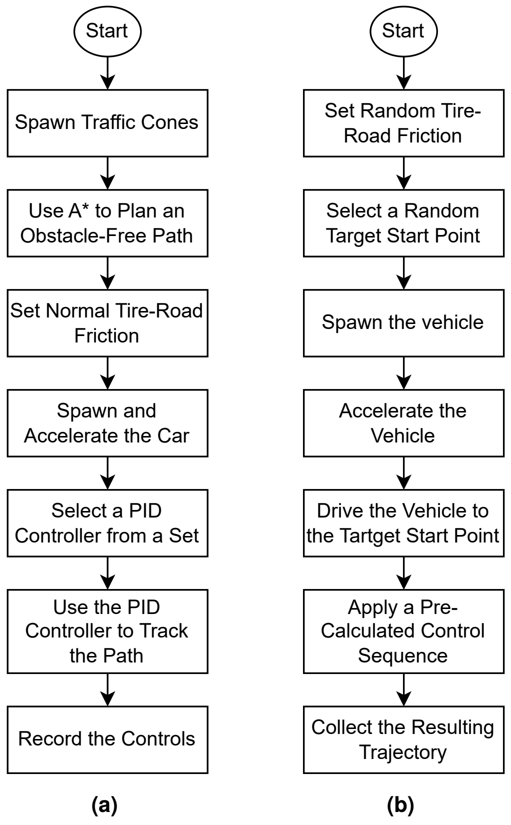

Start
Spawn Traffic Cones
Use A* to Plan an Obstacle-Free Path
Set Normal Tire-Road Friction
Spawn and Accelerate the Car
Select a PID Controller from a Set
Use the PID Controller to Track the Path
Record the Controls
(a)
Start
Set Random Tire-Road Friction
Select a Random Target Start Point
Spawn the vehicle
Accelerate the Vehicle
Drive the Vehicle to the Tartget Start Point
Apply a Pre-Calculated Control Sequence
Collect the Resulting Trajectory
(b)

FIGURE 9: Flowcharts of simulation experiments. (a) is a flowchart that shows how to calculate the pre-determined control signals. In the pre-calculation stage, we set normal tire-road friction in the simulator. We spawn traffic cones to represent obstacles. Then, we use A* planning algorithm to plan an obstacle-free path and use spline interpolation to generate waypoints for the vehicle to follow. We choose a PID controller from a predefined set of controllers and use the PID controller to track the path. (b) shows how to apply the pre-calculated control signals to collect the empirical trajectories under uncertain tire-road friction.

Given our objective of predicting future trajectories and quantifying trajectory uncertainty during extreme maneuvers, we have designed a moose test track for testing obstacle avoidance scenarios, as illustrated in Fig. 10. To simulate these obstacle avoidance scenarios effectively, we have placed traffic cones along the track. These cones serve as obstacles that our system must navigate around. In order to assess trajectory uncertainty during extreme maneuvers, we have set the enter speed to be 65km/h, a notably high-velocity setting.

The uncertainty sources that we consider in our data collection process primarily come from the friction levels between the car's wheels and the ground. We introduce environmental uncertainties by setting a random friction level ranging from 0.2 to 1.0 at each data collection episode. The tire-road friction level is unknown for training and testing. We have pre-defined a collision-free path during the global path planning stage, and the path is represented by waypoints. While the planned path is designed to be collision-free, environmental uncertainties can lead to deviations from this path, potentially resulting in collisions with obstacles. Hence, our objective is to develop a prediction model that can quantify the trajectory uncertainties and estimate the distribution of the actual path taken by the vehicle when a specific control sequence is executed.

As the vehicle starts with zero initial speed at the spawn point, we have designed an acceleration lane to facilitate its motion. The acceleration lane is straight, which results in relatively low uncertainty in the vehicle's lateral motion; its length is sufficient to allow the vehicle ample time to accelerate to the specified entry speed for the test track. In a data collection procedure, the vehicle accelerates from a fixed spawn point, gradually reaching a speed of 65 km/h while traveling along the acceleration lane. It is then driven to a predetermined starting point on the testing track, under the control of a PID controller. The starting point is selected from the set of waypoints along the pre-defined path. Because environmental uncertainties may lead to variations in the vehicle's behavior, we have incorporated a threshold mechanism to detect when the vehicle approaches the designated start point. If the vehicle's start cannot be driven to the designated start point, we terminate that particular data collection episode and start a new one.

