SAE INTERNATIONAL

ARTICLE INFO
Article ID: 12-08-01-0009

doi:10.4271/12-08-01-0009

# Toward Explainability in Urban Motion Prediction—Survey and Outlook

***Ilma Okanovic,[1] Michael Stolz,[2] and Bernhard Hillbrand[2]***

[1]*Virtual Vehicle Research GmbH, Autonomous Systems, Austria*
[2]*Virtual Vehicle Research GmbH, Austria*

## Abstract

With the influx of artificial intelligence (AI) models aiding the development of autonomous driving (AD), it has become increasingly important to analyze and categorize aspects of their operation. In conjunction with the high predictive power innate to AI solutions, due to the safety requirements inherent to automotive systems and the demands for transparency imposed by legislature, there is a natural demand for explainable and predictable models. In this work, we explore the various strategies that reveal the inner workings of these models at various component levels, focusing on those adapted at the modeling stage. Specifically, we highlight and review the use of explainability in state-of-the-art AI-based scenario understanding and motion prediction methods, which represent an integral part of any AD system. We break the discussion down across three key axes that are inherent to any AI solution: the data, the model architecture, and the loss optimization. For each of the axes, we outline the general methodologies for introducing explainability, and reference and review some practical realizations for each methodology. We conclude the article by identifying several strategies that we believe are yet to be fully explored, such as physics-inspired machine learning methods, neural network pretraining, graph neural networks designed using domain-specific priors, and end-to-end trainable networks based on differentiable kinematic models.

## History

Received: 02 Feb 2024
Revised: 18 Apr 2024
Accepted: 30 Jul 2024
e-Available: 24 Aug 2024

## Keywords

Scenario understanding, Motion prediction, Explainability, Transparency, Autonomous driving, Autonomous vehicles

## Citation

Okanovic, I., Stolz, M., and Hillbrand, B., "Toward Explainability in Urban Motion Prediction—Survey and Outlook," *SAE Int. J. of CAV* 8(1):109–123, 2025, doi:10.4271/12-08-01-0009.

ISSN: 2574-0741
e-ISSN: 2574-075X

This article is part of a focus issue on the Safety of AI-Based Systems.





# Introduction

Despite its largely unexplored potential, artificial intelligence (AI) has already proven its importance in furthering the development of autonomous driving (AD). Its use has been shown to bring about benefits ranging from increased road traffic safety, improvements in performance and efficiency, positive environmental effects, and increased accessibility and affordability. These benefits have been demonstrated mainly through the deployment of the so-called narrow AI systems that can perform "one or a few domain-specific tasks" [1]. One such task expected to be greatly improved by the use of AI is the task of scenario understanding and motion prediction. Solving it means accounting for the intricacies and uncertainties inherent in road infrastructure, the motion of the traffic participants, and the interactions that can occur between the traffic participants and their environment. Given the highly probabilistic nature of the task, AI emerges as an ideal approach, with its ability to capture and learn from heterogeneous traffic data sources. Accomplishing scenario understanding and motion prediction in complex environments can certainly contribute to the overall traffic safety, but it should not come at the cost of reducing the safety of the society at large.

Envisioned as the world's first horizontal regulatory framework for AI development, market placement, and use, the AI Act is currently in its final stages of adoption with enforcement expectations set in 2024. The 2021 draft proposal [2] of the European Commission laid out the goal of establishing a technology-neutral definition of AI systems and the adoption of a "risk-based" classification approach. The level of risk would correspond to the level of legal requirements imposed on the product or a system containing AI systems. These requirements are not conceived in isolation and will complement and strengthen existing and forthcoming regulations, particularly those on transparency, technical documentation, and record-keeping. Through delegated acts, the AI Act will request changes within sector-specific safety legislation, including the sector of the automobile industry, thereby influencing the regulatory landscape surrounding autonomous vehicles (AVs).

The incentive to extend existing sector-specific legislation also stems from the proposal put forth by the European Automobile Manufacturers Association (ACEA) in its 2020 position paper on AI in the automobile industry [1]. Reflecting the views of Europe's sixteen major automobile manufacturers, the document provided the EU Commission with recommendations to effectively balance the risks and opportunities the use of AI systems can bring. However, to achieve legal certainty, ACEA also emphasized the need for a regulatory approach that takes into account the level of vehicle automation (in the context of SAE levels [3]), rather than the technology used. The classification and risk assessment should, therefore, be unique to the application of the AI system and its intended use. Such an approach ensures that narrow AI systems utilized in AVs will not be at risk of misclassification and subsequent overregulation. Stringent ex ante conformity assessment procedures (e.g., Type Approval EU Regulation 2081/858), ex post surveillance mechanisms (e.g., conformity of production, in-service compliance, etc.), and regulations governing functionality updates (e.g., the UN Regulation 156 for automotive software updates) are already in force. Additionally, the automotive industry relies on the standards issued by major standardization bodies, exemplified by the Safety of the Intended Functionality (SOTIF) 21448 standard [4], which concerns functionalities that require situational awareness for safety purposes. For a more comprehensive overview of standards concerning the human safety and information exchange in AD systems, see [5]. Whether through standards, codes of conduct, or best practices, automobile manufacturers are committed to achieving another goal: fostering greater trust in transport automation. For that reason, ACEA supports initiatives that will be based on principles such as human centricity, responsibility, transparency, explainability, and privacy.

Parallel with safety validation, automobile manufacturers endorsed the Ethics Guidelines of the High-Level Expert Group on Artificial Intelligence (AI HLEG) [6], which delineates the aforementioned principles characterizing Trustworthy AI. Guided by the contributions of AI HLEG, the focus of this article will be placed on the principle of explainability, defined under the transparency requirement as "the ability to explain both the technical processes of an AI system and the related human decisions." To complement their definition, we will also reference two recent definitions of AI explainability established by the ISO 22989 [7] and ISO 24028 standards [8]. According to the ISO 22989 standard on AI concepts and terminology, AI explainability is defined as a system property to express important "factors influencing a decision [...] in a way that humans can understand." In terms of what is human-understandable, the ISO 24028 standard on the trustworthiness of AI provides a detailed clarification, recognizing that the requirements of different stakeholders will be reflected in different approaches to explainability. These approaches can be analyzed depending on the development stage at which explanations can be generated (pre-modeling, modeling, and post-modeling), the scope of explanations (local and global), and their granularity. This article seeks to explore global approaches capable of revealing the inner workings of the model at various component levels, focusing on those adapted at the modeling stage. However, one final distinction has to be made. The review will focus on approaches that make explainability inherent to the functioning of the AI model and not the subtask it has to solve. Latter approaches are out of the scope of this article, but interested readers can find more information on their use in AD in [5], [9], and [10].

The principle of explainability will provide a basis upon which we will survey the state-of-the-art methods proposed for the task of scenario understanding and motion prediction. Incorporating various heterogeneous sources of scenario information would suggest the use of data-driven methods. However, there remains a wealth of expert knowledge that can still be utilized in the model development for which explainable approaches can be the appropriate choice. Furthermore, the significant impact these solutions can have on the safety of individuals may lead to a push toward remodeling the existing solutions into their more explainable variants. Whether motivated by the complexity or criticality of the task, solutions that satisfy the adopted definition of explainability will be reviewed in the following sections.

## State of the Art

Addressing all the challenges present in solving the scenario understanding and motion prediction task has led to the definition of different survey approaches. Some surveys focus on solutions that tackle one or a few specific intricacies of these tasks, such as predicting the motion of a single class of traffic participants. Others argue for a classification and comparison based on the methodology used. This approach requires authors to recognize key methodological differences between proposed solutions resulting in classifications at varying levels of detail. Additionally, scenario understanding and motion prediction can be analyzed in conjunction with other tasks, whether by combining them with the upstream task of perception or the downstream tasks of decision-making and planning. However, past surveys indicate that more often they are performed by considering multiple criteria in parallel, some of which have been previously mentioned.

One of the initial efforts to classify methods for predicting vehicular motion was undertaken by [11]. They employed a classification approach that identified factors influencing motion prediction, resulting in a three-fold classification consisting of physics-based, maneuver-based, and interaction-aware classes. Although limited advances were made in the interaction-aware class at the time, a more recent review [12] focused exclusively on this class. Classification by [12], encompassing intention-aware in addition to interaction-aware class, went beyond methodology considerations and included algorithmic and dataset dimensions. Additionally, the authors refined the classification method by distinguishing between model-based and data-driven subclasses and adding layers based on the method's input and output types.

