**Academic Journal of Science and Technology**
ISSN: 2771-3032 | Vol. 22, No. 3, 2026

# A Review of Trajectory Prediction for Autonomous Driving from Accuracy to Robustness and Trustworthiness

**Yongli Li[1, 2], Zhixian Zhang[1, 2], Mingyao Gong[1, *], Shuochen Ji[1, 2], Xu Zhao[1, 2] and Lirong Zhou[1]**

[1]School of Aeronautical Engineering, Jilin University of Chemical Technology, Jilin 132000, China
[2]College of Mechanical and Electrical Engineering, Jilin University of Chemical Technology, Jilin 132000, China
[*] Corresponding author

**Abstract:** Trajectory prediction links environmental perception, behavior understanding, and decision-making and planning in autonomous driving. Its value depends not only on geometric error on standard test sets but also on model stability in open traffic environments, probabilistic reliability, and contributions to planning safety. With advances in interaction modeling, high-definition maps, multimodal generation, and Transformers, trajectory prediction has evolved from temporal extrapolation for a single vehicle into a spatiotemporal reasoning task that integrates multi-agent interactions, road semantics, and behavioral uncertainty. However, open-loop metrics such as Average Displacement Error (ADE), Final Displacement Error (FDE), minimum Average Displacement Error (minADE), and minimum Final Displacement Error (minFDE) primarily reflect average prediction accuracy. They reveal little about failures under perception noise, map deviations, out-of-distribution scenarios, long-tail interactions, and adversarial perturbations, and they cannot determine whether model probabilities align with actual errors and planning risks. Taking integrated robustness and trustworthiness as its organizing theme, this review first briefly surveys the trajectory prediction task and representative methods. It then focuses on robustness under input perturbations, scene changes, cross-domain generalization, and adversarial attacks, and discusses uncertainty quantification and propagation, probabilistic calibration, failure detection, and prediction-planning closed-loop evaluation. We further develop a unified framework comprising input-level robustness, scene-level robustness, distribution-level trustworthiness, and planning-level trustworthiness, bringing prediction accuracy, probabilistic quality, calibration reliability, failure detection, and planning safety within a single evaluation framework. Finally, we outline future research directions in unified evaluation, propagation of multi-source uncertainty, calibratable multimodal prediction, long-tail risk, and joint closed-loop evaluation.

**Keywords:** Autonomous driving; trajectory prediction; robustness; trustworthiness; uncertainty quantification; probabilistic calibration; failure detection; closed-loop evaluation.

## 1. Introduction

Trajectory prediction sits between scene perception and motion planning in an autonomous-driving stack. Perception and mapping supply tracked states and road context; the planner then uses predicted motion to choose maneuvers, estimate collision risk, and set safety margins. A prediction error is therefore not an isolated geometric miss. Its confidence and failure mode can change a downstream decision. This is why a low average error on a benchmark, although useful, does not by itself establish deployment readiness.

Three properties make the task especially difficult. A single observed history can support several credible futures, including lane keeping, a lane change, braking, turning, or avoidance. Those futures also depend on road topology, traffic rules, and the responses of other agents, so extrapolating one vehicle in isolation is rarely adequate. In addition, deployed systems receive estimated rather than perfect states: perception, localization, tracking, and maps all introduce error, and the operating distribution changes across cities, road layouts, traffic cultures, and rare interactions. Method development reflects these demands. MultiPath++ [1], CoverNet[2], and TNT[3] represent alternative future hypotheses; VectorNet[4] and LaneGCN [5] couple agents with map structure; AgentFormer[6], HiVT[7], and QCNet[8] use Transformer or query-based designs to resolve multi-agent dependencies over space and time.

These advances have improved standard open-loop results, but the usual metrics answer only part of the safety question. ADE and FDE average geometric error over samples that may be dominated by routine motion. minADE and minFDE score the candidate nearest to the observed future, even if a model assigns substantial probability to a different and unsafe mode. Perception faults, incomplete maps, distribution shift, and adversarial inputs can also produce confident errors. A probability output becomes useful for risk-sensitive planning only when its confidence is consistent with observed error, empirical coverage, and planning cost.

This review uses robustness and trustworthiness as the link between method design and system use. Section 2 examines how predictors encode interactions, map constraints, and multimodal futures. Section 3 asks how those designs behave when inputs or operating conditions change. Section 4 then considers uncertainty, calibration, failure detection, and evaluation with planning in the loop. Together, these sections assess whether an accurate predictor also supplies information that a planner can use safely.

# 2. Classification of Trajectory Prediction Methods for Autonomous Driving

## 2.1. Problem Formulation for Trajectory Prediction in Autonomous Driving

The basic task of trajectory prediction for autonomous driving is as follows: given the state sequences of a target traffic participant and the surrounding participants over a historical time window, predict their position sequences or trajectory distributions over a future time window. Suppose a traffic scene contains $N$ traffic participants, and the state of the $i$ th participant at historical time $t$ can be represented as

$$s_i^t = [x_i^t, y_i^t, v_i^t, a_i^t, \cdots] \tag{1}$$

where $(x_i^t, y_i^t)$ denotes spatial position, while $v_i^t$ and $a_i^t$ denote velocity and acceleration, respectively. A trajectory prediction model typically takes the historical state sequence over the preceding $T_h$ time steps as input:

$$\mathrm{X}_1^1 = \{s_i^t | i = 1, \cdots N; t = 1, \cdots, T_h\} \tag{2}$$

It outputs the trajectory of the target participant or multiple participants over the next $T_f$ time steps:

$$\widehat{\mathrm{Y}}_1^t = \left\{\hat{x}_i^t, \hat{y}_i^t | t = T_h + 1, \cdots, T_h + T_f\right\} \tag{3}$$

In deterministic prediction, the model typically outputs the single most likely future trajectory. In multimodal prediction, the model must output multiple candidate trajectories and their corresponding probabilities:

$$P = \left(\widehat{Y}_i^k, p_i^k\right)_{k=1}^K \tag{4}$$

where $K$ denotes the number of candidate trajectory modes, and $p_i^k$ denotes the confidence score or occurrence probability of the $k$ th predicted trajectory.

