

## Robustness in trajectory prediction for autonomous vehicles a survey

Hagenus, Jeroen; Mathiesen, Frederik Baymler; Schumann, Julian F.; Zgonnikov, Arkady

**DOI**

[10.1109/IV55156.2024.10588389](https://doi.org/10.1109/IV55156.2024.10588389)

**Publication date**

2024

**Document Version**

Final published version

**Published in**

Proceedings of the 35th IEEE Intelligent Vehicles Symposium, IV 2024

**Citation (APA)**

Hagenus, J., Mathiesen, F. B., Schumann, J. F., & Zgonnikov, A. (2024). Robustness in trajectory prediction for autonomous vehicles: a survey. In *Proceedings of the 35th IEEE Intelligent Vehicles Symposium, IV 2024* (pp. 969-976). (IEEE Intelligent Vehicles Symposium, Proceedings). IEEE.  
<https://doi.org/10.1109/IV55156.2024.10588389>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

***Green Open Access added to TU Delft Institutional Repository***

***'You share, we take care!' - Taverne project***

***<https://www.openaccess.nl/en/you-share-we-take-care>***

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

# Robustness in trajectory prediction for autonomous vehicles: a survey

Jeroen Hagenus<sup>1</sup>, Frederik Baymler Mathiesen<sup>2</sup>, Julian F. Schumann<sup>1</sup>, Arkady Zgonnikov<sup>1</sup>

**Abstract**— Autonomous vehicles rely on accurate trajectory prediction to inform decision-making processes related to navigation and collision avoidance. However, current trajectory prediction models show signs of overfitting, which may lead to unsafe or suboptimal behavior. To address these challenges, this paper presents a comprehensive framework that categorizes and assesses the definitions and strategies used in the literature on evaluating and improving the robustness of trajectory prediction models. This involves a detailed exploration of various approaches, including data slicing methods, perturbation techniques, model architecture changes, and post-training adjustments. In the literature, we see many promising methods for increasing robustness, which are necessary for safe and reliable autonomous driving.

## I. INTRODUCTION

A critical technological aspect in the control of autonomous vehicles is forecasting future states of the surrounding environment to facilitate safe navigation through dynamic surroundings. The predicted future trajectories of traffic participants play an important role in detecting potential hazards in advance and are essential for the design of decision-making or planning algorithms [1], [2]. This predictive task relies on utilizing perceived data, such as camera and LiDAR data [3]–[6]. This data is mostly tracked across multiple timestamps, transformed into trajectories, and then integrated into maps to enable accurate predictions [3], [5].

The task of predicting future trajectories of traffic participants is a complex time series regression problem [7]. It involves intricate contextual elements, including directional information, velocity, position, and time. Furthermore, such trajectories inherently convey semantic and behavioral information, facilitating inference of the future movement of road users [7].

Deep learning models have proven effective in trajectory prediction [8]–[12]. These models consider dynamic factors, model map information, and can incorporate interaction-related factors, allowing adaptation to complex scenes [1]. However, many deep learning models are known to be sensitive to small errors and susceptible to external attacks, potentially resulting in undesirable behavior and decreased performance [13]. Therefore, ensuring reliable performance (often called *robustness*) is key to the safe deployment of these deep learning models. In the context of trajectory prediction, robustness implies the ability to produce predictions unaffected by spurious features (e.g., sensor noise, data

collection errors, lack of understanding of object intentions, etc.) as well as prevention of overfitting, i.e. learned patterns that fail to generalize to new environments [14].

Over the past few years, there has been a surge in research focused on evaluating and improving the robustness of deep learning trajectory prediction models. However, despite multiple surveys discussing trajectory prediction methods in general [1], [2], [15], currently *the literature lacks a detailed and comprehensive analysis of robustness strategies for trajectory prediction*. To address this gap, this paper aims to provide a structured overview of the current research on robustness in trajectory prediction. To do this, we first formalize the problem of trajectory prediction and its robustness (Section II). We then introduce a framework that covers existing strategies for evaluating and improving robustness in trajectory prediction (Section III). For the evaluation/improvement strategies, detailed insights are then provided into specific methods for making trajectory prediction models more robust (Section IV). Finally, we summarize the surveyed literature and suggest concrete directions for future research on robustness in trajectory prediction (Sections V and VI).

## II. TRAJECTORY PREDICTION AND ROBUSTNESS

The objective of trajectory prediction is to model the future trajectory of other road users (vehicles, pedestrians and cyclists), often referred to as agents, taking into account their past states. The state  $x_i^t$  of an agent  $i$  usually contains its position and potentially velocity, acceleration, and heading angle [16]. Additionally, environmental contexts, such as road lines and sidewalks, are sometimes considered as well [16]. The process involves capturing a sequence of observed states for each agent at a fixed time interval  $\Delta t$  and generating the projected future trajectory for each agent. Some models produce a single predicted trajectory, while others generate a set of feasible trajectories along with associated confidence levels [17].