Similarly, [13] conducted a review of vehicular motion prediction along the same dimensions, albeit with variations in the hierarchy bringing the prediction methodology, input, and output type to the top, and including the interactions as an input type. This permutation of classification criteria was observed in another review of pedestrian motion prediction methods [14]. Both [13, 14] renamed the model-based subclass into a physics-based class, while the data-driven subclass was further divided into multiple classes. This enabled a comparison of the results along the methodology dimension between these two review papers, with some differences only observable along the contextual cues [14] or contextual factors [13] dimension. Notably, greater importance is often given to these contextual cues in reviews of pedestrian motion prediction, occasionally even forming the basis of the classification approach. One such review [15], an extension of the [16], classified the methods based on the inputs provided considering even the psychological knowledge of human behavior.

While [14] focused on pedestrian prediction methods, their intention in defining the classification approach was for it to be independent of the traffic participant's type. Their proposal was adopted by [17], whose review included motion prediction methods specific to or generalized for different traffic participants. Based on the motion modeling and causes, their taxonomy divides the methods into physics-based (defining a dynamic model based on Newton's laws), pattern-based (learning motion patterns from data), and planning-based (reasoning on motion intent). Taking inspiration from the classification definition of both [11, 14], another review paper including different types of traffic participants was proposed [18]. Interestingly, the concept of contextual cues introduced by [14] was unified and renamed situational awareness, a term we will also reference as part of the prediction task. For defining their classification approach, the authors also credited the classification proposed by [19]. Both reviews explored methods along similar dimensions, but the [19] review primarily focused on deep learning (DL) methods. Complementing their work on DL methods, a paper by [20] was published providing an overview within a broader scope of the AV pipeline. With perception and planning recognized as key components of the AV pipeline, the situational awareness and prediction task became an integral part of their implementation. Staying within the broader scope of AV tasks, but delineating motion prediction as a distinct task, was adopted by the review [21]. In addition to proposing the classification based on multiple method features, the authors introduced keywords that allowed the grouping of the methods specific to the motion prediction task and those used for other modeling tasks, such as state and intention estimation.

Among the papers presented, only [17, 18] have provided an introduction to explainability in their classification results. Reference [17] identified explainability as one of the features that was then generalized across exemplary methods representing each of the defined classes. Additionally, their lower-level methodology classification allowed for a summary of methods to which different subclasses could be assigned simultaneously

and regardless of the belonging superclasses. While not explicitly referencing the term explainability (or, often interchangeably used, interpretability), the classification approach by [18] also provided a summary of the reviewed methods combining different subclasses. These reviews can serve as a foundation for exploring explainability at the method level since certain classes can be determined to be fully explainable or unexplainable. However, the method-level analysis might not be sufficient, as explainability could be introduced at an even lower level. Motivated to introduce explainability to existing solutions, researchers are defining new techniques that can gradually change the previously identified characteristics of the method classes.

The pioneering paper [22] proposed to further analyze machine learning solutions based on three biases: observational, learning, and inductive. These biases can be incorporated through the data, loss, or architecture of the model used, respectively. The bias-based approach to the classification is already being utilized and extended in ML and RL review papers, such as [23, 24]. While the survey by [24] relied on biases to offer an additional perspective to physics-informed RL (PIRL), [23] went further, decomposing the three bias-based classes into five, recognizing model initialization and residual modeling as two additional classes. The trend of defining the taxonomy based on the model components, where priors can be introduced, continued in more recent surveys. Therefore, parallels can be drawn between the physics-informed data enhancement, optimization, and architecture design of [25] and the original biases proposed by [22]. A similar conclusion can be made for the data, model, and objective classes of [26] except for added optimizer and inference classes. Notably, despite certain differences between proposed taxonomies, [23, 25, 26] all argue their usability on a broad scope of applications. It is due to the broad scope considered in these papers that only a limited number of motion prediction methods were reviewed. Restricting the scope of the application on motion prediction in AD introduces priors that could lead to application-specific explainable solutions. However, among available surveys on explainability in AD, only those focusing on approaches that generate explanations can be found [5, 9, 10]. To our knowledge, no survey has specifically focused on the aspect of explainability as described in the Introduction section. Therefore, motivated by past taxonomy proposals, our survey will narrow down on the scenario understanding and motion prediction task defined within the context of AD and identify specific groups of explainable approaches integrated into key model components. For a clear overview of the referenced surveys, Table 1 offers a comparison based on their scope, classification dimensions, principal methodologies, and additional contributions. Note that the scope of the survey determines the contents of the traffic participant class, output abstraction, and interactions considered columns.

# Problem Definition

For the sake of technical clarity, a definition of motion prediction is a necessary prerequisite. In order to respect a large variety of use cases, we give a broadly applicable definition and later point out the common assumptions and special cases that occur in practice. The first step toward the general definition starts with the definition of motion itself.

In the broadest sense, motion can be described as the change in the state of a system through time. However, this definition is needlessly general and offers no meaningful distinction from similar concepts (e.g., state evolution in dynamical systems). With this in mind, we impose an additional constraint on the definition where the state is only informed by the mechanics (kinematics and dynamics) of the system, i.e., the position, velocity, acceleration, forces, torques, and so on. These variables become sensible only with respect to a reference coordinate frame, so a global frame $\mathcal{O}$ must be specified. This frame is usually rooted at the position of the observer, or, in other words, at the location of the sensor. The state of the system at any time $t$ is expressed via the state vector $\boldsymbol{S}(t)$, which is formed by aggregating all the variables of interest. The function $\boldsymbol{S}: t \to \boldsymbol{S}(t)$ is called the state function of the system. Given the state function, a trajectory may also be defined: Given two timestamps $t_s \leq t_f$, the set

$$\phi_{[t_s,t_f]} := \left\{ \left( t, \boldsymbol{S}(t) \right) : t_s \leq t \leq t_f \right\} \qquad \text{Eq. (1)}$$

is the trajectory of the system from $(t_s, S(t_s))$ to $(t_f, S(t_f))$. In digital systems, signals are most often discrete and finite, so this definition is further relaxed. The set

$$\phi_{\mathcal{T}_{[t_s,t_f]}} := \left\{ \left( t, \boldsymbol{S}(t) \right) : t \in \mathcal{T}_{[t_s,t_f]} \right\} \qquad \text{Eq. (2)}$$

where $\mathcal{T}_{[t_s,t_f]}$ consists of timestamps in $[t_s, t_f]$ and additionally $t_s$, $t_f \in \mathcal{T}$, is considered a trajectory.

With motion itself defined, the task may also be defined. At time $t$, we are given the trajectory $\phi_{\mathcal{T}_{[t-\Delta\tau,t]}}$ and contextual problem-specific information $\mathcal{I}(t)$. The estimator should output a predicted trajectory $\phi_{\mathcal{T}_{[t,t+\Delta T]}}$. Both $\Delta\tau$ and $\Delta T$ are configurable window size parameters, and $\mathcal{I}(t)$ can be anything relevant to the problem. Note that sometimes a distinction is drawn between motion and trajectory estimation. In this interpretation, the trajectory consists only of the position over time, whereas motion includes other relevant kinematic and dynamic quantities. However, according to our definition, these quantities can be introduced into the definition of the state. Hence, we do not make a distinction between the two problems.