Starting from the initial state, we record both the control action sequence and the state sequence of the vehicle. We have chosen a time interval of 0.05 seconds, and each data collection episode consists of 20 steps.

To collect the empirical trajectory samples under environmental uncertainty, we employ the same control sequence, starting from identical initial states for each data collection episode. The control sequence is calculated offline using PID controllers operating under the normal tire-road friction level. To enhance the diversity and robustness of our dataset, we have developed multiple PID controllers. The controllers are split into a training set and a testing set.

4.2 Data Processing

In the data collection phase, we gather data to develop and evaluate our probabilistic trajectory prediction models. We have collected 6020 vehicle trajectories for training and 2162 trajectories for testing.

The Seq2Seq GP model and the single-step GP model require different data formats. Thus, data preprocessing, which transforms the collected data into the required format for each model, is essential.

In the Seq2Seq model, the GP input is the vector containing the initial state and the control action sequence; the GP output is the modeling bias sequence.

Figure 11 visualizes the sequential bias data from an example scene. The orange dotted line represents the observed state sequence collected from the CARLA simulator. The green dotted line indicates the baseline state sequence predicted by the baseline physics-based model. The bias data, denoted by the blue arrows, are the difference between the observed trajectory and the baseline reference trajectory.

In Seq2Seq modeling, each collected trajectory contributes

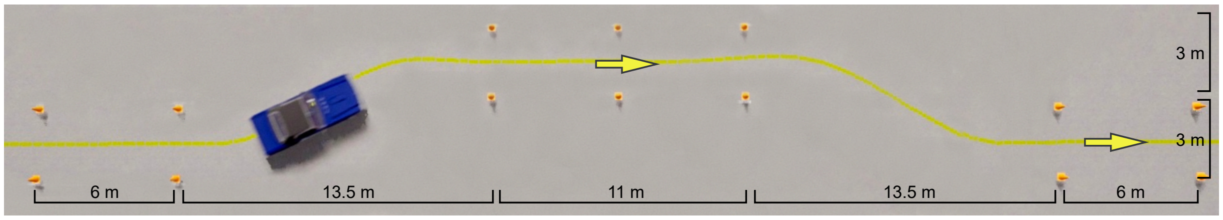


FIGURE 10: Obstacle avoidance track

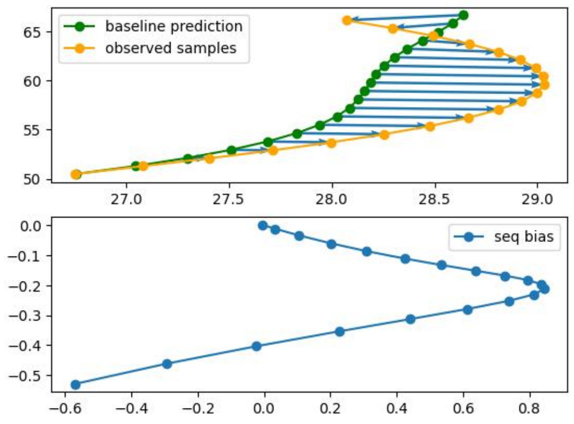


FIGURE 11: Sequential bias data visualization

to a single data instance. On the other hand, in the single-step GP model, each trajectory can be divided into T data instances, where T represents the horizon length. Consequently, data preprocessing in the single-step GP model yields more training samples. Figure 12 visualizes the bias data for the single-step GP model.

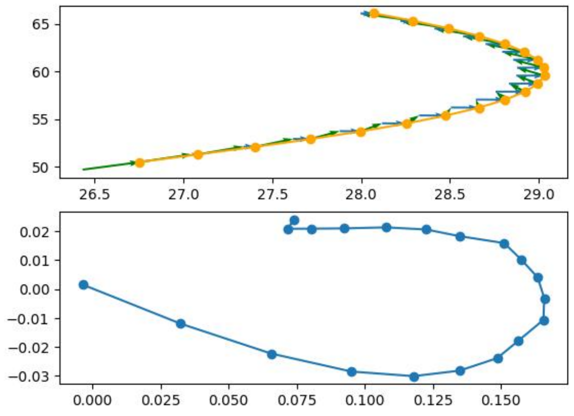


FIGURE 12: Single-step bias data visualization

In the Seq2Seq model, the modeling bias of the baseline model accumulates over the prediction horizon, resulting in a progressively more significant bias as time progresses. On the other hand, in the single-step model, the GP model predicts the modeling bias for a single step ahead, and as a result, the bias remains relatively small. The single-step modeling bias is negligible when the road curvature is small, while the baseline model exhibits noticeable modeling bias in scenarios characterized by large road curvature.

