

## Trajectory-Prediction with Vision: A Survey

Apoorv Singh

Motional

Carnegie Mellon University

apoorv.singh@motional.com

### Abstract

To plan a safe and efficient route, an autonomous vehicle should anticipate the future trajectories of other agents around it. Trajectory prediction is an extremely challenging task recently gaining much attention in the autonomous vehicle research community. Trajectory prediction forecasts the future state of all the dynamic agents in the scene, given their current and past states. A good prediction model can prevent collisions on the road, hence the ultimate goal for autonomous vehicles: Collision rate: collisions per Million miles. This paper aims to provide an overview of the field trajectory-prediction. We categorize the relevant algorithms into different classes so that researchers can follow through the trends in the trajectory-prediction research field. Moreover, we also touch upon the background knowledge required to formulate a trajectory-prediction problem.

### 1. Introduction

Modern approaches to self-driving divide the problem into four steps: detection, object-tracking, trajectory-prediction, and path-planning; used in sequence as shown in Fig.1. In this paper we will go through the trajectory prediction problem, which is responsible for forecasting the trajectories and intentions of all the dynamic agents on the scene. This module finds its usage in any dynamic robotic platform, the self-driving industry is one of the biggest ones. To fully solve the trajectory-prediction problem, social intelligence is very important, since we need to bound possibilities, to limit our infinite search space, given our social intelligence. For example, An unattended child at an intersection with a pedestrian's red signal would have more probability of coming on the road compared to a full-grown adult in the same situation.

*Why is prediction important?* Localization and classification (a.k.a perception) of the objects on the road is one thing, but we also need to understand the dynamics of the agents and their surroundings to predict their future

behavior and prevent any crashes. Given the criticality of the problem, the self-driving industry has a dedicated module for this on top of the perception and motion-planning module, nowadays.

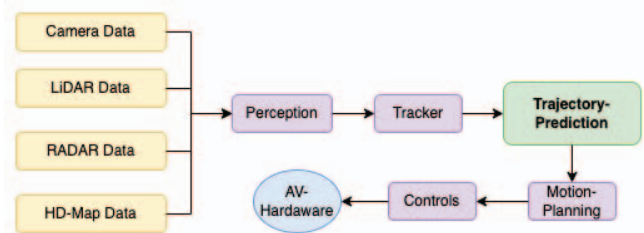


Figure 1. High-level architecture of an Autonomous vehicle stack. In this survey, we focus on the green color-coded block. Traditionally, *prediction module* takes the input of tracks from the *tracker module*; and outputs tracks with future trajectories to the *motion-planning module*.

Here are some key components and approaches used in trajectory prediction for autonomous cars:

- **Sensor Inputs:** Autonomous cars use a variety of sensors, such as LiDAR, cameras, radar, and GPS, to gather data about the surrounding environment. This sensor data forms the input for trajectory prediction algorithms.
- **Data Fusion:** The data from different sensors is combined using sensor fusion techniques to create a comprehensive and accurate representation of the environment.
- **Motion Models:** Trajectory prediction often employs motion models to estimate the future paths of objects based on their current state and historical behavior. Different types of models, such as kinematic, dynamic, and probabilistic models, can be used to account for different levels of vehicle or pedestrian movement complexity.

- **Behavior Prediction:** Understanding the intentions and behaviors of other road users is crucial. Behavioral models can be developed to predict how pedestrians and vehicles are likely to move based on past observations and common driving or walking patterns.
- **Environmental Context:** Consideration of environmental factors, such as road layout, traffic rules, traffic signs, and weather conditions, is essential for accurate trajectory prediction.
- **Uncertainty Estimation:** Trajectory predictions should account for uncertainties in the sensor measurements and the model's predictions. Probabilistic approaches like Monte Carlo methods or Kalman filters can help quantify and manage these uncertainties.
- **Interactive Prediction:** Trajectory prediction should consider interactions between different road users. For example, a pedestrian's trajectory could be influenced by the behavior of nearby vehicles.
- **Real-time Processing:** Autonomous cars need to perform trajectory prediction in real-time to make timely decisions. Therefore, algorithms should be computationally efficient.
- **Validation and Testing:** Trajectory prediction algorithms need rigorous testing and validation under various real-world scenarios to ensure their reliability and safety.
- **Human Behavior:** Predicting the behavior of human drivers and pedestrians introduces challenges due to their inherent variability and occasional irrational actions. Models need to account for this uncertainty.