Finally, note that the most common formulation of the problem uses a uniform sampling scheme to discretize

**TABLE 1** Overview of referenced surveys proposing taxonomies dependent on the scope, classification approach, and additional contributions.

| Survey | Year | Traffic participants | Output abstraction | Interactions considered | Classification | Dimensions | Methods | Other contributions |
|---|---|---|---|---|---|---|---|---|
| [11] | 2014 | Vehicles | Motion | Yes | Novel | Methods | All | Risk evaluation |
| [12] | 2022 | All | Motion | Yes | Follows [11] | Methods, input, output, algorithms, datasets | All | Datasets, evaluation metrics, future directions |
| [19] | 2020 | Vehicles | Motion | Yes | Novel | Method, input, output | DL | Evaluation metrics, future directions |
| [14] | 2020 | Pedestrians | Trajectory | Yes | Novel | Methods, contextual cues | All | Datasets, evaluation metrics, future directions |
| [17] | 2022 | All | Motion | Yes | Follows [14] | Methods | All | Discussion |
| [18] | 2021 | Pedestrians and vehicles | Motion | Yes | Follows [11, 14, 19] | Modeling approach, output type, situational awareness | All | Future direction |
| [13] | 2022 | Vehicles | Trajectory | Yes | Novel | Method, input (contextual factors), output | All | Datasets, evaluation metrics, future directions |
| [16] | 2016 | Pedestrians | Probability grid | Yes | Novel | Input | All | Evaluation metrics |
| [15] | 2018 | Pedestrians | Motion | Yes | Follows [16] | Input | All | Datasets, future directions |
| [21] | 2020 | Vehicles | Trajectory | Yes | Novel | Architecture, training, theory, scope, evaluation | All | Higher-level classification |
| [20] | 2020 | AV | | Yes | Novel | Methods | DL | Discussion, future directions |
| [24] | 2023 | AV | | Yes | Novel | Physics information, PIRL methods, RL pipeline | RL | Benchmark, future directions |
| [22] | 2021 | General | | | Novel | Data, loss, model | PIML | Software, future directions, datasets |
| [23] | 2020 | General | | | Novel | Loss, initialization, architecture design, residual modeling, hybrid PML models | PIML | Discussion |
| [25] | 2022 | General | | | Novel | Architecture design, model fusion, optimization | PIML | Future directions |
| [26] | 2022 | General | | | Novel | Data, objective, optimizer, model, inference | PIML | Future directions |



the time window. In this case, we may also represent the trajectory as an indexed list, namely

$$\phi_{T_{[t_s,t_f]}} := \left(S^1, S^2, \ldots, S^n\right) \quad \text{Eq. (3)}$$

$$\text{with} \quad S^k := S\left(t_s + \frac{k-1}{n-1}\left(t_f - t_s\right)\right) \quad \text{Eq. (4)}$$

# Explainability of Motion Prediction Methods

Given the problem definition of motion prediction, the methods proposed for its solution will be analyzed leveraging the ideas from the referenced surveys on physics-informed machine learning (PIML) [22, 23, 25, 26]. Building on their classification proposal, the following subsections will explore the methods along three dimensions: data, model design, and model optimization. Providing a direct connection between the user and the model, the choice of input and output data serves as the simplest method of integrating domain knowledge into the model. With an explainable input and output format, the model's objective can be abstracted to a higher level but at the potential cost of obscuring its complexity. Such an abstraction could still mask a "black-box" model necessitating the inclusion of the domain knowledge within the model itself. Alternatively, both model design and optimization present an alternative to modifying the model by directly incorporating known properties of a considered problem or a subproblem. Relying solely on the data,

the model would be expected to learn the underlying physical law or property connecting its input and output. However, approximations of certain physical systems are already well-established and their mathematical description can be translated into loss constraints or even the structure of the model. Depending on the prior knowledge considered, the two remaining dimensions offer significant flexibility for its integration.

Each of the mentioned dimensions will be further divided based on identified approaches that allowed for the enhancement of the method's explainability. With that, the review is not meant to be exhaustive, but rather provide a concise summary of recent explainability approaches employed in scenario understanding and motion prediction. By considering the dimensions of data, model design, and model optimization, researchers can develop solutions that are explainable, taking into account the potential need for trade-offs that may arise.

## Data

Previously, we have emphasized the complexities that need to be addressed to perform scenario understanding and motion prediction in an urban environment. These complexities are reflected in the scenario information that can be made available to the motion prediction model, but also in the expectations set for the predicted output. Therefore, the following subsection will discuss both the collected input data and the generated output data of the reviewed motion models. Note that our analysis encompassing both input and output data strays from the approach proposed by [22, 25] since they only consider the enhancement of input data. However, a connection can be established between the following subsection and the categories of data and inference model components taken into account for the introduction of explainability in [26].

**Input Data** The simplest approach to incorporate prior knowledge is to provide the model with access to various sources of scenario data. It is in such cases that data-driven models showcase the performance benefits that place them among the current state-of-the-art. In practice, such cases are rare and can cause a reduction of performance, unless the ability to generalize across the input domain is integrated in remaining model components. Availability of scenario data is not the only issue that has to be addressed in motion prediction. The heterogeneous nature of the scenario data raises the question of pertinent data preprocessing that is directly connected to the question of input layers of the model. Whether the choice of the input data format falls on raw or preprocessed, it has to be founded on the assumption that the model's input layer will be able to extract scenario features relevant to motion prediction.

Unless the task of motion prediction is combined with the task of perception, the starting assumption is the availability of tracked participants' trajectory states (e.g., 2D position or velocity vectors) either given in the world or the image coordinate system. Without exceptions, a history of past states up to a certain time horizon is passed to the model's input introducing the past dynamics as the first factor to influence the motion prediction. Certain preprocessing steps could be applied to the state vectors such as calculating relative participant positions [27], but the use of state vectors as inputs often suggests feature extraction within the input layer of the model. This is the case for graph neural networks (GNNs) whose nodes will contain the said states captured at different time steps [28, 29, 30, 31, 32, 33, 34]. The previously mentioned calculation of relative position could also be delegated to the input layer of the model. In doing so, the influencing factor of spatial interactions between traffic participants can be included.

Information on traffic participant states could be extended with contextual information and represented as a raster images [29, 35, 36, 37, 38, 39, 40], which would allow leveraging the convolutional neural networks (CNNs). Raster images are often generated as a bird's-eye view of the scenario color-coding the information deemed significant for the motion prediction. Such information includes oriented bounding boxes of the tracked traffic participants, polygons of lanelet networks, and lanelet permissibility dependent on traffic rules coupled with respective temporal progression. An alternative to the rasterized HD maps that come with computational burdens, methods of [41, 42, 43, 44] offer vectorized representation of the HD maps. This input representation encodes the traffic participant states and road infrastructure through the use of geometric primitives such as polylines, which can then be stored as vectors. Using such input representation does require additional modeling of the spatial and temporal information, which can be tracked within the input layer.

It is important to note that all the input representations presented are given in an inherently human-understandable format. However, which format (if any) can yield the best model performance has still to be explored.

**Output Data** Explainability requirements placed on the model output can go beyond the visualization of motion predictions. These requirements can be enforced directly on the output domain and often stem from human-like reasoning. Fulfilling certain requirements such as timeliness of the predictions or their scalability to the number of tracked traffic participants is mandatory for a successful motion prediction. However, others can be the subject of a discussion on output representation. Two example models, [35, 45], incorporate the uncertainty of their trajectory predictions as part of the model output, which proves beneficial for the downstream tasks of decision-making and planning. However, the difference between their approaches is in the modality of the outputted trajectories. While [45] predicts an unimodal trajectory, [35] generates a multimodal trajectory. The argument for

the extension from unimodal to multimodal prediction lies in the participants' awareness of potential routes observed that participants could choose. A human participant would then identify the most probable route, which is accomplished in the latter model as well as in the model of [44].

An Interesting approach to output representation is shown in [33], whose model generates intermediate output along the final trajectory prediction. This intermediate output represents the latent intention of the tracked vehicles, which is another human-like expectation set on the model output. Finally, motion prediction outputs modeled as occupancy grids [37] or heatmaps [42] can be visualized accounting for output explainability, but can also be used for further inference of trajectory and its uncertainty.

## Model Design

Embedding domain priors into the model design can be performed at a higher and a lower level. The higher-level design considers modules, among which some can be fully explainable and others unexplainable. Assuming that an unexplainable module can be replaced by one inherently explainable, fusing these modules at input–output connections increases the explainability of the overall model. However, among the fused modules some may already incorporate modifications at a lower design level. Module properties are then used to encode consistencies based on the prior domain. Which module properties can be modified varies according to the underlying methodology of the module. Some methodologies already show desirable and explainable properties such as time invariance encoded in recurrent neural networks (RNNs) and the rotation, translation, and scale invariance encoded in CNNs. Notably, custom convolutional layers in CNN can also be defined to encode domain priors. In addition to module operators, the module variables can be assigned a semantic or physical meaning, while the module parameters can be constrained to a certain value. Propagating such variables and parameters through the module could aid in uncovering the effects individual operators have on their inputs. Again, the choice of variables and parameters will depend on the priors that will be integrated. Therefore, the following subsections will provide examples of both high- and low-level design modifications drawing on scenario understanding and motion prediction domain knowledge.