## 2.2. Interaction Based Trajectory Prediction

Traffic participants react to one another. A target vehicle approaching a merge, for example, may respond to a vehicle in the destination lane rather than simply continue its recent motion. Similar dependencies arise in car following, overtaking, yielding, and obstacle avoidance, and they may involve vehicles, cyclists, or pedestrians. Interaction-aware prediction therefore treats the scene as a coupled system whose relevant relationships change with the maneuver and the road context.

Early deep models represented these relationships through local aggregation. Social LSTM[9] introduced social pooling, in which hidden states from nearby agents contribute to the target agent's prediction. Convolutional Social Pooling[10] adapted this idea to highway traffic by placing neighboring vehicle histories on a spatial grid and applying convolution. The grid gives the model a structured view of vehicles ahead, behind, and in adjacent lanes, but it also fixes the neighborhood through a predefined spatial partition.

A fixed grid is an imperfect proxy for influence. At an intersection or merge, a distant agent with right of way can matter more than a nearby agent on a nonconflicting path. This limitation motivated nonlocal, attention-based, and graph formulations. Non-local Social Pooling[11] aggregates dependencies beyond the local neighborhood. Graph models instead assign agents to nodes and encode proximity, relative motion, or another interaction relation as edges; graph convolution or attention then updates the node representations[12].

Graphs are useful here because the representation can change with the scene. Node types may distinguish vehicles, pedestrians, and cyclists, while edge weights may reflect relative distance, velocity, lane connectivity, collision risk, or learned attention. Message passing transfers information along these relations. CRAT-Pred[13], for instance, combines graph convolution with multi-head self-attention and does not require a complex map representation to identify influential vehicles. Dynamic graph attention allows the adjacency and interaction strength to vary as the scene evolves[14].

The progression from pooling to graphs changes the modeling question. Rather than asking only which agents are near the target, a predictor must decide which agents can alter its feasible behavior in the current scene. That distinction matters during yielding, merging, lane changes, and avoidance, where irrelevant neighbors add noise while a small number of conflict-relevant agents determine the maneuver. Interaction modeling must thus balance broad scene coverage with selective attention to the relationships that affect the target.

## 2.3. Trajectory Prediction Based on Map Semantics and Lane Structure

Agent interactions do not determine a trajectory independently of the road. Lane geometry, connectivity, drivable areas, and traffic rules restrict where a vehicle can go and which maneuvers are legal. A turn must follow the topology of an intersection; a lane change requires a connected adjacent lane; and a predicted path should not cross a curb or another non-drivable region. Map information therefore separates road-vehicle forecasting from unconstrained sequence extrapolation.

Map-aware predictors differ mainly in how they encode this structure. Raster models convert the local road scene into a bird's-eye-view semantic image that can be processed with a CNN or Transformer alongside agent history. The representation integrates naturally with image features, but its resolution and finite crop can obscure fine geometry or distant lane connections. Vector models retain lane boundaries, centerlines, and trajectories as polylines. VectorNet[4] uses line segments for both HD maps and agent motion and then learns local and global relations with a graph network. This preserves the map's geometric organization more directly than a raster grid.

Lane-conditioned methods narrow the prediction space before generating a trajectory. They first identify one or more plausible target lanes and then produce futures that follow those candidates. LAformer[15] implements this strategy with a lane-aware module that selects high-potential lane segments before multimodal refinement. The benefit is not

merely a smaller search space: candidate lanes also provide an explicit check against paths that cross boundaries, reverse lane direction, or violate the road topology.

Road constraints can also enter the loss or the output representation. Off-road penalties and related auxiliary objectives teach a model to respect drivable regions during training[16]. In graph-based designs, lane nodes and agent nodes are connected so that road topology shapes the agent representation. GOHOME[17] takes a different route: it predicts a future-position distribution on an HD-map graph through a graph-oriented heatmap. The map then constrains both spatial feasibility and the way multimodal uncertainty is expressed.

These representations answer two related questions: which locations are reachable, and which road-conforming paths lead to them. History-only models lack this explicit feasibility test, especially at intersections, ramps, and multilane roads. Richer scene annotations in Argoverse, nuScenes, and the Waymo Open Motion Dataset have made it practical to model lane topology and semantic constraints together with agent motion. The remaining issue is how strongly a predictor should rely on a map that may itself be incomplete or inaccurate.

### 2.4. Multimodal Generative Trajectory Prediction

Observed motion usually leaves more than one future unresolved. A vehicle may continue in its lane, change lane, or yield; a pedestrian may cross, wait, or turn away. A deterministic regressor collapses these alternatives to one path and can place that path between realistic modes. Multimodal prediction instead represents a set or distribution of plausible futures, together with information about their relative likelihood.

One early solution tied each future to a maneuver label. Maneuver LSTM-based models generate separate trajectories for classes such as lane keeping, left lane change, and right lane change[18]. The labels make the output easy to interpret, but the mode inventory is chosen in advance. A small fixed set is suitable for structured highway maneuvers and less able to represent the range of behavior found in urban intersections.

Anchor and trajectory-set methods replace maneuver labels with a discrete library of futures. MultiPath[19] scores probabilistic anchor trajectories, whereas CoverNet[2] treats prediction as classification over a feasible trajectory set. Both constrain an otherwise continuous output space and can stabilize training. Their coverage, however, is bounded by the library: if the anchors omit an important maneuver or geometry, no probability score can recover the missing future.