Formally, we will use the following description of the trajectory prediction problem [16], [18]. Denote by  $X^t = (x_1^t, \dots, x_N^t)$  the joint state of the scene at time  $t \leq 0$  where  $N$  represents the total number of agents. Let  $H$  denote the length of the observed time window, then  $X = (X^{-H+1}, \dots, X^0)$  denote the set of states observed. Future states are denoted by  $Y^t = (y_1^t, \dots, y_N^t)$  where, similarly to the past,  $y_i^t$  represent the state of agent  $i$  at time step  $t$ . The task is to predict, using prediction model  $M_\theta$  parameterized by  $\theta$ , the trajectory of all agents in a finite prediction horizon  $T$ , i.e. predict the joint trajectory  $Y = (Y^1, \dots, Y^T) \approx \hat{Y} =$

<sup>1</sup>Department of Cognitive Robotics, <sup>2</sup>Delft Center for Systems and Control, TU Delft, 2628 CB Delft, The Netherlands {J.Hagenus@student., F.B.Mathiesen@, J.F.Schumann@, A.Zgonnikov@tudelft.nl

$M_\theta(X)$ . The dataset consists of multiple combinations of inputs, denoted as  $D = \{(X, Y), \dots\}$ .

#### A. Standard evaluation

To train and evaluate the prediction model  $M_\theta$ , a train/test split of the dataset  $D$  is established [19]. The training set is utilized to train the model by adjusting the parameters  $\theta$  to optimize a loss function. The optimized model  $M_{\theta^*}$  with parameters  $\theta^*$  is evaluated on the test split of the data. Within the test set, historical states are used to generate future predictions using  $M_{\theta^*}$ . Subsequently, a measure is applied to compare these predictions with the ground truth trajectories of the test set.

#### B. Performance measures

To assess the performance of the trajectory prediction model, it is crucial to quantify the disparity between the ground truth values  $Y$  and the predicted values  $\hat{Y}$ . For this a number of measures have been developed [20]–[25]. When examining the measures, a critical distinction lies in whether they consider all future data points or only the endpoint. Commonly used measures include the Average Displacement Error, where the average Euclidean distance between predictions and ground truth is calculated for all time steps [20], and the Final Displacement Error, which considers only the endpoint [21]. The Miss Rate is the ratio in which the Euclidean prediction at the endpoint exceeds a defined threshold [22]. The Collision Rate calculates the percentage of samples in which at least one collision occurs in the predicted trajectories between the candidate agent and its neighbors [23]. The Off-Road Rate is the probability of all predicted trajectories having at least one point off-road [24], [25]. The Mean Inter-Endpoint Distance is the average distance from all predicted endpoints to the mean endpoint [25].

#### C. Robustness in trajectory prediction

Intuitively, the robustness of a prediction model  $M_\theta$  requires that it produces consistent outputs  $\hat{Y} = (\hat{Y}^1, \dots, \hat{Y}^T)$  even when presented with a slightly perturbed version  $\tilde{X} = (\tilde{X}^{-H+1}, \dots, \tilde{X}^0)$  of the history  $X$ . Robustness against spurious features in trajectory prediction can be categorized into adversarial and natural robustness [26], [27].

Adversarial robustness involves robustness to adversarial attacks — small, worst-case perturbations to model inputs [28]. To reliably evaluate and train models that are robust to adversarial attacks, it is essential to specify an attack model, providing a precise definition of the type of attack to which the model should be resistant [16]. An example of an attack model in the context of perception is presented by Eykholt et al. [29], where black and white stickers serve as adversarial examples on road signs, achieving high targeted misclassification rates against standard-architecture road sign classifiers.

Natural robustness of trajectory prediction implies robustness against perturbations that adhere to the physical constraints of actual vehicle driving and reflect normal driving

behavior rather than adversarial maneuvers [16], [30]. If generated trajectories appear unnatural to an autonomous vehicle, they may exploit this property to predict an anomaly or an attack [30].

Moreover, ensuring the robustness of trajectory prediction model  $M_\theta$  requires it to learn the true underlying relationship between the observed data  $X = (X^{-H+1}, \dots, X^0)$  and the future predictions  $Y = (Y^1, \dots, Y^T)$ . Assessing its resistance to overfitting can be measured through the train/test split. The key is to determine whether the training and test data are identical and independent samples from the same data distribution [31]. However, this condition is rarely satisfied in practice, resulting in distributional shifts [31]. These shifts, characterized by differences between the training and test distributions, can substantially undermine the accuracy of deep learning models deployed in real-world scenarios [32].

### III. STRATEGIES FOR EVALUATION AND IMPROVEMENT OF ROBUSTNESS

We propose a general framework for training and evaluating trajectory prediction models that includes robustness evaluation and improvement strategies (Figure 1). This overarching framework is based on a synthesis of existing literature, incorporating approaches currently available in the field.

#### A. Evaluation strategies

To assess the robustness of a trajectory prediction model in specific settings, the test data can be modified [17]. We distinguish between two general strategies of evaluating robustness through data modification (Figure 1).

1) *Test on sliced data (E.1):* To assess robustness against overfitting, testing can be performed on slices [33]. This evaluation strategy involves using benign inputs from the test set under specific settings to assess the robustness of the model on crucial subsets [33] (Figure 1). The evaluation includes testing individual traffic situations sorted by factors such as safety-criticality [34], trajectory type [17], [35] or geographical location [36]. Whenever the model might underperform on these subsets, it poses a challenge in terms of generalizing to out-of-distribution data [31].

2) *Test on perturbed data (E.2):* To assess the adversarial robustness of the model, perturbations denoted as  $\tilde{X}$  can be introduced into the test data [30]. This evaluation method involves the trained trajectory prediction model  $M_{\theta^*}$  making predictions on perturbed test samples rather than benign ones (Figure 1). While models often perform well on benign inputs, they may be vulnerable to adversarial attacks and perturbations that manipulate the input, resulting in incorrect outputs [37]. This strategy provides insights into the robustness of the model on unseen data that have been slightly altered, offering a realistic assessment of its resilience to perturbations [28]. However, a notable limitation is the reliance on simulating realistic agent behavior, which can introduce a simulation-to-real gap [34].

