Noname manuscript No.
(will be inserted by the editor)

Explainability of vision-based autonomous driving systems: Review and challenges

Éloi Zablocki[*,1] · **Hédi Ben-Younes**[*,1] · **Patrick Pérez**[1] · **Matthieu Cord**[1,2]

Abstract This survey reviews explainability methods for vision-based self-driving systems. 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. Second, major recent state-of-the-art approaches to develop self-driving systems are quickly presented. Third, methods providing explanations to a black-box self-driving system in a post-hoc fashion are comprehensively organized and detailed. Fourth, 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 interpretabililty

Éloi Zablocki
E-mail: eloi.zablocki@valeo.com

Hédi Ben-Younes
E-mail: hedi.ben-younes@valeo.com

Patrick Pérez
E-mail: patrick.perez@valeo.com

Matthieu Cord
E-mail: matthieu.cord@valeo.com

[*] equal contribution
[1] Valeo.ai
[2] Sorbonne Université

1 Introduction

1.1 Self-driving systems

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; 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. On the industry side, several manufacturers are already producing cars equipped with advanced computer vision technologies for automatic lane following, assisted parking, or collision detection among other things. Meanwhile, constructors are working on and designing prototypes with level 4 and 5 autonomy. The development of autonomous vehicles has the potential to reduce congestions, fuel consumption, and crashes, and it can increase personal mobility and save lives given that nowadays the vast majority of car crashes are caused by human error (Anderson et al, 2014).

The first steps in the development of autonomous driving systems are taken with the collaborative European project PROMETHEUS (Program for a European Traffic with Highest Efficiency and Unprecedented Safety) (Xie et al, 1993) at the end of the '80s and the Grand DARPA Challenges in the late 2000s. At these times, systems are heavily-engineered pipelines (Urmson et al, 2008; Thrun et al, 2006) and their modular aspect decomposes the task of driving into several smaller tasks — from perception to planning — which has the advantage to offer interpretability and transparency to the processing. Nevertheless, modular pipelines have also known limitations such as the lack of flexibility, the need for handcrafted representations, and the risk of error propagation. In the 2010s, we observe an interest in approaches aiming to *train* driving systems, usually in the form of neural networks, either by leveraging large quantities of expert recordings (Bojarski et al, 2016; Codevilla et al, 2018; Ly and Akhloufi, 2020) or through simulation (Espié 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 produce 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.

1.2 Need for explainability

The need for explainability is multi-factorial and depends on the concerned people, whether they are end-users, legal authorities, or self-driving car designers. End-users and citizens need to trust the autonomous system and to be reassured (Choi and Ji, 2015). Moreover, designers of self-driving models need to understand the limitations of current models to validate them and improve future versions (Tian et al, 2018). Besides, regarding legal and regulator bodies, it is needed to access explanations of the system for liability purposes, especially in the case of accidents (Rathi, 2019; Li et al, 2018c).

The fact that autonomous self-driving systems are not inherently interpretable has two main origins. On the one hand, models are designed and trained within the deep learning paradigm which has known explainability-related limitations: datasets contain numerous biases and are generally not precisely curated, the learning and generalization capacity remains empirical in the sense that the system may learn from spurious correlation and overfit on common situations, also, the final trained model represents a highly-non-linear function and is non-robust to slight changes in the input space. On the other hand, self-driving systems have to simultaneously solve intertwined tasks of very different natures: perception tasks with detection of lanes and objects, planning and reasoning tasks with motion forecasting of surrounding objects and of the ego-vehicle, and control tasks to produce the driving end-commands. Here, explaining a self-driving system thus means disentangling predictions of each implicit task, and to make them human-interpretable.

1.3 Research questions and focus of the survey

Two complementary questions are the focus of this survey and they guide its organization:

1. Given a trained self-driving model, coming as a black-box, how can we explain its behavior?
2. How can we design learning-based self-driving models which are more interpretable?

Regardless of driving considerations, these questions are asked and answered in many generic machine learning papers. Besides, some papers from the vision-based autonomous driving literature propose interpretable driving systems. In this survey, we bridge the gap between general X-AI methods that can be applied for the self-driving literature, and driving-based approaches claiming explainability. In practice, we reorganize and cast the autonomous driving literature into an X-AI taxonomy that we introduce. Moreover, we detail generic X-AI approaches — some have not been used yet in the autonomous driving context — and that can be leveraged to increase the explainability of self-driving models.

