



Explainability of Deep Vision-Based Autonomous Driving Systems: Review and Challenges

Éloi Zablocki¹ · Hédi Ben-Younes¹ · Patrick Pérez¹ · Matthieu Cord^{1,2}

Received: 19 January 2021 / Accepted: 16 July 2022 / Published online: 7 August 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

This survey reviews explainability methods for vision-based self-driving systems trained with behavior cloning. The concept of *explainability* has several facets and the need for explainability is strong in driving, a safety-critical application. Gathering contributions from several research fields, namely computer vision, deep learning, autonomous driving, explainable AI (X-AI), this survey tackles several points. First, it discusses definitions, context, and motivation for gaining more interpretability and explainability from self-driving systems, as well as the challenges that are specific to this application. Second, methods providing explanations to a black-box self-driving system in a post-hoc fashion are comprehensively organized and detailed. Third, approaches from the literature that aim at building more interpretable self-driving systems by design are presented and discussed in detail. Finally, remaining open-challenges and potential future research directions are identified and examined.

Keywords Autonomous driving · Explainability · Interpretability · Black-box · Post-hoc interpretability

1 Introduction

1.1 Explainability in the Context of Autonomous Driving

1.1.1 Call for Explainable Autonomous Driving

The need to explain self-driving behaviors is multi-factorial. To begin with, autonomous driving is a high-stake and safety-critical application. It is thus natural to ask for performance guarantees, from a societal point-of-view. However, self-

driving models are not completely testable under all scenarios as it is not possible to exhaustively list and evaluate every situation the model may possibly encounter. As a fallback solution, this motivates the need for *explanation* of driving decisions.

Moreover, explainability is also desirable for various reasons depending on the performance of the system to be explained. For example, as detailed by Selvaraju et al. (2020), when the system works poorly, explanations can help engineers and researchers to improve future versions by gaining more information on corner cases, pitfalls, and potential failure modes (Tian et al., 2018; Hecker et al., 2020). Moreover, when the system's performance matches human performance, explanations are needed to increase users' trust and enable the adoption of this technology (Lee and Moray, 1992; Choi and Ji, 2015; Shen et al., 2020; Zhang et al., 2020). In the future, if self-driving models largely outperform humans, produced explanations could be used to teach humans to better drive and to make better decisions with machine teaching (Mac Aodha et al., 2018).

Besides, from a machine learning perspective, it is also argued that the need for explainability stems from a mismatch between training objectives on the one hand, and the more complex real-life goal on the other hand, I.E., *driving* (Lipton, 2018; Doshi-Velez and Kim, 2017). Indeed, the predictive performance on test sets does not perfectly represent

Communicated by Tinne Tuytelaars.

Éloi Zablocki and Hédi Ben-Younes have contributed equally to this work.

✉ Éloi Zablocki
eloi.zablocki@valeo.com
Hédi Ben-Younes
hedi.ben-younes@valeo.com
Patrick Pérez
patrick.perez@valeo.com
Matthieu Cord
matthieu.cord@valeo.com

¹ Valeo.ai, Paris, France

² Sorbonne Université, Paris, France

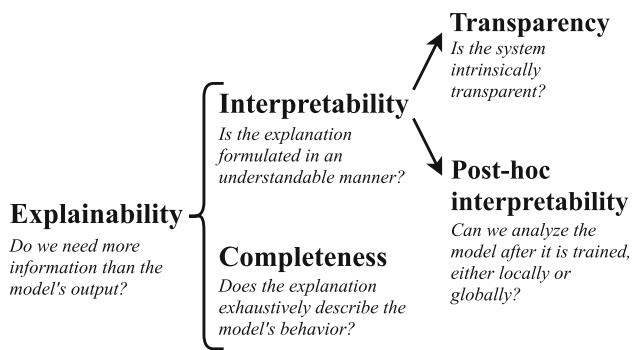


Fig. 1 Taxonomy of explainability terms adopted in this survey. *Explainability* is the combination of *interpretability* (= comprehensible by humans) and *completeness* (= exhaustivity of the explanation) aspects. There are two approaches to have interpretable systems: approaches intrinsic to the design of the system, which increases its *transparency*, and *post-hoc* approaches that justify decisions afterwards for any black-box system

performances an actual car would have when deployed to the real world. For example, this may be due to the fact that the environment is not stationary, and the i.i.d. assumption does not hold as actions made by the model alter the environment. In other words, Doshi-Velez and Kim (2017) argue that the need for explainability arises from incompleteness in the problem formalization: machine learning objectives are flawed proxy functions towards the ultimate goal of driving. Prediction metrics alone are not sufficient to fully characterize the learned system (Lipton, 2018): extra information is needed, *explanations*. Explanations thus provide a way to check if the hand-designed objectives which are optimized enable the trained system to drive as a by-product.

Finally, as Autonomous Vehicles rely more and more on deep neural networks processing visual streams (Janai et al., 2020b), it is of critical importance to study the explainability of driving models from a computer vision perspective. The visual input space is usually of very high dimensionality, potentially built from multiple sensor types, and it does not explicitly express any semantic concepts (Ribeiro et al., 2016). These considerations induce extra challenges in explaining the behavior of vision-based driving models.

1.1.2 Explainability: Taxonomy of Terms

Many terms are related to the *explainability* concept and several definitions have been proposed for each of these terms. The boundaries between concepts are fuzzy and constantly evolving. To clarify and narrow the scope of the survey, we detail here common definitions of key concepts related to explainable AI, and how they are related to one another as illustrated in Fig. 1.

In human-machine interactions, *explainability* is defined as the ability for the human user to understand the agent's logic (Rosenfeld and Richardson, 2020). The explanation

is based on how the human user understands the connections between inputs and outputs of the model. According to Doshi-Velez and Kortz (2017), an explanation is a human-interpretable description of the process by which a decision-maker took a particular set of inputs and reached a particular conclusion. In practice, Doshi-Velez and Kortz (2017) state that an explanation should answer at least one of the three following questions: *what were the main factors in the decision?* *Would changing a certain factor have changed the decision?* and *Why did two similar-looking cases get different decisions, or vice versa?*

The term *explainability* often co-occurs with the concept of *interpretability*. While some recent work (Beaudouin et al., 2020) advocate that the two are synonyms, (Gilpin et al., 2018) use the term *interpretability* to designate to which extent an explanation is understandable by a human. For example, an exhaustive and completely faithful explanation is a description of the system itself and all its processing: this is a complete explanation although the exhaustive description of the processing may be incomprehensible. Gilpin et al. (2018) state that an explanation should be designed and assessed in a trade-off between its *interpretability* and its *completeness*, which measures how accurate the explanation is as it describes the inner workings of the system. The whole challenge in explaining neural networks is to provide explanations that are both interpretable and complete.

Interpretability may refer to different concepts, as explained by Lipton (2018). In particular, interpretability regroups two main concepts: model *transparency* and *post-hoc* interpretability. Increasing *model transparency* amounts to gaining an understanding of *how the model works*. For example, Guidotti et al. (2018) explain that a decision model is transparent if its decision-making process can be directly understood without any additional information; if an external tool or model is used to explain the decision-making process, the provided explanation is not transparent according to Rosenfeld and Richardson (2020). For Choi and Ji (2015), the system transparency can be measured as the degree to which users can understand and predict the way autonomous vehicles operate. On the other hand, *post-hoc* interpretability relates to the fact that system comprehension is gained *after* the model has been trained or after the system has been produced. This can be the case for a specific instance, I.E., local interpretability, or, more generally, to explain the whole model and/or its processing and representations.

An important aspect for explanations is the notion of *correctness* or *fidelity*. They designate whether the provided explanation accurately depicts the internal process leading to the output/decision (Xie et al., 2020). In the case of transparent systems, explanations are faithful by design, however, this is not guaranteed with post-hoc explanations which may be chosen and optimized their capacity to persuade users instead of accurately unveiling the system's inner workings.

Table 1 The four W's of explainable driving AI

Who?	Why?	What?	When?
End user, citizen	Trust, situation management	Intrinsic explanations, post-hoc explanations, persuasive explanations	Before/After
Designer, certification body	Debug, understand limitations and shortcomings, improve future versions, machine teaching	Stratified evaluation, corner cases, intrinsic explanations, post-hoc explanations	Before/After
Justice, regulator, insurance	Liability, accountability	Exhaustive and precise explanations, complete explanations, post-hoc explanations, training and validation data	After

Who needs explanations? What kind? For what reasons? When?

Finally, it is worth mentioning that explainability in general—and interpretability and transparency in particular—serve and assist broader concepts such as traceability, auditability, liability, and accountability (Beaudouin et al., 2020).

1.1.3 Contextual Elements of an Explanation

The relation with Autonomous Vehicles differs a lot given who is interacting with the system: surrounding pedestrians and end-users of the ego-car put their life in the hand of the driving system and thus need to gain trust in the system; designers of self-driving systems seek to understand limitations and shortcomings of the developed models to improve next versions; insurance companies and certification organizations need guarantees about the autonomous system. These categories of stakeholders have varying expectations and thus the need for explanations has different motivations. The discussions of this subsection are summarized in Table 1.

Car users, citizens and trust.

There is a long and dense line of research trying to define, characterize, evaluate, and increase the trust between an individual and a machine (Lee and Moray, 1992, 1994; Lee and See, 2004; Choi and Ji, 2015; Shariff et al., 2017; Du et al., 2019; Shen et al., 2020; Zhang et al., 2020). Importantly, trust is a major factor for users' acceptance of automation, as was shown in the empirical study of Choi and Ji (2015). Lee and See (2004) define trust between a human and a machine as “*the attitude that an agent will help achieve an individual's goal, in a situation characterized with uncertainty and vulnerability*”. According to Lee and Moray (1992), human-machine trust depends on three main factors. First, performance-based trust is built relatively to how well the system performs at its task. Second, process-based trust is a function of how well the human understands the methods used by the system to complete its task. Finally, purpose-based trust reflects the designer's intention in creating the system.

In the more specific case of autonomous driving, Choi and Ji (2015) define three dimensions for trust in an Autonomous Vehicle. The first one is *system transparency*, which refers to which extent the individual can predict and understand the operating of the vehicle. The second one is *technical competence*, I.E., the perception by the human of the vehicle's performance. The third dimension is *situation management*, which is the belief that the user can take control whenever desired. As a consequence of these three dimensions of trust, Zhang et al. (2020) propose several key factors to positively influence human trust in Autonomous Vehicles. For example, improving the system performance is a straightforward way to gain more trust. Another possibility is to increase *system transparency* by providing information that will help the user understand how the system functions. Therefore, it appears that the capacity to explain the decisions of an Autonomous Vehicle has a significant impact on user trust, which is crucial for broad adoption of this technology. Besides, as argued by Haspiel et al. (2018), explanations are especially needed when users' expectations have been violated as a way to mitigate the damage.

Research on human-computer interactions argues that the timing of explanations is important for trust. (Haspiel et al., 2018; Du et al., 2019) conducted a user study showing that, to promote trust in the Autonomous Vehicle, explanations should be provided *before* the vehicle takes action rather than after. Apart from the moment when the explanation should appear, Rosenfeld and Richardson (2020) advocate that users are not expected to spend a lot of time processing the explanation, which is why it should be concise and direct. This is in line with other findings of Shariff et al. (2017); Koo et al. (2015) who show that although transparency can improve trust, providing too much information to the human end-user may cause anxiety by overwhelming the passenger and thus decrease trust. Lastly, Rezvani et al. (2016) show that expressing uncertainty about the autonomous system may lead to a drop of trust from users.

System designers, certification, debugging and improvement of models.

Driving is a high-stake critical application, with strong safety requirements. The concept of Operational Design Domain (ODD) is often used by carmakers to designate the conditions under which the car is expected to behave safely. Thus, whenever a machine learning model is built to address the task of driving, it is crucial to know and understand its failure modes, I.E., in the case of accidents (Chan et al., 2016; Zeng et al., 2017; Suzuki et al., 2018; Kim et al., 2019; You and Han, 2020; Li et al., 2018b), and to verify that these situations do not overlap with the ODD. To this end, explanations can provide technical information about the current limitations and shortcomings of a model.

The first step is to characterize the performance of the model. While performance is often measured as an averaged metric on a test set, it may not be enough to reflect the strengths and weaknesses of the system. A common practice is to stratify the evaluation into situations, so that failure modes could be highlighted. This type of method is used by the European New Car Assessment Program (Euro NCAP) to test and assess assisted driving functionalities in new vehicles. Such evaluation method can also be used at the development step, as in (Bansal et al., 2019) where authors build a real-world driving simulator to evaluate their system on controlled scenarios. When these failure modes are found in the behavior of the system, the designers of the model can augment the training set with these situations and re-train the model (Pei et al., 2019).

However, even if these global performance-based explanations are helpful to improve the model's performance, this virtuous circle may stagnate and not be sufficient to solve some types of mistakes. It is thus necessary to delve deeper into the inner workings of the model and to understand *why* it makes those errors. Practitioners will look for explanations that provide insights into the network's processing. Researchers may be interested in the regions of the image that were the most useful for the model's decision (Bojarski et al., 2018), the number of activated neurons for a given input (Tian et al., 2018), the measure of bias in the training data (Torralba and Efros, 2011), etc.

This being said, conducting a rigorous validation of a machine learning-based system is a hard problem, mainly because it is not trivial to specify the requirements a neural network should meet (Borg et al., 2019).

Regulators and legal considerations.

In the European General Data Protection Regulation (GDPR),¹ it is stated that “data subjects” have the right to obtain explanations from automated decision-making systems that may significantly affect them. These explanations should provide “*meaningful information about the logic involved*” in the decision-making process. Algorithms are

expected to be available for the scrutiny of their inner workings (possibly through counterfactual interventions (Wachter et al., 2017; Rathi, 2019)), and their decisions should be available for contesting and contradiction. This should prevent unfair and/or unethical behaviors of algorithms. In the context of autonomous driving, the issues raised by personal data are generally related to storage and anonymization of raw data that contain people or license plates. This has indeed triggered interesting research problems, such as automatic anonymization (Sun et al., 2018; Maximov et al., 2020) or training with anonymized data (Ren et al., 2018; Tomei et al., 2021), but they are beyond the scope of our survey.

Legal institutions are interested in explanations for *liability* and *accountability* purposes, especially when a self-driving system is involved in a car accident. As noted in (Beaudouin et al., 2020), detailed explanations of all aspects of the decision process could be required to identify the reasons for a malfunction. This aligns with the guidelines towards algorithmic transparency and accountability published by the Association for Computing Machinery (ACM), which state that system auditability requires logging and record keeping (Garfinkel et al., 2017). In contrast with this *local* form of explanations, a more *global* explanation of the system's functioning could be required in a lawsuit. It consists in full or partial disclosure of source codes, training or validation data, or thorough performance analysis. It may also be important to provide information about the system's general logic that could be understandable, such as the goals of the loss function.

Notably, explanations generated for legal or regulatory institutions are likely to be different from those addressed to the end-user. Here, explanations are expected to be exhaustive and precise, as the goal is to take a deep delve into the inner workings of the system. These explanations are directed towards experts who will likely spend large amounts of time studying the system (Rosenfeld and Richardson, 2020), and who are thus inclined to receive rich explanations with great amounts of detail.

1.2 Autonomous Driving: Learning-Based Self-Driving Models

The development of autonomous vehicles has the potential to reduce crashes, fuel consumption, congestions, and to increase personal mobility (Anderson et al., 2014). Research on autonomous vehicles is blooming thanks to recent advances in deep learning and computer vision (Krizhevsky et al., 2012; LeCun et al., 2015), as well as the development of autonomous driving datasets and simulators (Geiger et al., 2013; Dosovitskiy et al., 2017; Caesar et al., 2020; Yu et al., 2020). The number of academic publications on this subject is rising in most machine learning, computer vision, robotics and transportation conferences, and journals.

¹ <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN>.

On the industry side, manufacturers are already producing cars equipped with advanced computer vision technologies for automatic lane following, assisted parking, or collision detection among other things; several automotive companies are designing prototypes with level 4 and 5 autonomy.