During the model training phase, the initial state, control sequence, and empirical trajectory samples are revealed. In the testing phase, the model takes the initial state and control sequence as input, aiming to estimate the distribution of the state sequence.

4.3 Results

Building upon the data collected in the previous subsection, we have developed various prediction models and conducted comparisons using a set of evaluation metrics. The methods for comparison include:

- Baseline vehicle dynamics model: This physics-based model serves as a fundamental reference, relying on simplified vehicle kinematics to make point predictions.
- Seq2Seq Model: Leveraging sequence-to-sequence learning and residual learning, this model uses GP as a backbone, and utilizes the baseline model as a reference, aiming to capture sequential dependencies in the data for trajectory prediction.
- Sequential Monte Carlo Model: Employing the Sequential Monte Carlo method, this model also uses GP as a backbone, and utilizes the baseline model as a reference, aiming to capture the trajectory uncertainty by propagating particles along the prediction horizon.
- Combined Model: Our hybrid model merges the strengths of the Seq2Seq model as the mean estimator and the Sequential Monte Carlo model as the covariance estimator.

4.3.1 Numerical Evaluation Metrics. To evaluate the performance of each prediction method, we present three evaluation metrics. The average displacement error (ADE) metric computes the average of the L2 norm between the ground truth waypoints and the predicted mean point over the prediction horizon,

$$ADE = \frac{1}{T}\sum_{t=1}^{T} \|\mathbf{s}_t - \hat{\mathbf{s}}_t\|_2. \tag{21}$$

The final displacement error (FDE) computes the displacement error at time step T,

$$FDE = \|\mathbf{s}_T - \hat{\mathbf{s}}_T\|_2. \tag{22}$$

The continuous ranked probability score (CRPS) measures the accuracy of the probabilistic prediction,

$$CRPS(F, \mathbf{s}) = \int_{-\infty}^{\infty} (F(y) - \mathbb{1}(y - \mathbf{s}))^2 dy. \tag{23}$$

where 𝟙 is the Heaviside step function and denotes a step function along the real line that attains: the value of 1 if the real argument is positive or zero, the value of 0 otherwise. ADE and FDE are metrics for point prediction accuracy; CRPS is a metric for probabilistic prediction accuracy.

Empirical results of the evaluation metrics to compare the prediction accuracy of different methods are presented in Table 1. The baseline model assumes that the tire slip angle remains small. This assumption holds well under normal conditions but proves inadequate when the tire-road friction level is low or when the vehicle is engaged in extreme maneuvers, causing the baseline model to exhibit suboptimal performance on the test data. The Seq2Seq model, which utilizes GP to capture correlations between the input (initial state and control sequence) and output (future state sequence), has demonstrated the most favorable performance when evaluated using point prediction metrics. The Sequential Monte Carlo method demonstrates its ability to compensate for the modeling bias of the baseline model, although it falls short of outperforming the Seq2Seq model in terms of point prediction accuracy. However, when we integrate the Sequential Monte Carlo method as a covariance estimator alongside the Seq2Seq model as a mean estimator within the combined model, we observe a notable improvement. The combined model exhibits the best CRPS, which indicates the best prediction performance in a probabilistic sense.

TABLE 1: Trajectory Prediction Empirical Results

Method Name	Metrics		
	ADE	*FDE*	*CRPS*
Baseline	0.6117	1.3627	0.3055
Seq2Seq	**0.1532**	**0.4404**	0.0701
Mean	-	-	0.0752
Monte Carlo	0.4815	1.2030	0.2106
Combined	-	-	**0.0576**

4.3.2 Evaluation of Uncertainty Propagation. It's noteworthy that the Sequential Monte Carlo method excels in capturing uncertainties along the prediction horizon. To demonstrate this, we present the metric of total variation, which measures dispersion in the data. The total variation is the trace of the population variance-covariance matrix,

$$V = \sum_{j=1}^{p} \sigma_j^2, \tag{24}$$

where p is the state dimension.

Starting from the known initial state, the system state uncertainty tends to grow as time progresses. Moreover, when encountering driving scenarios with large road curvature, the vehicle tends to exhibit more significant deviations from the planned path. Fig. 15a is an example of when the total variation of the state increases steadily over the time horizon. The Sequential Monte Carlo method uses particles to propagate the uncertainty, and it captures the growing uncertainty well. Fig. 15b is an example of when the total variation remains relatively stable over time, which is also captured by the Sequential Monte Carlo method.