1.4 Positioning

Many works advocate for the need of *explainable* driving models (Ly and Akhloufi, 2020) and published reviews about explainability often mention autonomous driving as an important application for X-AI methods. However, there are only a few works on interpretable autonomous driving systems, and, to the best of our knowledge, there exists no survey focusing on the interpretability of autonomous driving systems. Our goal is to bridge this gap, to organize and detail existing methods, and to present challenges and perspectives for building more interpretable self-driving systems.

This survey is the first to organize and review 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 (2020) 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, and Kiran et al (2020) review self-driving models based on deep reinforcement learning.

Besides, 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).

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.

1.5 Contributions and outline

Throughout the survey, we review explainability-related definitions from the X-AI literature and we gather a large number of papers proposing self-driving models that are explainable or interpretable to some extent, and organize them within an explainability taxonomy we define. Moreover, 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.

This survey is organized as follows: Section 2 contextualizes and motivates the need for interpretable autonomous driving models and presents a taxonomy of explainability methods, suitable for self-driving systems; Section 3 gives an overview of neural driving systems and explores reasons why it is challenging to explain them; Section 4 presents post-hoc methods providing explanations to any black-box self-driving model; Section 5 turns to approaches providing more transparency to self-driving models, by adding explainability constraints in the design of the systems; this section also presents potential future directions to increase further explainability of self-driving systems. Section 6 presents the particular use-case of explaining a self-driving system by means of natural language justifications.

2 *Explainability* in the context of autonomous driving

This section contextualizes the need for interpretable driving models. In particular, we present the main motivations to require increased explainability in Section 2.1, we define and organize explainability-related terms in Section 2.2 and, in Section 2.3, we answer questions such as *who needs explanations? what kind? for what reasons? when?*

2.1 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 in machine learning 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 performances an actual car would have when deployed to

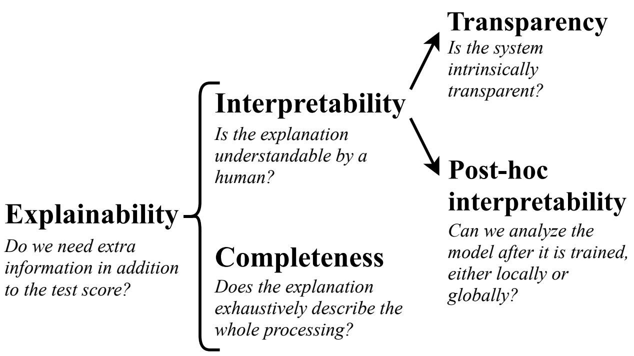


Fig. 1: **Taxonomy of explainability terms adopted in this survey**. *Explainability* is the combination of *interpretability* (= comprehensible by humans) and *completeness* (= exhaustivity of the explanation) aspects. There are two approaches to have interpretable systems: approaches intrinsic to the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

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.

2.2 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 Figure 1.

In human-machine interactions, *explainability* is defined as the ability for the human user to understand the agent's logic (Rosenfeld and Richardson, 2019). 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 (2019). 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, gaining *post-hoc interpretability* amounts to acquiring extra information in addition to the model metric, generally after the driving decision is made. 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.

Besides, 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).

2.3 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.

2.3.1 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 (2019) 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.

2.3.2 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), 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

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

Table 1: **The four W's of explainable driving AI**. Who needs explanations? What kind? For what reasons? When?

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).

2.3.3 Regulators and legal considerations

In the European General Data Protection Regulation (GDPR)[1], it is stated that users have the right to obtain explanations from automated decision-making systems. 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 (Rathi, 2019; Wachter et al, 2017)), and their decisions should be available for contesting and contradiction. This should prevent unfair and/or unethical behaviors of algorithms. Even though these questions are crucial for the broad machine learning community in general, the field of autonomous driving is not directly impacted by such problems as systems do not use personal data.

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

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

who will likely spend large amounts of time studying the system (Rosenfeld and Richardson, 2019), and who are thus inclined to receive rich explanations with great amounts of detail.

3 Self-driving cars

In this section, we present an overview of the main approaches tackling autonomous driving, regardless of explainability concerns, in Section 3.1. Moreover, in Section 3.2, we delineate the explainability challenges toward the design of interpretable self-driving systems.