**Module Fusion** This section is concerned with methods that combine multiple modules into a single cohesive model. The canonical example for motion estimation fuses a data-driven module and a motion model, with the goal to ensure physics-consistent motion estimates. This serves two purposes: to improve the quality of predictions by respecting known physical laws and to improve the predictability and explainability of the model.

We differentiate two main approaches to embed a physics-informed motion model. The former, more common, approach keeps the data-driven and physics modules separate and combines their predictions in some way. The latter approach embeds the motion model into or on top of the network itself, forcing the output to conform to the physics. The latter approach produces more explainable predictions when compared to the former, but this comes at the cost of constraining the model and can lead to difficulty during training.

One variant of the former approach combines the physics features with data-driven features within the network, which offers no guarantees on the physicality of the output. For example, in [46] a social force module is combined with the output of a vehicle trajectory encoder and passed through a non-linearity to produce the predicted trajectories. Similarly, in [47] the output of a physics-based model, motivated by the physics of shockwaves, is concatenated with an output from prior learning-based layers. The concatenation is fed into posterior learning-based layers to predict the output trajectory.

A significantly more explainable approach is presented in [48], where the motion model is embedded in a probabilistic manner, namely modeled as a dynamic Bayesian network (DBN). The preferred path is generated from the probabilistic model, and is therefore not guaranteed to be physically feasible, but is likely to respect the motion model to some extent. Likewise, [49] uses an interacting multiple models (IMM) methodology, which combines a motion model and a Bayesian network. Although the output trajectory is obtained from an interaction of the two models, there is nonetheless no guarantee that it respects the motion model. In [50], the dynamical model is embedded into a Bayesian neural network (BNN), and the joint recurrent model is trained holistically. Similarly to [48], while the Bayesian model incorporates the motion model, there is no guarantee of the physicality of the trajectories. Nonetheless, the predictions are expected to be physically feasible on average.

On the other side of the fence, in the latter approach, trajectories are usually outputted directly from the physics-inspired motion model, and are thus guaranteed to respect the underlying physics. The data-driven module is then mostly used to predict inputs to the motion model. For example, in [36] the state transitions of the motion model are unrolled into kinematic layers, which take in the output of an unconstrained data-driven model. Due to the differentiability of the kinematic layers, the whole model can be trained end-to-end. In [29], gated recurrent unit (GRU) cells of the decoder network output the parameters of a bivariate Gaussian over the control actuations. The control activations are integrated to obtain the predicted trajectory.

The approach in [44] utilizes a different methodology, whereby possible future trajectories are sampled using a motion planner and a model is trained to select the

predictions. Environmental and kinematic constraints are enforced in the sampling-based planner, thus producing physically feasible trajectories, ensuring that the prediction respects the motion model.

Another set of approaches leverages the explainability of Kalman filtering, using it to enforce the motion model. For example, [51] proposes a learnable Kalman filter that utilizes a long–short-term memory (LSTM) network for command prediction. In [52] a Kalman neural network takes in interaction-aware trajectories, produced by a motion layer, and fuses them with dynamic trajectories to predict future trajectories. The motion layer is based on a dynamic model of a vehicle, and it takes as input interaction-aware accelerations produced by a data-driven interaction layer. The entire pipeline is trained end-to-end. Note that the vehicle model is used in the motion layer as a transformation, and the Kalman filter models a linear system.

**Graph Neural Networks** GNNs are a class of DL methods designed to operate on graph-structured data. Such a data structure allows for both explicit and implicit incorporation of the domain knowledge into the structure of the model. The domain of scenario understanding and motion prediction is not an exception, as distinct entities and their relationships can be identified from a traffic scenario. Therefore, a GNN and its variants could provide a high-level representation of the traffic scenario considered. However, in modeling the traffic scenario, a trade-off between the granularity level at which entities and relationships are defined and model performance can be observed.

Translating the entities and relationships identified would require the definition of nodes $V$ and edges $E$ resulting in a graph structure $G = (V, E)$. The definition of nodes and edges of a GNN model outputting traffic participants' motion prediction often starts with assumptions on factors that influence the predicted trajectories or intentions. Given an $N$ number of tracked traffic participants, a GNN could capture the interaction between homogeneous (e.g., pedestrian–pedestrian, vehicle–vehicle) or heterogeneous (e.g., pedestrian–vehicle) traffic participants, modeling each as a node and their mutual impact as an edge. The assumption of interacting traffic participants was adopted in [27, 28, 53] for pedestrian motion prediction, while for vehicular motion prediction, it was considered in [32, 33]. For the motion prediction method to be applicable in an urban environment, previous methods would have to be extended to include heterogeneous traffic participants and their interactions. Examples of methods for which motion prediction of different classes of traffic participants was accomplished are presented in [29, 30, 31, 34]. However, the representation of a scenario taking place in an urban environment is not only specified by the interacting traffic participants but also by its complex infrastructure. To capture the influence the road infrastructure may have on motion prediction, [42] constructs a road graph whose nodes encode lanelet features, while its edges preserve the geometric and connectivity information. The GNN component of this method will finally output scores of the most probable future lanelets, which are then used to recreate a trajectory for each tracked participant. The advantage of such an approach to trajectory prediction is that it guarantees the feasibility of the predicted trajectories with respect to the lanelet network. Since the trajectory is reconstructed using the endpoints belonging to the most likely lanelets, it was not necessary to represent the previous trajectories of the participants as nodes. In contrast, [43] opted to model both traffic participants ("actors") and lanes within respective graphs, which allowed for the definition of four types of interactions: actor-to-lane, lane-to-actor, lane-to-lane, and actor-to-actor. Considering all four types of interactions not only addressed the complexity of road infrastructure itself but also the dependence of actors on the map information. Similarly, the history of trajectory states and the road components are modeled as nodes [41], but with all nodes contained within the same graph structure. However, to retain the semantic and spatial information, the graph structure was divided into local and global graphs with the global graph capturing the high-level interactions between the local subgraphs. The scenario context information can be further expanded with other static and dynamic infrastructure components should such information be available. Reference [38] proposes a graph structure that comprises both traffic participant nodes and traffic element nodes such as traffic signs and lights. Interestingly, their graph structure also allowed for inference of the unknown traffic light states based on the information available in neighboring nodes.

Among the previously considered papers, most edge modeling strategies were based on representing high-level semantic interactions between nodes. However, other strategies defined at a lower level could reveal more of the graph's internal operations. Such strategies were demonstrated in [29, 30, 31, 34] with edges constructed to specifically represent temporal and spatial relationships. While temporal edges existed only between the nodes representing the same traffic participants, spatial edges were often defined by imposing a proximity rule. In [29], a spatial edge only existed between two traffic participants if their $l_2$ distance was below a predefined threshold. Additionally, comprising spatial edges were of the directed type, which allowed for modeling of the asymmetric influences between traffic participants. Another example of proximity constraints was set by [27] establishing a directed spatial edge only if the neighboring traffic participant was in the view area of the considered traffic participant. The view area was based on the approximation of the head orientation and the predefined view angle, which successfully accounted for the asymmetric influences the relative position of neighboring participants can have on the considered participant.

Notably, representation learning utilized in reviewed methods focused on edge representation, especially in the cases of high-level interactions, i.e., domain priors were primarily reflected in the node definition. This approach made the hand-engineered node features inherently explainable, but it often came at the cost of reduced performance. This result can be explained through the limitation of the node's flexibility to model key entity features. To address this limitation, [27, 33, 40] choose to perform node representation learning, as well as edge representation learning. Given the requirement for a method scalable to the arbitrary number of traffic participants, the learned representation parameters should be shared not only between edges but also the nodes of the same type. Nonetheless, with learned node and edge feature extraction the explainability of the subsequent graph operations will be reduced.