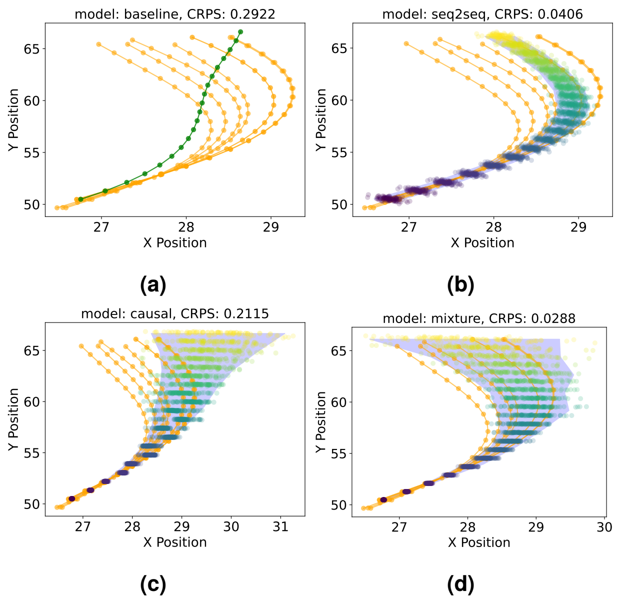


FIGURE 13: predicted trajectory with different models (a) baseline model. (b) seq2seq model. (c) particle-based model. (d) combined model. When the road curvature is relatively large, the trajectory uncertainty becomes significant.

This explains why the combined model outperforms other methods, particularly when considering the CRPS metric. The metrics for point prediction, ADE and FDE, demonstrate that the Seq2Seq model excels in point prediction. Figure 15 shows that the Sequential Monte Carlo method captures uncertainties well. The combined model achieves the best overall performance by employing the Seq2Seq model as the mean estimator and the Sequential Monte Carlo model as the covariance estimator.

5. CONCLUSION

In conclusion, we have introduced and evaluated several methods for constructing probabilistic models for vehicle trajectory prediction. The Seq2Seq model, leveraging sequential learning techniques, excels in point prediction accuracy. Meanwhile, the Sequential Monte Carlo (SMC) method effectively propagates uncertainty step-by-step, making it proficient in quantifying uncertainty along the prediction horizon. Both models incorporate a physics-based model as a baseline for reference, which notably improves data efficiency.

Our simulation results demonstrate the strengths of each method: the Seq2Seq model's precision in point predictions and the SMC model's capability in uncertainty quantification. However, the most notable finding is that the combined model, integrating the Seq2Seq model as the mean estimator and the SMC model as the covariance estimator, surpasses all other methods in overall performance.

By accounting for uncertainties introduced by the environment, we aim to provide a probabilistic understanding of vehicle trajectories, thus enhancing safety and enabling effective obstacle avoidance in real-world scenarios where environmental

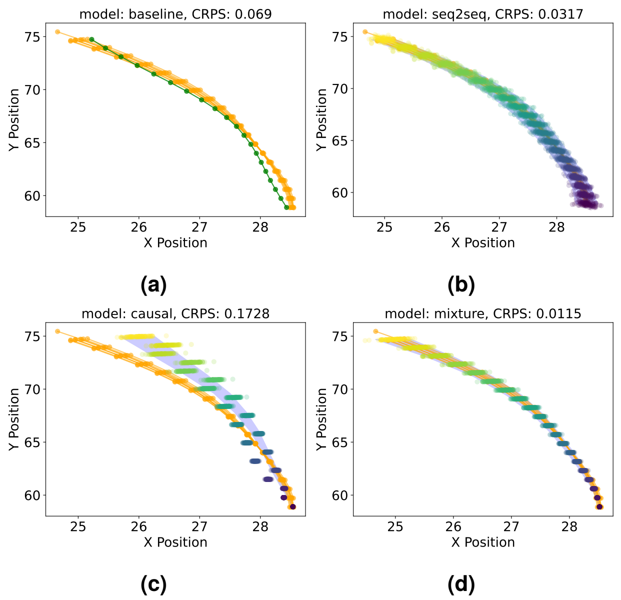


FIGURE 14: When the road curvature is relatively small, the trajectory uncertainty under varying tire-road friction levels tends to be small. Our model is able to capture this.