3.1 Autonomous driving: learning-based self-driving models

This subsection gives an outlook over the historical shift from modular pipelines towards end-to-end learning based models (Section 3.1.1); the main architectures used in modern driving systems are presented (Section 3.1.2), as well as how they are trained and optimized (Section 3.1.3). Finally, the main public datasets used for training self-driving models are presented in Section 3.1.4.

3.1.1 From historical 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. The common point between STANLEY, BOSS, and the vast majority of the other approaches at this time (Leonard et al, 2008) is the *modularity*. Leveraging strong suites of sensors, these systems are 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, they lack flexibility to account for real-world uncertainties and to generalize to unplanned scenarios. 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 machine learning-based driving systems, and in particular on deep neural networks. In this survey, we focus on these deep learning systems for autonomous driving.

3.1.2 Driving architecture

We now present the different components constituting most of the existing learning-based driving systems. As illustrated in Figure 2, we can distinguish four key elements involved in the design of a neural driving system: input sensors, input representations, output type, and learning paradigm.

Sensors. Sensors are the hardware interface through which the neural network perceives its environment. Typical neural driving systems rely on sensors from two families: *proprioceptive* sensors and *exteroceptive* sensors. Proprioceptive sensors provide information about the internal vehicle state such as speed, acceleration, yaw, change of position, and velocity. They are measured through tachometers, inertial measurement units (IMU), and odometers. All these sensors communicate through the controller area network (CAN) bus, which allows signals to be easily accessible. In contrast, exteroceptive sensors acquire information about the surrounding environment. They include cameras, radars, LiDARs, and GPS:

– *Cameras* are passive sensors that acquire a color signal from the environment. They provide RGB videos that can be analyzed using the vast and growing computer vision literature treating video signals. Despite being very cheap and rich sensors, there are two major downsides to their use. First, they are sensitive to illumination changes. It implies that

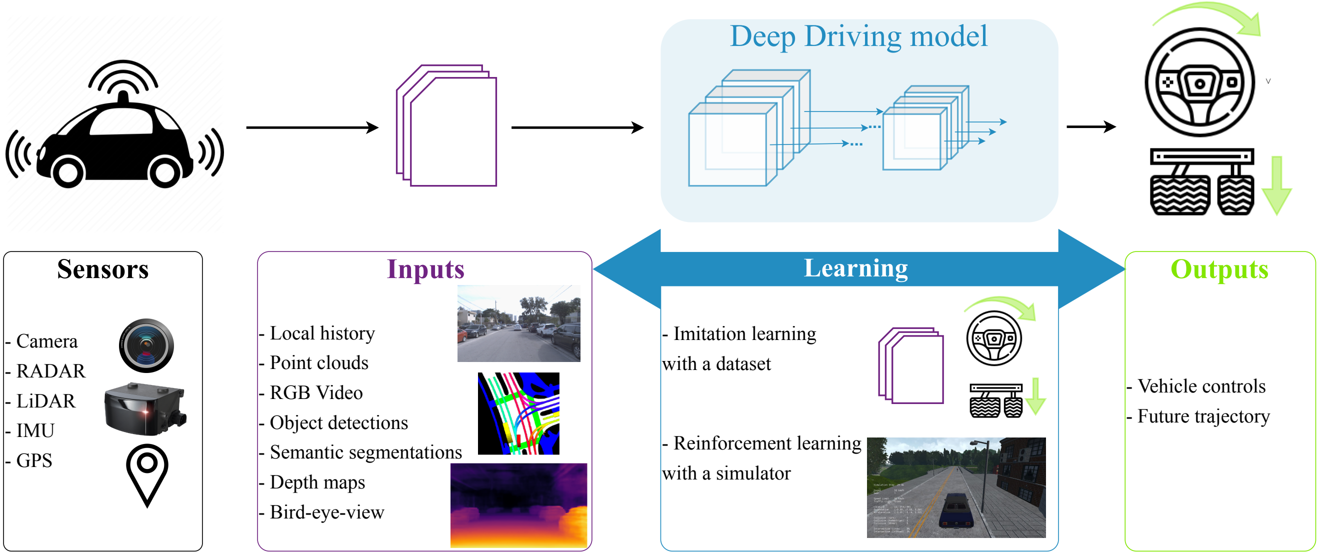


Fig. 2: Overview of neural network-based autonomous driving systems.

day/night changes, in particular, have a strong impact on the performance of downstream algorithms, even if this phenomenon is tackled by some recent work on domain adaptation (Romera et al, 2019). Second, they perceive the 3D world through 2D projection, making depth sensing with a single view challenging. This is an important research problem in which deep learning has shown promising results (Godard et al, 2017, 2019; Guizilini et al, 2020), but is still not robust enough.