**Probabilistic Graphical Model** Probabilistic graphical models (PGMs) offer a highly versatile and insightful representation of probability distributions and the way they are factored. Due to the predictable behavior of these models and the well-understood theoretic underpinnings, both for inference and learning, they are highly explainable by construction. These models are of great historic significance, with their ability to express complex probability distributions finding a wide range of use, most notably in computer vision [54]. With the advent of modern machine learning that offers a much greater predictive ability, the relevance and popularity of PGMs has gone down in general. Nonetheless, the methodology has found some use in motion prediction, as we will outline in this section.

The classification of normal and dangerous driving scenarios is handled by two hidden Markov models (HMMs) in [55]. Conditioned on the driving scenario, the trajectory prediction is statistically generated from the appropriate driving dataset. An approach based on IMMs is proposed in [49] and leverages a pair of strongly explainable models: a DBN for maneuver-based predictions in conjunction with an unscented Kalman filter (UKF) for physics-based predictions. The method is then interpreted as a Markov switching system, which alternates between the two models.

In [56], a HMM is used to classify vehicle maneuvers. The classifier is combined with a trajectory prediction module and a vehicle interaction module. The trajectory prediction model combines a maneuver-specific variational Gaussian mixture model (VGMM) and a IMM-based motion model. The interaction module takes the input from the HMM and the trajectory prediction module and produces the final estimates by solving an integer linear programming problem.

**Social Force Model** Social force models (SFMs) model the traffic participants (usually pedestrians) as particles, and define forces that act on these particles, which then move in ways that are consistent with the physics of motion. Although originally introduced as purely physics-driven approaches, machine learning has been introduced into these models. In this new type of SFMs, the forces are usually generated via neural networks, and the mechanics of motion are modeled using physics. This combination maintains the explainability of the underlying physical model, while leveraging the predictive power of neural networks.

For example, a graph-based SFM-based approach is employed in [57], where pedestrian–pedestrian and pedestrian–obstacle interactions are modeled as interactions in a graph network. This approach is combined with a collision avoidance learner, which better captures the patterns of pedestrian motion, and the model is trained using a student–teacher co-forcing training algorithm.

A similar approach is featured in [58], where the social forces are modeled via three neural network components: a goal attraction component, drawn toward a sampled goal location, an interagent repulsive component, and an environmental repulsive component based on a segmentation map of the environment. In tandem with the SFM, motion stochasticity and observation noise are modeled explicitly using a conditional variational auto-encoder (CVAE). The combined network is trained through a multi-stage training process that demonstrated faster training convergence.

Finally, [59] models pedestrian dynamics via SFM where each force component is replaced by a small neural network. Namely, one for the acceleration force that drives the pedestrian toward the desired location, one for repulsion from boundaries, one for repulsion between pedestrians, and one for the group force, which consists of the group coherence force and the force based on the desire for each pedestrian to keep others in their field-of-view. It should be noted that, unlike [57, 58] that are trained on real-world data, this model is only trained on synthetic data, which was generated by another social force model in a closed simulation environment.

An alternative approach to SFM uses networks to predict a potential field, rather than forces. The potential is then differentiated and the result is used to predict motion. An example of this methodology is presented in [60], where a potential field is constructed from three components: an environment potential field, an inertial potential field, and a social force field. The environment captures lane-following behavior and obstacle avoidance, and is combined with the inertial potential field to obtain a motion field. The motion field is then combined with the social force field and speed prediction to compute a displacement field, which is differentiated to obtain the velocity. The velocity is finally integrated to compute the trajectory prediction. Another result [61] uses a data-free potential field SFM approach as a baseline, and uses a neural network to correct the acceleration prediction within the motion model state transitions.

**Attention Mechanism** Attention mechanisms allow ML models to assign importance to input data often based on some contextual information. They can provide insights into which parts of the input data are relevant to the prediction and, consequently, enhance its explainability. Although they were first introduced for natural language processing (NLP), they have found use in various AD tasks including motion prediction.

Reviewed papers reveal that different variants of the attention mechanism can be integrated into models relying on GNNs. With graph nodes representing pedestrians, [28] uses a soft attention mechanism to capture the influence of each pedestrian on another. Calculating attention weights over spatial edges connecting the pedestrians serves as an alternative to edge definition based on the pedestrians' proximity. Rather than being an alternative, [29] follows the definition of the edge based on the proximity threshold with an attention mechanism to obtain the aggregated influence of all neighboring traffic participants on the participant considered. Similar to [28, 34] integrates the soft attention mechanism to distribute weights across all spatial edges connecting heterogeneous instance nodes. In addition to defining an instance layer to their graph, [34] also defines a category layer intended to learn similarities between instances of the same type for which it employs a self-attention mechanism. The graph proposed in [41] also proves the benefit of capturing high-level interactions within a global graph with a self-attention mechanism.

A temporal attention mechanism is proposed in [62], which assigns weights to the LSTM encoder output at each time step. This aims to address problems occurring due to speed variations in a pedestrian's motion and due to temporal misalignment. The output of the temporal attention module is a fixed-length feature vector computed as a weighted combination of feature vectors. Combining layers of LSTM and attention mechanisms is also employed in the Evaluator component of the PRIME model [44]. Once the LSTMs encode entities representing the static and dynamic environment components, namely, tracked participants' trajectory, set of their reachable paths, and feasible future trajectories, variants of attention mechanism learn their interactions. The adopted attention mechanism is constructed of four modules: path to track (P2T), path to future (P2F), agent to agent (A2A), and future to future (F2F). The goal of the proposed attention mechanism is to fuse spatiotemporal information from varying number of agents in a scenario, which is argued as a requirement to generate scores for the possible future trajectories.

With the aim of modeling non-local vehicle interactions, [63] propose a multi-head attention mechanism that captures the importance of global dependencies on a target vehicle. In particular, it captures the attention put on each grid position when predicting the motion of a certain vehicle.

**Social Pooling** Social pooling is an approach to modeling social interactions between agents in machine learning models. In contrast to a purely data-driven approach, embedding social priors introduces a very limited type of explainability into the model. That is, while the interactions themselves may not necessarily be explainable, the existence and intensity of the interactions may provide certain insights into the behavior of the model.

The baseline approach considers a (rectangular) neighborhood around each agent and aggregates the features of the neighbors. This aggregate is then used to extract information about agent-to-agent social interactions. For example, in [64] the hidden states of the neighbor agent LSTMs are concatenated to form a hidden state tensor. The hidden state tensor and the coordinates of the neighbors are embedded, and the concatenation of these embeddings is passed forward as an input to the LSTM cell of the corresponding trajectory. In addition to human-to-human pooling of LSTM hidden states, [65] also proposes a distance-aware context-aware pooling between human and static objects in the scene. According to [65], while equal weighting of human neighbors is justified due to collision avoidance, (distance-dependent) weights are needed to model how much static elements can influence a person's movement. Another approach based on LSTM hidden states is presented in [62], where the positional offsets between neighbors and the cosine similarity of the hidden information are combined to obtain weighted information on the interactions.

In [33, 66], neighboring trajectory features are assembled into a social tensor, which is then processed by convolutional social pooling layers to obtain the social context. Likewise, [67] proposes spatially aware social pooling approach, where the historic trajectories of neighboring vehicles are encoded and combined into a spatial tensor, which preserves the spatial relations between the vehicles. Convolutional layers are used to compute a social tensor, which captures spatially aware multivehicle interactions.

Other strategies have also been proposed, which consider more than just the local neighborhood around each agent. This is in line with intuitive notions regarding the problem, where distant pedestrians might still impact one another's movements. With this in mind, a non-local approach to human-to-human interaction is proposed in [68]. Rather than restricting the analysis to a local neighborhood, the model considers the relative positions between all pedestrians in the scene. This information is concatenated with each person's hidden state, processed using a multilayer perceptron (MLP) and passed through a symmetric pooling function. In [63], the non-local social pooling module is constructed using two components: a convolution layer for capturing local interactions and a multi-head attention mechanism for capturing distant dependencies.