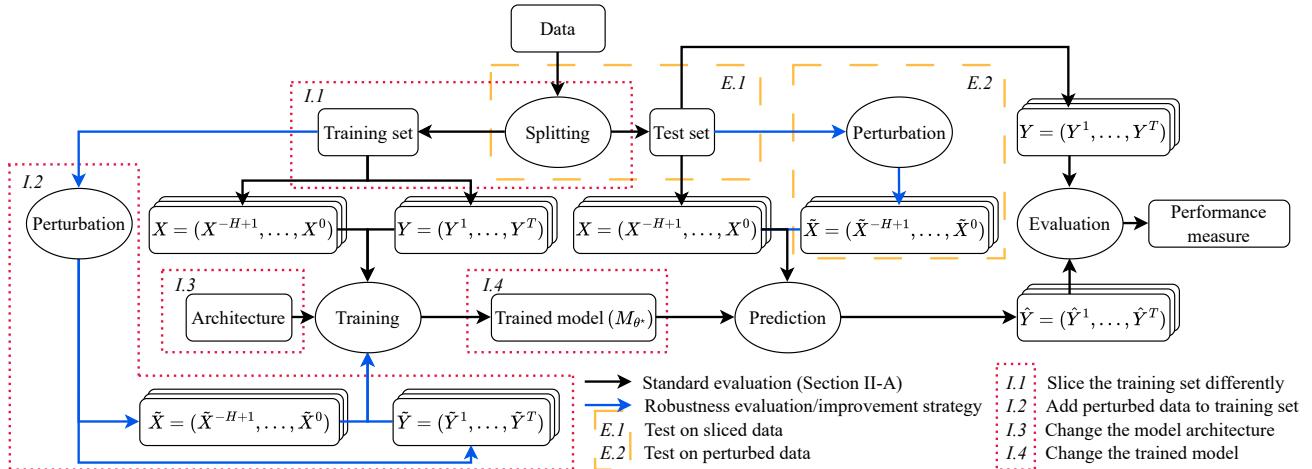


Fig. 1. A general framework for training and evaluating (robust) trajectory prediction models. Robustness evaluation (*E*) and improvement (*I*) components are highlighted by dashed and dotted boxes respectively.

### B. Robustness improvement strategies

Based on the performance measure results obtained from both standard and robustness evaluations, the selection of an improvement strategy becomes crucial. Improvements in robustness can be achieved via four general strategies (Figure 1).

1) *Slice the training set differently (I.1)*: This strategy involves modifying the training set to prevent the model from overfitting. The implementation of this approach occurs during the preparation of the train/test split (Figure 1). It includes two key steps: firstly, mining specific scenarios from the dataset, and secondly, designing a remediation strategy to mitigate the impact of distribution shift [34]. The objective is to emphasize the overall quality of the model while improving slice-specific measures [33]. However, extracting valuable scenes poses two main challenges: first, most data is not interesting from a scenario/safety perspective, and second, extensive data processing is required, impeding the utilization of large real-world datasets [35].

2) *Add perturbed data to training set (I.2)*: This strategy involves modifying the training set by introducing perturbations to enhance resistance against spurious features. It includes modifying the training data or augmenting it with perturbations (Figure 1). These perturbations can be incorporated into the historical states  $X$  [7], [16], [18], [23], [38] and optionally into the future states  $Y$  [14], [24], [39], denoted as  $\tilde{X}$  and  $\tilde{Y}$ . The objective is for the trained model to exhibit comparable behavior for perturbed and benign examples [40]. An important consideration for this strategy is that training the model on perturbations to improve its robustness often leads to degraded performance on clean, unperturbed data [41]. The balance between benign and perturbed variants becomes crucial for safety-critical tasks that autonomous vehicles handle [18].

3) *Change the model architecture (I.3)*: This strategy involves modifying the model architecture, focusing on the building blocks of the deep learning model (Figure 1). The building blocks of the prediction model are designed to

extract features from historical states, aiming to learn the motion dynamics of each agent and the social interactions between agents [16]. The model must describe how the past states of the agent influence its future movements or decisions, as well as how the state of one agent can influence the states or actions of other agents [16]. Considering the data-driven approach employed during training for real-world scenarios, it becomes apparent that prediction model can face challenges in extracting features from unseen domains [42]. To address this issue, modifying the model architecture can be essential in enhancing resistance to spurious features [42] and mitigating model overfitting [43]. An effective method for obtaining a deeper understanding of how these building blocks affect robustness is through ablation studies [44].

4) *Change the trained model (I.4)*: This strategy involves the refinement of the trained model, with the modification taking place post-training (Figure 1). Existing literature suggests that employing techniques such as model compression can enhance generalization performance on previously unseen data [45]. A method used for this purpose is pruning, which optimizes the model by systematically removing parameters [46]. The prunability of a model serves as a crucial indicator of its generalization capabilities. The greater the proportion of parameters that can be pruned, the more effectively the model is expected to generalize [45].

## IV. METHODS FOR EVALUATION AND IMPROVEMENT OF ROBUSTNESS

For robustness evaluation/improvement strategies *E.1*, *E.2*, *I.1*, *I.2*, and *I.3* (Figure 1, Section III), this section describes concrete methods. At the time of writing, no papers explicitly used strategy *I.4* to enhance the robustness of trajectory prediction models.