– *Radars* are active sensors that emit radio waves and measure the travel time and frequency shift of the received reflected waves. They can provide information about the distance and speed of other vehicles at long range, and are not sensitive to weather conditions. However, their accuracy can be quite poor.
– *LiDARs* work similarly as radars but emit light waves instead of radio waves. They are much more accurate than radars and can be used to construct a 3D representation of the surrounding scene. However, contrary to radars, they do not measure the relative speed of objects and are affected by bad weather (snow and heavy fog in particular). Also, the price and bulk of high-end LiDARs make them unsuited until now for the majority of the car market.
– *GPS* receivers can estimate precise geolocation, within an error range of 30 centimeters, by monitoring multiple satellites to determine the precise position of the receivers.

For a more thorough review of driving sensors, we refer the reader to (Yurtsever et al, 2020).

Input representation. Once sensory inputs are acquired by the system, they are processed before being passed to the neural driving architecture. Approaches differ by the way they process the raw signals before feeding them to the network, and this step constitutes an active research topic. Focusing on cameras, recent work proposed to use directly the raw image pixels (Bojarski et al, 2016; Codevilla et al, 2018). But most successful methods build a structured representation of the scene using computer vision models. This type of approach is referred to as *mediated perception* (Ullman, 1980): several perception systems provide their understanding of the world, and their outputs are aggregated to build an input for the driving model. An example of such vision tasks is object detection, which aims at finding and classifying relevant objects in a scene (cars, bicycles, pedestrians, stop signs, *etc.*). Popular object detectors such as Faster-RCNN (Ren et al, 2015) and YOLO (Redmon et al, 2016; Redmon and Farhadi, 2017, 2018) operate at the image level, and the temporality of the video can be leveraged to jointly detect and track objects (Behrendt et al, 2017; Li et al, 2018a; Fernandes et al, 2021). See (Feng et al, 2019) for a comprehensive survey on object detection and semantic segmentation for autonomous driving, including datasets, methods using multiple sensors and challenges. In addition to detecting and tracking objects, understanding the vehicle's environment involves extracting depth information, i.e. knowing the distance that separates the vehicle from each point in the space. Approaches to depth estimation vary depending on the sensors that are available: direct LiDAR measurements (Xu et al, 2019; Tang et al, 2019; Jaritz et al, 2018; Park et al, 2020), stereo cameras (Chang and Chen, 2018; Kendall et al, 2017) or even single monocular cameras (Fu et al, 2018; Kuznietsov et al, 2017; Amiri et al, 2019; Godard et al, 2017; Zhou et al, 2017; Casser et al, 2019; Godard et al, 2019; Guizilini et al, 2020). Other types of semantic information can be used to complement and enrich inputs such as the recognition of pedestrian intent (Abughalieh and Alawneh, 2020; Rasouli et al, 2019).

Mediated perception contrasts with the *direct perception* approach (Gibson, 1979), which instead extracts visual *affordances* from an image. Affordances are scalar indicators that describe the road situation such as curvature, deviation to neighboring lanes, or distances between ego and other vehicles. These human-interpretable features are usually recognized using neural networks (Chen et al, 2015; Sauer et al, 2018; Xiao et al, 2020). Then, they are passed at the input of a driving controller which is usually hard-coded, even if some recent approaches use affordance recognition to provide compact inputs to learning-based driving systems (Toromanoff et al, 2020).

Outputs. Ultimately, the goal is to generate vehicle controls. Some approaches, called end-to-*end*, tackle this problem by training the deep network to directly output the commands (Pomerleau, 1988; Bojarski et al, 2016; Codevilla et al, 2018). However, in practice most methods instead predict the future trajectory of the autonomous vehicle; they are called end-to-*mid* methods. The trajectory is then expected to be followed by a low-level controller, such as the proportional–integral–derivative (PID) controller. The different choices for the network output, and their link with explainability, are reviewed and discussed in Section 5.3.