1.2.1 From Modular Pipelines to End-to-End Learning

The history of autonomous driving systems started in the late '80s and early '90s with the European Eureka project called Prometheus (Dickmanns, 2002). This has later been followed by driving challenges proposed by the Defense Advanced Research Projects Agency (DARPA). In 2005, STANLEY (Thrun et al., 2006) is the first autonomous vehicle to complete a Grand Challenge, which consists in a race of 142 miles in a desert area. Two years later, DARPA held the Urban Challenge, where autonomous vehicles had to drive in an urban environment, taking into account other vehicles and obeying traffic rules. BOSS won the challenge (Urmson et al., 2008), driving 97 km in an urban area, with a speed up to 48 km/h. STANLEY, BOSS and the vast majority of the other approaches at this time (Leonard et al., 2008) are systems composed of several sub-modules, each completing a very specific task. Broadly speaking, these sub-tasks deal with sensing the environment, forecasting future events, planning, taking high-level decisions, and controlling the vehicle.

As pipeline architectures split the driving task into easier-to-solve problems, they offer somewhat interpretable processing of sensor data through specialized modules (perception, planning, decision, control). However, these approaches have several drawbacks. First, they rely on human heuristics and manually-chosen intermediate representations, which are not proven to be optimal for the driving task. Second, their handcrafted nature limits their ability to account for real-world uncertainties, and therefore to generalize to scenarios that were not envisioned by system designers. Moreover, from an engineering point of view, these systems are hard to scale and to maintain as the various modules are entangled together (Chen et al., 2020a). Finally, they are prone to error propagation between the multiple sub-modules (McAllister et al., 2017).

To circumvent these issues, and nurtured by the deep learning revolution (Krizhevsky et al., 2012; LeCun et al., 2015), researchers focus more and more on learning-based driving systems, and in particular on neural networks. Inspired by the seminal work of Pomerleau (1988) who designed the first vision-based neural driving model, networks are now trained either by leveraging large quantities of expert recordings (Bojarski et al., 2016; Codevilla et al., 2018; Ly and Akhloufi, 2020) or through simulation (Espíe et al., 2005; Toromanoff et al., 2020; Dosovitskiy et al., 2017). In both cases, these systems learn a highly complex transformation that operates over input sensor data and produces end-commands

(steering angle, throttle). While these neural driving models overcome some of the limitations of the modular pipeline stack, they are sometimes described as *black-boxes* for their critical lack of transparency and interpretability. Indeed, as they are trained within the deep learning paradigm, they fall into the known shortcomings associated with these architectures. In this paper, we focus on models learned by *behavior cloning*, which leverage datasets of human driving sessions, as opposed to *reinforcement learning* approaches which train models through trial-and-error simulation.

1.2.2 Autonomous Vehicles as Machine Learning Models

Explainability hurdles of self-driving models are shared with most deep learning models, across many application domains. Indeed, decisions of deep systems are intrinsically hard to explain as the functions these systems represent, mapping from inputs to outputs, are not transparent. In particular, although it may be possible for an expert to broadly understand the structure of the model, the parameter values, which have been *learned*, are yet to be explained.

From a machine learning perspective, there are several factors giving rise to interpretability problems for self-driving systems, as machine learning researchers do not have exhaustive control over the dataset, the trained model, and the learning phase. Explainability methods aim at providing cues to pass these barriers, and answer some of the questions detailed in Fig. 2.

First, the dataset used for training brings interpretability problem, with questions such as: *Has the model encounter situations like X?* Indeed, a finite training dataset cannot exhaustively cover all possible driving situations and it will likely under- and over-represent some specific ones (Tommasi et al., 2017). Moreover, datasets contain numerous biases of various nature (omitted variable bias, cause-effect bias, sampling bias), which also gives rise to explainability issues related to fairness (Mehrabi et al., 2019).

Second, the trained model, and the mapping function it represents, is poorly understood. The model is highly non-linear and does not provide any robustness guarantee as small input changes may dramatically change the output behavior. Also, these models are known to be prone to adversarial attacks (Morgulis et al., 2019; Deng et al., 2020). Explainability issues thus occur regarding the generalizability and robustness aspects: *How will the model behave under these new scenarios?*

Third, the learning phase is not perfectly understood. Among other things, there are no guarantees that the model will settle at a minimum point that generalizes well to new situations, and that the model does not underfit on some situations and overfit on others. Also, the model may learn to ground its decisions on spurious correlations during training instead of leveraging causal signals (Codevilla et al., 2019;

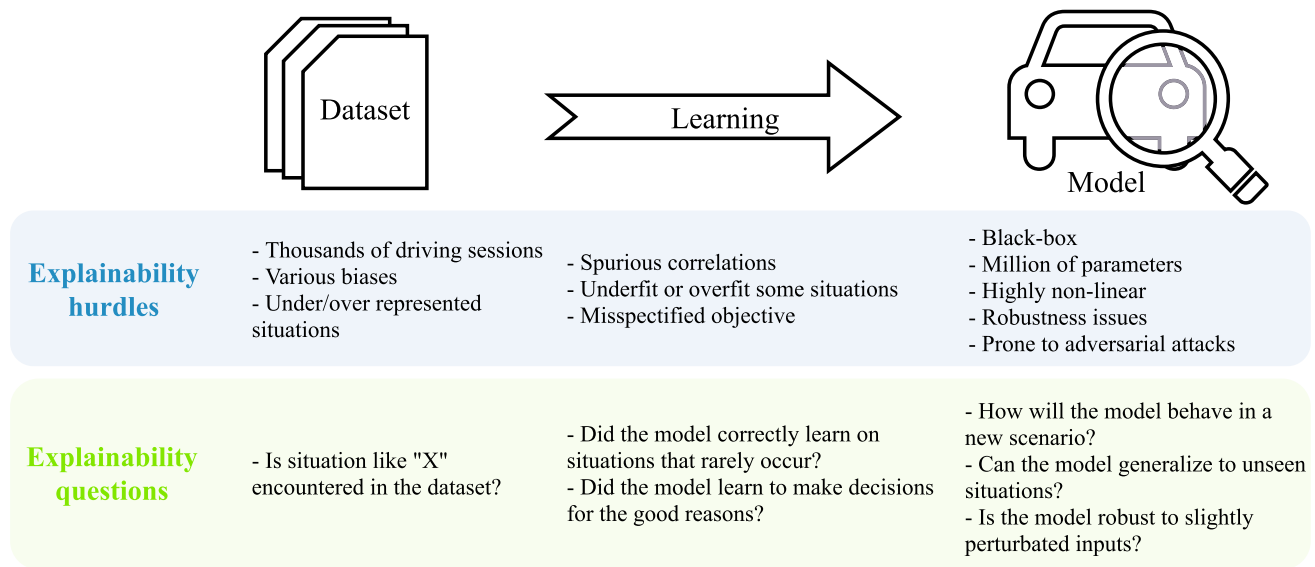


Fig. 2 Explainability hurdles and questions for autonomous driving models, as seen from a machine learning point of view

de Haan et al., 2019). We aim at finding answers to questions like *Which factors caused this decision to be taken?*

The three aforementioned barriers to model understanding are common to the vast majority learning-based systems that consume real-world data. Additionally, some specific traits of the self-driving application should be considered. In particular, self-driving systems have to simultaneously solve intertwined tasks of very different natures, including perception tasks, motion forecasting, planning and control. Explaining a self-driving system thus means disentangling these implicit task, and to make them human-interpretable.

1.2.3 Autonomous Vehicles as Heterogeneous Systems

For humans, the complex task of driving involves solving many intermediate sub-problems, at different levels of hierarchy (Michon, 1984). In the effort towards building an autonomous driving system, researchers aim at providing the machine with these intermediate capabilities. Thus, explaining the general behavior of Autonomous Vehicles inevitably requires understanding how each of these intermediate steps is carried and how it interacts with others, as illustrated in Fig. 3. We can categorize these capabilities into three types:

- **Perception** information about the system’s understanding of its local environment. This includes the objects that have been recognized and assigned to a semantic label (persons, cars, urban furniture, driveable area, crosswalks, traffic lights), their localization, properties of their motion (velocity, acceleration), intentions of other agents, etc.;

- **Reasoning** information about how the different components of the perceived environment are organized and assembled by the system. This includes global explanations about the rules that are learned by the model, instance-wise explanation showing which objects are relevant in a given scene (Bojarski et al., 2018), traffic pattern recognition (Zhang et al., 2013), object occlusion reasoning (Wojek et al., 2011, 2013);
- **Decision** information about how the system processes the perceived environment and its associated reasoning to produce a decision. This decision can be a high-level goal such as “*the car should turn right*”, a prediction of the ego vehicle’s trajectory, its low-level relative motion or even the raw controls, etc.

While the separation between perception, reasoning, and decision is clear in modular driving systems, recent end-to-end neural networks blur the lines and perform these simultaneously (Bojarski et al., 2016; Zeng et al., 2019; Casas et al., 2021; Chitta et al., 2021). However, despite the efficiency and flexibility of end-to-end approaches, they leave small room for structured modeling of explanations, which would give the end-user a thorough understanding of how each step is achieved. Indeed, when an explanation method is developed for a neural driving system, it is often not clear whether it attempts to explain the perception, the reasoning, or the decision step. Considering the nature of neural networks architecture and training, disentangling perception, reasoning, and decision in neural driving systems constitutes a non-trivial challenge.

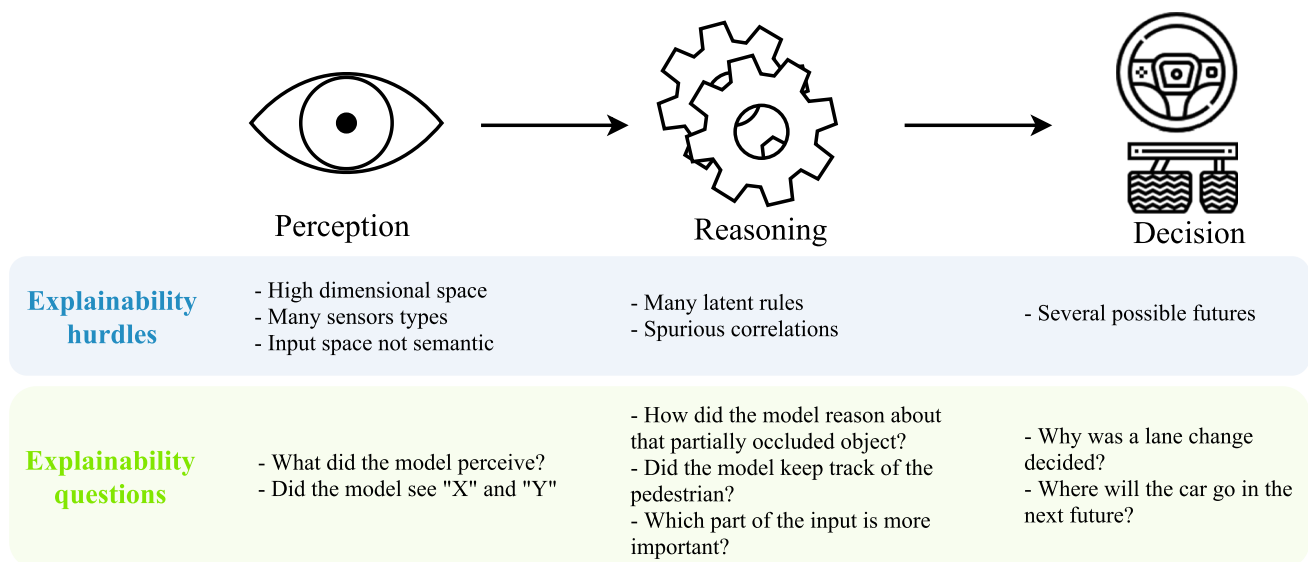


Fig. 3 Explainability hurdles and questions for autonomous driving models, as seen from an autonomous driving point of view

1.3 Survey Organization

In this survey, we organize and review deep and vision-based self-driving models under the light of explainability. The scope is thus different from papers that review self-driving models in general. For example, Janai et al. (2020a) review vision-based problems arising in self-driving research, Di and Shi (2020) provide a high-level review on the link between human and automated driving, Ly and Akhloufi (2020) review imitation-based self-driving models, Manzo et al. (2020) survey deep learning models for predicting steering angle, Brown et al. (2020) review approaches modeling dynamics of interactive multi-agent traffic, and Kiran et al. (2020) review self-driving models based on deep reinforcement learning.

Likewise, there exist reviews on X-AI, interpretability, and explainability in machine learning in general (Beaudouin et al., 2020; Gilpin et al., 2018; Adadi and Berrada, 2018; Das and Rad, 2020). Among others, Xie et al. (2020) give a pedagogic review for non-expert readers while Vilone and Longo (2020) offer the most exhaustive and complete review on the X-AI field. Moraffah et al. (2020) focus on *causal* interpretability in machine learning. Moreover, there also exist reviews on explainability applied to decision-critical fields other than driving. This includes interpretable machine learning for medical applications (Tjoa and Guan, 2019; Fellous et al., 2019). Concurrently to our work, the very recent survey of Omeiza et al. (2021) demonstrates a growing interest in the research community for explainability of autonomous driving models. Finally, while there are links between the fields of model *validation* (Tian et al., 2018; Koren et al., 2018; Corso et al., 2019) and model *explainability*, we restrict the survey to the latter.

Overall, the goal of this survey is diverse, and we hope that it contributes to the following:

- Interpretability and explainability notions are clarified in the context of autonomous driving, depending on the type of explanations and how they are computed;
- Legal and regulator bodies, engineers, technical and business stakeholders can learn more about explainability methods and approach them with caution regarding presented limitations;
- Self-driving researchers are encouraged to explore new directions from the X-AI literature such as causality, to foster explainability and reliability of self-driving systems;
- The quest for interpretable models can contribute to other related topics such as fairness, privacy, and causality, by making sure that models are taking good decisions for good reasons.

Throughout the survey, we identify limitations and shortcomings from X-AI methods and propose several future research directions to have potentially more transparent, richer, and more faithful explanations for upcoming generations of self-driving models. For the sake of simplicity and with autonomous driving research in mind, we classify methods into two main categories. Section 2 presents the first category, I.E., explainability methods that are applied to an already-trained deep network and are designed to provide *post-hoc* explanations. Section 3 turns to approaches of the second category: methods providing upfront more transparency and interpretability to self-driving models' processing, by adding explainability constraints in the design of the systems. This section also presents potential future

Table 2 Selected references aiming at explaining a learning-based driving model

Approach	Explanation type	Section	Selected references
Local	Saliency map	2.1.1	VisualBackprop (Bojarski et al., 2018, 2017)
			Causal filtering (Kim and Canny, 2017)
			Grad-CAM (Sauer et al., 2018)
			Meaningful Perturbations (Liu et al., 2020)
Global	Counterfactual interventions	2.1.2	Shifting objects (Bojarski et al., 2017)
			Removing objects (Li et al., 2020c)
			Causal factor identification (Bansal et al., 2019)
			∅
Global	Model translation	2.2.1	∅
	Representations	2.2.2	Neuron coverage (Tian et al., 2018)

directions to increase further explainability of self-driving systems. Finally, Sect. 4 presents the particular use-case of explaining a self-driving system by means of natural language justifications.

2 Explaining a Deep Driving Model

When a deep learning model in general—or a self-driving model more specifically—comes as an opaque black-box as it has not been designed with a specific explainability constraint, *post-hoc* methods have been proposed to gain interpretability from the network processing and its representations. Post-hoc explanations have the advantage of giving an interpretation to black-box models without conceding any predictive performance. In this section, we assume that we have a pre-trained model f . Two main categories of post-hoc methods can be distinguished to explain f : *local* methods which explain the prediction of the model for a specific instance (Sect. 2.1), and *global* methods that seek to explain the model in its entirety (Sect. 2.2), I.E., by gaining a finer understanding on learned representations and activations. Selected references from this section are reported in Table 2.

2.1 Local Explanations

Given an input image x , a *local explanation* aims at justifying why the model f gives its specific prediction $y = f(x)$. In particular, we distinguish two types of approaches: post-hoc saliency methods which determine regions of image x influencing the most the decision (Sect. 2.1.1) and counterfactual analysis which aims to find the cause in x that made the model predict $f(x)$ (Sect. 2.1.2).