## Model Optimization

Constraints derived from motion dynamics can be used as priors to construct new loss functions. Opposed to the calculation of the loss based on just the error between the ground truth and the solution, the loss has an added term penalizing the violation of the set constraints. With the modified loss equation, the model is simultaneously learning to fit the ground truth and satisfy prior constraints. The following subsection will present examples of modifications introduced to the motion prediction model's loss.

**Loss** A common approach to introduce explainability into a machine learning model is to introduce priors in the loss term during training. The priors bias the model toward learning to predict physically feasible behavior, or behavior that conforms to problem-specific expert knowledge or intuitions.

An example of the former is physics-inspired neural networks (PINNs) [69]. While this is a broad umbrella term that includes many kinds of models that utilize physical priors, in the canonical example a physics-inspired regularization term is added to the loss term during training

$$\mathcal{L} = \lambda \mathcal{L}_{\text{data}} + \left(1-\lambda\right)\mathcal{L}_{\text{phy}} \qquad \text{Eq. (5)}$$

with $\lambda \in [0,1]$ biasing the model toward the two extreme ends: either fully consistent with the data ($\lambda = 1$) or fully consistent with the physics ($\lambda = 0$). The physics-inspired loss term acts as a soft (implicit) constraint on the solution, which unlike a hard constraint does not make the model fully explainable. However, it still introduces some explainability when compared to a purely data-driven approach. A direct application of this methodology for trajectory prediction can be seen in [70], where a transformer is trained using a loss that includes a physics-inspired regularization term, which favors solutions similar to the predictions of a motion model. From the works we analyzed, this approach of embedding physicality is markedly less popular than embedding the motion model into the network itself or introducing physics features into the model input, both of which we analyze in different parts of the article.

In the latter approach, the loss term is based on problem-specific expert knowledge rather than physical constraints. For example, to account for the multimodality of trajectory prediction, which is not captured by a standard regression formulation, [35] uses a modified multiple-trajectory prediction (MTP) loss, which reformulates the task as a mode selection problem. A neural network outputs a fixed number of output trajectories, and the best matching mode $m^\star$ is selected based on its proximity to the ground truth. The loss can then be decomposed into a mode classification part and a per-mode regression part

$$\mathcal{L} = \mathcal{L}_{\text{class}} + \sum_{m=1}^{M} I_{m=m^*} L\left(\boldsymbol{\tau}, \boldsymbol{\tau}_m\right) \qquad \text{Eq. (6)}$$

where $I_{m=m^*}$ is a binary indicator function (of the condition $m = m^\star$) and $\mathcal{L}_{\text{class}}$ is the cross-entropy loss

$$\mathcal{L}_{\text{class}} = -\sum_{m=1}^{M} I_{m=m^*} \log p_m \qquad \text{Eq. (7)}$$

In addition to the main task of trajectory prediction, the model in [41] is also trained to recover randomly masked-out node features from its neighbors. The intention behind the auxiliary task is to improve the ability of the model to learn context-dependent node interactions. The loss is constructed as a combination of the two tasks

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \alpha \mathcal{L}_{\text{node}} \qquad \text{Eq. (8)}$$

where $\alpha$ is the loss weighting factor, which controls the relative importance of the auxiliary loss term compared to the trajectory estimation loss term. It should be noted that the auxiliary loss alone is not sufficiently informative to claim any kind of explainability, but in combination with the graph structure, it provides some additional insight into the model behavior.

The work of [71] extracts spectral information from the Laplacian matrices of its road agent proximity disk graphs. The (weighted) Laplacian is expressed as $L = D - A$, where $A$ is the adjacency matrix given as

$$A_{i,j} := \begin{cases} e^{-d\left(v_i, v_j\right)}, & \text{if} \quad i \neq j \wedge d\left(v_i, v_j\right) < \mu \\ 0, & \text{otherwise} \end{cases}$$

with $\mu$ as the radius of the disk graph and $D$ as the degree matrix, which is a diagonal matrix with the node degrees on the main diagonal

$$D_{i,i} := \sum_{j=1}^{n} A_{i,j}$$

The spectra, which consist of the top $k$ eigenvectors of $L$ (by eigenvalue), are then utilized in two ways. First, they are used to predict the behavior of road agents, classifying them as underspeeding, neutral, or overspeeding. Second, and more relevant to the point of discussion, the loss function is regularized by additional spectral clustering terms, pertaining to the mean and variance of the clusters.

In [58], a randomness-modeling CVAE is trained in tandem with a social force model. This leads to the modification of the loss function, to not only match the predicted and ground-truth trajectories but also push the model to learn the distribution of randomness. The overall loss is computed as the sum of two components

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \mathcal{L}_{\text{tvae}} \qquad \text{Eq. (9)}$$

# Future Directions

Comparison of the approaches specified under general PIML taxonomies of [22, 23, 25, 26], and the approaches identified for the task of scenario understanding and motion prediction suggests that certain approaches have yet to be explored. Primarily, the approaches that fall under the optimization enhancement appear to have gained limited use in motion prediction despite, for example, PINNs offering a clear mechanism to account for constraints on participant motion. Complexities of urban scenarios cannot be captured in their entirety, leading to certain edge cases being excluded from the training data. To address this issue, the proposed model requires the ability to generalize over edge cases that have not been encountered before. The benefit of generalization offered by PINNs could be one of the approaches to expand the operational domain of the model. Also classified as an approach for optimization enhancement is the initialization of the model's starting state. Since poor initialization can cause models to anchor in local minima, the model would benefit from initial weight definition based on domain knowledge. Unfortunately, no such approach was found in use for the scenario understanding and motion prediction task. Given the complexity of the reviewed models, it is expected that model-level initialization would pose a significant challenge, but it could become possible at the module level. An example of module-level initialization could be employed for models that use the CNN modules for processing raster images. The learned weights of existing CNN models trained on shapes used to encode the scenario information can be set as initialization weights of the CNN module.

Among the approaches identified under the model design enhancement, those based on GNNs offer a promising direction for continued advances. A contributing factor to their use is the intuitiveness of scenario representation that allows for simple expansion of the graph structure should additional scenario information become available. The intuitiveness of the graph structure is also reflected in the modeling of the influencing interactions as graph edges. Additionally, interaction modeling can be further enhanced by another reviewed design-level group of approaches. The attention mechanisms have proven to successfully translate assumptions about interactions between different scenario components. It should be noted that attention mechanisms present more recent advances introduced to the ML field. The future may bring novel design approaches that will allow for new ways to incorporate domain knowledge, whether by considering more informative and evolving graph topologies, developing more sophisticated global attention mechanisms, or implicitly embedding problem-informed invariants into the model (e.g., [72]).

Finally, we note that the embedding of kinematic models into neural networks and the consequent end-to-end learning of these hybrid models represents a strongly explainable and high-potential approach for future work. We believe that there is high value in exploring the introduction of new implicit constraints into neural networks, whether differentiable and hence back-propagation friendly or non-differentiable and combined with adjoint state methods for computing gradients. The great deal of physical explainability that these methods offer makes up for their disadvantages, which is evidenced by the prior popularity of these approaches in the state-of-the-art, despite the existence of methods that are simpler to employ.

# Conclusion

In this work, we outlined the increasing importance of explainability in AI solutions, further emphasized by the development of the AI Act. Automotive applications, which are especially safety-sensitive, present themselves as a natural domain of interest. Being part of any holistic automotive system, we have highlighted motion prediction as a critical component in the pursuit of safe explainable AI solutions. With this in mind, we have summarized and discussed the various ways in which explainability has been used in state-of-the-art motion prediction methods. This methodology stands in contrast to previous surveys, which seldom even mention explainability or treat it as a secondary detail. As a work focusing primarily on explainability, our main goal was to present the established methods for those looking for practical solutions, as well as to categorize them based on the corresponding general methodology. For each of the methods outlined in the article, we debated the degree to which they can be considered explainable. With the natural trade-off between explainability and predictive accuracy, we also discussed the wider considerations when choosing between models. Finally, we outlined some gaps in the state-of-the-art, where certain methodologies have not been fully explored and might offer a promising avenue for further research.

# Acknowledgements

This work has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No. 101076754—Althena project.

The publication was written at Virtual Vehicle Research GmbH in Graz and partially funded within the COMET K2 Competence Centers for Excellent Technologies from the Austrian Federal Ministry for Climate Action (BMK), the Austrian Federal Ministry for Labour and Economy (BMAW), the Province of Styria (Dept. 12), and the Styrian Business Promotion Agency (SFG). The Austrian Research Promotion Agency (FFG) has been authorized for programme management.