3.1.3 Learning

Two families of methods coexist for training self-driving neural models: *behavior cloning* approaches, which leverage datasets of human driving sessions (Section 3.1.3), and *reinforcement learning* approaches, which train models through trial-and-error simulation (Section 3.1.3).

Behavior cloning (BC). These approaches leverage huge quantities of recorded human driving sessions to learn the input-output driving mapping by imitation. In this setting, the network is trained to mimic the commands applied by the expert driver (end-to-end models), or the future trajectory (end-to-mid models), in a supervised fashion. The objective function is defined in the output space (vehicle controls, future trajectories, ...) and minimized on the training set composed by human driving sessions. Initial attempt to behavior cloning of vehicle controls was made in (Pomerleau, 1988), and continued later in (Chen et al, 2015; Bojarski et al, 2016; Codevilla et al, 2018). For example, DESIRE (Lee et al, 2017) is the first neural trajectory prediction model based on behavior cloning.

Even if it seems satisfactory to train a neural network based on easy-to-acquire expert driving videos, imitation learning methods suffer from several drawbacks. First, in the autoregressive setting, the test distribution is different to the train distribution due to the *distributional shift* (Ross et al, 2011) between expert training data and online behavior (Zeng et al, 2019; Codevilla et al, 2019). At train time, the model learns to make its decision from a state which is a consequence of previous decisions of the expert driver. As there is a strong correlation between consecutive expert decisions, the network finds and relies on this signal to predict future decisions. At deployment, the loop between previous prediction and current input is closed and the model can no longer rely on expert previous decisions to take an action. This phenomenon gives low train and test errors, but very bad behavior at deployment. Second, supervised training is harmed by biases in datasets: a large part of real-world driving consists of a few simple behaviors and only rare cases require complex reasoning. Also, systems trained with supervised behavior cloning suffer from causal confusion (de Haan et al, 2019), such that spurious correlations cannot be distinguished from true causal relations between input elements and outputs. Besides, behavior cloning methods are known to poorly explore the environment, they are data-hungry, requiring massive amounts of data to generalize. Finally, behavior cloning methods are unable to learn in situations that are not contained in driving datasets: these approaches have difficulties dealing with dangerous situations that are never demonstrated by experts (Chen et al, 2020a).

Reinforcement learning (RL). Alternatively, researchers have explored using RL to train neural driving systems (Kiran et al, 2020; Toromanoff et al, 2020). This paradigm learns a policy by balancing self-exploration and reinforcement (Chen et al, 2020a). This training paradigm does not require a training set of expert driving but relies instead on a simulator. In (Dosovitskiy et al, 2017), the autonomous vehicle evolves in the CARLA simulator, where it is asked to reach a high-level goal. As soon as it reaches the goal, collides with an object, or gets stuck for too long, the agent receives a reward, positive or negative, which it tries to maximize. This reward is a scalar value that combines speed, distance traveled towards the goal, collision damage, overlap with sidewalk, and overlap with the opposite lane.

In contrast with BC, RL methods do not require any annotations and have the potential to achieve superhuman performances through exploration. However, these methods are inefficient to train, they necessitate a simulator, and the design of the reward function is delicate. Besides, as shown in (Dosovitskiy et al, 2017), RL-based systems achieve lower performance than behavior cloning training. More importantly, even if driving in simulation can provide insights about system design, the ultimate goal is to drive in the real world. Promising results have been provided in (Kendall et al, 2019) to training an RL driving system in the real world, but the problem is not solved yet. A detailed review of reinforcement learning models is provided in (Kiran et al, 2020).

It is also worth mentioning the family of *Inverse Reinforcement Learning (IRL)* methods, which use both expert driving data and simulation. IRL is based on the assumption that humans drive optimally. These techniques aim at discovering the unknown reward function justifying human driving behavior (Ng and Russell, 2000; Sharifzadeh et al, 2016; Kiran et al, 2020). On standard control tasks, IRL approaches are particularly efficient in the low data regime, i.e. when few expert trajectories are available (Ho and Ermon, 2016). In the context of autonomous driving, IRL has been mostly employed for learning on driving-related subtasks such as highway driving (Abbeel and Ng, 2004; Syed and Schapire, 2007), automatic parking lot navigation (Abbeel et al, 2008), urban driving (Ziebart et al, 2008), lane changing (Sharifzadeh et al, 2016) and comfortable driving (Kuderer et al, 2015). Unfortunately, IRL algorithms are expensive to train as they involve a reinforcement learning step between cost estimation to policy training and evaluation (Kiran et al, 2020).