There are several published survey papers on behavior analysis. For example, [22] looked at the problem based on the perspective-view approaches, but this coordinate space is rarely used in the autonomous vehicles industry. All the modeling is done in the Bird's Eye View (BEV) space instead. Historically camera-based features were computed in perspective view, but modern techniques [20, 19, 7] have enabled perception features to be generated in the BEV itself. Hence, it makes more intuitive sense to operate trajectory-prediction module in the BEV space, as is required by its customers viz., motion-planning module shown in Fig. 1. [13] is a little outdated in terms of State-of-the-art (SoTA) methods; with the major focus on HMMs, SVMs, Bayesian-based methods et al. A lot of top-performing prediction methods are based on deep-learning-based approaches, which also reflects the current perception models [15, 18, 25]. However, to the best of the author's knowledge, this is the first work that covers modern DL-based forecasting approaches targeted for AV space using

all the sensory data that are available on board to model accurate prediction models for dynamic agents on the road.

The contributions of this work are summarized as follows:

- Paper goes through the basics of the trajectory-prediction problem statement along with all the terminologies around it in section 2.
- Paper goes through top-trending techniques in trajectory-prediction, primarily focusing on SoTA (State of The Art) methods viz., Transformers and Goal-based models in section 3.
- Paper provokes researchers with possible future directions and highlights current research gaps in section 4.

## 2. Basics and Challenges of Trajectory Prediction

In this section, we go through the challenges in the trajectory-prediction problem statement and some basic background information required to dive deeper into top-trending techniques for modern trajectory-prediction methods.

### 2.1. Challenges

Prediction in the self-driving domain is a complex problem due to the following characteristics:

- **Interdependence:** There are inter-dependencies in the behavior of the agents i.e. future behavior of one agent may affect the future behavior of other agents in the vicinity. Hence, we need to take into account the entire surrounding scene of the road including traffic rules for making the agent's trajectory prediction. This makes prediction modeling a joint optimization problem for all the agents.
- **Real-time Requirement:** We need to design a bulky deep-learning module that does joint optimization of the agent's trajectories. However, autonomous vehicles need to operate in real-time ( $\sim 10\text{hz}$ ) giving the prediction module a very tight run-time budget.
- **Accumulating Errors:** Prediction module comes after the perception and tracking module in the self-driving software stack. This means that there would be some errors already accumulated by the other modules. Hence prediction module's performance is dependent on how well previous models perform.
- **Dynamic Nature:** Both the ego-vehicle and agents are moving in the scene. Future trajectories of the agents depend on the motion of the ego vehicle as well. Hence, ego-vehicle motion compensation needs

to be modeled while dealing with temporal data from the sensors. Modeling everything in BEV makes this problem somewhat simpler.

- **Multi-modal Behavior:** Multi-modal behavior of agents, that is given a past history of an agent, there could be multiple possible future trajectories. For example, a pedestrian who just stepped onto the crosswalk with a pedestrian red-signal may continue walking or may turn around. A comprehensive predictor needs to evaluate all the possible trajectories for each event with their likelihood score.

## 2.2. Prediction task Problem Statement

The prediction task can be divided into two sections as per [22]:

- **Intention:** This is a classification task where we pre-design a set of intention classes for an agent. For example, for a vehicle, it could be: *stopped*; *parked*; or *moving*. We generally treat it as a supervised learning problem, where we need to annotate the possible classification intents of the agent.
- **Trajectory:** This division needs to predict a set of possible future locations for an agent in the next  $T_{pred}$  future frames, referred to as way points. This constitutes their interaction with other agents as well as with the road as shown in Fig. 2.