Generative models provide a less rigid alternative. Trajectron++[20] combines heterogeneous agents, scene context, and dynamic feasibility in a multimodal multi-agent model. MultiPath++[1] emphasizes information fusion and trajectory aggregation across perception, map, and mode representations. More recent work changes the generation mechanism itself. MotionDiffuser [21] samples diverse joint futures through controlled iterative denoising, while MotionLM [22] converts future motion into tokens and predicts them autoregressively. These models expand the output space, but they also make probability interpretation and cross-agent consistency more important.

A useful multimodal output needs more than a large number of samples. Its modes should represent distinct behavior, include a trajectory near the realized future, respect vehicle dynamics and road constraints, and attach scores that correspond to predictive reliability. Uncertainty-aware approaches distinguish epistemic and aleatoric components rather than treating every source of variation alike[23]. Joint prediction adds another requirement: futures for different agents should remain mutually compatible. QCNeXt[24] addresses this setting by modeling interactions across agents and future time steps instead of forecasting each participant independently.

Multimodal methods therefore trade a single point estimate for a structured account of ambiguity. This avoids the unrealistic mean trajectory that can arise when mutually exclusive behaviors are averaged together. The gain is useful to planning only if the alternatives are feasible and their scores can support risk comparisons. Candidate diversity, probability calibration, and joint consistency are consequently part of the prediction problem rather than optional evaluation details.

## 3. Robustness of Trajectory Prediction for Autonomous Driving

### 3.1. Definition and Major Sources of Failure

For trajectory prediction, robustness describes how performance changes when the assumptions behind a standard test set no longer hold. Relevant departures include noisy states, missing or inaccurate map information, unfamiliar scenes, distribution shift, and deliberate attacks. The concern is both the magnitude of degradation and the form of the resulting failure, particularly when it affects planning. Adding noise and reporting a new ADE or FDE is therefore only one part of a robustness assessment.

The first source of mismatch is the input pipeline. Historical tracks, neighboring-agent states, HD-map elements, and traffic signals are outputs of perception, localization, tracking, and mapping rather than error-free ground truth. Occlusion may fragment a track; state estimates may drift; a lane boundary may be displaced; and a map may lag behind road changes. A predictor that relies heavily on one of these inputs can amplify its error. Work on state, map, and observation uncertainty shows why explicit error modeling is needed to avoid overconfident predictions that pass risk downstream [25, 26, 27].

Robustness also varies by scene. Car following, merges, roundabouts, unprotected turns, and mixed pedestrian-vehicle traffic impose different constraints and levels of interaction. Success on straight, weakly interactive segments does not establish stability at a dense intersection. INTERACTION [28], inD [29], round [30], and exiD [31] provide complementary settings for such sliced evaluation, including intersections, roundabouts, ramps, and other multi-agent situations.

A third mismatch appears between data domains. Road geometry, traffic rules, driving style, and participant mix vary across cities and datasets, so an in-distribution split can hide failures that emerge elsewhere. Cross-domain tests and Out-of-Distribution (OOD) detection examine this gap directly. They ask whether a model transfers to an unseen environment and whether it can recognize that the new input differs from its training distribution [32,33].

At system level, robustness is judged by the decisions a forecast supports. ADE and FDE can be dominated by common, easy cases, while minADE and minFDE reward the closest candidate without evaluating how probability is

distributed over the remaining modes. Neither view identifies a rare error that changes collision risk. A deployment-oriented assessment should therefore report scene-sliced and long-tail behavior, probability quality, rule compliance, and the effect of forecast errors on closed-loop planning.

## 3.2. Robustness to Distribution Shifts and Cross Domain Generalization

Dataset splits usually preserve much of the collection environment, yet deployment changes road layout, traffic density, agent composition, and driving behavior at the same time. An aggregate ADE or FDE from a familiar test set can then overstate robustness because routine samples outweigh merges, roundabouts, lane changes, and other difficult interactions. Distribution-shift evaluation must separate these conditions instead of relying on a single average.

The highD[34], inD[29], rounD[30], and exiD[31] datasets illustrate the scale of the domain change. They cover highways, ramps, intersections, and roundabouts, with different traffic compositions and interaction patterns. A model's rank on one of these settings therefore cannot be read as evidence of comparable performance on the others.

Cross-dataset comparison makes this limitation measurable. UniTraj[32] evaluates multiple models under a unified data and evaluation framework and shows that an advantage observed in distribution can shrink across datasets, cities, or road topologies. Reporting both In-Distribution (ID) and cross-domain or OOD results is thus necessary to distinguish benchmark fit from transferable prediction.

Proposed responses target different causes of shift. Polynomial representations reduce sensitivity to coordinate scale and road geometry by encoding trajectory shape in a more stable form[33]. Analyses on rounD show that road configuration, traffic density, and collection conditions can each alter generalization[35]. CILF and related causal representation methods attempt to retain features that remain informative across environments[36,37], while RuleFuser adds traffic-rule and evidential constraints to improve plausibility under OOD inputs[38]. These methods address transfer, but a deployed system must also detect when transfer is failing. Runtime OOD scores can provide that signal and support confidence adjustment or conservative planning[39].

Distribution robustness therefore has two components: limiting performance loss under a new domain and identifying inputs for which the model's usual confidence is no longer credible. Evaluations should report both across changes in dataset, city, road structure, and scene type. Aggregate displacement error alone measures neither the source of the loss nor the system's ability to respond to it.

## 3.3. Robustness to Perturbations and Adversarial Attacks

Inputs can depart from training conditions through ordinary noise or through an attack. Sensor error, occlusion, tracking failures, localization drift, and map inaccuracies produce incidental perturbations. An adversary instead selects a change to historical states, map elements, or training records in order to induce a particular error. The change may be numerically small, yet optimization can place it where the predictor is most vulnerable.

Historical-state attacks demonstrate this asymmetry: a small targeted change in a past trajectory can produce a much larger forecasting error and alter downstream planning[40]. AdvDO[41] restricts the perturbed history with vehicle kinematics, road boundaries, and traffic rules, making the attack closer to a realizable motion sequence. A robustness test should therefore report physical feasibility as well as perturbation magnitude.