### A. Slicing techniques (*E.1*, *I.1*)

The utilization of conventional real-world data for training trajectory prediction models is attractive as it captures the authentic behaviors of road users [34]. However, a challenge

arises during data analysis: data distribution often does not favor varied scenes [47], safety-critical incidents [48] and geographical diversity [36], a phenomenon also known as the "Curse of Rarity" [34], [49]. The data typically contains minimal information about rare events, and acquiring sufficient information requires a substantial increase in the volume of data [49]. Slicing the data can be a useful method to extract valuable scenes from the dataset. These scenes are then segregated into a separate set, serving as a representation of valuable and rare, long-tailed events [34]. After these scenes are identified, the model is trained using a remediation strategy [34]. This process optimizes the benefits of existing data rather than necessitating the generation of new scenarios [34]. The data can be sliced using a search that is scenario or location specific.

1) *Scenario importance sampling*: To operate safely, an automated vehicle must anticipate the evolving environment around it. To achieve this, it is crucial to determine the most suitable prediction models for various situations [17]. Addressing this challenge involves employing scenario extraction, which requires a substantial amount of data to identify common and uncommon scenarios [35]. The initial step involves the predefinition of tags describing road user activities and their interactions with each other and the environment [17], [35], [50]. Subsequently, scenarios are extracted by searching for combinations of these predefined tags [35], [50].

To enhance the effectiveness of the sampling process, a more focused search can prove beneficial. An approach involves identifying and studying the most safety-critical scenarios within a given dataset [34]. The relevance of safety warrants emphasizing scenarios in which observed agents exhibit safety-critical or near-safety-critical behavior, as well as instances in which agents avoid such critical situations through subtle and proactive maneuvers. Sampling these scenarios can be accomplished by assigning scores based on their individual state and interaction features. Individual states are derived from the relative positional data of a trajectory, including speed, acceleration, and jerk. The interaction features utilize surrogate safety metrics such as time headway, time-to-collision, and deceleration rate to avoid a crash [51], [52].

2) *Geographic importance sampling*: Datasets can be captured in various locations worldwide. The dominance of the USA, Western Europe, and East Asia leads to a bias where autonomous driving systems are overfitted to environmental conditions typical of these regions [36]. To address this concern, one viable approach involves the development of region-specific models that are trained on data specific to the targeted operational area. Slicing the data according to geographic regions can enhance the model's ability to comprehend and adapt to the unique characteristics of specific locations.

### B. Perturbation techniques (E.2, I.2)

As previously mentioned, a method to evaluate and improve model robustness against unseen situations involves

collecting more data that includes the missing safety-critical scenarios, which is costly, time-consuming, and lacks scalability [38]. The necessity for additional data arises due to the prevalent issue of current prediction models being frequently overfitted on limited patterns within datasets and not capturing the long-tailed distribution of driving scenarios and behaviors [53]. To overcome this overfitting issue, a common practice is to use augmentation techniques to enrich existing datasets [54]. Another method involves partially perturbing the training data. These perturbations can take two primary forms: complementing perturbations, designed to enrich recorded data with previously unseen situations, and disruptive perturbations, which simulate complications like sensor noise [38]. For trajectory prediction techniques, these perturbations involve the removal of agents [14], [38], geometric transformations [39], adversarial attacks/training [16], [23], [30], [53], and contextual map perturbations [24], [55].

1) *Removal of agents*: This method involves enhancing the data by removing historical tracked locations of agents to prevent the model from relying solely on extrapolating historical data [25]. Two options can be considered for this task. The initial option is to completely remove historical information from a selected group of agents [25], optionally based on their social interactions [14]. Understanding how agents influence each other within the scene and labeling each agent according to its influence on interaction can be crucial for this purpose [14]. This can reveal whether a given agent is non-causal (an agent whose deletion does not alter the ground truth trajectory of a given target agent), causal (an agent capable of influencing the target trajectory of the agent), and static (an agent that remain motionless in the scene) [14].

Another option involves partially removing a selection of historical positions by employing methods such as dropouts [56] or late detection [38]. Using dropouts simulates perceptual loss, reflecting situations in the real world where the target might be lost in some timestamps or not tracked for a sufficient duration [56]. Late detection results in no historical data to exploit, considering only the most recent observation [38]. The significance of this improvement method lies in the need for the prediction model to handle imperfect trajectories robustly, rather than being constrained by fixed-duration tracking inputs [56].

2) *Geometric transformations*: Geometric transformations augment both input and ground truth labels. Rotating and mirroring data are methods commonly employed in various applications, such as segmentation problems [57]. These techniques can also be applied to trajectory prediction data. When each trajectory is subjected to random rotation, the directional bias can be reduced [58] and the network can be trained to be rotation-invariant [39]. Mirroring involves flipping the trajectory over the x-axis or y-axis with a certain probability. When no rotation is applied, mirroring could potentially augment the effects of random rotations [39]. Another geometric transformation method involves adding positional noise (typically Gaussian) to every point in the tra-

jectory. This approach is intended to enhance the robustness of the network to minor perturbations and inaccuracies [39]. Similarly, adding noise to the heading angle can enhance robustness against noisy heading angle measurements caused by faulty sensors or adverse weather conditions [38].

3) *Adversarial attacks/training:* In this class of methods, an adversary can inject any perturbation within some limit, called the attack model, into historical trajectories [16], [53]. These perturbations have the potential to significantly misguide the prediction of future trajectories, leading to unsafe planning [18]. Effectively training the model on adversarial attacks becomes crucial for improving the safety of autonomous vehicles in uncertain scenarios [18].