2.1.1 Post-hoc Saliency Methods

X-Ai Background. A post-hoc *saliency* method aims at explaining which input image's regions influence the most

the output of the model. These methods produce a *saliency map* (A.K.A. *heat map*) that highlights regions on which the model relied the most for its decision. There are three main lines of methods to obtain a saliency map for a trained network, namely *back-propagation methods*, *perturbation-based methods* and *local approximation* methods.

Back-propagation methods retro-propagate output information back into the network and evaluate the gradient of the output with respect to the input, or intermediate feature-maps, to generate a heat-map of the most contributing regions. These methods include DeConvNet (Zeiler and Fergus, 2014) and improved versions (Simonyan et al., 2014; Oramas et al., 2019), Class Activation Mapping (CAM) (Zhou et al., 2016), Grad-CAM (Selvaraju et al., 2020), Layer-Wise Relevance Propagation (LRP) (Bach et al., 2015), deepLift (Shrikumar et al., 2017), and Integrated Gradients (Sundararajan et al., 2017).

Perturbation-based methods estimate the importance of an input region by observing how modifications in this region impacts the prediction. These modifications include editing methods such as pixel (Zeiler and Fergus, 2014) or super-pixel (Ribeiro et al., 2016) occlusion, greying out (Zhou et al., 2015a) or blurring (Fong and Vedaldi, 2017) image regions.

Local approximation methods approach the behavior of a trained model in the vicinity of the instance to be explained, with a simpler model. In practice, a separate model, inherently interpretable, is built to act as a proxy for the input/output mapping of the main model locally around the instance of interest. In the Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016), this simpler model is defined as a decision tree or a linear model, whereas if-then rules extraction is explored in (Ribeiro et al., 2018). SHAP (Lundberg and Lee, 2017) has been introduced to generalize LIME, and provides more consistent results.

Applications For Driving Models. In the autonomous driving literature, post-hoc saliency methods have been employed to highlight image regions that influence the most driving decisions. By doing so, these methods mostly explain



Fig. 4 Example of salient pixels (in green) obtained by VisualBackprop (Bojarski et al., 2018). They highlight the central lane delimitation, the right edge of the road and other vehicles. Credits to (Bojarski et al., 2017)

the perception part of the driving architectures. The first post-hoc saliency method to visualize the input influence in the context of autonomous driving has been developed by Bojarski et al. (2018). The VisualBackprop method they propose identifies sets of pixels by backpropagating activations from both late layers, which contain relevant information for the task but have a coarse resolution, and early layers which have a finer resolution. The algorithm runs in real-time and can be embedded in a self-driving car. This method has been used by Bojarski et al. (2017) to explain PilotNet (Bojarski et al., 2016), a deep end-to-end opaque self-driving architecture. As seen in Fig. 4, they qualitatively validate that the model correctly grounds its decisions on lane markings, edges of the road (delimited with grass or parked cars), and surrounding cars.

The VisualBackprop procedure has also been employed by Mohseni et al. (2019) to gain more insights into the PilotNet architecture and its failures in particular. They use saliency maps to predict model failures by training a student model that operates over saliency maps and tries to predict the error made by the PilotNet. They find that saliency maps given by the VisualBackprop are better suited than raw input images to predict model failure, especially in case of adverse conditions.

Kim and Canny (2017) propose a post-hoc saliency visualization method for self-driving models built with an attention mechanism. They explain that attention maps comprise “blobs” and argue that while some input blobs have a true causal influence on the output, others are spurious. Thus, they propose to segment and filter out about 60% spurious blobs to produce simpler *causal* saliency maps, derived from attention maps in a post-hoc analysis. To do so, they measure a decrease in performance when a local visual blob from an input raw image is masked out. Qualitatively, they find that the network cues on features that are also used by humans while driving, including surrounding cars and lane markings for example.

Recently, Sauer et al. (2018) propose to condition the post-hoc saliency visualization on a variety of driving features, namely driving “affordances”. They employ the Grad-CAM saliency technique (Selvaraju et al., 2020) on an end-to-mid self-driving model trained to predict driving affordances on

a dataset recorded from the CARLA simulator (Dosovitskiy et al., 2017). They argue that post-hoc saliency methods are particularly well suited for this type of architecture on the contrary to end-to-end models, as all of the perception (E.G., detection of speed limits, red lights, cars, *etc.*) is mapped to a single control output for those models. Instead, in their case, they can analyze the post-hoc saliency in the input image for each affordance, E.G., “hazard stop” or “red light”.

Still in the context of driving scenes, although not properly for explaining a self-driving model, it is worth mentioning that Liu et al. (2020) use the perturbation-based masking strategy of Fong and Vedaldi (2017) to obtain saliency maps for a driving scene classification model trained on the HDD dataset (Ramanishka et al., 2018).

Challenges. While post-hoc saliency methods enable visual explanations for deep black-box models, they come with some limitations. First, they are hard to evaluate. For example, human evaluation can be employed (Ribeiro et al., 2016; Lu et al., 2021) but this comes with the risk of selecting methods which are more *persuasive*, I.E., plausible and convincing and not necessarily *faithful*.

Another possibility to evaluate post-hoc saliency methods is to use automated metrics. In the work of Samek et al. (2017), saliency maps are evaluated by measuring the decrease of the output score induced by the removal of salient pixels of the validation set. Closely related, the Remove and Retrain (ROAR) scheme (Hooker et al., 2019) hides the most salient pixels of training images in addition to the validation set, and re-trains image classifier on this modified training set. The intuition is that the new model should perform poorly as it has been trained without salient regions that were considered important for the task. Other methods involve additional ground-truth locations of image regions that cause the output. These annotations can either be annotated by humans (Fong and Vedaldi, 2017), or obtained automatically with synthetic data (Oramas et al., 2019; Arras et al., 2022). Second, Adebayo et al. (2018) indicate that the generated heat maps may be misleading as some post-hoc saliency methods are independent both of the model and the data. Indeed, they show that some post-hoc saliency methods behave like edge-detectors even when they are applied to a randomly initialized model. Third, Ghorbani et al. (2019) show that it is possible to attack visual saliency methods so that the generated heat-maps do not highlight important regions anymore, while the predicted class remains unchanged. Lastly, different saliency methods produce different results and it is not obvious to know which one is correct, or better than others. In that respect, a potential research direction is to learn to combine explanations coming from various explanation methods.

Interestingly, these saliency methods provide explanations that can be processed very rapidly, as they are visual in essence. Thus, they constitute relevant candidates to be shown to regular car users, who will not spend too much time

analyzing explanations. However, as it has been discussed in Sect. 1.1.3, the purpose of these explanations geared towards car users is mainly to build and enhance trust in the system. Considering the limitations we just discussed, a special care should be given in designing and providing saliency maps that indeed reflect the true behavior of the model.

2.1.2 Counterfactual Explanation

X-Ai Background. Recently, a lot of attention has been put on *counterfactual analysis*, a field from the causal inference literature (Pearl, 2009; Moraffah et al., 2020). A counterfactual analysis aims at finding features X within the input x that *caused* the decision $y = f(x)$ to be taken, by imagining a new input instance x' where X is changed and a different outcome y' is observed. The new imaginary scenario x' is called a *counterfactual example* and the different output y' is a *contrastive class*. The new counterfactual example, and the change in X between x and x' , constitute *counterfactual explanations*. In other words, a counterfactual example is a modified version of the input, in a minimal way, that changes the prediction of the model to the predefined output y' . For instance, in an autonomous driving context, it corresponds to questions like “What should be different in this scene, such that the car would have stopped instead of moving forward?” We highlight the difference between saliency methods and counterfactual explanations. Saliency methods answer a “*where* question” as they are limited to find input regions that are the most influential for the decision. On the contrary, counterfactual explanations answer a more precise “*what* question” as they seek minimal modifications of the input that switch the decision of the model.

Several requirements should be imposed to find counterfactual examples. First, the prediction $f(x')$ of the counterfactual example must be close to the desired contrastive class y' . Second, the counterfactual change must be *minimal*, I.E., the new counterfactual example x' must be as similar as possible to x , either by making sparse changes or in the sense of some distance. Third, the counterfactual change must be *relevant*, I.E., new counterfactual instances must be likely in the underlying input data distribution. The simplest strategy to find counterfactual examples is the naive trial-and-error strategy, which finds counterfactual instances by randomly changing input features. More advanced protocols have been proposed, for example Wachter et al. (2017) propose to minimize both the distance between the model prediction $f(x')$ for the counterfactual x' and the contrastive output y' and the distance between x and x' .

Traditionally, counterfactual explanations have been developed for classification tasks, with a low-dimensional semantic input space, such as the credit application prediction task (Wachter et al., 2017). It is worth mentioning that there also exist *model-based* counterfactual explanations which

aim at answering questions like “What decision would have been taken if this model component was not part of the model or designed differently?” (Narendra et al., 2018; Harradon et al., 2018). To tackle this task, the general idea is to model the deep network as a Functional Causal Model (FCM) on which the causal effect of a model component can be computed with causal reasoning on the FCM (Pearl, 2009). For example, this has been employed to gain an understanding of the latent space learned in a variational autoencoder (VAE) or a generative adversarial network (GAN) (Besserve et al., 2020), or in RL to explain agent’s behavior with counterfactual examples by modeling them with an SCM (Madumal et al., 2020). Counterfactual explanations have the advantage that they do not require access to the dataset nor the model to be computed. This aspect is important for automotive stakeholders who own datasets and industrial property of their model and who may lose a competitive advantage by being forced to disclose them. Moreover, counterfactual explanations are GDPR compliant (Wachter et al., 2017). A potential limit of counterfactual explanations is that they are not unique: distinct explanations can explain equally well the same situation while contradicting each other.

When dealing with a high-dimensional input space—as it is the case with images and videos—counterfactual explanations are very challenging to obtain as naively producing examples under the requirements specified above leads to new instances x' that are imperceptibly changed with respect to x while having output $y' = f(x')$ dramatically different from $y = f(x)$. This can be explained given that the problem of adversarial perturbations arises with high dimensional input space of machine learning models, neural networks in particular (Szegedy et al., 2014). To mitigate this issue in the case of image classification, Goyal et al. (2019) use a specific instance, called a *distractor* image, from the predefined target class and identify the spatial region in the original input such that replacing them with specific regions from the distractor image would lead the system to classify the image as the target class. In addition, Hendricks et al. (2018) provide counterfactual explanations by staying at the attribute level and by augmenting the training data with negative examples created with hand-crafted rules.

The automatic evaluation of counterfactual explanations must be conducted on several fronts. First, counterfactual changes must change the decision of the main model. Second, they must be realistic and the resulting altered image or video must belong to the data distribution, as measured with the Frechet Inception Distance (FID) (Heusel et al., 2017). Lastly, counterfactual changes must be minimal and sparse, for example, as measured by an oracle model estimating the number of attribute changes between two inputs (Singla et al., 2020; Rodríguez et al., 2021).

Applications For Driving Models. Regarding the autonomous driving literature, there only exists a limited

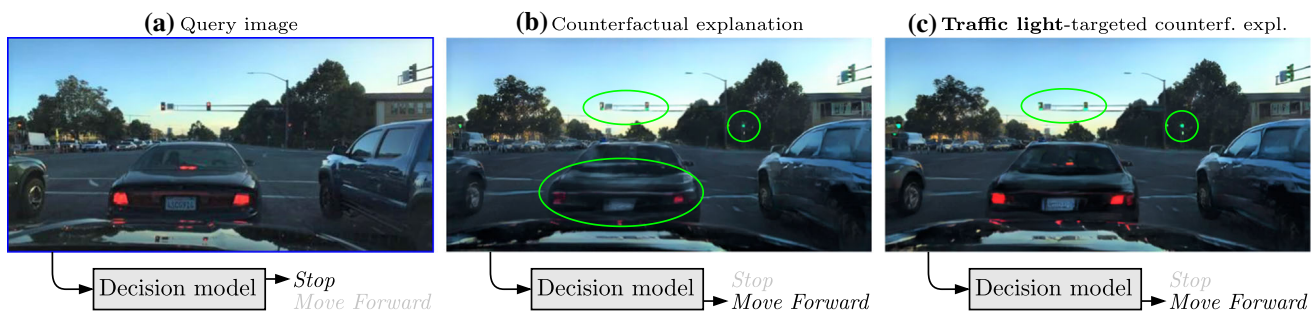


Fig. 5 Counterfactual explanations generated by STEEX (Jacob et al., 2021). The ‘Decision model’ is a binary classifier that predicts whether or not it is possible to move forward. To explain that model, STEEX gen-

erates counterfactual explanations (b). It can also provide explanations that target a specified region (c)

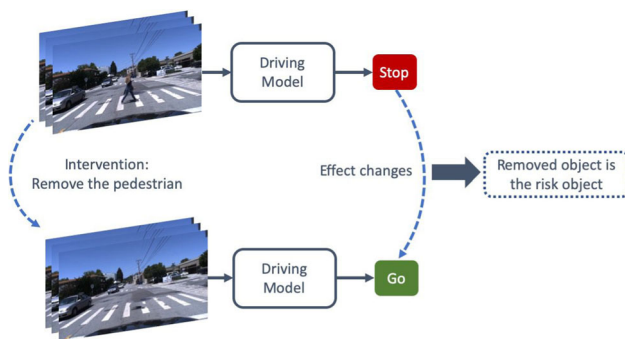


Fig. 6 Counterfactual intervention to measure the causal impact of an input region. Removing a pedestrian through an object-level manipulable driving model induces a change in the driver’s decision from *Stop* to *Go*, which indicates that the pedestrian is a risk-object, I.E., the cause for *Stop*. Credits to (Li et al., 2020c)

number of approaches involving counterfactual interventions. When the input space has semantic dimensions and can thus be easily manipulated, it is easy to check for the causality of input factors by intervening on them (removing or adding). For example, Bansal et al. (2019) investigate the causal factors for specific outputs: they test the ChauffeurNet model under hand-designed inputs where some objects have been removed. With a high-dimensional input space (E.G., pixels), Bojarski et al. (2017) propose to check the causal effect that image parts have, with a saliency visualization method. In particular, they measure the effect of *shifting* the image regions that were found salient by VisualBackProp on the PilotNet architecture. They observe that translating only these image regions, while maintaining the position of other non-salient pixels, leads to a significant change in the steering angle output. Moreover, translating non-salient image regions, while maintaining salient ones, leads to almost no change for the output of PilotNet. This analysis indicates a causal effect of the salient image regions.

More recently, Li et al. (2020c) introduce a causal inference strategy for the identification of “risk-objects”, I.E., objects that have a causal impact on the driver’s behavior (see Fig. 6). The task is formalized with an FCM and objects

are removed in the input stream to simulate causal effects, the underlying idea being that removing non-causal objects will not affect the behavior of ego vehicles. Under this setting, they do not require strong supervision about the localization of risk-objects, but only the high-level behavior label (‘go’ or ‘stop’), as provided in the HDD dataset (Ramanishka et al., 2018) for example. They propose a training algorithm with interventions, where some objects are randomly removed in scenes where the output is ‘go’. The object removal is instantiated with partial convolutions (Liu et al., 2018). At inference, in a sequence where the car predicts ‘stop’, the risk-object is found as the one which gives the higher score to the ‘go’ class. Finally, the model STEEX (Jacob et al., 2021) presents a counterfactual method to explain deep visual models. In their case, the driving model is highly simplified and instantiated with a binary classifier *Stop/Move Forward* that operates on images taken by a frontal camera. With the help of a pretrained generative model, STEEX generates counterfactual explanations where the scene structure is preserved and only the style of each region is allowed to be modified. Moreover, their architecture can generate counterfactual explanation that target a user-specified region of the query image. Results are shown in Fig. 5.

Challenges. We call the reader’s attention to the fact that analyzing driving scenes and building driving models using causality is far from trivial as it requires the capacity to *intervene* on the model’s inputs. This, in the context of driving, is a highly complex problem to solve for three main reasons.

First, the data is composed of high-dimensional tensors of raw sensor inputs (such as the camera or LiDAR signals) and scalar-valued signals that represent the current physical state of the vehicle (velocity, yaw rate, acceleration, etc.). Performing controlled interventions on these input spaces require the capacity to modify the content of raw high-dimensional inputs (E.G., videos) realistically: changes in the input space such that counterfactual examples still belong to the data distribution, without producing meaningless perturbations alike adversarial ones. Even though some recent works explore realistic alterations of visual content (Gao et al., 2020), this

is yet to be applied in the context of self-driving and this open challenge, shared by other interpretability methods, is discussed in more details in Sect. 3.1.2. Interestingly, as more and more neural driving systems rely on semantic representations, alterations of the input space are simplified as the realism requirement is removed, and synthetic examples can be passed to the model as it has been done in (Bansal et al., 2019).