The attack surface extends beyond history. Future-state perturbations near a behavior boundary can change mode assignment, collapse alternatives, or shift high confidence to the wrong mode[42]. Changes to lane centerlines, boundaries, or topology can likewise degrade predictors that depend heavily on an HD map[43]. Map-aware forecasting should consequently model the map as a fallible input, even though the same map supplies valuable feasibility priors.

Training data create a separate exposure. Naturalistic poisoning and backdoor attacks can preserve ordinary test accuracy while activating targeted errors only when a trigger appears[44]. Because the trigger may be absent from a standard benchmark, conventional summary metrics provide little evidence about this failure mode.

Certification methods seek a different form of evidence. Randomized smoothing and related formal analyses bound how much a prediction can change within a specified perturbation region[45,46,47]. Much of this literature studies pedestrian forecasting, so its guarantees cannot be transferred to vehicle prediction without checking the motion model and threat assumptions. The underlying aim, however, is relevant: define a perturbation range within which stability is supported rather than inferred from a few attacks.

A complete attack evaluation should cover the historical trajectory, map, and training pipeline, because each can fail in a different way. It should also trace the effect beyond displacement error to mode probabilities and planning behavior. This broader view separates benign noise sensitivity from worst-case vulnerability and clarifies which safety claim a defense actually supports.

## 3.4. Model Improvement Strategies for Robustness

Robustness improvements intervene at several points in the forecasting pipeline. Some change the training data, some seek features that transfer across domains, and others impose road or physical constraints. A separate group models input uncertainty, while system-level approaches combine adaptation with risk-sensitive responses. Their value should be compared against the failure source they are designed to address.

Robust training exposes a model to noise, domain variation, or adversarial examples before deployment. Adversarial training, domain augmentation, and trajectory smoothing can reduce sensitivity to local perturbations and some forms of shift[48][40]. The benefit is not free: training becomes more expensive, and improving worst-case behavior may reduce accuracy on unperturbed data.

Representation-based methods try to remove domain-specific shortcuts. CILF, Curb Your Attention, and CaDeT use causal feature decomposition, attention gating, or disentanglement to emphasize motion and interaction factors expected to remain stable across scenes[36,49,50]. This differs from simply adding more samples because the objective is to change which features the predictor relies on.

Explicit priors can constrain outputs when data support is weak. RuleFuser incorporates traffic rules in an evidential model for OOD prediction[38]. PiP uses planning information, vehicle kinematics, and drivable-area constraints to discourage dynamically or geographically infeasible

futures[51]. Such constraints can improve plausibility, although their reliability still depends on the correctness of the map and rule inputs.

Uncertainty-aware methods address errors entering from upstream modules. State-uncertainty propagation carries perception uncertainty into the forecast distribution, while NATRA and GFTD target noisy observed trajectories[27][52]. EgoTraj-Bench argues for evaluation with noise derived from onboard perception rather than only synthetic perturbations[53]. These approaches treat imperfect inputs as part of the prediction problem instead of assuming that perception and maps are exact.

Adaptation and risk-responsive forecasting operate closer to deployment. Self-supervised pretraining, pseudo-labeling, and lightweight adaptation can revise a representation for an unfamiliar environment. Diffusion-based multimodal forecasting and risk-responsive methods additionally connect coverage of possible futures with downstream safety[21,54]. The number of generated modes is not a robustness guarantee; their probabilities and safety implications still need calibration.

No single intervention covers every failure source. Augmentation targets perturbation exposure, stable representations target transfer, priors restrict infeasible motion, and uncertainty modeling communicates input and model limitations. A deployment claim should specify which of these mechanisms was tested, under what shift or perturbation, and with which planning consequence. Combining them is useful only when the evaluation keeps those roles visible.

# 4. Trustworthiness and Comprehensive Evaluation of Trajectory Prediction

## 4.1. Foundations of Trustworthy Trajectory Prediction

A trajectory forecast is consumed by a planner, so trustworthiness concerns the information carried with the path as well as its geometric error. The planner needs to know whether a forecast remains reliable in the current scene and how an error would affect collision avoidance, route choice, or an interactive maneuver[55,56]. Evaluation must therefore connect predictive outputs to their use in a driving system.

Open-loop benchmarks commonly report ADE, FDE, minADE, minFDE, Root Mean Square Error (RMSE), and Miss Rate (MR), including in Argoverse and the Waymo Open Motion Dataset. These metrics quantify trajectory fit, but averages understate rare failures and best-of-set scores do not judge whether probability mass is placed on the correct mode[2,19,20]. A trustworthy predictor also needs to represent uncertainty, align confidence with error or empirical coverage, recognize OOD or high-risk inputs, and provide outputs that support stable closed-loop planning[57,58].

The resulting evaluation problem has five linked parts: geometric accuracy, uncertainty representation, calibration, failure detection, and planning impact. Treating any one of them as a substitute for the others leaves a gap between benchmark performance and operational use.

## 4.2. Uncertainty Quantification and Propagation

Uncertainty enters trajectory prediction at several points rather than appearing only in the final trajectory. Agent intent is unresolved, road and interaction context may be ambiguous, and upstream states may already contain error. The ego vehicle's future action introduces another dependency. A useful uncertainty model should therefore identify which source is being represented and how it reaches the planner.

Behavioral ambiguity is commonly represented with multiple trajectories or distributions, as in MultiPath[19], CoverNet[2], Trajectron++[20], and MultiPath++[1]. Epistemic uncertainty has a different origin: training data may not cover the current scene, particularly in long-tail or OOD conditions. Models that separate aleatoric and epistemic uncertainty aim to distinguish variation in the observations from limits in learned knowledge[23]. Upstream perception adds uncertainty to position, velocity, and heading[25], while map construction and lane matching affect the feasible trajectory space[26]. Interaction uncertainty is coupled across agents because one participant's future changes the responses available to another[20].