The primary objective of such attacks is to manipulate predictions at each time step or partially, with the intention of causing the autonomous vehicle plan to execute unsafe driving behaviors [16]. The technical challenges lie in devising an attack framework where the threat model synthesizes adversarial trajectories feasible subject to the physical constraints of the real vehicle and closely resembling nominal trajectories [16], [53]. Nominal trajectories are particularly important, as a significant perturbation in the trajectory history can obfuscate whether changes in future predictions result from the vulnerability of the prediction model or more fundamental perturbations to the meaning of the history [16]. Once the threat model is defined, adversarial training can be applied to the trajectory models to fortify their resilience against such attacks [37].

The challenges in devising effective adversarial attacks come from the fact that historical trajectories are often generated by upstream tracking pipelines and sparsely queried using discrete time intervals due to computational limitations. On the one hand, due to the sparse sampling of these trajectories, correctly estimating the parameters of the dynamics becomes challenging, making it difficult to determine the realism of a trajectory [16]. Reconstructing a realistic dense trajectory from a sampled trajectory in the dataset can mitigate this problem [16]. On the other hand, adversarial trajectories often exhibit changes in vehicle velocity or acceleration as a characteristic. A practical solution to mitigate the issue of unstable velocities or acceleration involves the application of trajectory smoothing. This approach proves beneficial in rendering the perturbation more realistic and aligning it with real-world scenarios [30]. Both approaches ensure that adversarial trajectories are physically feasible to reproduce in the context of driving a real car. Therefore, adversarial attacks can be viewed as a method to discover the most realistic yet worst cases for the prediction model [30].

4) *Contextual map perturbations:* Recent advances in trajectory prediction models have incorporated contextual map information [59]. The inclusion of this map helps prediction algorithms recognize specific scenarios, which contributes to the improved accuracy of predictions [24]. However, the exclusion of map information allows for the analysis of scenarios in which the road information is uncertain [38]. Furthermore, prediction models are susceptible to a specific type of attack targeting context maps [59]. These map-based

attacks involve the introduction of adversarial perturbations to the original map encoding [59]. To ensure that changes in adversarial maps are imperceptible, the generated adversarial perturbations need to be constrained [59].

Another map-based method employs a generative model for adversarial scene generation. In this approach, the generative model introduces realistic modifications to induce failures in prediction models when presented with an observed scene [24]. The generated scenarios must adhere to physical constraints and feasibility; otherwise, they cannot accurately represent potential real-world cases and can be subject to anomaly detection. Training the model on these attacks proves beneficial in improving robustness against contextual map perturbations [24], [38].

### C. Model architecture (I.3)

Modifying the model architecture can help improve various aspects of robustness in trajectory prediction. Prediction models typically consist of multiple building blocks that can be altered or expanded to improve predictive capabilities. These alterations can include a defense mechanism against perturbations [59], scene extrapolation [25], [43], [60], and comprehension of social interactions [8], [9], [42], [61]–[65].

1) *Defense mechanism:* Map-based attacks can significantly impact trajectory prediction models. Rather than training models on these perturbations, [59] advises to employ a defense mechanism. The defense strategy depends on how the trajectory prediction model extracts features from the map. Whether using images or node representation, defense mechanisms such as convolution-based filters or smoothers can be applied to resist these attacks [59].

2) *Scene extrapolation:* The diverse geometry and topology of map elements across different scenes pose a challenge for trajectory prediction models. Current works often neglect domain generalization, and their performance tends to degrade when transferred to a different dataset or scenario [60]. Extracting domain-independent features, such as motion patterns and social interactions, rather than memorizing domain-specific map features, is beneficial for enhancing robustness against domain shift problems [43]. Utilizing Frenet-based coordinate systems can mitigate trajectory coordinate variations across domains by exploiting local coordinates relative to lane centerlines [43]. Another issue related to scene extrapolation is the significant impact of captured speed differences [60]. Extreme speed cases, such as very low or high speeds, are infrequent in the training data, causing the model to struggle when tested under such conditions. Including a velocity refinement module in the model can help address this challenge.

3) *Social interaction:* Vehicle trajectories encapsulate detailed semantic and behavioral information, describing the actions and intentions of road users. Among these, the understanding of social behavior and learning interactions between individuals are core challenges in trajectory prediction. Using social pooling [65], global scene information [9], and spatio-temporal features [42], [62], [64] can help models capture interaction between road users.

TABLE I

SURVEYED PAPERS AND THEIR UTILIZATION OF PERFORMANCE MEASURES (SECTION II-B), ROBUSTNESS STRATEGIES (SECTION III), AND ROBUSTNESS METHODS (SECTION IV)

Performance measures	Zamboni et al. (2020) [39]	Wang et al. (2021) [60]	Sánchez et al. (2022) [17]	Bahari et al. (2022) [23]	Saadatnejad et al. (2022) [18]	Cao et al. (2022) [24]	Roelofs et al. (2022) [16]	Zhang et al. (2022) [14]	Stoler et al. (2022) [30]	Sánchez et al. (2023) [34]	Zheng et al. (2023) [38]	Jiao et al. (2023) [59]	Tan et al. (2023) [53]	Vishnu et al. (2023) [43]	Ye et al. (2023) [66]	Hallgarten et al. (2023) [25]
Average Displacement Error	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Final Displacement Error	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Miss Rate					✓		✓								✓	
Collision Rate						✓				✓						
Off-Road Rate																✓
Mean Inter Endpoint Distance																✓
Robustness strategies																
Test on sliced data (E.1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Test on perturbed data (E.2)						✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Slice the training set differently (I.1)	✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Add perturbed data (I.2)	✓	✓														✓
Change the model architecture (I.3)																✓
Change the trained model (I.4)																✓
Robustness methods																
Scenario importance sampling					✓					✓					✓	✓
Geographic importance sampling						✓					✓					✓
Removal of agents	✓					✓				✓						
Geometric transformation							✓				✓					
Adversarial attacks/training								✓				✓				
Contextual map perturbations									✓				✓			
Defense mechanism										✓						
Scene extrapolation	✓										✓					
Social interaction												✓				

## V. DISCUSSION

Based on the survey, we identified 17 research papers that directly address the topics of evaluating and/or improving the robustness of trajectory prediction (Table I). From this selection, we identified four general developments.