Second, modified inputs must be coherent and respect the underlying causal structure of the data generation process. Indeed, the different variables that constitute the input space are interdependant, and performing an intervention on one of these variables implies that we can simulate accordingly the reaction of other variables. As an example, we may be provided with a driving scene that depicts a green light, pedestrians waiting and vehicles passing. A simple intervention consisting of changing the state of the light to red would imply massive changes on the other variables to be *coherent*: pedestrians should start crossing the street and vehicles should stop at the red light. The very recent and promising work of Li et al. (2020d) tackles the issue of unsupervised *causal discovery* in videos. They discover a structural causal model in the form of a graph that describes the relational dependencies between variables. Interestingly, this causal graph can be leveraged to perform interventions on the data (E.G., specify the state of one of the variables), leading to an evolution of the system that is coherent with this inferred graph. We believe that the adaptation of this type of approach to real driving data is crucial for the development of causal explainability.

Finally, even if we are able to perform realistic and coherent interventions on the input space, we would need to have annotations for these new examples. Indeed, whether we use those altered examples to train a driving model on or to perform exhaustive and controlled evaluations, expert annotations would be required. Considering the nature of the driving data, it might be hard for a human to provide these annotations: they would need to imagine the decision they would have taken (control values or future trajectory) in this newly generated situation.

2.2 Global Explanations

Global explanations contrast with local explanation methods as they attempt to explain the behavior of a model in general by summarizing the information it contains. They aim at increasing global and holistic *model interpretability*. We cover two families of methods to provide global explanations: *model translation* techniques, which aim at transforming an opaque neural network into a more interpretable model (Sect. 2.2.1) and *representations explanation* which analyze the knowledge contained in the data structures of the model (Sect. 2.2.2).

2.2.1 Model Translation

X-Ai Background. The idea of *model translation* is to transfer the knowledge contained in the main opaque model into a separate machine learning model that is inherently interpretable. Concretely, this involves training an explainable model to mimic the input-output mapping of the black-box function. Despite sharing the same spirit with local approximation methods presented in Sect. 2.1.1, model translation methods are different as they should approximate the main function *globally* across the data distribution. In the work of Zhang et al. (2018a) an explanatory graph is built from a pre-trained convolutional neural net to understand how the patterns memorized by its filters are related to object parts. This graph aims at providing a global view of how visual knowledge is organized within the hierarchy of convolutional layers in the network. Deep neural networks have also been translated into soft decision trees (Frosst and Hinton, 2017) or rule-based systems (Zilke et al., 2016; Sato and Tsukimoto, 2001). The recent work of Harradon et al. (2018) presents a causal model used to explain the computation of a deep neural network. Human-understandable concepts are first extracted from the neural network of interest, using auto-encoders with sparsity losses. Then, the causal model is built using those discovered human-understandable concepts and can quantify the effect of each concept on the network's output.

Challenges. To the best of our knowledge, such strategies have not been used in the autonomous driving literature to visualize and interpret the rules learned by a neural driving system. Indeed, one of the limit of such a strategy lies in the disagreements between the interpretable translated model and the main self-driving model. These disagreements are inevitable as rule-based models or soft-decision trees have a lower capacity than deep neural networks. Moreover, these methods are typically designed to explain deep networks that perform a classification task, which is usually not the case of self-driving models. Being able to adapt these techniques to driving models could certainly be beneficial to engineers and regulatory institutions. Indeed, these methods would provide a direct understanding on the knowledge captured by learning-based algorithms, which could help assess their safety and conformity to societal rules and ethics.

2.2.2 Explaining Representations

X-Ai Background. Representations in deep networks take various forms as they are organized in a hierarchy that encompasses individual units (neuron activation), vectors, and layers (Gilpin et al., 2018). The aim of explaining representations is to provide insights into what is captured by the internal data structures of the model, at different granularities. Representations are of practical importance in transfer learning scenarios, I.E., when they are extracted from a deep

network trained on a task and transferred to bootstrap the training of a new network optimizing a different task.

In practice, the quality of intermediate representations can be evaluated, and thus made partially interpretable, with a proxy transfer learning task (Razavian et al., 2014). At another scale, some works attempt to gain insights into what is captured at the level of an individual neuron (Bau et al., 2017; Zhang and Zhu, 2018). For example, a neuron's activation can be interpreted by accessing input patterns which maximize its activation, for example by sampling such input images (Zhou et al., 2015b; Castrejón et al., 2016), with gradient ascent (Erhan et al., 2009; Simonyan et al., 2014), or with a generative network (Nguyen et al., 2016). To gain more understanding of the content of vector activations, the t-Distributed Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton, 2008) has been proposed to project high-dimensional data into a space of lower dimension (usually 2d or 3d). This algorithm aims at preserving the distances between points in the new space where points are projected. t-SNE has been widely employed to visualize and gain more interpretability from representations, by producing *scatter plots* as explanations. This has for example been employed for video representations (Tran et al., 2015), or deep Q-networks (Zahavy et al., 2016).

Challenges. In the autonomous driving literature, such approaches have not been widely used to the best of our knowledge. The only example we can find is reported in (Tian et al., 2018) which uses the neuron *coverage* concept from (Pei et al., 2019). The neuron coverage is a testing metric for deep networks, that estimates the amount of logic explored by a set of test inputs: more formally the neuron coverage of a set of test inputs is the proportion of unique activated neurons, among all network's neurons for all test inputs. With the idea of detecting erroneous behaviors of deep self-driving models that could lead to potential accidents, Tian et al. (2018) partition the input space according to the neuron coverage by assuming that the model decision is the same for inputs that have the same neuron coverage. With the aim of increasing neuron coverage of the model, they compose a variety of transformation of the input image stream, each corresponding to a synthetic but realistic editing of the scene: linear (E.G., change of luminosity/contrast), affine (E.G., camera rotation) and convolutional (E.G., rain or fog) transformations. This enables them to automatically discover many—synthetic but realistic—scenarios where the predictions are incorrect. Interestingly, they show that the insights obtained on erroneous corner cases can be leveraged to successfully retrain the driving model on the synthetic data to obtain an accuracy boost. Despite not giving explicit explanations about the self-driving model, such predictions help to understand the model's limitations. Overall, inspecting neurons and intermediate representations is a tedious and lengthy operation. The detailed information provided by this

inspection can be very precious to machine learning experts, but certainly not to regular car users.

3 Designing an Explainable Driving Model

In the previous section, we saw that it is possible to explain the behavior of a machine learning model locally or globally, using post-hoc tools that make little to no assumption about the model. Interestingly, these tools operate on models whose design may have completely ignored the requirement of explainability. A good example of such models is PilotNet (Bojarski et al., 2016, 2020), which consists in a convolutional neural network operating over a raw video stream and producing the vehicle controls at every time step. Understanding the behavior of this system is only possible through external tools, such as the ones presented in Sect. 2, but cannot be done directly by observing the model itself.

On the other hand, a recent trend propose to integrate explainability constraints directly *during* the design of deep architectures such that they exhibit increased interpretability levels (Zhang et al., 2018b; Chen et al., 2019; Lee et al., 2018). Drawing inspiration from modular systems, recent architectures place a particular emphasis on conveying understandable information about their inner workings, in addition to their performance imperatives. As was advocated in (Xu et al., 2020), the modularity of pipelined architectures allows for forensic analysis, by studying the quantities that are transferred between modules (E.G., semantic and depth maps, forecasts of surrounding agent's future trajectories, etc.). Moreover, finding the right balance between modular and end-to-end systems can encourage the use of simulation, for example by training separately perception and driving modules (Müller et al., 2018).

These modularity-inspired models exhibit some forms of interpretability, which can be enforced at three different levels in the design of the driving system. We first review *input-level explanations* (Sect. 3.1), which aim at communicating which perceptual information is used by the model. We then study *intermediate-level explanations* (Sect. 3.2) which force the network to produce supplementary information as it drives. Selected references from this section are reported in Table 3.

3.1 Input

Input-level explanations aim at enlightening the user on which perceptual information is used by the model to take its decisions. We identified two families of approaches that ease interpretation at the input level: attention-based models (Sect. 3.1.1) and models that use semantic inputs (Sect. 3.1.2).

Table 3 Selected references to design an explainable driving model

Approach	Explanation type	Section	Selected references
Input interpretability	Attention maps	3.1.1	Visual attention (Kim and Canny, 2017) Object centric (Wang et al., 2019) Attentional Bottleneck (Kim and Bansal, 2020) DESIRE (Lee et al., 2017)
	Semantic inputs	3.1.2	ChauffeurNet (Bansal et al., 2019) MTP (Djuric et al., 2020; Cui et al., 2019)
Intermediate representations	Auxiliary branch	3.2.1	Affordances/action primitives (Mehta et al., 2018) Detection/forecast of vehicles (Zeng et al., 2019) Multiple auxiliary losses (Bansal et al., 2019) Auto-regressive likelihood map (Srikanth et al., 2019; Bansal et al., 2019)
	Output	3.2.1	Segmentation of future track in Bird'S-Eye-View (Caltagirone et al., 2017) Cost-volume (Zeng et al., 2019) Sequences of points (Lee et al., 2017) Classes (Phan-Minh et al., 2020)
	NLP	3.2.2	Natural language (Kim et al., 2018; Mori et al., 2019)

3.1.1 Attention-Based Models

X-Ai Background. Attention mechanisms, initially designed for NLP application (Bahdanau et al., 2015), learn a function that scores different regions of the input depending on whether or not they should be considered in the decision process. This scoring is often performed based on some contextual information that helps the model decide which part of the input is relevant to the task at hand. Xu et al. (2015) are the first to use an attention mechanism for a computer vision problem, namely, image captioning. In this work, the attention mechanism uses the internal state of the language decoder to condition the visual masking. The network knows which words have already been decoded, and seeks for the next relevant information inside of the image. Many of such attention models were developed for other applications since then, for example in Visual Question Answering (VQA) (Xu and Saenko, 2016; Lu et al., 2016; Yang et al., 2016). These systems, designed to answer questions about images, use a representation of the question as a context to the visual attention module. Intuitively, the question tells the VQA model where to look to answer the question correctly. Not only do attention mechanisms boost the performance of machine learning models, but also they provide insights into the inner workings of the system. Indeed, by visualizing the attention weight associated with each input region, it is possible to know which part of the image was deemed relevant to make the decision.

Applications For Driving Models. Attention-based models recently stimulated interest in the self-driving community, as they supposedly give a hint about the internal reasoning of the neural network. In (Kim and Canny, 2017), an attention mechanism is used to weight each region of an image, using information about previous frames as a context. A different version of attention mechanisms is used in (Mori et al., 2019), where the model outputs a steering angle and a throttle command prediction for each region of the image. These local predictions are used as attention maps for visualization and are combined through a linear combination with learned parameters to provide the final decision.

Visual attention can also be used to select objects defined by bounding boxes, as in (Wang et al., 2019). In this work, a pre-trained object detector provides regions of interest (RoIs), which are weighted using the global visual context, and aggregated to decide which action to take; their approach is validated on both simulated GTAV (Krähenbühl, 2018) and real-world BDDV (Xu et al., 2017) datasets. Cultrera et al. (2020) also use attention on RoIs in a slightly different setup with the CARLA simulator (Dosovitskiy et al., 2017), as they directly predict a steering angle instead of a high-level action. Recently, Kim and Bansal (2020) extended the ChauffeurNet (Bansal et al., 2019) architecture by building a visual attention module that operates on a bird-eye view semantic scene representation. Interestingly, as shown in Fig. 7, combining visual attention with information bottleneck results in sparser saliency maps, making them more interpretable.

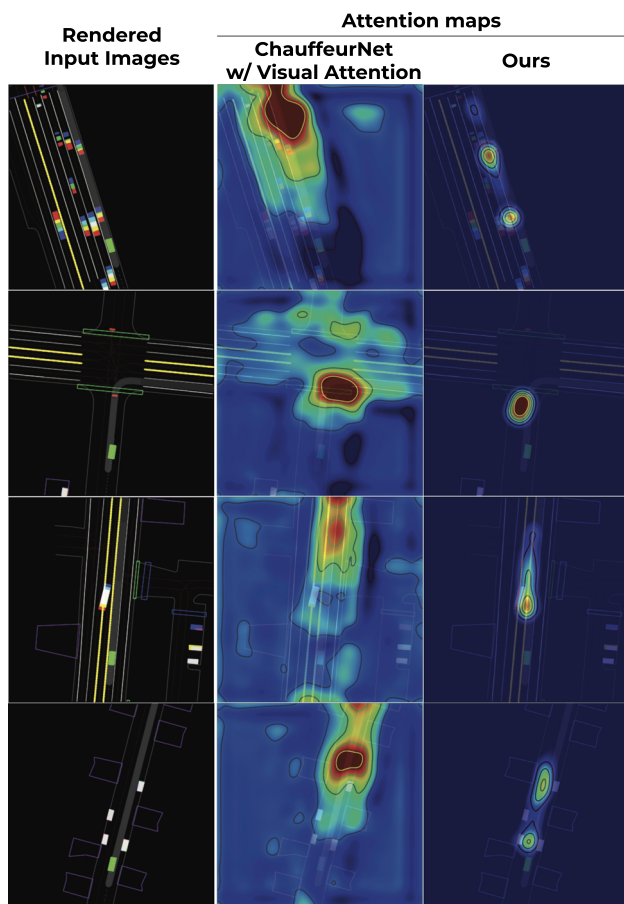


Fig. 7 Comparison of attention maps with classical visual attention (middle column) and maps produced by the attentional bottleneck method (Kim and Bansal, 2020) (right column). Attentional bottleneck provides sparser saliency maps with tighter modes that focus on objects of interest such as surrounding cars. Credits to (Kim and Bansal, 2020)

Challenges. While these attention mechanisms are often thought to make neural networks more transparent, the recent work of Jain and Wallace (2019) mitigates this assumption. Indeed, they show, in the context of natural language, that learned attention weights poorly correlate with multiple measures of feature importance. Moreover, they show that randomly permuting the attention weights usually does not change the outcome of the model. They even show that it is possible to find adversarial attention weights that keep the same prediction while weighting the input words very differently. Even though some works attempt to tackle these issues by learning to align attention weights with gradient-based explanations (Patro et al., 2020), all these findings cast some doubts on the faithfulness of explanations based on attention maps. These limitations restrict the use of attention maps for engineers and legal institutions, who have high requirements for fidelity (see Sect. 1.1.3). However, similarly to post-hoc visual saliency methods presented in Sect. 2.1.1, interpretable-by-design attention maps provide acceptable explanations to increase the trust of regular car users, as they

provide an instantaneous glimpse into the inner workings of the model.

3.1.2 Semantic Inputs

Some traditional machine learning models such as linear and logistic regressions, decision trees, or generalized additive models are considered interpretable by practitioners (Molnar, 2019). However, as was remarked by Alvarez-Melis and Jaakkola (2018), these models tend to consider each input dimension as the fundamental unit on which explanations are built. Consequently, the input space must have a semantic nature such that explanations become interpretable. Intuitively, each input dimension should *mean something* independently of other dimensions. In general machine learning, this condition is often met, for example with categorical and tabular data. However, in computer vision, when, dealing with images, videos, and 3D point clouds, the input space has not an interpretable structure. Overall, in self-driving systems, the lack of semantic nature of inputs impacts the interpretability of machine learning systems.

This observation has motivated researchers to design, build, and use more interpretable input spaces, for example by enforcing more structure or by imposing dimensions to have an underlying high-level meaning. The promise of a more interpretable input space towards increased explainability is diverse. First, the visualization of the network's attention or *post-hoc* saliency heat maps in a semantic input space is more interpretable as it does not apply to individual pixels but rather to higher-level object representations. In practice, visualizing input information in a semantic space can improve the trust of regular car users in the driving model. Second, counterfactual analysis, useful to regulators and engineers (cf. Sect. 1.1.3), is simplified as the input can be manipulated more easily without the risk of generating meaningless imperceptible perturbations, akin to adversarial attacks.