These sources propagate differently. An uncertain state broadens or shifts the predictor's input; a map error changes which paths appear feasible; interaction uncertainty changes the joint distribution over agents. The forecast then influences the planner's estimate of risk. CCTR[59] illustrates the consequence of leaving this chain uncalibrated: a model may output probabilities or variances that cause the planner to understate or overstate risk. Quantification is therefore incomplete unless the source, calibration, and downstream use of uncertainty are specified.

## 4.3. Probabilistic Calibration and Confidence Evaluation

A multimodal predictor may attach probabilities to candidate paths or parameterize a Gaussian, mixture distribution, or heatmap. The presence of a probability field does not establish that the score is reliable. A model can be highly confident in the wrong maneuver or assign too little mass to alternatives in a complex interaction, leading a planner to discount a material risk.

Calibration compares stated confidence with observed frequency or coverage. Guo et al.[60] describe Expected Calibration Error (ECE), reliability diagrams, and temperature scaling for classification confidence. Continuous prediction requires an analogous coverage check: if a method reports a 90% prediction interval, observations should fall inside it approximately 90% of the time under the conditions for which the interval is claimed[61]. The metric must match the output type; mode confidence and spatial coverage are related but not interchangeable.

CCTR connects trajectory calibration to planning and uses sample-dependent temperature adjustment to revise mode probabilities for uncertainty-aware decisions[59]. Conformal prediction offers prediction regions with statistical coverage guarantees under stated assumptions[62], and CUQDS studies this approach when the data distribution shifts[63]. These methods provide different objects, calibrated scores or covered regions, so their guarantees should be interpreted at the level they actually address.

For trajectory prediction, a calibration study should test three relations: whether confidence orders trajectories by error, whether prediction regions attain nominal coverage, and whether the score tracks failure or planning risk. A confidence value becomes operationally meaningful only when the evaluation identifies which of these relations holds and under which scene distribution.

### 4.4. Failure Detection and Risk Identification

Low mean error does not rule out severe failures in a dense interaction, after a sensor fault, or under an OOD input. At runtime the system must decide whether the current forecast is credible before the future trajectory is observed. Failure detection therefore relies on signals available at prediction time and should distinguish an unfamiliar input from a forecast whose error is likely to affect planning.

Offline studies can define failure by an ADE or FDE threshold because ground truth is available. Deployment cannot. Online detection instead uses confidence, predicted uncertainty, or disagreement across modes or models as a proxy for future error. Self-aware trajectory prediction makes this reliability estimate an explicit model output[64]. Its value depends on calibration against the failures the system is intended to catch.

Geometric error and task risk need not rank cases in the same order. Farid et al.[65] combine prediction error with planning cost in task-relevant failure detection. A displacement error for a distant vehicle may leave the ego plan unchanged, whereas a smaller error at a merge or pedestrian crossing can alter collision risk. Failure criteria should therefore reflect the downstream decision as well as the distance between trajectories.

OOD detection addresses the related question of whether an input resembles the model's training distribution. Confidence, feature distance, and energy scores have been used to flag unfamiliar scenes, including work that combines real-time OOD awareness with uncertainty estimation for trajectory prediction[39]. Such a flag does not prove that a forecast is wrong, but it can justify a more conservative planning response.

The purpose of failure detection is thus prospective rather than retrospective. The system needs an actionable warning that the forecast may be unreliable and that the potential error matters to the current plan. Accuracy thresholds, OOD scores, uncertainty estimates, and task cost each supply part of that judgment.

### 4.5. Trustworthiness Evaluation in the Prediction Planning Loop

Open-loop evaluation holds the recorded scene fixed: the model predicts from observed history, and its output is compared with the recorded future. This makes models easy to compare, but it omits feedback. In operation, a prediction changes the ego plan, and that plan may change how another agent responds. A low open-loop error can therefore coexist with poor closed-loop behavior.

Studies have reported mismatches between minADE or minFDE and vehicle-control or planning performance. L2 distance also does not directly represent collision, off-road, or interaction risk[66,67]. These findings do not make open-loop metrics irrelevant; they show that trajectory similarity and planning utility are different evaluation targets.

nuPlan evaluates planning in simulation with measures such as collisions, drivable-area compliance, comfort, progress, and task completion[58]. nuPlan-R adds reactive agents so that surrounding traffic responds to the ego vehicle[68]. PiP[51] and DIPP[69] connect prediction and planning at model level: PiP conditions forecasting on planning information, whereas DIPP jointly optimizes the two components through differentiable computation. These approaches test whether predictive information changes decisions, not merely whether a path resembles the recorded future.

Open- and closed-loop protocols should therefore be read together. The open-loop result isolates forecasting against a common reference trajectory. The closed-loop result measures how the forecast behaves when decisions and other agents can respond. Combining them preserves comparability while exposing collision, comfort, rule-compliance, and progress effects that a displacement metric cannot capture.

## 5. Challenges and Future Research Directions

Although trajectory prediction for autonomous driving has advanced substantially in interaction modeling, map-semantic fusion, multimodal generation, and Transformer architectures, a central challenge remains: moving from accurate prediction to robust and trustworthy prediction. Current models are often evaluated using open-loop error metrics such as ADE, FDE, minADE, and minFDE, which inadequately reflect reliability under perception noise, map deviations, OOD scenes, long-tail interactions, adversarial perturbations, and closed-loop planning. Moreover, although multimodal predictors output candidate trajectories with probability scores, these probabilities do not necessarily agree with empirical frequencies, prediction errors, or planning risks. They may remain uncalibrated and overconfident, conceal failures, or underestimate high-risk modes. The principal challenge is therefore no longer merely to reduce average prediction error, but to achieve stable, calibrated, interpretable, and monitorable prediction that supports safe planning in complex open traffic environments.