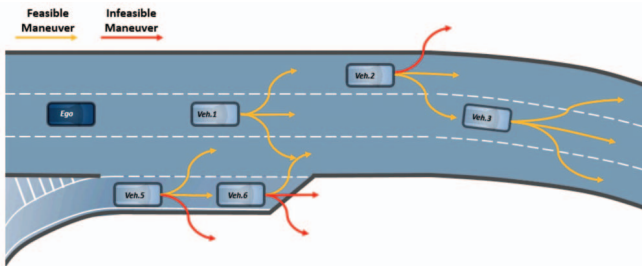


Figure 2. [26] shows how to model trajectory predictions for dynamic agents on the road by making them interaction-aware and road-aware. *Veh.2's* trajectory is dependent on *veh.3's* trajectory and vice versa.

Trajectories and intentions need to be interaction-aware. For instance, it's a fair assumption that on-coming cars might break a little if you aggressively try to enter a highway on a traffic-packed highway road. Generally, trajectory prediction can be modeled in image-view (a.k.a perspective-view) or BEV; but nowadays it is preferred to be done in the BEV space itself. Reason being, in BEV space we can assign a dedicated distance range in the form of a grid for our Region of Interest (RoI). However, Image-view can have theoretically infinite RoI because of the vanishing lines in the perspective-view. It is easier to model occlusions

in BEV space as motion is more linearly modeled. Ego-motion compensation can be easily done with pose-change (translation and rotation of ego-vehicle) in BEV. Moreover, this space preserves the motion and scale of the agents i.e. a vehicle will occupy the same number of BEV pixels irrespective of how far it is from the ego-vehicle; which is not the case with image-view.

To predict what will happen in the future, we need to have a good idea of the past. This can be commonly done by using the output of the tracker, or it can also be done using historically aggregated BEV features. Goal-based prediction [1] has recently been trending in the literature. This approach argues that in order to predict the future of an agent, we need to have an idea of the individual agent's goal.

## 2.3. Datasets

For trajectory-prediction any large-scale perception dataset can be used which includes sequential data viz., nuScenes [3]; Waymo Open Dataset [21]; Lyft [6]; ArgoVerse [4]. However, these datasets don't include annotations for intent. LOKI [5] is a commonly used intention prediction dataset. Trajectory-prediction task can make use of auto-annotations as well if we have sequential unlabeled data; as long we have a good offline perception and tracker model to detect agents and generate temporal link between them.

## 2.4. Input-data formats

Input for prediction can be defined in multiple forms. The simplest way is to send in the sparse tracks from the tracker. A more complex prediction model can feed in BEV representation of the scene; which could be either defined by an occupancy grid or a deep-learning-based latent space.

## 2.5. Evaluation Metrics and Losses

**Intent Prediction** is a classification task hence Binary cross entropy/ Focal loss [11] can be used. For evaluation metrics: *precision*, *recall*, *F1-score*, *mean average precision* can be used.

$$Precision = TP / (TP + FP) \quad (1)$$

$$Recall = TP / (TP + FN) \quad (2)$$

Key: *TP*: True Positive; *FP*: False Positive; *FN*: False Negative.

**Trajectory Prediction:** This is a regression problem where we try to regress future way-points of the agent as close as possible to the ground-truth value. For loss computation, some versions of the L1/L2 norm can be used. For evaluation metrics, there are multiple ways:

- **Final Displacement Error (FDE):** It measures the distance between the predicted final location and the true

final location.

$$FDE = |\hat{y}_{final}^t - y_{final}^t| \quad (3)$$

- *Mean Absolute Error (MAE)*: It measures the average magnitude of the prediction error based on the root mean square value (RMSE) value.
- *Minimum of K Metric*: This is used when  $K$  trajectories are predicted per agent by the model. The metric is calculated based on one of the  $K$  trajectories which minimizes the metric error value.