3.1.3 Using Semantic Inputs

Apart from camera inputs processed with deep CNNs in (Bojarski et al., 2016; Codevilla et al., 2018), different approaches have been developed to use semantic inputs in a self-driving model, depending on the types of signals at hand.

3D point clouds, provided by LiDAR sensors, can be processed to form a top-view representation of the car surroundings. For instance, Caltagirone et al. (2017) propose to flatten the scene along the vertical dimension to form a top-down map, where each pixel in the Bird's-Eye-View corresponds to a 10 cm × 10 cm square of the environment. While this input representation provides information about the presence or absence of an obstacle at a certain location,

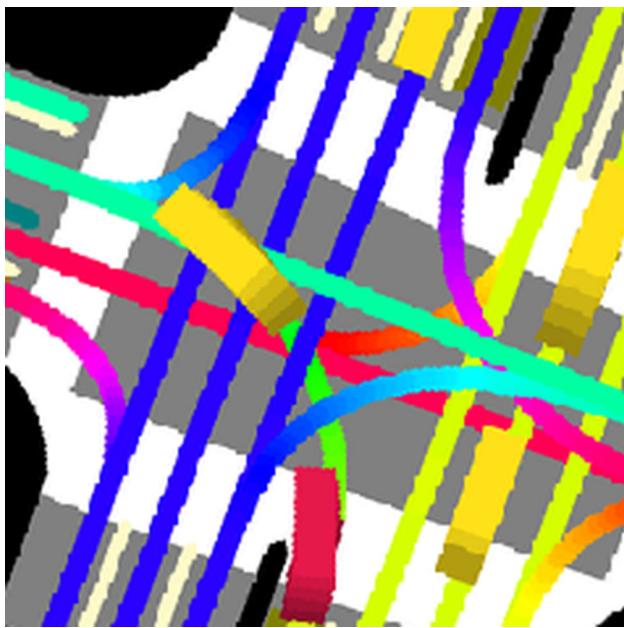


Fig. 8 RGB Bird'S-Eye-View. Semantic information is rasterized within an RGB image that constitutes the input of a convolutional driving backbone. For example, surrounding cars are represented as yellow rectangles and their previous positions are also depicted with a shade. The lanes are encoded with colors corresponding to their orientation. Credits to (Djuric et al., 2020)

it crucially lacks semantics as it ignores the nature of the obstacles (sidewalks, cars, pedestrians, etc.). This lack of high-level scene information is attenuated in DESIRE (Lee et al., 2017), where the output of an image semantic segmentation model is projected to obtain labels in the top-down view generated from the LiDAR point cloud. In DESIRE, static scene components are projected within the top-down view image (E.G., road, sidewalk, vegetation), and moving agents are represented along with their tracked present and past positions.

The ChauffeurNet model (Bansal et al., 2019) relies on a similar top-down scene representation, however instead of originating from a LiDAR point cloud, the Bird'S-Eye-View is obtained from city map data (such as speed limits, lane positions, and crosswalks), traffic light state recognition and detection of surrounding cars. These diverse inputs of the network are gathered into a stack of several images, where each channel corresponds to a rendering of a specific semantic attribute. This contrasts with more recent approaches that aggregate all information into a single RGB top-view image, where different semantic components correspond to different color channels (Djuric et al., 2020; Cui et al., 2019). While the information is still semantic, having a 3-channel RGB image allows leveraging the power of pre-trained convolutional networks. An example RGB semantic image is shown in Fig. 8.

3.1.4 Towards More Control on the Input Space

Having a manipulable input space where we can play on semantic dimensions (E.G., controlling objects' attributes, changing the weather, removing a specific car) is a very desirable feature for increased explainability of self-driving models. Importantly, having such a feature would nicely synergies with many of the post-hoc explainability methods presented in Sect. 2. For example, to produce counterfactual examples, without falling into adversarial meaningless perturbations, it is desirable to have an input space on which we can apply semantic modifications at a pixel-level. As other examples, local approximation methods such as LIME (Ribeiro et al., 2016) would highly benefit from having a controllable input space as a way to ease the sampling of locally similar scenes.

Manipulating inputs can be done at different semantic levels. First, at a global level, changes can include the scene lighting (night/day) and the weather (sun/rain/fog/snow) of the driving scene (Tian et al., 2018), and more generally any change that separately treats style and texture from content and semantics (Geng et al., 2020); such global changes can be done with video translation models (Tulyakov et al., 2018; Bansal et al., 2018; Chen et al., 2020b). At a more local level, possible modifications include adding or removing objects (Li et al., 2020c; Chang et al., 2019; Yang et al., 2020), or changing attributes of some objects (Lample et al., 2017). Recent video inpainting works (Gao et al., 2020) can be used to remove objects from videos. Finally, at an intermediate level, we can think of other semantic changes to be applied to images, such as controlling the proportion of classes in an image (Zhao et al., 2020; Moing et al., 2021). Manipulations could be done by playing on attributes (Lample et al., 2017), by inserting virtual objects in real scenes (Alhaija et al., 2018; Tan et al., 2021), by augmenting existing images with dynamic objects extracted from other scenes (Chen et al., 2021), or by the use of textual inputs with GANs (Li et al., 2020a,b).

We note that having a semantically controllable input space can have lots of implications for areas connected with interpretability. For example, to validate models, and towards having a framework to certify models, we can have a fine-grain stratified evaluation of self-driving models. This can also be used to automatically find failures and corner cases by easing the task of exploring the input space with manipulable inputs (Tian et al., 2018). Finally, to aim for more robust models, we can even use these augmented input spaces to train more robust models, as a way of data augmentation with synthetically generated data (Bowles et al., 2018; Bailo et al., 2019).

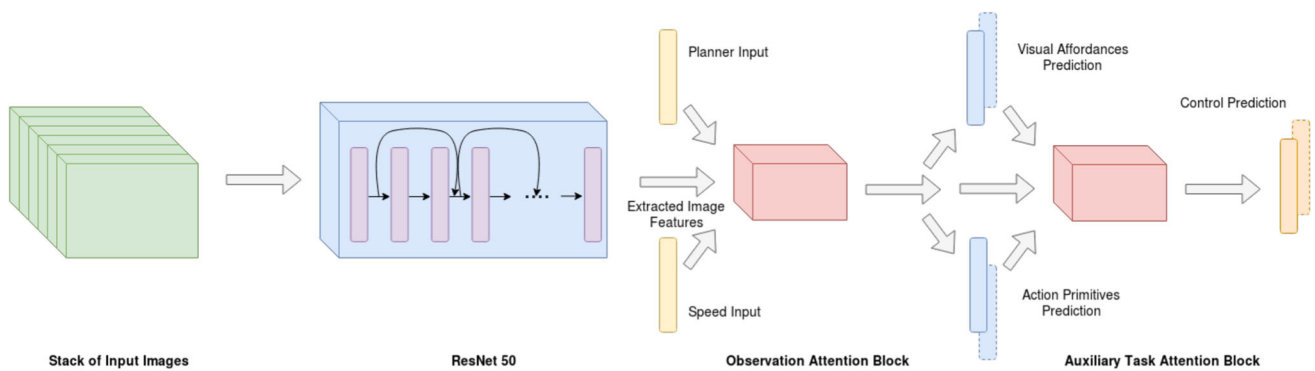


Fig. 9 Predicting intermediate representations. The end-to-end driving model operates over stacked images as well as inputs from the planner to produce control commands. Auxiliary objectives are employed for the

joint prediction of visual affordances (abstract description of the visual scene) and action primitives (abstract description of possible high-level actions required for driving). Credits to (Mehta et al., 2018)

3.2 Supervising Intermediate Representations

A neural network makes its decisions by automatically constructing intermediate representations of the data. These intermediate representations can be constrained to encode for extra driving information such as underlying high-level objectives of the driving system (Sect. 3.2.1), to mimic explanations that humans would provide in similar situations (Sect. 3.2.2). Without explicit constraint, these representations can still be visualized or disentangled for more interpretability (Sect. 3.2.3).

3.2.1 Providing Auxiliary Driving Information

The task of autonomous driving consists in continuously producing the suitable vehicle commands, I.E., steering angle, brake, and throttle controls. A very appealing solution is to train a neural network to directly predict these values (Pomerleau, 1988; Bojarski et al., 2016; Codevilla et al., 2018). However, having a system that directly predicts these command values may not be satisfactory in terms of interpretability, as it may fail to communicate the precise understanding of the surroundings from the perception models, as well as local objectives that the vehicle is attempting to attain.

One way of creating interpretable driving models is to enforce that some information, different than the one directly needed for driving, is present in these features. Doing so, the prediction of a driving decision can be accompanied by an auxiliary output that provides a human-understandable view of the information contained in the intermediate features. Moreover, as was stated in (Zhou et al., 2019), sensorimotor agents benefit, in terms of accuracy, from predicting explicit intermediate scene representations in parallel to their main task. In (Mehta et al., 2018), as illustrated in Fig. 9, a neural network learns to predict control outputs from input images. Its training is helped with auxiliary tasks that aim at recognizing

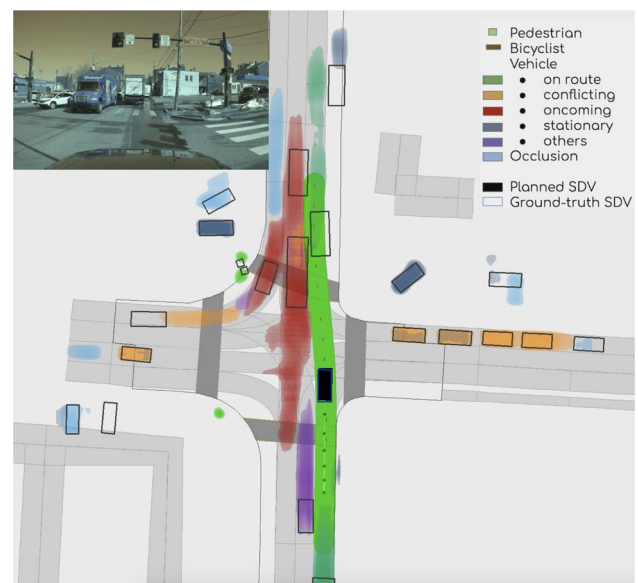


Fig. 10 Semantic occupancy maps. This semantic map is predicted with an auxiliary objective while training the end-to-end motion planner. Colors represent the probability of the future occupancy of various semantic classes (pedestrian, bicycles, vehicles, etc.) Credits to (Sadat et al., 2020)

ing high-level action primitives (E.G., “stop”, “slow down”, “turn left”, *etc.*) and visual affordances in the CARLA simulator (Dosovitskiy et al., 2017).

In (Zeng et al., 2019), a neural network predicts the future trajectory of the ego-vehicle using a top-view Lidar point-cloud. In parallel to this main objective, they learn to produce an interpretable intermediate representation composed of 3D detections and future trajectory predictions. Multi-task in self-driving has been explored deeply in (Bansal et al., 2019), where the authors design a system with ten losses that, besides learning to drive, also forces internal representations to contain information about on-road/off-road zones and future positions of other objects. In (Sadat et al., 2020), an end-to-end motion planning model produces interpretable

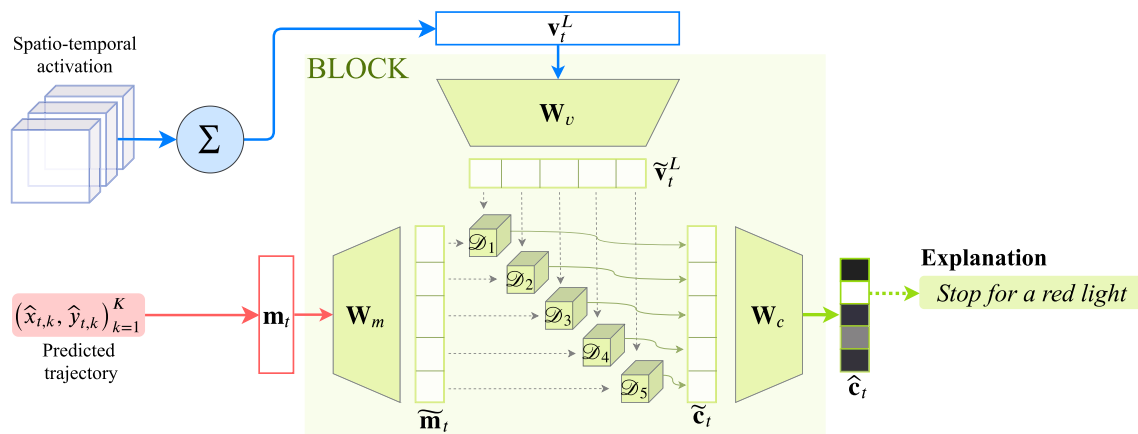


Fig. 11 Multi-level fusion for explanation prediction. The explanation (in green) for a driving decision is expressed as a fusion between the predicted trajectory, I.E., the driving output (in red) and perceptual fea-

tures, I.E., mid-level features from the backbone (in blue) that bring complementary information. Credits to (Ben-Younes et al., 2020)

intermediate representations in the form of probabilistic semantic occupancy maps over space and time. These spatio-temporal maps express the future locations of objects of different classes, as illustrated in Fig. 10, in a probabilistic and instance-free fashion. Their goal is to capture whether and when a discretized spatial region is occupied by an agent at a certain time step. These maps have the potential to remove the need for both detection and tracking while offering similar levels of interpretability.

Most of recent models do not directly output driving commands, but rather some form of intermediate goals that the model should reach. Those are then passed to a controller that finds the suitable steering, brake and acceleration commands to reach the required position. The intermediate goals can be visualized in the same coordinate system as the input representation, which helps the human user interpret the prediction with respect to scene elements (road structure, surrounding agents, etc.).

Output representations of neural trajectory prediction systems can be split into two categories: analytical representations and spatial grid representations.

Systems that output an analytical representation of the future trajectory provide one or more predictions in the form of points or curves in the 2D space. This output structure is commonly used in motion forecasting models (Lee et al., 2017; Cui et al., 2019; Phan-Minh et al., 2020; Salzmann et al., 2020; Ben-Younes et al., 2022). For instance, Lee et al. (2017) propose to predict multiple future trajectories for each agent of the scene. More specifically, recurrent models are trained to sample trajectories as sequences of 2D points in a bird's-eye-view basis, rank them, and refine them according to perceptual features. In MTP (Cui et al., 2019), each trajectory consists of a set of 2D points and a confidence score. CoverNet (Phan-Minh et al., 2020) poses the trajectory pre-

diction problem as a classification one, where each possible class is a predefined trajectory profile.

In the second family of trajectory prediction systems, the network scores regions of the spatial grid according to their likelihood of hosting the car in the future. In (Caltagirone et al., 2017), the network is trained to predict the track of the future positions of the vehicle, in a semantic segmentation fashion. Differently, ChauffeurNet (Bansal et al., 2019) predicts the next vehicle position as a probability distribution over the spatial coordinates. The Neural Motion Planner (Zeng et al., 2019) contains a neural network that outputs a cost volume, which is a spatio-temporal quantity indicating the cost for the vehicle to reach a certain position at a certain moment. Trajectories are sampled from a set of dynamically possible paths (straight lines, circles, and clothoids) and scored according to the cost volume. Interestingly, the cost volume can be visualized, and thus provides a human-understandable view of what the system considers feasible.

3.2.2 Using Human Explanations

Instead of supervising intermediate representations with scene information, other approaches propose to directly use explanation annotations as an auxiliary branch. The driving model is trained to simultaneously decide *and* explain its behavior. In the work of Xu et al. (2020), the BDD-OIA dataset was introduced, where clips are manually annotated with *authorized* actions and their associated explanation. Action and explanation predictions are expressed as multi-label classification problems, which means that multiple actions and explanations are possible for a single example. While this system is not properly a driving model (no control or trajectory prediction here, but only high-level classes such as “stop”, “move forward” or “turn left”), Xu et al. (2020)

were able to increase the performance of action decision making by learning to predict explanations as well.