Future research should develop systematic methods and evaluation frameworks for integrated robustness and trustworthiness. First, unified benchmarks should incorporate cross-dataset generalization, scene-slice evaluation, long-tail risk tests, input-perturbation tests, map-uncertainty tests, and adversarial-attack tests within a common protocol. Second, research should strengthen the quantification and propagation of uncertainty from multiple sources so that perception-state errors, map-semantic uncertainty, epistemic uncertainty, and stochastic traffic behavior are modeled explicitly during prediction. Third, multimodal probabilistic calibration and runtime failure detection should be improved so that confidence accurately reflects error magnitude, risk, and planning utility. Finally, trajectory prediction should progress from open-loop evaluation alone toward joint evaluation in the prediction-planning loop, focusing on how predictions affect collision risk, rule compliance, comfort, traffic efficiency, and safety-critical decisions. Overall, the field is likely to move from model comparison centered on accuracy metrics toward coordinated evaluation of robustness, trustworthiness, and system safety for real-world deployment.

## References


[1] Varadarajan, B., et al. (2022). MultiPath++: Efficient information fusion and trajectory aggregation for behavior prediction. 2022 International Conference on Robotics and Automation (ICRA), 7814–7821. https://doi.org/10.1109/ICRA46639.2022.9812107.

[2] Phan-Minh, T., Grigore, E. C., Boulton, F. A., Beijbom, O., & Wolff, E. M. (2020). CoverNet: Multimodal behavior prediction using trajectory sets. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 14062–14071. https://doi.org/10.1109/CVPR42600.2020.01408

[3] Zhao, H., Gao, J., Lan, T., & et al. (2021). TNT: Target-driven trajectory prediction. Conference on Robot Learning, 895–904. PMLR.

[4] Gao, J., et al. (2020). VectorNet: Encoding HD maps and agent dynamics from vectorized representation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11522–11530. https://doi.org/10.1109/CVPR42600.2020.01154

[5] Liang, M., et al. (2020). Learning lane graph representations for motion forecasting. In A. Vedaldi, H. Bischof, T. Brox, & J. M. Frahm (Eds.), Computer vision – ECCV 2020 (Vol. 12347, pp. 541–556). Springer. https://doi.org/10.1007/978-3-030-58536-5_32

[6] Yuan, Y., Weng, X., Ou, Y., & Kitani, K. (2021). AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 9793–9803. https://doi.org/10.1109/ICCV48922.2021.00967

[7] Zhou, Z., Ye, L., Wang, J., Wu, K., & Lu, K. (2022). HiVT: Hierarchical vector transformer for multi-agent motion prediction. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8813–8823. https://doi.org/10.1109/CVPR52688.2022.00862

[8] Zhou, Z., Wang, J., Li, Y., & Huang, Y. (2023). Query-centric trajectory prediction. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 17863–17873. https://doi.org/10.1109/CVPR52729.2023.01713

[9] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social LSTM: Human trajectory prediction in crowded spaces. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 961–971. https://doi.org/10.1109/CVPR.2016.110

[10] Deo, N., & Trivedi, M. M. (2018). Convolutional social pooling for vehicle trajectory prediction. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 1549–1558. https://doi.org/10.1109/CVPRW.2018.00196

[11] Messaoud, K., Yahiaoui, I., Verroust-Blondet, A., & Nashashibi, F. (2019). Non-local social pooling for vehicle trajectory prediction. 2019 IEEE Intelligent Vehicles Symposium (IV), 975–980. https://doi.org/10.1109/IVS.2019.8813829

[12] Mo, X., Xing, Y., & Lv, C. (2021). Graph and recurrent neural network-based vehicle trajectory prediction for highway driving. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 1934–1939. https://doi.org/10.1109/ITSC48978.2021.9564929

[13] Schmidt, J., Jordan, J., Gritschneder, F., & Dietmayer, K. (2022). CRAT-Pred: Vehicle trajectory prediction with crystal graph convolutional neural networks and multi-head self-attention. 2022 International Conference on Robotics and Automation (ICRA), 7799–7805. https://doi.org/10.1109/ICRA46639.2022.9811637

[14] Dong, B., Liu, H., Bai, Y., Lin, J., Xu, Z., Xu, X., & Kong, Q. (2021). Multi-modal trajectory prediction for autonomous driving with semantic map and dynamic graph attention network (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2103.16273

[15] Liu, M., et al. (2024). LAformer: Trajectory prediction for autonomous driving with lane-aware scene constraints. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2039–2049. https://doi.org/10.1109/CVPRW63382.2024.00209

[16] Li, Z., & Yu, H. (2024). Trajectory prediction for autonomous driving using a transformer network. arXiv.

[17] Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., & Moutarde, F. (2022). GOHOME: Graph-oriented heatmap output for future motion estimation. 2022 International Conference on Robotics and Automation (ICRA), 9107–9114. https://doi.org/10.1109/ICRA46639.2022.9812253

[18] Deo, N., & Trivedi, M. M. (2018). Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs. 2018 IEEE Intelligent Vehicles Symposium (IV), 1179–1184. https://doi.org/10.1109/IVS.2018.8500493

[19] Chai, Y., Sapp, B., Bansal, M., & et al. (2019). Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. arXiv preprint arXiv:1910.05449.

[20] Salzmann, T., Ivanovic, B., Chakravarty, P., & Pavone, M. (2020). Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In A. Vedaldi, H. Bischof, T. Brox, & J. M. Frahm (Eds.), Computer vision – ECCV 2020 (Vol. 12363, pp. 683–700). Springer. https://doi.org/10.1007/978-3-030-58523-5_40

[21] Jiang, S., Cornman, A., Park, K., Sapp, B., Zhou, D., & Anguelov, D. (2023). MotionDiffuser: Controllable multi-agent motion prediction using diffusion. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9644–9653. https://doi.org/10.1109/CVPR52729.2023.00930

[22] Seff, A., et al. (2023). MotionLM: Multi-agent motion forecasting as language modeling. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 8545–8556. https://doi.org/10.1109/ICCV51070.2023.00788

[23] Suk, H., & Kim, S. (2025). Uncertainty-aware multimodal trajectory prediction via a single inference from a single model. Sensors, 25(1), 217. https://doi.org/10.3390/s25010217

[24] Zhou, Z., Wen, Z., Wang, J., & et al. (2023). Qcnext: A next-generation framework for joint multi-agent trajectory prediction. arXiv preprint arXiv:2306.10508.