### A. Analysis of recent developments

First, training and testing models on perturbations is the most common strategy to robustness in trajectory prediction. Perturbations are valuable tools for evaluating the performance of trajectory prediction models because they can be easily constructed and applied to discrete input data. This makes it easier to assess the robustness of the model and identify potential weaknesses.

Second, the performance in benign situations tends to degrade after including perturbations in model training [14], [16], [30], [38], [53]. Therefore, it is crucial to consistently evaluate the retrained model in benign situations to ascertain if it maintains the same performance as before [38]; mixing of perturbed scenarios with benign ones during training can help mitigate this issue [53], [67].

Third, we found that papers related to enhancing robustness in trajectory prediction often overlook the aspect of modifying the trained model (Table I). This gap is evident

not only in the discussed papers but also appears to be a general trend within the trajectory prediction literature.

Fourth, most existing work uses simplistic performance measures such as Average Displacement Error and Final Displacement Error. However, the relevance of these measures remains limited, particularly in evaluating adversarial attacks. These measures are designed for benign settings and are insufficient to demonstrate the implications for autonomous systems under attack [16]. For instance, a large Average Displacement Error in prediction does not directly entail concrete consequences, such as a collision [16]. Furthermore, when analyzing measures using only endpoints such as the Final Displacement Error, significant details regarding the trajectory shape, curvature, and point-to-point transitions are overlooked. Additionally, the measures only consider pointwise predictions whereas distributions over trajectory predictions could potentially provide a better understanding of the non-deterministic nature of human behavior [68].

In summary, the current research primarily focuses on evaluating and improving robustness of trajectory prediction through data perturbation using mostly ADE and FDE as performance measures (Table I). Adversarial attacks are extensively used to generate perturbations, while adversarial training aims to enhance robustness.

## B. Potential research directions

Following a comprehensive review of evaluation and enhancement strategies, we identified five key research gaps that we believe should be addressed in future literature on robustness of trajectory prediction.

First, upon analyzing modifications that can enhance the robustness of trajectory prediction methods, it becomes apparent that there is a lack of in-depth analysis regarding the strategy of modifying the trained trajectory prediction model. Research focused on evaluating the robustness of convolutional neural networks has demonstrated that post-training techniques, such as pruning, can significantly improve model robustness [45]. Further research on this topic may reveal methods for enhancing robustness of trajectory prediction models.

Second, the papers mainly focus on enhancing robustness based on a chosen measure while evaluating the effect of the enhancements on the same measure. However, improving robustness for specific measures may conflict with one another. Investigating the influence of improving robustness with one measure over other measures can provide insights into how the measures correlate.

Third, after applying adversarial attacks during training, pre-processing these attacks during both training and testing phases can be essential to enhance the robustness of the model. Previous studies have indicated that managing the trade-off between clean and perturbed data [53], exploring defense strategies [59], and smoothing the train and test data [30], [53] can significantly influence the resilience of the model against adversarial examples. A comprehensive examination of all pre-processing possibilities for adversarial data in trajectory prediction models can be conducted.

Fourth, when analyzing papers related to adversarial training in trajectory prediction [16], [30], [53], [66], the focus is usually on evaluating the performance of the same models. To assess the effectiveness of adversarial training strategies across diverse algorithms, a direct comparison involving different prediction models and datasets can be undertaken. This comparative analysis can help investigate the generalizability of adversarial training for trajectory prediction models.

Fifth, the majority of papers present only their method, and do not explore how it combines with existing robustness methods. As a result, the interactions of various robustness methods remain unclear. Designing a unified framework that allows for the combination and comparison of these methods will facilitate the exploration of their synergies and the identification of underexplored methods.

## VI. CONCLUSION

This survey presents a framework that categorizes literature related to the evaluation and improvement of robustness in trajectory prediction. We found that existing literature in the field mainly focuses on several mainstream general strategies and a number of common methods, as well as shares some key limitations. In particular, we observed the widespread use of perturbations for evaluating and improving robustness, the limited adoption of post-training techniques

to enhance robustness, and the lack of diversity in evaluation measures to validate prediction performance. We believe that addressing the identified challenges will advance the development of robust and reliable trajectory prediction models for autonomous vehicles.