Very recently, Ben-Younes et al. (2020) propose to explain the behavior of a driving system by fusing high-level decisions with mid-level perceptual features. The fusion, depicted in Fig. 11, is performed using BLOCK (Ben-Younes et al., 2019), a tensor-based fusion technique designed to model rich interactions between heterogeneous features. Their model is trained on the HDD dataset (Ramanishka et al., 2018), where 104 hours of human driving are annotated with a focus on driver behavior. In this dataset, video segments are manually labeled with classes that describe the goal of the driver (E.G., “turn left”, “turn right”, etc.) as well as an explanation for its stops and deviations (E.G., “stop for a red light”, “deviate for a parked car”, etc). The architecture of Ben-Younes et al. (2020) is initially developed to provide explanations in a classification setup, and they show an extension of it to generate natural language sentences (see Sect. 4.1).

3.2.3 Visualizing and Disentangling Representations

Visualizing the predictions of an auxiliary head is an interesting way to give the human user an idea of what information is contained in the intermediate representation. Indeed, observing that internal representations of the driving network can be used to recognize drivable areas, estimate pedestrian attributes (Mordan et al., 2020), detect other vehicles, and predict their future positions strengthens the trust one can give to a model. Yet, it is important to keep in mind that information contained in the representation is not necessarily used by the driving network to make its decision. More specifically, the fact that we can infer future positions of other vehicles from the intermediate representation does not mean that these forecasts were actually used to make the driving decision. Overall, one should be cautious about such auxiliary predictions to interpret the behavior of the driving model, as the causal link between these auxiliary predictions and the driving output is not enforced.

Interestingly, models have been developed to learn and discover disentangled latent variables without using auxiliary supervision. These unsupervised methods aim at capturing underlying salient data factors, such that each individual variable represents a single interpretable attribute (Bengio et al., 2013; Chen et al., 2016). While the application of these approaches remains scarce in autonomous driving (Morton and Kochenderfer, 2017), their popularity is rising in image synthesis as they offer a way to control the generation over human-understandable variation factors (Pu et al., 2016; Karas et al., 2019).

4 Use Case: Natural Language Explanations

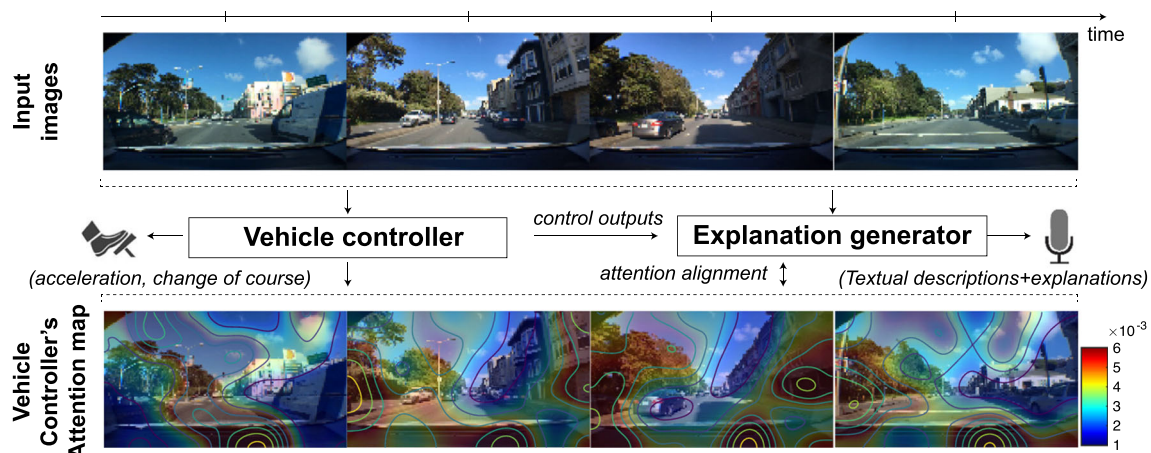
As was stated in Sect. 1.1.3, some of the main requirements of explanations targeted at non-technical human users are conciseness and clarity. To meet these needs, some research efforts have been geared at building models that provide explanations of their behavior in the form of natural language sentences. In Sect. 4.1, we review the methods proposed by the community to generate natural language explanations of machine learning models. The limits of such techniques are discussed in Sect. 4.2.

4.1 Generating Natural Language Explanations

X-Ai Background. The first attempt to explain the predictions of a deep network with natural language was in the context of image classification, where Hendricks et al. (2016) train a neural network to generate sentence explanations from image features and class label. These explanations are forced to be *relevant* to the image, I.E., to mention elements that are present in the image, and also *class-discriminative*, which means they can spot specific visual elements that separate one class from another. This work is further extended in (Hendricks et al., 2018), where a list of candidate explanations is sorted with respect to how noun phrases are visually-grounded.

In the field of natural language processing (NLP), Liu et al. (2019) build an explanation-producing system for long review text classification. In particular, they tackle the problem of independence between the prediction and its explanation and try to strengthen the connection between both. To do so, they pre-train a classifier that takes as input an explanation and predicts the class of the associated text input, and they use this classifier to measure and optimize the difference between true and generated explanations. Moreover, Camburu et al. (2018) propose to learn from human-provided explanations at train time for a natural language inference task. Similarly, Rajani et al. (2019) gather a dataset of human natural language explanations for a common-sense inference task and learn a model that jointly classifies the correct answer and generates the correct explanation.

In the field of vision-and-language applications, Park et al. (2018) build ACT-X and VQA-X, two datasets of multimodal explanations for the task of action recognition and visual question answering. More specifically, VQA-X (resp. ACT-X) contains textual explanations that justify the answer (resp. the action), as well as an image segmentation mask that shows areas that are relevant to answer the question (resp. recognize the action). Both textual and visual explanations are manually annotated. Related to this work, Zellers et al. (2019) design a visual commonsense reasoning task where a question is asked about an image, and the answer is a sentence to choose among a set of candidates. Each example



Example of textual descriptions + explanations:

Ours: “The car is driving forward + because there are no other cars in its lane”

Human annotator: “The car heads down the street + because the street is clear.”

Fig. 12 Generating natural language explanations. The end-to-end driving model (‘vehicle controller’) predicts the control commands (acceleration and change of direction). The predicted control outputs are provided to the ‘Explanation generator’ that also sees the visual

scene. This module is trained to provide natural language justifications describing the scene and explaining the driving decision taken by the driving model. The explanation generator is forced to align its attention on the one learnt by the vehicle controller. Credits to (Kim et al., 2018)

is also associated with another set of sentences containing candidate justifications of the answer and describing the reasoning behind a decision.


Applications For Driving Models. In the context of self-driving, Kim et al. (2018) learn to produce textual explanations justifying decisions from a self-driving system. Based on the video material of BDDV (Xu et al., 2017), the authors built the BDD-X dataset where dash-cam video clips are annotated with a sentence that describes the driving decision (E.G., “the car is deviating from its main track”), and another one that explains why this is happening (E.G., “because the yellow bus has stopped”). An end-to-end driving system equipped with visual attention is first trained on this dataset to predict the vehicle controls for each frame, and, in a second phase, an attention-based video-to-text captioning model is trained to generate natural language explanations justifying the system’s decisions. The attention of the captioning explanation module is constrained to align with the attention of the self-driving system. We show an overview of their system in Fig. 12. Notably, this model is akin to a post-hoc explanation system as the explanation-producing network is trained after the driving model.

The BDD-X dataset is also used by Ben-Younes et al. (2020) as they adapt their explanation classification method to the setup of natural language generation. Interestingly, they study the impact of the temperature parameter in the decoding softmax, classically used to control the diversity of generated sentences, on the variability of sampled explanations for the same situation. In particular, they show that for reasonably low values of the temperature, the model justifies

a driving situation with semantically consistent sentences. These explanations differ from each other only *syntactically* and with respect to their *completeness* (some explanations are more exhaustive and precise than others), but not *semantically*. Looking at the example shown in Table 4, we see that all the explanations are correct as they correspond to the depicted scene, but the level of detail they convey may be different.

Interestingly, Ben-Younes et al. (2020) draw a parallel between VQA (Antol et al., 2015; Agrawal et al., 2017; Malinowski et al., 2017) and the task of explaining decisions of a self-driving system with natural language: similarly to the way the question is combined with visual features in VQA, in their work, decisions of the self-driving system are combined with perceptual features encoding the scene. For the VQA task, the result is the answer to the question and, in the case of the driving explanations, the result is the justification why the self-driving model produced its decision. More generally, we believe that recent VQA literature can inspire more explainable driving works. In particular, there is a strong trend to make VQA models more interpretable (Li et al., 2018a; Riquelme et al., 2020; Alipour et al., 2020), to unveil learned biases (Agrawal et al., 2018; Ramakrishnan et al., 2018; Cadène et al., 2019b), and to foster reasoning mechanisms (Johnson et al., 2017; Hu et al., 2017; Cadène et al., 2019a). Lastly, towards the long-term goal of having human-machine dialogs and more interactive explanations, the VQA literature can also be a source of inspiration (Alipour et al., 2020).

Table 4 Various samples of generated explanations

	
Extracted frame	
GT	Because traffic is moving now
T = 0	Because the light is green and traffic is moving
T = 0.3	As the light turns green and traffic is moving
T = 0.3	Because the light is green and traffic is moving
T = 0.3	Because traffic is moving forward
T = 0.3	Because the light turns green
T = 0.3	Because the light turned green and traffic is moving

GT stands for the the ground-truth (human gold label). Other lines are justifications generated by the model BEEF, obtained with with various decoding temperature T: T = 0 corresponds to the greedy decoding and the lines with T = 0.3 correspond to random decoding with a temperature of 0.3. A higher temperature implies an increased diversity in the generated justifications, both syntactically and in the completeness level, with some sentences mentioning the light and/or the moving traffic. However, pushing this parameter too high may also lead to an undesired semantic drift. Credits to (Ben-Younes et al., 2020)

We remark that driving datasets that are designed for explainability purposes have poor quality on the automated driving side. For instance, they include only one camera, the sensor calibration is often missing, etc. We argue that better explainability datasets should be proposed, by building on high-quality driving datasets, such as nuScenes (Caesar et al., 2020). Regarding the lack of high-quality driving datasets containing explanations, another research direction lies in transfer learning for explanation: the idea would be to separately learn to drive on big driving datasets and to explain on more limited explanation datasets. The transfer between the two domains would be done by fine-tuning, by using multi-task objectives, or by leveraging recent transfer learning works.

4.2 Limits of Mimicking Natural Language Explanations

Using annotations of explanations to supervise the training of a neural network seems natural and effective. Yet, this practice has some strong assumptions and the generated explanations may be limited in their faithfulness. From a data point-of-view, as was noted in (Kim et al., 2018), acquiring the annotations for explanations can be quite difficult: ground-truth explanations are often post-hoc rationales generated by an external observer of the scene and not by the person who took the action. Beyond this, explanation annotations correspond to the reasons why a *person* made an action. Using these annotations to explain the behavior

of a *machine learning model* is an extrapolation that should be made carefully. Indeed, applying some type of behavior cloning method on explanations assumes that the reasons behind the model decision must be the same as the one of the human performing the action. This assumption prevents the model to discover new cues on which it can ground its decision. For example, in medical diagnosis, it has been found that machine learning models can discover new visual features and biomarkers, which are linked to the diagnosis through a causal link unknown to medical experts (Makino et al., 2020). In the context of driving, however, it seems satisfactory to make models rely on the same cues human drivers would use.

Beyond the aforementioned problems, evaluating natural language explanations constitutes a challenge per se. Most approaches (Kim et al., 2018; Hendricks et al., 2016; Camburu et al., 2018; Rajani et al., 2019) evaluate generated natural language explanations based on human ratings or by comparing them to ground-truth explanation of humans (using automated metrics like BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), or CIDEr (Vedantam et al., 2015) scores). As argued by Hase et al. (2020); Gilpin et al. (2018), the evaluation of natural language explanations is delicate and automated metric and human evaluations are not satisfying as they cannot guarantee that the explanation is faithful to the model's decision-making process. These metrics rather evaluate the plausibility of the explanation regarding human evaluations (Jacovi and Goldberg, (2020a)).

Overall, this evaluation protocol encourages explanations that match human expectation and it is prone to produce *persuasive explanations* (Herman, 2017; Gilpin et al., 2018), I.E., explanations that satisfy the human users regardless of their faithfulness to the model processing. Similarly to what is observed in (Adebayo et al., 2018) with saliency maps, the human observer is at risk of confirmation bias when looking at outputs of natural language explainers. Potential solutions to tackle the problem of persuasive explanations can be inspired by recent works in NLP. Indeed, in this field, several works have recently advocated for evaluating the *faithfulness* of explanations rather than their *plausibility* (Jacovi and Goldberg, (2020b); Ding and Koehn, 2021). For example, Hase et al. (2020) propose the leakage-adjusted simulatability (LAS) metric, which is based on the idea that the explanation should be helpful to predict the model's output without leaking direct information about the output.

5 Conclusion

In this survey, we presented the challenges of explainability raised by the development of modern, deep-learning-based self-driving models. In particular, we argued that the need for explainability is multi-factorial, and it depends on the person needing explanations, on the person's expertise level, as well as on the available time to analyze the explanation. We gave a quick overview of recent approaches to build and train modern self-driving systems and we specifically detailed why these systems are not explainable *per se*. First, many shortcomings come from our restricted knowledge on deep learning generalization, and the black-box nature of learned models. Those aspects do not spare self-driving models. Moreover, as being very heterogeneous systems that must simultaneously perform tasks of very different natures, the willingness to disentangle implicit sub-tasks appears natural.

As an answer to such problems, many explanation methods have been proposed, and we organized them into two categories. First, *post-hoc* methods which apply on a trained driving model to locally or globally explain and interpret its behavior. These methods have the advantage of not compromising driving performances since the explanation models are applied afterward; moreover, these methods are usually architecture-agnostic to some extent, in the sense that they can transfer from a network to another one. However, even if these techniques are able to exhibit spurious correlations learned by the driving model, they are not meant to have an impact on the model itself. On the other hand, directly *designing* interpretable self-driving models can provide better control on the quality of explanations at the expense of a potential risk to degrade driving performances. Explainability is contained in the neural network architecture itself and is generally not transferable to other architectures

Evaluating explanations is a huge challenge. Automated evaluation methods usually depend on the explainability scheme and the type of data, and there is a lack of unified evaluation methods. Concerning crowd-sourced evaluations, for example to evaluate natural language explanations with a human rating, is not satisfying as it does not scale well for future methods and as it can lead to persuasive explanations, especially if the main objective is to increase users' trust. In particular, this is a serious pitfall for approaches that learn to mimic human explanations (E.G., imitation learning for explanations) such as models in (Kim et al., 2018; Hendricks et al., 2016; Park et al., 2018), but also for post-hoc saliency methods (Adebayo et al., 2018). A solution to this issue could be to measure and quantify the *uncertainty of explanations*, I.E., answering the question "*how much can we trust explanations?*". Related to this topic is the recent work of Corbière et al. (2020), which learns the confidence of predictions made by a neural network with an auxiliary model called ConfidNet, or the work of Bykov et al. (2020) which applies explanation methods to Bayesian neural networks instead of classical deep networks, thus providing built-in modeling of uncertainties for explanations. Besides, evaluating *complete* explanations of a driving decision is especially challenging. Indeed, more complete explanations may involve *compositional* explanations, I.E., that combine several atomic logical facts. Such explanations are particularly hard to evaluate, as argued by Jansen et al. (2021), because they cannot be satisfyingly evaluated against gold-labels as the number of combinations explodes with the number of facts. Overall, finding ways to evaluate explanations with respect to key concepts such as human-interpretability, completeness level, or faithfulness to the model's processing is essential to design better explanation methods in the future.

Writing up this survey, we observe that many X-AI approaches have not been used—or in a very limited way—to make neural driving models more interpretable. This is the case for example for local approximation methods, for counterfactual interventions, or model translation methods. Throughout the survey, we hypothesized the underlying reasons that make it difficult to apply off-the-shelf X-AI methods for the autonomous driving literature. One of the main hurdles lies in the type of input space at hand, its very high dimensionality, and the rich semantics contained in a visual modality (video, 3D point clouds). Indeed, many X-AI methods have been developed assuming either the interpretability of each of the input dimensions or a limited number of input dimensions. Because of the type of the input space for self-driving models, many X-AI methods do not trivially transpose to make self-driving models more interpretable. For example, one will obtain meaningless adversarial perturbations if naively generating counterfactual explanations on driving videos and we thereby observe a huge gap between the profuse literature for generating counterfactual examples

for low-dimensional inputs and the scarce literature on counterfactual explanations for high-dimensional data (images and videos). As another example, it seems impractical to design a sampling function in the video space to locally explore around a particular driving video and learn a local approximation of the self-driving model with methods presented in Sect. 2.1.1. We believe that ways to bridge this gap, detailed in Sect. 3.1.2, include making raw input spaces more controllable and manipulable, and designing richer input semantic spaces that have human-interpretable meaning.