[25] Ivanovic, B., Lin, Y., Shrivastava, S., Chakravarty, P., & Pavone, M. (2022). Propagating state uncertainty through trajectory forecasting. 2022 International Conference on Robotics and Automation (ICRA), 2351–2358. https://doi.org/10.1109/ICRA46639.2022.9811776

[26] Gu, X., Song, G., Gilitschenski, I., Pavone, M., & Ivanovic, B. (2024). Producing and leveraging online map uncertainty in trajectory prediction. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 14521–14530. https://doi.org/10.1109/CVPR52733.2024.01376

[27] Li, R., et al. (2025). NATRA: Noise-agnostic framework for trajectory prediction with noisy observations. 2025 IEEE/CVF International Conference on Computer Vision (ICCV), 27872–27884. https://doi.org/10.1109/ICCV51701.2025.02587

[28] Zhan, W., Sun, L., Wang, D., & et al. (2019). Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. arXiv preprint arXiv:1910.03088.

[29] Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., & Eckstein, L. (2020). The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections. 2020 IEEE Intelligent Vehicles Symposium (IV), 1929–1934. https://doi.org/10.1109/IV47402.2020.9304839

[30] Krajewski, R., Moers, T., Bock, J., Vater, L., & Eckstein, L. (2020). The rounD dataset: A drone dataset of road user trajectories at roundabouts in Germany. 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 1–6. https://doi.org/10.1109/ITSC45102.2020.9294728

[31] Moers, T., Vater, L., Krajewski, R., Bock, J., Zlocki, A., & Eckstein, L. (2022). The exiD dataset: A real-world trajectory

dataset of highly interactive highway scenarios in Germany. 2022 IEEE Intelligent Vehicles Symposium (IV), 958–964. https://doi.org/10.1109/IV51971.2022.9827305

[32] Feng, L., Bahari, M., Amor, K. M. B., Zablocki, É., Cord, M., & Alahi, A. (2025). UniTraj: A unified framework for scalable vehicle trajectory prediction. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, & G. Varol (Eds.), Computer vision – ECCV 2024 (Vol. 15070, pp. 1–18). Springer. https://doi.org/10.1007/978-3-031-73254-6_7

[33] Yao, Y., Yan, S., Goehring, D., Burgard, W., & Reichardt, J. (2024). Improving out-of-distribution generalization of trajectory prediction for autonomous driving via polynomial representations. 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 488–495. https://doi.org/10.1109/IROS58592.2024.10801535

[34] Krajewski, R., Bock, J., Kloeker, L., & Eckstein, L. (2018). The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems. 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2118–2125. https://doi.org/10.1109/ITSC.2018.8569552

[35] Zhang, Z. (2024). Exploring rounD dataset for domain generalization in autonomous vehicle trajectory prediction. Sensors, 24(23), 7538. https://doi.org/10.3390/s24237538

[36] Li, S., Xue, Q., Zhang, Y., & Li, X. (2023). CILF: Causality inspired learning framework for out-of-distribution vehicle trajectory prediction. In H. Lu, M. Blumenstein, S. B. Cho, C. L. Liu, Y. Yagi, & T. Kamiya (Eds.), Pattern recognition (Vol. 14407, pp. 149–164). Springer. https://doi.org/10.1007/978-3-031-47637-2_12

[37] Bagi, S. S. G., Gharaee, Z., Schulte, O., & Crowley, M. (2023). Generative causal representation learning for out-of-distribution motion forecasting. Proceedings of the 40th International Conference on Machine Learning, 202, 31596–31612. JMLR.org.

[38] Patrikar, J., Veer, S., Sharma, A., & et al. (2024). Rulefuser: Injecting rules in evidential networks for robust out-of-distribution trajectory prediction. arXiv preprint arXiv:2405.11139.

[39] Guo, T., Banerjee, T., Liu, R., & Su, L. (2026). Building real-time awareness of out-of-distribution in trajectory prediction for autonomous vehicles. ACM Transactions on Cyber-Physical Systems. https://doi.org/10.1145/3803009

[40] Zhang, Q., Hu, S., Sun, J., Chen, Q. A., & Mao, Z. M. (2022). On adversarial robustness of trajectory prediction for autonomous vehicles. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15138–15147. https://doi.org/10.1109/CVPR52688.2022.01473

[41] Cao, Y., Xiao, C., Anandkumar, A., Xu, D., & Pavone, M. (2022). AdvDO: Realistic adversarial attacks for trajectory prediction. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), Computer vision – ECCV 2022 (Vol. 13665, pp. 48–65). Springer. https://doi.org/10.1007/978-3-031-20065-6_3

[42] Schumann, J., Hagenus, J., Mathiesen, F., & Zgonnikov, A. (2026). Realistic adversarial attacks for robustness evaluation of trajectory prediction models via future state perturbation. ACM Journal on Autonomous Transportation Systems, 3(4), Article 23. https://doi.org/10.1145/3799894

[43] Zheng, Z., Ying, X., Yao, Z., & Chuah, M. C. (2023). Robustness of trajectory prediction models under map-based attacks. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 4530–4539. https://doi.org/10.1109/WACV56688.2023.00452

[44] Pourkeshavarz, M., Sabokrou, M., & Rasouli, A. (2024). Adversarial backdoor attack by naturalistic data poisoning on trajectory prediction in autonomous driving. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 14885–14894. https://doi.org/10.1109/CVPR52733.2024.01410

[45] Bahari, M., Saadatnejad, S., Farsangi, A. A., & et al. (2025). Certified human trajectory prediction. 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12301–12311.