## Contact Information

**Ilma Okanovic**, corresponding author
MSc., Researcher
Virtual Vehicle Research GmbH
ilma.okanovic@v2c2.at

## References

1. European Automobile Manufacture Association, "ACEA Position Paper Artificial Intelligence in the Automobile Industry," 2020, accessed January 10, 2024, https://www.acea.auto/uploads/publications/ACEA_Position_Paper-Artificial_Intelligence_in_the_automotive_industry.pdf.
2. European Commission, "Proposal for a Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts," 2021, accessed January 10, 2024, https://eur-lex.europa.eu/legalcontent/EN/TXT/?uri=CELEX:52021PC0206.
3. SAE International, "SAE Levels of Driving Automation™ Refined for Clarity and International Audience," 2021, accessed January 10, 2024, https://www.sae.org/blog/sae-j3016-update.
4. International Organization for Standardization, "Road Vehicles—Safety of the Intended Functionality," Standard, Geneva, June 2022.
5. Omeiza, D., Webb, H., Jirotka, M., and Kunze, L., "Explanations in Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 8 (2021): 10142-10162.
6. High-Level Expert Group on Artificial Intelligence, "Ethics Guidelines for Trustworthy AI," 2019, accessed January 10, 2024, https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai.
7. International Organization for Standardization, "Information Technology—Artificial Intelligence—Artificial Intelligence Concepts and Terminology," Standard, Geneva, July 2022.
8. International Organization for Standardization, "Information Technology—Artificial Intelligence—Overview of Trustworthiness in Artificial Intelligence," Standard, Geneva, May 2020.
9. Zablocki, É., Ben-Younes, H., Pérez, P., and Cord, M., "Explainability of Deep Vision-Based Autonomous Driving Systems: Review and Challenges," *International Journal of Computer Vision* 130, no. 10 (2022): 2425-2452.
10. Atakishiyev, S., Salameh, M., Yao, H., and Goebel, R., "Explainable Artificial Intelligence for Autonomous Driving: A Comprehensive Overview and Field Guide for Future Research Directions," arXiv preprint arXiv:2112.11561, 2021.
11. Lefèvre, S., Vasquez, D., and Laugier, C., "A Survey on Motion Prediction and Risk Assessment for Intelligent Vehicles," *ROBOMECH Journal* 1, no. 1 (2014): 1-14.
12. Benrachou, D.E., Glaser, S., Elhenawy, M., and Rakotonirainy, A., "Use of Social Interaction and Intention to Improve Motion Prediction within Automated Vehicle Framework: A Review," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 12 (2022): 22807-22837.
13. Huang, Y., Du, J., Yang, Z., Zhou, Z. et al., "A Survey on Trajectory-Prediction Methods for Autonomous Driving," *IEEE Transactions on Intelligent Vehicles* 7, no. 3 (2022): 652-674.
14. Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M. et al., "Human Motion Trajectory Prediction: A Survey," *The International Journal of Robotics Research* 39, no. 8 (2020): 895-935.
15. Ridel, D., Rehder, E., Lauer, M., Stiller, C. et al., "A Literature Review on the Prediction of Pedestrian Behavior in Urban Scenarios," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, 2018, 3105-3112.
16. Brouwer, N., Kloeden, H., and Stiller, C., "Comparison and Evaluation of Pedestrian Motion Models for Vehicle Safety Systems," in *Proceeding IEEE International Conference on Intelligent Transportation Systems*, Rio de Janeiro, Brazil, 2016, 2207-2212.
17. Karle, P., Geisslinger, M., Betz, J., and Lienkamp, M., "Scenario Understanding and Motion Prediction for Autonomous Vehicles—Review and Comparison," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 10 (2022): 16962-16982.
18. Gulzar, M., Muhammad, Y., and Muhammad, N., "A Survey on Motion Prediction of Pedestrians and Vehicles for Autonomous Driving," *IEEE Access* 9 (2021): 137957-137969.
19. Mozaffari, S., Al-Jarrah, O.Y., Dianati, M., Jennings, P. et al., "Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 1 (2020): 33-47.
20. Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G., "A Survey of Deep Learning Techniques for Autonomous Driving," *Journal of Field Robotics* 37, no. 3 (2020): 362-386.
21. Brown, K., Driggs-Campbell, K., and Kochenderfer, M.J., "A Taxonomy and Review of Algorithms for Modeling and Predicting Human Driver Behavior," arXiv preprint arXiv:2006.08832, 2020.
22. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P. et al., "Physics-Informed Machine Learning," *Nature Reviews Physics* 3, no. 6 (2021): 422-440.
23. Willard, J., Jia, X., Xu, S., Steinbach, M. et al., "Integrating Physics-Based Modeling with Machine Learning: A Survey," arXiv preprint arXiv:2003.04919, 2020.
24. Banerjee, C., Nguyen, K., Fookes, C., and Raissi, M., "A Survey on Physics Informed Reinforcement Learning:

Review and Open Problems," arXiv preprint arXiv:2309.01909, 2023.