Despite their differences, all the methods reviewed in this survey share the objective of exposing the causes behind model decisions. Yet, only very few works directly borrow tools and concepts from the field of causal modeling (Pearl, 2009). Taken apart methods that attempt to formulate counterfactual explanations, applications of causal inference methods to explain self-driving models are rare. As discussed in Sect. 2.1.2, inferring the causal structure in driving data has strong implications in explainability. It is also a very promising way towards more robust neural driving models. As was stated in (de Haan et al., 2019), a driving policy must identify and rely solely on true causes of expert decisions if we want it to be robust to distributional shift between training and deployment situations. Building neural driving models that take the right decisions for the right identified reasons would yield inherently robust, explainable, and faithful systems.

Funding This survey was funded by Valeo and no other funding was received to assist with the preparation of this manuscript.

Declarations

Conflict of interests The authors have no relevant financial or non-financial interests to disclose.

References

- Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*
- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I.J., Hardt, M., & Kim, B. (2018). Sanity checks for saliency maps. In *NeurIPS*
- Agrawal, A., Batra, D., Parikh, D., & Kembhavi, A. (2018). Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*
- Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C.L., Parikh, D., & Batra, D. (2017). VQA: visual question answering - www.visualqa.org. *IJCV*
- Alhajj, H.A., Mustikovela, S.K., Mescheder, L.M., Geiger, A., & Rother, C. (2018). Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *IJCV*
- Alipour, K., Schulze, J.P., Yao, Y., Ziskind, A., & Burachas, G. (2020). A study on multimodal and interactive explanations for visual question answering. In *SafeAI@AAAI*
- Alvarez-Melis, D., & Jaakkola, T.S. (2018). Towards robust interpretability with self-explaining neural networks. In *NeurIPS*
- Anderson, J.M., Nidhi, K., Stanley, K.D., Sorensen, P., Samaras, C., & Oluwatola, O.A. (2014). Autonomous vehicle technology: A guide for policymakers
- Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., & Parikh, D. (2015). VQA: visual question answering. In *ICCV*
- Arras, L., Osman, A., & Samek, W. (2022). Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations. *Inf Fusion*, 81(C), 14–40. <https://doi.org/10.1016/j.inffus.2021.11.008>
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS one*
- Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR*
- Bailo, O., Ham, D., & Shin, Y.M. (2019). Red blood cell image generation for data augmentation using conditional generative adversarial networks. In *CVPR Workshops*
- Banerjee, S., & Lavie, A. (2005). METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization @ACL*
- Bansal, M., Krizhevsky, A., & Ogale, A.S. (2019). Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Robotics: Science and Systems*
- Bansal, A., Ma, S., Ramanan, D., & Sheikh, Y. (2018). Recycle-gan: Unsupervised video retargeting. In *ECCV*
- Bau, D., Zhou, B., Khosla, A., Oliva, A., & Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*
- Beaudouin, V., Bloch, I., Bounie, D., Cléménçon, S., d'Alché-Buc, F., Eagan, J., Maxwell, W., Mozharovskiy, P., & Parekh, J. (2020). Flexible and context-specific AI explainability: A multidisciplinary approach. *CoRR*
- Bengio, Y., Courville, A.C., & Vincent, P. (2013). Representation learning: A review and new perspectives. *TPAMI*
- Ben-Younes, H., Cadene, R., Thome, N., & Cord, M. (2019). Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *AAAI*
- Ben-Younes, H., Éloi, Z., Zablocki, P., Pérez, P., Cord, M. (2020). Driving behavior explanation with multi-level fusion. *Machine Learning for Autonomous Driving Workshop ML4AD@NeurIPS*
- Ben-Younes, H., Zablocki, É., Chen, M., Pérez, P., & Cord, M. (2022). Raising context awareness in motion forecasting. *CVPR Workshop on Autonomous Driving (WAD)*
- Besserve, M., Mehrjou, A., Sun, R., & Schölkopf, B. (2020). Counterfactuals uncover the modular structure of deep generative models. In *ICLR*
- Bojarski, M., Chen, C., Daw, J., Degirmenci, A., Deri, J., Firner, B., Flepp, B., Gogri, S., Hong, J., Jackel, L.D., Jia, Z., Lee, B.J., Liu, B., Liu, F., Muller, U., Payne, S., Prasad, N.K.N., Provodin, A., Roach, J., Rvachov, T., Tadimet, N., van Engelen, J., Wen, H., Yang, E., & Yang, Z. (2020). The NVIDIA pilotnet experiments. *CoRR*
- Bojarski, M., Choromanska, A., Choromanski, K., Firner, B., Ackel, L.J., Muller, U., Yeres, P., & Zieba, K. (2018). Visualbackprop: Efficient visualization of cnns for autonomous driving. In *ICRA*
- Bojarski, M., Testa, D.D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). End to end learning for self-driving cars. *CoRR*
- Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L.D., & Muller, U. (2017). Explaining how a deep neural network trained with end-to-end learning steers a car. *CoRR*
- Borg, M., Englund, C., Wnuk, K., Durann, B., Lewandowski, C., Gao, S., Tan, Y., Kaijser, H., Lönn, H., & Törnqvist, J. (2019). Safely entering the deep: A review of verification and validation for

- machine learning and a challenge elicitation in the automotive industry. *Journal of Automotive Software Engineering*
- Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R. N., Hammers, A., Dickie, D. A., del C Valdés Hernández, M., Wardlaw, J. M., & Rueckert, D. (2018). GAN augmentation: Augmenting training data using generative adversarial networks. *CoRR*
- Brown, K., Driggs-Campbell, K., & Kochenderfer, M. J. (2020). A taxonomy and review of algorithms for modeling and predicting human driver behavior. arXiv preprint [arXiv:2006.08832](https://arxiv.org/abs/2006.08832)
- Bykov, K., Höhne, M. M., Müller, K., Nakajima, S., Kloft, M. (2020). How much can I trust you? - quantifying uncertainties in explaining neural networks. *CoRR*
- Cadène, R., Ben-younes, H., Cord, M., & Thome, N. (2019a). MUREL: multimodal relational reasoning for visual question answering. In *CVPR*
- Cadène, R., Dancette, C., Ben-younes, H., Cord, M., & Parikh, D. (2019b). Rubi: Reducing unimodal biases for visual question answering. In *NeurIPS*
- Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In *CVPR*
- Caltagirone, L., Bellone, M., Svensson, L., & Wahde, M. (2017). Lidar-based driving path generation using fully convolutional neural networks. In *ITSC*
- Camburu, O., Rocktäschel, T., Lukasiewicz, T., & Blunsom, P. (2018). e-snli: Natural language inference with natural language explanations. In *NeurIPS*
- Casas, S., Sadat, A., & Urtasun, R. (2021). MP3: A unified model to map, perceive, predict and plan. In *CVPR*
- Castrejón, L., Aytar, Y., Vondrick, C., Pirsaviash, H., & Torralba, A. (2016). Learning aligned cross-modal representations from weakly aligned data. In *CVPR*
- Chan, F., Chen, Y., Xiang, Y., & Sun, M. (2016). Anticipating accidents in dashcam videos. In *ACCV*
- Chang, Y., Liu, Z. Y., & Hsu, W. H. (2019). Vornet: Spatio-temporally consistent video inpainting for object removal. In *CVPR Workshops*
- Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*
- Chen, J., Li, S. E., & Tomizuka, M. (2020a). Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *CoRR*
- Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., & Su, J. (2019). This looks like that: Deep learning for interpretable image recognition. In *NeurIPS*
- Chen, Y., Rong, F., Duggal, S., Wang, S., Yan, X., Manivasagam, S., Xue, S., Yumer, E., & Urtasun, R. (2021). Geosim: Realistic video simulation via geometry-aware composition for self-driving. In *CVPR*
- Chen, X., Zhang, Y., Wang, Y., Shu, H., Xu, C., & Xu, C. (2020b). Optical flow distillation: Towards efficient and stable video style transfer. In *ECCV*
- Chitta, K., Prakash, A., & Geiger, A. (2021). NEAT: neural attention fields for end-to-end autonomous driving. In *ICCV*
- Choi, J. K., & Ji, Y. G. (2015). Investigating the importance of trust on adopting an autonomous vehicle. *IJHCI*
- Codevilla, F., Miiller, M., López, A., Koltun, V., & Dosovitskiy, A. (2018). End-to-end driving via conditional imitation learning. In *ICRA*
- Codevilla, F., Santana, E., López, A. M., Gaidon, A. (2019). Exploring the limitations of behavior cloning for autonomous driving. In *ICCV*
- Corbière, C., Thome, N., Saporta, A., Vu, T., Cord, M., & Pérez, P. (2020). Confidence estimation via auxiliary models. *PAMI*
- Corso, A., Du, P., Driggs-Campbell, K. R., Kochenderfer, M. J. (2019). Adaptive stress testing with reward augmentation for autonomous vehicle validation. In *ITSC*
- Cui, H., Radosavljevic, V., Chou, F., Lin, T., Nguyen, T., Huang, T., Schneider, J., & Djuric, N. (2019). Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *ICRA*
- Cultrera, L., Seidenari, L., Becattini, F., Pala, P., & Bimbo, A. D. (2020). Explaining autonomous driving by learning end-to-end visual attention. In *CVPR Workshops*
- Das, A. & Rad, P. (2020). Opportunities and challenges in explainable artificial intelligence (XAI): A survey. *CoRR*
- de Haan, P., Jayaraman, D., & Levine, S. (2019). Causal confusion in imitation learning. In *NeurIPS*
- Deng, Y., Zheng, J. X., Zhang, T., Chen, C., Lou, G., & Kim, M. (2020). An analysis of adversarial attacks and defenses on autonomous driving models. In *PerCom*
- Di, X. & Shi, R. (2020). A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to ai-guided driving policy learning. *CoRR*
- Dickmanns, E. D. (2002). The development of machine vision for road vehicles in the last decade. In *IV*
- Ding, S. & Koehn, P. (2021). Evaluating saliency methods for neural language models. In *NAACL*
- Djuric, N., Radosavljevic, V., Cui, H., Nguyen, T., Chou, F., Lin, T., Singh, N., & Schneider, J. (2020). Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In *WACV*
- Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *CoRR*
- Doshi-Velez, F., & Kortz, M. A. (2017). Accountability of ai under the law: The role of explanation. *CoRR*
- Dosovitskiy, A., Ros, G., Codevilla, F., López, A., & Koltun, V. (2017). CARLA: an open urban driving simulator. In *CoRL*
- Du, N., Haspiel, J., Zhang, Q., Tilbury, D., Pradhan, A. K., Yang, X. J., & Robert Jr, L.P. (2019). Look who's talking now: Implications of av's explanations on driver's trust, av preference, anxiety and mental workload. Transportation research part C: emerging technologies
- Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2009). Visualizing higher-layer features of a deep network. *Technical Report*, University of Montreal
- Espíe, E., Guionneau, C., Wymann, B., Dimitrakakis, C., Coulom, R., & Sumner, A. (2005). Torcs, the open racing car simulator
- Fellous, J. M., Sapiro, G., Rossi, A., Mayberg, H. S., & Ferrante, M. (2019). Explainable artificial intelligence for neuroscience: Behavioral neurostimulation. *Frontiers in Neuroscience*
- Fong, R. C., & Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*
- Frosst, N. & Hinton, G. E. (2017). Distilling a neural network into a soft decision tree. In *Workshop on comprehensibility and explanation in AI and ML @AI*IA 2017*
- Gao, C., Saraf, A., Huang, J., & Kopf, J. (2020). Flow-edge guided video completion. In *ECCV*
- Garfinkel, S., Matthews, J., Shapiro, S. S., & Smith, J. M. (2017). Toward algorithmic transparency and accountability. *Communications ACM*
- Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *IJRR*
- Geng, Z., Cao, C., & Tulyakov, S. (2020). Towards photo-realistic facial expression manipulation. *IJCV*
- Ghorbani, A., Abid, A., & Zou, J. Y. (2019). Interpretation of neural networks is fragile. In *AAAI*
- Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *DSSA*

- Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., & Lee, S. (2019). Counterfactual visual explanations. In *ICML*
- Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computer Survey*
- Harradon, M., Druce, J., & Ruttenberg, B.E. (2018). Causal learning and explanation of deep neural networks via autoencoded activations. *CoRR*
- Hase, P., Zhang, S., Xie, H., & Bansal, M. (2020). Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In T. Cohn, Y. He, Y. Liu (Eds.) *EMNLP (Findings)*
- Haspel, J., Du, N., Meyerson, J., Jr L. P. R., Tilbury, D. M., Yang, X. J., & Pradhan, A. K. (2018). Explanations and expectations: Trust building in automated vehicles. In *HRI*
- Hecker, S., Dai, D., Liniger, A., & Gool, L.V. (2020). Learning accurate and human-like driving using semantic maps and attention. *CoRR*
- Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., & Darrell, T. (2016). Generating visual explanations. In *ECCV*
- Hendricks, L. A., Hu, R., Darrell, T., & Akata, Z. (2018). Grounding visual explanations. In *ECCV*
- Herman, B. (2017). The promise and peril of human evaluation for model interpretability. *CoRR*
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*
- Hooker, S., Erhan, D., Kindermans, P., & Kim, B. (2019). A benchmark for interpretability methods in deep neural networks
- Hu, R., Andreas, J., Rohrbach, M., Darrell, T., & Saenko, K. (2017). Learning to reason: End-to-end module networks for visual question answering. In *ICCV*
- Jacob, P., Zablocki, É., Ben-Younes, H., Chen, M., Pérez, P., & Cord, M. (2021). STEEX: steering counterfactual explanations with semantics. *CoRR arXiv: abs/2111.09094*
- Jacovi, A. & Goldberg, Y. (2020a). Aligning faithful interpretations with their social attribution. *TACL*
- Jacovi, A. & Goldberg, Y. (2020b). Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *ACL*
- Jain, S. & Wallace, B. C. (2019). Attention is not explanation. In *NAACL*
- Janai, J., Güney, F., Behl, A., & Geiger, A. (2020a). Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Found Trends Computer Graph Vision*
- Janai, J., Güney, F., Behl, A., & Geiger, A. (2020b). Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*. 12(1), 1–308
- Jansen, P., Smith, K., Moreno, D., & Ortiz, H. (2021). On the challenges of evaluating compositional explanations in multi-hop inference: Relevance, completeness, and expert ratings. *CoRR arXiv:2109.03334*
- Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., & Girshick, R. B. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*
- Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *CVPR*
- Kim, J. & Canny, J. F. (2017). Interpretable learning for self-driving cars by visualizing causal attention. In *ICCV*
- Kim, J., & Bansal, M. (2020). Attentional bottleneck: Towards an interpretable deep driving network. In *CVPR Workshops*
- Kim, H., Lee, K., Hwang, G., & Suh, C. (2019). Crash to not crash: Learn to identify dangerous vehicles using a simulator. In *AAAI*
- Kim, J., Rohrbach, A., Darrell, T., Canny, J. F., & Akata, Z. (2018). Textual explanations for self-driving vehicles. In *ECCV*
- Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S. K., & Pérez, P. (2020). Deep reinforcement learning for autonomous driving: A survey. *CoRR*
- Koo, J., Kwac, J., Ju, W., Steinert, M., Leifer, L., & Nass, C. (2015). Why did my car just do that? explaining semi-autonomous driving actions to improve driver understanding, trust, and performance. *IJDeM*
- Koren, M., Alsaif, S., Lee, R., & Kochenderfer, M. J. (2018). Adaptive stress testing for autonomous vehicles. In *IV*
- Krähenbühl, P. (2018). Free supervision from video games. In *CVPR*
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*
- Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., & Ranzato, M. (2017). Fader networks: Manipulating images by sliding attributes. In *NIPS*
- LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature*
- Lee, J. & Moray, N. (1992). Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*
- Lee, J. D., & Moray, N. (1994). Trust, self-confidence, and operators' adaptation to automation. *International Journal of Human-computer Studies*
- Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*
- Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H. S., & Chandraker, M. (2017). DESIRE: distant future prediction in dynamic scenes with interacting agents. In *CVPR*
- Lee, R., Kochenderfer, M. J., Mengshoel, O. J., & Silbermann, J. (2018). Interpretable categorization of heterogeneous time series data. In *SDM*
- Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., & Koch, O. (2008). A perception-driven autonomous urban vehicle. *Journal of Field Robotics*
- Li, C., Chan, S. H., & Chen, Y. (2020c). Who make drivers stop? towards driver-centric risk assessment: Risk object identification via causal inference. In *IROS*
- Li, Z., Motoyoshi, T., Sasaki, K., Ogata, T., & Sugano, S. (2018b). Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability. *CoRR*
- Li, B., Qi, X., Lukasiewicz, T., & Torr, P. H. S. (2020a). Manigan: Text-guided image manipulation. In *CVPR*
- Li, B., Qi, X., Torr, P. H. S., & Lukasiewicz, T. (2020b). Lightweight generative adversarial networks for text-guided image manipulation. In *NeurIPS*
- Li, Q., Tao, Q., Joty, S. R., Cai, J., & Luo, J. (2018a). VQA-E: explaining, elaborating, and enhancing your answers for visual questions. In *ECCV*
- Li, Y., Torralba, A., Anandkumar, A., Fox, D., & Garg, A. (2020d). Causal discovery in physical systems from videos. *NeurIPS*
- Lipton, Z. C. (2018). The mythos of model interpretability. *Communications ACM*
- Liu, Y., Hsieh, Y., Chen, M., Yang, C. H., Tegnér, J., & Tsai, Y. J. (2020). Interpretable self-attention temporal reasoning for driving behavior understanding. In *ICASSP*
- Liu, G., Reda, F. A., Shih, K. J., Wang, T., Tao, A., & Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In *ECCV*
- Liu, H., Yin, Q., & Wang, W. Y. (2019). Towards explainable NLP: A generative explanation framework for text classification. In *ACL*
- Lu, X., Tolmachev, A., Yamamoto, T., Takeuchi, K., Okajima, S., Takebayashi, T., Maruhashi, K., & Kashima, H. (2021). Crowdsourcing evaluation of saliency-based XAI methods. In *ECML-PKDD*
- Lu, J., Yang, J., Batra, D., & Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. In *NIPS*
- Lundberg, S. M., & Lee, S. (2017). A unified approach to interpreting model predictions. In *NIPS*