### 3. SoTA Prediction Methods

Previous surveys have classified behavior prediction models based on *physics-based*, *maneuver-based*, and *interactions-aware models*. *Physics-based* models [16] constitute dynamic equations that model hand-designed motions for different categories of agents. This approach fails to model the hidden state of the entire scene and tends to focus only on a particular agent at a time. However, this trend used to be SOTA in the pre-deep-learning era. *Maneuver-based models* are the niche models based on the intended motion type of the agents. *Interactions-aware models* are generally ML-based systems [27] that can pairwise reason every single agent in the scene and produce interaction-aware predictions for all the dynamic agents. There is a high correlation between the trajectories of different agents present nearby in a scene. This approach has shown that it can generalize better by modeling complex attentions module on the agent’s trajectories that are heuristically hard to model by a human-designed approach. For the rest of the paper, we will focus on this category of work. Inspired from [13] we would like to classify trajectory-prediction models based on the input representations:

- *Tracklets*: A Perception module predicts the current state of all the dynamic agents. This state includes attributes like 3d-center, dimensions, velocity, acceleration, etc. Tracker’s role is to consume this data and associate it temporally so that each track can contain the state history of all the agents. Each track now represents the past movement of that agent. This is the simplest form of prediction model as it contains only sparse tracks as input. A good tracker is able to track an agent even if that agent is occluded in the current frame because of its inherent logic. Traditional trackers are non-ML-based networks, so it can be hard to make the models end-to-end with this approach. A lot of work has been done for interaction-aware modeling with this approach. [2] considers tracks of multiple agents in the close range of the agents with a fixed number of surrounding vehicles for which we are trying to predict the trajectory. They also claim that in-

creasing the receptive field of the tracks helps in improving the behavior prediction performance. [9] has looked upon this problem in distance terms for short-listing all the agents by specifying a distance range around the target agent. [23] adds attention modeling for different agent classes and their weights.

- *BEV Representation*: This branch of networks generates a BEV representation of the current perception output, history of perception outputs, and road state with the information from the HD-map all in a single-stacked BEV representation which is consumed by the prediction model. [8] uses camera and RADAR to model BEV representation as an occupancy grid. Dynamic Occupancy Grid Map (DOGMa) [14] is created from the data fusion of various sensors and provides a BEV image of the environment. The channels of this grid contain the probability of the occupancy and velocity vector for each cell in the grid. As a drawback, this category still suffers from the problem of accumulation of errors, form perception, and tracker module.
- *Raw-sensor Data*: This is an end-to-end approach where the model takes in raw-sensor data information and directly predicts trajectory prediction for each agent in the scene. This approach may or may not have auxiliary outputs and their losses to supervise the complex training as in [10, 12]. [24] extends the previous approach by adding collaborative perception-prediction for multiple AVs on the road with a vehicle-to-vehicle (V2V) communications platform. The drawback of this category is that it is computationally expensive due to dense information for input. Also, since it combines three problems together, i.e., perception, tracking, and prediction, the model becomes very hard to develop and even harder to converge.

**Goal-based Prediction** Along with the scene context, the behavioral intention prediction is commonly influenced by different intended goals and should be inferred with interpretability. For the goal-conditioned future prediction, the *goal* is modeled as the future state (defined as destination coordinates) [1] or moving types that an agent desires [17]. To break down this problem into two categories: The first one is *epistemic*: to answer the question *Where are the agents going?* The second is *aleatoric* to answer the question *how is this agent going to reach its goal?*

### 4. Further Extensions and Conclusion

With perception tasks being solved by more than 90% on their respective metrics and planning tasks being considered to be easily solved as long as accurate predictions are provided; solving trajectory-prediction remains one

of the key challenges to unleash autonomous vehicles at the large scale. Accurate trajectory modeling is the key to preventing possible future collisions; it also needs to run in real-time so that collision avoidance action can be taken within physical limits and reasonable-comfort-level of the users. In this work, we make a thorough review of existing methods of trajectory prediction for self-driving. We divide different approaches into various classes to easily follow through the trends. Moreover, we went through the trajectory-prediction basics so readers can easily follow through the paper. We hope we can highlight key concerns and trends with this survey paper, which can provoke further research in the field.