## REFERENCES

- [1] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, “A survey on trajectory-prediction methods for autonomous driving,” in *IEEE Trans. on Intell. Veh.*, 2022.
- [2] V. Bhatilya and N. Kumar, “Machine learning for autonomous vehicle’s trajectory prediction: A comprehensive survey, challenges, and future research directions,” in *Veh. Commun.*, 2024.
- [3] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuScenes: A multimodal dataset for autonomous driving,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2020.
- [4] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, “Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset,” in *IEEE/CVF Int. Conf. on Comput. Vis.*, 2021.
- [5] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, “Argoverse: 3d tracking and forecasting with rich maps,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2019.
- [6] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, “INTERACTION dataset: An INTERnational, adversarial and cooperative moTION dataset in interactive driving scenarios with semantic maps,” 2019. arXiv:1910.03088.
- [7] R. Jiao, X. Liu, B. Zheng, D. Liang, and Q. Zhu, “TAE: A semi-supervised controllable behavior-aware trajectory generator and predictor,” in *IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, 2022.
- [8] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, “AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting,” in *IEEE/CVF Int. Conf. on Comput. Vis.*, 2021.
- [9] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data,” in *Eur. Conf. on Comput. Vis.*, 2020.
- [10] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, D. Weiss, B. Sapp, Z. Chen, and J. Shlens, “Scene transformer: A unified architecture for predicting multiple agent trajectories,” 2022. arXiv:2106.08417.
- [11] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, “Multimodal motion prediction with stacked transformers,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2021.
- [12] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, “CoverNet: Multimodal behavior prediction using trajectory sets,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2020.
- [13] P. Xiong, S. Buffet, S. Iqbal, P. Lamontagne, M. Mamun, and H. Molyneaux, “Towards a robust and trustworthy machine learning system development: An engineering perspective,” in *J. Inf. Secur. Appl.*, 2022.
- [14] R. Roelofs, L. Sun, B. Caine, K. S. Refaat, B. Sapp, S. Ettinger, and W. Chai, “CausalAgents: A robustness benchmark for motion forecasting using causal relationships,” 2022. arXiv:2207.03586.
- [15] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, “Deep learning-based vehicle behavior prediction for autonomous driving applications: A review,” in *IEEE Trans. on Intell. Transp. Syst.*, 2022.
- [16] Y. Cao, C. Xiao, A. Anandkumar, D. Xu, and M. Pavone, “AdvDO: Realistic adversarial attacks for trajectory prediction,” in *Eur. Conf. on Comput. Vis.*, 2022.
- [17] M. Muñoz Sánchez, J. Elfring, E. Silvas, and R. van de Molengraft, “Scenario-based evaluation of prediction models for automated vehicles,” in *IEEE 25th Int. Conf. on Intell. Transp. Syst.*, 2022.
- [18] Y. Cao, D. Xu, X. Weng, Z. Mao, A. Anandkumar, C. Xiao, and M. Pavone, “Robust trajectory prediction against adversarial attacks,” in *PMLR The 6th Conf. on Robot Learn.*, 2023.

[19] G. Ian, Y. Bengio, and C. Aaron, “Deep learning,” 2016. <https://www.deeplearningbook.org/>.

[20] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in *IEEE 12th Int. Conf. on Comput. Vis.*, 2009.

[21] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human trajectory prediction in crowded spaces,” in *IEEE Conf. on Comput. Vis. Pattern Recognit.*, 2016.

[22] J. Schmidt, T. Monninger, J. Jordan, and K. Dietmayer, “LMR: Lane distance-based metric for trajectory prediction,” in *IEEE Intell. Veh. Symp.*, 2023.

[23] S. Saadatnejad, M. Bahari, P. Khorsandi, M. Saneian, S.-M. Moosavi-Dezfooli, and A. Alahi, “Are socially-aware trajectory prediction models really socially-aware?,” in *Transp. Research Part C: Emerg. Technol.*, 2022.

[24] M. Bahari, S. Saadatnejad, A. Rahimi, M. Shaverdikondori, A. H. Shahidzadeh, S.-M. Moosavi-Dezfooli, and A. Alahi, “Vehicle trajectory prediction works, but not everywhere,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2022.

[25] M. Hallgarten, I. Kisa, M. Stoll, and A. Zell, “Stay on track: A frenet wrapper to overcome off-road trajectories in vehicle motion prediction,” 2023. arxiv:2306.00605.

[26] A. Tocchetti, L. Corti, A. Balayn, M. Yurrita, P. Lippmann, M. Brambilla, and J. Yang, “A.i. robustness: a human-centered perspective on technological challenges and opportunities,” 2022. arXiv:2210.08906.

[27] N. Drenkow, N. Sani, I. Shpitser, and M. Unberath, “A systematic review of robustness in deep learning for computer vision: Mind the gap?,” 2022. arXiv:2112.00639.

[28] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2015. arXiv:1412.6572.

[29] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, “Robust physical-world attacks on deep learning visual classification,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2018.

[30] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, “On adversarial robustness of trajectory prediction for autonomous vehicles,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2022.

[31] Z. Liu, Y. Xu, Y. Xu, Q. Qian, H. Li, R. Jin, X. Ji, and A. B. Chan, “An empirical study on distribution shift robustness from the perspective of pre-training and data augmentation,” 2022. arXiv:2205.12753.

[32] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balasubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. A. Earnshaw, I. S. Haque, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang, “WILDS: A benchmark of in-the-wild distribution shifts,” 2021. arXiv:2012.07421.

[33] V. Chen, S. Wu, A. J. Ratner, J. Weng, and C. Ré, “Slice-based learning: A programming model for residual learning in critical data slices,” in *Adv. Neural Inf. Process. Syst.*, 2019.

[34] B. Stoler, I. Navarro, M. Jana, S. Hwang, J. Francis, and J. Oh, “SafeShift: Safety-informed distribution shifts for robust trajectory prediction in autonomous driving,” 2024. arXiv:2309.08889.

[35] D. Guo, M. Muñoz Sánchez, E. de Gelder, and T. P. J. van der Sande, “Scenario extraction from a large real-world dataset for the assessment of automated vehicles,” 2023. arXiv:2306.07815.