- Ly, A. O., & Akhloufi, M. A. (2020). Learning to drive by imitation: an overview of deep behavior cloning methods. *T-IV*
- Maaten, Lvd., & Hinton, G. (2008). Visualizing data using t-sne. *JMLR*
- Mac Aodha, O., Su, S., Chen, Y., Perona, P., & Yue, Y. (2018). Teaching categories to human learners with visual explanations. In *CVPR*
- Madumal, P., Miller, T., Sonenberg, L., & Vetere, F. (2020). Explainable reinforcement learning through a causal lens. In *AAAI*
- Makino, T., Jastrzebski, S., Oleszkiewicz, W., Chacko, C., Ehrenpreis, R., Samreen, N., Chhor, C., Kim, E., Lee, J., Pysarenko, K., Reig, B., Toth, H., Awal, D., Du, L., Kim, A., Park, J., Sodickson, D. K., Heacock, L., Moy, L., Cho, K., & Geras, K. J. (2020). Differences between human and machine perception in medical diagnosis. *CoRR*
- Malinowski, M., Rohrbach, M., & Fritz, M. (2017). Ask your neurons: A deep learning approach to visual question answering. *IJCV*
- Manzo, U. G., Chiroma, H., Aljojo, N., Abubakar, S., Popoola, S. I., & Al-Garadi, M. A. (2020). A survey on deep learning for steering angle prediction in autonomous vehicles. *IEEE Access*
- Maximov, M., Elezi, I., & Leal-Taixé, L. (2020). CIAGAN: conditional identity anonymization generative adversarial networks. In *CVPR*
- McAllister, R., Gal, Y., Kendall, A., van der Wilk, M., Shah, A., Cipolla, R., & Weller, A. (2017). Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In *IJCAI*
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. *CoRR*
- Mehta, A., Subramanian, A., & Subramanian, A. (2018). Learning end-to-end autonomous driving using guided auxiliary supervision. In *ICVGIP*
- Michon, J. (1984). A critical view of driver behavior models: What do we know, what should we do? *Human behavior and traffic safety*
- Mohseni, S., Jagadeesh, A., & Wang, Z. (2019). Predicting model failure using saliency maps in autonomous driving systems. *Workshop on Uncertainty and Robustness in Deep Learning @ICML*
- Moing, G. L., Vu, T., Jain, H., Pérez, P., & Cord, M. (2021). Semantic palette: Guiding scene generation with class proportions. In *CVPR*
- Molnar, C. (2019). Interpretable machine learning
- Moraffah, R., Karami, M., Guo, R., Raglin, A., & Liu, H. (2020). Causal interpretability for machine learning - problems, methods and evaluation. *SIGKDD Explorations*
- Mordan, T., Cord, M., Pérez, P., & Alahi, A. (2020). Detecting 32 pedestrian attributes for autonomous vehicles. *CoRR*
- Morgulis, N., Kreines, A., Mendelowitz, S., & Weisglass, Y. (2019). Fooling a real car with adversarial traffic signs. *CoRR*
- Mori, K., Fukui, H., Murase, T., Hirakawa, T., Yamashita, T., & Fujiyoshi, H. (2019). Visual explanation by attention branch network for end-to-end learning-based self-driving. In *IV*
- Morton, J. & Kochenderfer, M. J. (2017). Simultaneous policy learning and latent state inference for imitating driver behavior. In *ITSC*
- Müller, M., Dosovitskiy, A., Ghanem, B., & Koltun, V. (2018). Driving policy transfer via modularity and abstraction. In *CoRL*
- Narendra, T., Sankaran, A., Vijaykeerthy, D., & Mani, S. (2018). Explaining deep learning models using causal inference. *CoRR*
- Nguyen, A. M., Dosovitskiy, A., Yosinski, J., Brox, T., & Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *NIPS*
- Omeiza, D., Webb, H., Jiroka, M., & Kunze, L. (2021). Explanations in autonomous driving: A survey. *CoRR*
- Oramas, J., Wang, K., & Tuytelaars, T. (2019). Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks. In *ICLR*
- Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL*
- Park, D.H., Hendricks, L.A., Akata, Z., Rohrbach, A., Schiele, B., Darrell, T., & Rohrbach, M. (2018). Multimodal explanations: Justifying decisions and pointing to the evidence. In *CVPR*
- Patro, B. N., Anupriy, & Namboodiri, V. (2020). Explanation vs attention: A two-player game to obtain attention for VQA. In *AAAI*
- Pearl, J. (2009). Causality
- Pei, K., Cao, Y., Yang, J., & Jana, S. (2019). Deepxplore: automated whitebox testing of deep learning systems. *Communications ACM*
- Phan-Minh, T., Grigore, E. C., Boulton, F. A., Beijbom, O., & Wolff, E. M. (2020). Covernet: Multimodal behavior prediction using trajectory sets. In *CVPR*
- Pomerleau, D. (1988). ALVINN: an autonomous land vehicle in a neural network. In *NIPS*
- Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., & Carin, L. (2016). Variational autoencoder for deep learning of images, labels and captions. In *NIPS*
- Rajani, N. F., McCann, B., Xiong, C., & Socher, R. (2019). Explain yourself! leveraging language models for commonsense reasoning. In *ACL*
- Ramakrishnan, S., Agrawal, A., & Lee, S. (2018). Overcoming language priors in visual question answering with adversarial regularization. In *NeurIPS*
- Ramanishka, V., Chen, Y., Misu, T., & Saenko, K. (2018). Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *CVPR*
- Rathi, S. (2019). Generating counterfactual and contrastive explanations using SHAP. *Workshop on Humanizing AI (HAI) @IJCAI*
- Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*
- Ren, Z., Lee, Y. J., & Ryoo, M. S. (2018). Learning to anonymize faces for privacy preserving action detection. In: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.) *ECCV*
- Rezvani, T., Driggs-Campbell, K. R., Sadigh, D., Sastry, S. S., Seshia, S. A., & Bajcsy, R. (2016). Towards trustworthy automation: User interfaces that convey internal and external awareness. In *ITSC*
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “why should I trust you?”: Explaining the predictions of any classifier. In *SIGKDD*
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *AAAI*
- Riquelme, F., Goyeneche, A. D., Zhang, Y., Niebles, J. C., & Soto, A. (2020). Explaining VQA predictions using visual grounding and a knowledge base. *Image Vision Computer*
- Rodríguez, P., Caccia, M., Lacoste, A., Zamparo, L., Laradji, I. H., Charlin, L., Vázquez, D. (2021). Beyond trivial counterfactual explanations with diverse valuable explanations. *CoRR arXiv: abs/2103.10226*
- Rosenfeld, A. & Richardson, A. (2020). Why, who, what, when and how about explainability in human-agent systems. In *AAMAS*
- Sadat, A., Casas, S., Ren, M., Wu, X., Dhawan, P., & Urtasun, R. (2020). Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *ECCV*
- Salzmann, T., Ivanovic, B., Chakraborty, P., & Pavone, M. (2020). Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In H. Bischof, T. Brox, J. Frahm, & A. Vedaldi (Eds.) *Lecture Notes in Computer Science: ECCV*
- Samek, W., Binder, A., Montavon, G., Lapuschkin, S., & Müller, K. (2017). Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions Neural Networks Learning Systems*, 28(11), 2660–2673. <https://doi.org/10.1109/TNNLS.2016.2599820>
- Sato, M., & Tsukimoto, H. (2001). Rule extraction from neural networks via decision tree induction. In *IJCNN*
- Sauer, A., Savinov, N., & Geiger, A. (2018). Conditional affordance learning for driving in urban environments. In *CoRL*
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2020). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*

- Shariff, A., Bonnefon, J. F., & Rahwan, I. (2017). Psychological roadblocks to the adoption of self-driving vehicles. *Nature Human Behaviour*
- Shen, Y., Jiang, S., Chen, Y., Yang, E., Jin, X., Fan, Y., & Campbell, K. D. (2020). To explain or not to explain: A study on the necessity of explanations for autonomous vehicles. *CoRR*
- Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In *ICML*
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*
- Singla, S., Pollack, B., Chen, J., & Batmanghelich, K. (2020). Explanation by progressive exaggeration. In *ICLR*, OpenReview.net
- Srikanth, S., Ansari, J. A., R. K. R., Sharma, S., Murthy, J. K., & Krishna, K. M. (2019). INFER: intermediate representations for future prediction. In *IROS*
- Sun, Q., Ma, L., Oh, S. J., Gool, L. V., Schiele, B., & Fritz, M. (2018). Natural and effective obfuscation by head inpainting. In *CVPR*, *Computer Vision Foundation/IEEE Computer Society*
- Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *ICML*
- Suzuki, T., Kataoka, H., Aoki, Y., & Satoh, Y. (2018). Anticipating traffic accidents with adaptive loss and large-scale incident DB. In *CVPR*
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. In *ICLR*
- Tan, S., Wong, K., Wang, S., Manivasagam, S., Ren, M., & Urtasun, R. (2021). Scenegen: Learning to generate realistic traffic scenes. In *CVPR*
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., & Lau, K. (2006). Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*
- Tian, Y., Pei, K., Jana, S., & Ray, B. (2018). Deeptest: automated testing of deep-neural-network-driven autonomous cars. In *ICSE*
- Tjoa, E. & Guan, C. (2019). A survey on explainable artificial intelligence (XAI): towards medical XAI. *CoRR*
- Tomei, M., Baraldi, L., Bronzin, S., & Cucchiara, R. (2021). Estimating (and fixing) the effect of face obfuscation in video recognition. In *CVPR Workshops*
- Tommasi, T., Patricia, N., Caputo, B., & Tuytelaars, T. (2017). A deeper look at dataset bias. In *Domain adaptation in computer vision applications*
- Toromanoff, M., Wirbel, É., & Moutarde, F. (2020). End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *CVPR*
- Torralla, A., & Efros, A. A. (2011). Unbiased look at dataset bias. In *CVPR*
- Tran, D., Bourdev, L. D., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *ICCV*
- Tulyakov, S., Liu, M., Yang, X., & Kautz, J. (2018). Mocogan: Decomposing motion and content for video generation. In *CVPR*
- Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., Dolan, J., Duggins, D., Galatali, T., Geyer, C. & Gittleman, M. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*
- Vedantam, R., Zitnick, C. L., & Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *CVPR*
- Vilone, G. & Longo, L. (2020). Explainable artificial intelligence: a systematic review. *CoRR*
- Wachter, S., Mittelstadt, B. D., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*
- Wang, D., Devin, C., Cai, Q., Yu, F., & Darrell, T. (2019). Deep object-centric policies for autonomous driving. In *ICRA*
- Wojek, C., Walk, S., Roth, S., & Schiele, B. (2011). Monocular 3d scene understanding with explicit occlusion reasoning. In *CVPR*
- Wojek, C., Walk, S., Roth, S., Schindler, K., & Schiele, B. (2013). Monocular visual scene understanding: Understanding multi-object traffic scenes. *TPAMI*
- Xie, N., Ras, G., van Gerven, M., & Doran, D. (2020). Explainable deep learning: A field guide for the uninitiated. *CoRR*
- Xu, H. & Saenko, K. (2016). Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML*
- Xu, H., Gao, Y., Yu, F., & Darrell, T. (2017). End-to-end learning of driving models from large-scale video datasets. In *CVPR*
- Xu, Y., Yang, X., Gong, L., Lin, H., Wu, T., Li, Y., & Vasconcelos, N. (2020). Explainable object-induced action decision for autonomous vehicles. In *CVPR*
- Yang, Z., He, X., Gao, J., Deng, L., & Smola, A. J. (2016). Stacked attention networks for image question answering. In *CVPR*
- Yang, Z., Manivasagam, S., Liang, M., Yang, B., Ma, W., & Urtasun, R. (2020). Recovering and simulating pedestrians in the wild. *CoRL*
- You, T. & Han, B. (2020). Traffic accident benchmark for causality recognition. In *ECCV*
- Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., & Darrell, T. (2020). BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*
- Zahavy, T., Ben-Zrihem, N., & Mannor, S. (2016). Graying the black box: Understanding dqns. In *ICML*
- Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *ECCV*
- Zellers, R., Bisk, Y., Farhadi, A., & Choi, Y. (2019). From recognition to cognition: Visual commonsense reasoning. In *CVPR*
- Zeng, K., Chou, S., Chan, F., Niebles, J. C., & Sun, M. (2017). Agent-centric risk assessment: Accident anticipation and risky region localization. In *CVPR*
- Zeng, W., Luo, W., Suo, S., Sadat, A., Yang, B., Casas, S., & Urtasun, R. (2019). End-to-end interpretable neural motion planner. In *CVPR*
- Zhang, Q., & Zhu, S. (2018). Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*
- Zhang, Q., Cao, R., Shi, F., Wu, Y. N., & Zhu, S. (2018a). Interpreting CNN knowledge via an explanatory graph. In *AAAI*
- Zhang, H., Geiger, A., & Urtasun, R. (2013). Understanding high-level semantics by modeling traffic patterns. In: *ICCV*
- Zhang, Q., Wu, Y. N., & Zhu, S. (2018b). Interpretable convolutional neural networks. In *CVPR*
- Zhang, Q., Yang, X. J., & Robert, L. P. (2020). Expectations and trust in automated vehicles. In *CHI*
- Zhao, B., Yin, W., Meng, L., & Sigal, L. (2020). Layout2image: Image generation from layout. *IJCV*
- Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., & Torralba, A. (2015a). Object detectors emerge in deep scene cnns. In *ICLR*
- Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., & Torralba, A. (2015b). Object detectors emerge in deep scene cnns. In *ICLR*

- Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *CVPR*
- Zhou, B., Krähenbühl, P., & Koltun, V. (2019). Does computer vision matter for action? *Scientific Robotics*
- Zilke, J. R., Mencía, E. L., & Janssen, F. (2016). Deepred - rule extraction from deep neural networks. In *DS*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.