## References

- [1] Stefano V Albrecht, Cillian Brewitt, John Wilhelm, Balint Gyevnar, Francisco Eiras, Mihai Dobre, and Subramanian Ramamoorthy. Interpretable goal-based prediction and planning for autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1043–1049. IEEE, 2021.
- [2] Florent Althé and Arnaud de La Fortelle. An lstm network for highway trajectory prediction. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pages 353–359. IEEE, 2017.
- [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- [4] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8748–8757, 2019.
- [5] Harshayu Girase, Haiming Gang, Srikanth Malla, Jiachen Li, Akira Kanehara, Karttikeya Mangalam, and Chiho Choi. Loki: Long term and key intentions for trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9803–9812, 2021.
- [6] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Igloukov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, pages 409–418. PMLR, 2021.
- [7] Ankit Laddha, Shivam Gautam, Stefan Palombo, Shreyash Pandey, and Carlos Vallespi-Gonzalez. Mvfusenet: Improving end-to-end object detection and motion forecasting through multi-view fusion of lidar data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2865–2874, 2021.
- [8] Donghan Lee, Youngwook Paul Kwon, Sara McMains, and J. Karl Hedrick. Convolution neural network-based lane change intention prediction of surrounding vehicles for acc. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.
- [9] Xin Li, Xiaowen Ying, and Mooi Choo Chuah. Grip: Graph-based interaction-aware trajectory prediction. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3960–3966, 2019.
- [10] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11553–11562, 2020.
- [11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [12] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018.
- [13] Sajjad Mozaffari, Omar Y Al-Jarrah, Mehrdad Dianati, Paul Jennings, and Alexandros Mouzakitis. Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):33–47, 2020.
- [14] Dominik Nuss, Stephan Reuter, Markus Thom, Ting Yuan, Gunther Krehl, Michael Maile, Axel Gern, and Klaus Dietmayer. A random finite set approach for dynamic occupancy grid maps with real-time application. *The International Journal of Robotics Research*, 37(8):841–866, 2018.
- [15] Jongwoo Park, Apoorv Singh, and Varun Bankiti. 3m3d: Multi-view, multi-path, multi-representation for 3d object detection. *arXiv preprint arXiv:2302.08231*, 2023.
- [16] Carole G. Prevost, Andre Desbiens, and Eric Gagnon. Extended kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle. In *2007 American Control Conference*, pages 1805–1810, 2007.
- [17] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2821–2830, 2019.
- [18] Apoorv Singh. Transformer-based sensor fusion for autonomous driving: A survey. *arXiv preprint arXiv:2302.11481*, 2023.
- [19] Apoorv Singh. Vision-radar fusion for robotics bev detections: A survey. *arXiv preprint arXiv:2302.06643*, 2023.
- [20] Apoorv Singh and Varun Bankiti. Surround-view vision-based 3d detection for autonomous driving: A survey. *arXiv preprint arXiv:2302.06650*, 2023.
- [21] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

- [22] Izzeddin Teeti, Salman Khan, Ajmal Shahbaz, Andrew Bradley, and Fabio Cuzzolin. Vision-based intention and trajectory prediction in autonomous vehicles: A survey. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5630–5637. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Survey Track.
- [23] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pages 4601–4607. IEEE, 2018.
- [24] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 605–621. Springer, 2020.
- [25] Sergi Adipraja Widjaja, Venice Erin Baylon Liong, Zhuang Jie Chong, and Apoorv Singh. Machine learning-based framework for drivable surface annotation, Jan. 19 2023. US Patent App. 17/836,974.
- [26] Yookhyun Yoon, Taeyeon Kim, Ho Lee, and Jahngyun Park. Road-aware trajectory prediction for autonomous driving on highways. *Sensors*, 20(17), 2020.
- [27] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021.