[46] Schumann, J. F., Figueiredo, E., Mathiesen, F. B., & et al. (2026). Evaluating randomized smoothing as a defense against adversarial attacks in trajectory prediction. arXiv preprint arXiv:2603.10821.

[47] Zhang, L., et al. (2026). TRAJRS: Towards certified robustness in pedestrian trajectory prediction. ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3011–3015. https://doi.org/10.1109/ICASSP55912.2026.11462840

[48] Duan, A., Wang, R., Cui, Y., He, P., & Chen, L. (2024). Causal robust trajectory prediction against adversarial attacks for autonomous vehicles. IEEE Internet of Things Journal, 11(22), 35762–35776. https://doi.org/10.1109/JIOT.2023.3342788

[49] Ahmadi, E., Mercurius, R., Alizadeh, S., Rezaee, K., & Rasouli, A. (2025). Curb your attention: Causal attention gating for robust trajectory prediction in autonomous driving. 2025 IEEE International Conference on Robotics and Automation (ICRA), 779–786. https://doi.org/10.1109/ICRA55743.2025.11128367

[50] Pourkeshavarz, M., Zhang, J., & Rasouli, A. (2024). CaDeT: A causal disentanglement approach for robust trajectory prediction in autonomous driving. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 14874–14884. https://doi.org/10.1109/CVPR52733.2024.01409

[51] Song, H., Ding, W., Chen, Y., Shen, S., Wang, M. Y., & Chen, Q. (2020). PiP: Planning-informed trajectory prediction for autonomous driving. In A. Vedaldi, H. Bischof, T. Brox, & J. M. Frahm (Eds.), Computer vision – ECCV 2020 (Vol. 12366, pp. 568–584). Springer. https://doi.org/10.1007/978-3-030-58589-1_36

[52] Lin, H., Wang, Y., Huo, M., Peng, C., Liu, Z., & Tomizuka, M. (2024). Joint pedestrian trajectory prediction through posterior sampling. 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 5672–5679. https://doi.org/10.1109/IROS58592.2024.10802382

[53] Liu, J., Zhou, J., Ye, K., & et al. (2025). Egotraj-bench: Towards robust trajectory prediction under ego-view noisy observations. arXiv preprint arXiv:2510.00405.

[54] Liao, H., Kong, H., Li, Z., & et al. (2025). SafeCast: Risk-responsive motion forecasting for autonomous vehicles. AI Open, 6, 118–129. https://doi.org/10.1016/j.aiopen.2025.08.001

[55] Xu, M., Liu, Z., Wang, B., & Li, S. (2025). A survey of autonomous driving trajectory prediction: Methodologies, challenges, and future prospects. Machines, 13(9), 818. https://doi.org/10.3390/machines13090818

[56] Shi, J., Chen, J., Wang, Y., & et al. (2026). Motion forecasting for autonomous vehicles: A survey. International Journal of Machine Learning and Cybernetics, 17, 38. https://doi.org/10.1007/s13042-025-02859-8

[57] Phong, T., Wu, H., Yu, C., & et al. (2023). What truly matters in trajectory prediction for autonomous driving? Advances in Neural Information Processing Systems, 36, 71327–71339.

[58] Caesar, H., Kabzan, J., Tan, K. S., & et al. (2021). nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. arXiv preprint arXiv:2106.11810.

[59] Cao, C., Chen, X., Wang, J., & et al. (2024). CCTR: Calibrating trajectory prediction for uncertainty-aware motion planning in

autonomous driving. Proceedings of the AAAI Conference on Artificial Intelligence, 38(19), 20949–20957.

[60] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. Proceedings of the 34th International Conference on Machine Learning, 70, 1321–1330. JMLR.org.

[61] Kuleshov, V., Fenner, N., & Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. International Conference on Machine Learning, 2796–2804. PMLR.

[62] Angelopoulos, A. N., & Bates, S. (2023). Conformal prediction: A gentle introduction. Foundations and Trends in Machine Learning, 16(4), 494–591. https://doi.org/10.1561/2200000101

[63] Huang, H., He, S., & Miao, F. (2025). CUQDS: Conformal uncertainty quantification under distribution shift for trajectory prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 39(16), 17422–17430.

[64] Shao, W., Li, J., & Wang, H. (2023). Self-aware trajectory prediction for safe autonomous driving. 2023 IEEE Intelligent Vehicles Symposium (IV), 1–8. https://doi.org/10.1109/IV55152.2023.10186629

[65] Farid, A., Veer, S., Ivanovic, B., & et al. (2023). Task-relevant failure detection for trajectory predictors in autonomous vehicles. Conference on Robot Learning, 1959–1969. PMLR.

[66] Zhai, J. T., Feng, Z., Du, J., & et al. (2023). Rethinking the open-loop evaluation of end-to-end autonomous driving in nuScenes. arXiv preprint arXiv:2305.10430.

[67] Dauner, D., Hallgarten, M., Geiger, A., & et al. (2023). Parting with misconceptions about learning-based vehicle motion planning. Conference on Robot Learning, 1268–1281. PMLR.

[68] Peng, M., Yao, R., Guo, X., & Ma, J. (2026). nuPlan-R: A closed-loop planning benchmark for autonomous driving via reactive multi-agent simulation. 2026 IEEE Intelligent Vehicles Symposium (IV), 702–709. https://doi.org/10.1109/IV66570.2026.11623929

[69] Huang, Z., Liu, H., Wu, J., & Lv, C. (2024). Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving. IEEE Transactions on Neural Networks and Learning Systems, 35(11), 15222–15236. https://doi.org/10.1109/TNNLS.2023.3283542.