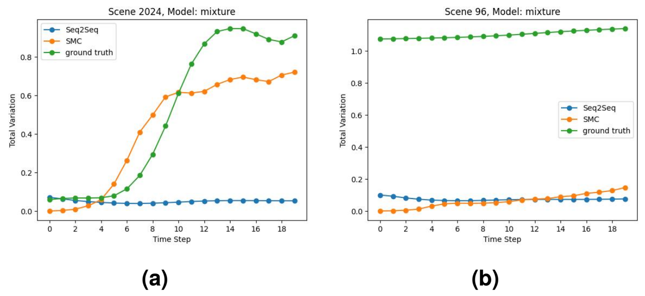


FIGURE 15: Total variation (a) Case I. (b) Case II.

uncertainties significantly impact vehicle trajectories.

There remain several limitations in our current study, which may direct our future research:

- The computational efficiency of the particle-based framework is relatively high, indicating a need for optimization.
- During the data collection phase, the simulator introduces initial variations in state variables, potentially leading to divergence in future trajectories. This issue requires further investigation and refinement.

REFERENCES

[1] Hu, Hongyu, Wang, Qi, Du, Laigang, Lu, Ziyang and Gao, Zhenhai. "Vehicle trajectory prediction considering aleatoric uncertainty." *Knowledge-Based Systems* Vol. 255 (2022): p. 109617. DOI https://doi.org/10.1016/j.knosys.2022.109617. URL https://www.sciencedirect.com/science/article/pii/S0950705122008140.

[2] Guo, Lulu, Shan, Ce, Shi, Tengfei, Li, Xuan and Wang, Fei-Yue. "A Vectorized Representation Model for Trajectory Prediction of Intelligent Vehicles in Challenging Scenarios." *IEEE Transactions on Intelligent Vehicles* (2023): pp. 1–6DOI 10.1109/TIV.2023.3317032.

[3] Chen, Hao, Liu, Yinhua, Zhao, Baixuan, Hu, Chuan and Zhang, Xi. "Vision-Based Real-Time Online Vulnerable Traffic Participants Trajectory Prediction for Autonomous Vehicle." *IEEE Transactions on Intelligent Vehicles* Vol. 8 No. 3 (2023): pp. 2110–2122. DOI 10.1109/TIV.2022.3227940.

[4] Liu, Kaiwen, Li, Nan, Tseng, H. Eric, Kolmanovsky, Ilya and Girard, Anouck. "Interaction-Aware Trajectory Prediction and Planning for Autonomous Vehicles in Forced Merge Scenarios." *IEEE Transactions on Intelligent Transportation Systems* Vol. 24 No. 1 (2023): pp. 474–488. DOI 10.1109/TITS.2022.3216792.

[5] Huang, Yanjun, Du, Jiatong, Yang, Ziru, Zhou, Zewei, Zhang, Lin and Chen, Hong. "A Survey on Trajectory-Prediction Methods for Autonomous Driving." *IEEE Transactions on Intelligent Vehicles* Vol. 7 No. 3 (2022): pp. 652–674. DOI 10.1109/TIV.2022.3167103.

[6] Xie, Guotao, Gao, Hongbo, Qian, Lijun, Huang, Bin, Li, Keqiang and Wang, Jianqiang. "Vehicle Trajectory Prediction by Integrating Physics- and Maneuver-Based Approaches Using Interactive Multiple Models." *IEEE Transactions on Industrial Electronics* Vol. 65 No. 7 (2018): pp. 5999–6008. DOI 10.1109/TIE.2017.2782236.

[7] Gao, Hongbo, Qin, Yechen, Hu, Chuan, Liu, Yuchao and Li, Keqiang. "An Interacting Multiple Model for Trajectory Prediction of Intelligent Vehicles in Typical Road Traffic Scenario." *IEEE Transactions on Neural Networks and Learning Systems* Vol. 34 No. 9 (2023): pp. 6468–6479. DOI 10.1109/TNNLS.2021.3136866.

[8] Ren, Lichuan and Xi, Zhimin. "Bias-Learning-Based Model Predictive Controller Design for Reliable Path Tracking of Autonomous Vehicles Under Model and Environmental Uncertainty." *Journal of Mechanical Design* Vol. 144 No. 9 (2022): p. 091706.

[9] Rajamani, R. *Vehicle Dynamics and Control*. Springer (2006). DOI 10.1007/0-387-28823-6.

[10] Dosovitskiy, Alexey, Ros, German, Codevilla, Felipe, Lopez, Antonio and Koltun, Vladlen. "CARLA: An Open Urban Driving Simulator." *Proceedings of the 1st Annual Conference on Robot Learning*: pp. 1–16. 2017.