25. Meng, C., Seo, S., Cao, D., Griesemer, S. et al., "When Physics Meets Machine Learning: A Survey of Physics-Informed Machine Learning," arXiv preprint arXiv:2203.16797, 2022.
26. Hao, Z., Liu, S., Zhang, Y., Ying, C. et al., "Physics-Informed Machine Learning: A Survey on Problems, Methods and Applications," arXiv preprint arXiv:2211.08064, 2022.
27. Zhang, L., She, Q., and Guo, P., "Stochastic Trajectory Prediction with Social Graph Network," arXiv preprint arXiv:1907.10233, 2019.
28. Vemula, A., Muelling, K., and Oh, J., "Social Attention: Modeling Attention in Human Crowds," in *Proceeding IEEE International Conference on Robotics and Automation*, Brisbane, Australia, 2018, 4601-4607.
29. Salzmann, T., Ivanovic, B., Chakravarty, P., and Pavone, M., "Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data," in *Proceeding European Conference on Computer Vision*, Virtual, 2020, 683-700.
30. Li, X., Ying, X., and Chuah, M.C., "Grip: Graph-Based Interaction-Aware Trajectory Prediction," in *Proceeding IEEE Intelligent Transportation Systems Conference*, Auckland, New Zealand, 2019, 3960-3966.
31. Li, X., Ying, X., and Chuah, M.C., "Grip++: Enhanced Graph-Based Interaction-Aware Trajectory Prediction for Autonomous Driving," arXiv preprint arXiv:1907.07792, 2019.
32. Diehl, F., Brunner, T., Le, M.T., and Knoll, A., "Graph Neural Networks for Modelling Traffic Participant Interaction," in *Proceeding IEEE Intelligent Vehicles Symposium*, Paris, France, 2019, 695-701.
33. Zhao, Z., Fang, H., Jin, Z., and Qiu, Q., "Gisnet: Graph-Based Information Sharing Network for Vehicle Trajectory Prediction," in *Proceeding IEEE International Joint Conference on Neural Networks*, Glasgow, UK, 2020, 1-7.
34. Ma, Y., Zhu, X., Zhang, S., Yang, R. et al., "Trafficpredict: Trajectory Prediction for Heterogeneous Traffic-Agents," in *Proceeding AAAI Conference on Artificial Intelligence*, Honolulu, HI, Vol. 33, 2019, 6120-6127.
35. Cui, H., Radosavljevic, V., Chou, F.-C., Lin, T.-H. et al., "Multimodal Trajectory Predictions for Autonomous Driving Using Deep Convolutional Networks," in *Proceeding IEEE International Conference on Robotics and Automation*, Montreal, QC, Canada, 2019, 2090-2096.
36. Cui, H., Nguyen, T., Chou, F.-C., Lin, T.-H. et al., "Deep Kinematic Models for Kinematically Feasible Vehicle Trajectory Predictions," in *Proceeding IEEE International Conference on Robotics and Automation*, Paris, France, 2020, 10563-10569.
37. Hong, J., Sapp, B., and Philbin, J., "Rules of the Road: Predicting Driving Behavior with a Convolutional Model of Semantic Interactions," in *Proceeding IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2019, 8454-8462.
38. Kumar, S., Gu, Y., Hoang, J., Haynes, G.C. et al., "Interaction-Based Trajectory Prediction over a Hybrid Traffic Graph," in *Proceeding IEEE International Conference on Intelligent Robots and Systems*, Prague, Czech Republic, 2021, 5530-5535.
39. Tang, C. and Salakhutdinov, R.R., "Multiple Futures Prediction," *Proc. Advances in Neural Information Processing Systems* 32 (2019).
40. Li, J., Yang, F., Tomizuka, M., and Choi, C., "Evolvegraph: Multi-Agent Trajectory Prediction with Dynamic Relational Reasoning," *Proc. Advances in Neural Information Processing Systems* 33 (2020): 19783-19794.
41. Gao, J., Sun, C., Zhao, H., Shen, Y. et al., "Vectornet: Encoding HD Maps and Agent Dynamics from Vectorized Representation," in *Proceeding IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 2020, 11525-11533.
42. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B. et al., "Gohome: Graph-Oriented Heatmap Output for Future Motion Estimation," in *Proceeding IEEE International Conference on Robotics and Automation*, Philadelphia, PA, 2022, 9107-9114.
43. Liang, M., Yang, B., Hu, R., Chen, Y. et al., "Learning Lane Graph Representations for Motion Forecasting," in *Proceeding European Conference on Computer Vision*, Virtual, 2020, 541-556.
44. Song, H., Luan, D., Ding, W., Wang, M.Y. et al., "Learning to Predict Vehicle Trajectories with Model-Based Planning," in *Proceeding Conference on Robot Learning*, Auckland, New Zealand, 2022, 1035-1045.
45. Djuric, N., Radosavljevic, V., Cui, H., Nguyen, T. et al., "Uncertainty-Aware Short-Term Motion Prediction of Traffic Actors for Autonomous Driving," in *Proceeding IEEE/CVF Winter Conference on Applications of Computer Vision*, Virtual, 2020, 2095-2104.
46. Li, H., Liao, Z., Rui, Y., Li, L. et al., "A Physical Law Constrained Deep Learning Model for Vehicle Trajectory Prediction," *IEEE Internet of Things Journal* 10, no. 24 (2023): 22775-22790.
47. Yao, H., Li, X., and Yang, X., "Physics-Aware Learning-Based Vehicle Trajectory Prediction of Congested Traffic in a Connected Vehicle Environment," *IEEE Transactions on Vehicular Technology* 72, no. 1 (2023): 102-112.
48. Ballan, L., Castaldo, F., Alahi, A., Palmieri, F. et al., "Knowledge Transfer for Scene-Specific Motion Prediction," in *Proceeding European Conference on Computer Vision*, Amsterdam, the Netherlands, 2016, 697-713.
49. Xie, G., Gao, H., Qian, L., Huang, B. et al., "Vehicle Trajectory Prediction by Integrating Physics- and Maneuver-Based Approaches Using Interactive Multiple Models," *IEEE Transactions on Industrial Electronics* 65, no. 7 (2018): 5999-6008.
50. Tang C., Chen J., and Tomizuka M., "Adaptive Probabilistic Vehicle Trajectory Prediction through Physically Feasible Bayesian Recurrent Neural Network," in *Proceeding IEEE International Conference on Robotics*

*and Automation*, Montreal, QC, Canada, 2019, 3846-3852.

51. Mercat, J., Zoghby, N.E., Sandou, G., Beauvois, D. et al., "Kinematic Single Vehicle Trajectory Prediction Baselines and Applications with the NGSIM Dataset," arXiv preprint arXiv:1908.11472, 2019.
52. Ju, C., Wang, Z., Long, C., Zhang, X. et al., "Interaction-Aware Kalman Neural Networks for Trajectory Prediction," in *Proceeding IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, 2020, 1793-1800.
53. Mohamed, A., Qian, K., Elhoseiny, M., and Claudel, C., "Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction," in *Proceeding IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 2020, 14424-14432.
54. Wang, C., Komodakis, N., and Paragios, N., "Markov Random Field Modeling, Inference & Learning in Computer Vision & Image Understanding: A Survey," *Computer Vision and Image Understanding* 117, no. 11 (2013): 1610-1627.
55. Liu, P., Kurt, A., and Özgüner, Ü., "Trajectory Prediction of a Lane Changing Vehicle Based on Driver Behavior Estimation and Classification," in *Proceeding IEEE Conference on Intelligent Transportation Systems*, Qingdao, China, 2014, 942-947.
56. Deo, N., Rangesh, A., and Trivedi, M.M., "How Would Surround Vehicles Move? A Unified Framework for Maneuver Classification and Motion Prediction," *IEEE Transactions on Intelligent Vehicles* 3, no. 2 (2018): 129-140.
57. Zhang, G., Yu, Z., Jin, D., and Li, Y., "Physics-Infused Machine Learning for Crowd Simulation," in *Proceeding ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington, DC, 2022, 2439-2449.
58. Yue, J., Manocha, D., and Wang, H., "Human Trajectory Prediction via Neural Social Physics," in *Proceeding European Conference on Computer Vision*, Tel Aviv, Israel, 2022, 376-394.
59. Hossain, S., Johora, F.T., Müller, J.P., Hartmann, S. et al., "SFMGNET: A Physics-Based Neural Network to Predict Pedestrian Trajectories," arXiv preprint arXiv:2202.02791, 2022.
60. Su, S., Peng, C., Shi, J., and Choi, C., "Potential Field: Interpretable and Unified Representation for Trajectory Prediction," arXiv preprint arXiv:1911.07414, 2019.
61. Particke, F., Zhou, J., Hiller, M., Hofmann, C. et al., "Neural Network Aided Potential Field Approach for Pedestrian Prediction," in *Proceeding Sensor Data Fusion: Trends, Solutions, Applications*, Bonn, Germany, 2019, 1-6.
62. Li, X., Liu, Y., Wang, K., and Wang, F.-Y., "A Recurrent Attention and Interaction Model for Pedestrian Trajectory Prediction," *IEEE/CAA Journal of Automatica Sinica* 7, no. 5 (2020): 1361-1370.
63. Messaoud, K., Yahiaoui, I., Verroust-Blondet, A., and Nashashibi, F., "Non-Local Social Pooling for Vehicle Trajectory Prediction," in *Proceeding IEEE Intelligent Vehicles Symposium*, Paris, France, 2019, 975-980.
64. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A. et al., "Social LSTM: Human Trajectory Prediction in Crowded Spaces," in *Proceeding IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, 961-971.
65. Bartoli, F., Lisanti, G., Ballan, L., and Del Bimbo, A., "Context-Aware Trajectory Prediction," in *Proceeding IEEE International Conference on Pattern Recognition*, Beijing, China, 2018, 1941-1946.
66. Deo, N. and Trivedi, M.M., "Convolutional Social Pooling for Vehicle Trajectory Prediction," in *Proceeding IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Salt Lake City, UT, 2018, 1468-1476.
67. Wang, Y., Zhao, S., Zhang, R., Cheng, X. et al., "Multi-Vehicle Collaborative Learning for Trajectory Prediction with Spatio-Temporal Tensor Fusion," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 1 (2022): 236-248.
68. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S. et al., "Social Gan: Socially Acceptable Trajectories with Generative Adversarial Networks," in *Proceeding IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, 2255-2264.
69. Raissi, M., Perdikaris, P., and Karniadakis, G., "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," *Journal of Computational Physics* 378 (2019): 686-707.
70. Geng, M., Li, J., Xia, Y., and Chen, X.M., "A Physics-Informed Transformer Model for Vehicle Trajectory Prediction on Highways," *Transportation Research Part C: Emerging Technologies* 154 (2023): 104272.
71. Chandra, R., Guan, T., Panuganti, S., Mittal, T. et al., "Forecasting Trajectory and Behavior of Road-Agents Using Spectral Clustering in Graph-LSTMS," *IEEE Robotics and Automation Letters* 5, no. 3 (2020): 4882-4890.
72. Qi, C.R., Su, H., Mo, K., and Guibas, L.J., "Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proceeding IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017, 652-660.