[36] M. Liu, E. Yurtsever, X. Zhou, J. Fossaert, Y. Cui, B. L. Zagar, and A. C. Knoll, “A survey on autonomous driving datasets: Data statistic, annotation, and outlook,” 2024. arXiv:2401.01454.

[37] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” 2019. arXiv:1706.06083.

[38] M. Muñoz Sánchez, E. Silvas, J. Elfring, and R. van de Molengraft, “Robustness benchmark of road user trajectory prediction models for automated driving,” in *22nd IFAC World Congr.*, 2023.

[39] S. Zamboni, Z. T. Kefato, S. Girdzijauskas, C. Norén, and L. Dal Col, “Pedestrian trajectory prediction with convolutional neural networks,” in *Pattern Recognit.*, 2022.

[40] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, “Recent advances in adversarial training for adversarial robustness,” in *Int. Jt. Conf. on Artif. Intell.*, 2021.

[41] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, and M. Jordan, “Theoretically principled trade-off between robustness and accuracy,” in *PMLR 36th Int. Conf. on Mach. Learn.*, 2019.

[42] C. Vishnu, V. Abhinav, D. Roy, C. Mohan, and C. Babu, “Improving multi-agent trajectory prediction using traffic states on interactive driving scenarios,” in *IEEE Robotics Autom. Lett.*, 2023.

[43] L. Ye, Z. Zhou, and J. Wang, “Improving the generalizability of trajectory prediction models with frenet-based domain normalization,” in *IEEE Int. Conf. on Robotics Autom.*, 2023.

[44] R. Meyes, M. Lu, C. W. de Puiseau, and T. Meisen, “Ablation studies in artificial neural networks,” 2019. arXiv:1901.08644.

[45] L. Kuhn, C. Lyle, A. N. Gomez, J. Rothfuss, and Y. Gal, “Robustness to pruning predicts generalization in deep neural networks,” 2021. arXiv:2103.06002.

[46] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, “What is the state of neural network pruning?,” 2020. arXiv:2003.03033.

[47] B. Ivanovic, G. Song, I. Gilitschenski, and M. Pavone, “trajdata: A unified interface to multiple human trajectory datasets,” 2023. arXiv:2307.13924.

[48] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, “A survey on safety-critical driving scenario generation—a methodological perspective,” in *IEEE Trans. on Intell. Transp. Syst.*, 2023.

[49] H. X. Liu and S. Feng, “‘curse of rarity’ for autonomous vehicles,” 2022. arXiv:2207.02749.

[50] E. de Gelder, J. Manders, C. Grappiolo, J.-P. Paardekooper, O. O. d. Camp, and B. De Schutter, “Real-world scenario mining for the assessment of automated vehicles,” in *IEEE 23rd Int. Conf. on Intell. Transp. Syst.*, 2020.

[51] C. Glasmacher, R. Krajewski, and L. Eckstein, “An automated analysis framework for trajectory datasets,” 2022. arXiv:2202.07438.

[52] K. Vogel, “A comparison of headway and time to collision as safety indicators,” in *Accident Anal. & Prev.*, 2003.

[53] R. Jiao, X. Liu, T. Sato, Q. A. Chen, and Q. Zhu, “Semi-supervised semantics-guided adversarial training for robust trajectory prediction,” in *IEEE/CVF Int. Conf. on Comput. Vis.*, 2023.

[54] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” in *J. Big Data*, 2019.

[55] L. Mi, H. Zhao, C. Nash, X. Jin, J. Gao, C. Sun, C. Schmid, N. Shavit, Y. Chai, and D. Anguelov, “HDMapGen: A hierarchical graph generative model of high definition maps,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2021.

[56] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, “Learning to predict vehicle trajectories with model-based planning,” in *PMLR 5th Conf. on Robot Learn.*, 2022.

[57] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015. arXiv:1505.04597.

[58] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, “What the constant velocity model can teach us about pedestrian motion prediction,” 2020. arXiv:1903.07933.

[59] Z. Zheng, X. Ying, Z. Yao, and M. C. Chuah, “Robustness of trajectory prediction models under map-based attacks,” in *IEEE/CVF Winter Conf. on Appl. Comput. Vis.*, 2023.

[60] Z. Wang, J. Guo, H. Zhang, R. Wan, J. Zhang, and J. Pu, “Bridging the gap: Improving domain generalization in trajectory prediction,” in *IEEE Trans. on Intell. Veh.*, 2023.

[61] J. Sun, Q. Jiang, and C. Lu, “Recursive social behavior graph for trajectory prediction,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2020.

[62] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, “Multi-agent tensor fusion for contextual trajectory prediction,” in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, 2019.

[63] J. Wang, C. Guo, M. Guo, and J. Chen, “Jointly learning agent and lane information for multimodal trajectory prediction,” in *2022 26th Int. Conf. on Pattern Recognit.*, 2022.

[64] Y. Zhu, D. Ren, M. Fan, D. Qian, X. Li, and H. Xia, “Robust trajectory forecasting for multiple intelligent agents in dynamic scene,” 2020. arXiv:2005.13133.

[65] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, “It is not the journey but the destination: Endpoint conditioned trajectory prediction,” in *Eur. Conf. on Comput. Vis.*, 2020.

[66] K. Tan, J. Wang, and Y. Kantaros, “Targeted adversarial attacks against neural network trajectory predictors,” 2022. arXiv:2212.04138.

[67] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” 2018. arXiv:1710.09412.

[68] B. Ivanovic and M. Pavone, “The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs,” in *IEEE/CVF Int. Conf. on Comput. Vis.*, 2019.