3.1.4 Driving datasets

We list here public datasets used for training self-driving models. We do not exhaustively cover all of them and refer the reader to (Janai et al, 2020) for more datasets. However, we focus on datasets that can be used for designing transparent driving systems thanks to extra annotations, or that can be used to learn to provide post-hoc explanations. Table 2 summarizes the main characteristics of these datasets.

Geiger et al (2013) have pioneered the work on multi-modal driving datasets with KITTI, which contains 1.5 hours of human driving acquired through stereo cameras and LiDAR sensors. The dataset offers 15k frames annotated with 3D bounding boxes and semantic segmentation maps. More recently, Caesar et al (2020) released the nuScenes dataset composed of one thousand clips of 20 seconds each. The acquisition was done through 6 cameras for a 360° field of view, 5 radars, and one LiDAR. Keyframes are sampled at 2Hz and fully annotated with 3D bounding boxes of 23 object classes. Besides, a human-annotated semantic map of 11 classes (e.g. *traffic light, stop line, drivable area*) is associated to the clips on keyframes, and can be used in combination with the precise localization data (with errors below 10 cm). Other multi-modal driving datasets have been released (e.g., Waymo Open Dataset (Sun et al, 2020), ArgoVerse (Chang et al, 2019a), Lyft L5 (Houston et al, 2020)) with a varying number of recorded hours, type and number of sensors, and semantic annotations. Contrasting with these datasets using a calibrated camera, in BDDV (Xu et al, 2017), the authors have collected a large quantity of dash-cam driving videos and explored the use of this low-quality data to learn driving models.

3.2 Challenges for explainable autonomous vehicles

Introducing explainability in the design of learning-based self-driving systems is a challenging task. These concerns arise from two aspects: modern self-driving systems are deep learning models, which brings known shortcomings associated with these trained architectures as detailed in Section 3.2.1. Besides, these systems are implicitly solving several heterogeneous subtasks at the same time as explained in Section 3.2.2.

3.2.1 Autonomous vehicles are 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 perfectly master the dataset, the trained model, and the learning phase. These barriers to explainability are reported in Figure 3.

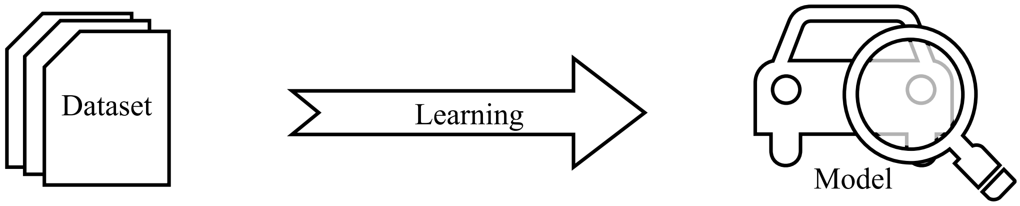
	Vol.	Sensors					Annotations
		Cameras	LiDAR	Radar	GPS/IMU	CAN	
KITTI (Geiger et al, 2013)	1.5 hours	2 RGB + 2 grayscale	✓	✗	✓	✓	2D/3D bounding boxes, tracking, pixel-level
Cityscapes (Cordts et al, 2016)	20K frames	2 RGB	✗	✗	✓	✗	Pixel-level
SYNTHIA (Ros et al, 2016)	200K frames	2 multi-cameras	✗	✗	✗	✗	Pixel-level, depth
HDD (Ramanishka et al, 2018)	104 hours	3 cameras	✓	✓	✓	✓	Driver behavior annotations (labels)
BDDV (Xu et al, 2017)	10K hours	dash-cam	✗	✗	✓	✗	✗
BDD100K (Yu et al, 2020)	100K × 40s	dash-cam	✗	✗	✓	✗	2D bounding boxes, tracking, pixel-level
BDD-A (Xia et al, 2018)	1232 × 10s	dash-cam	✗	✗	✓	✗	Human gaze
BDD-X (Kim et al, 2018)	7K × 40s	dash-cam	✗	✗	✓	✗	Textual explanations associated to video segments
BDD-OIA (Xu et al, 2020)	23K × 5s	dash-cam	✗	✗	✓	✗	Authorized actions, explanations (classif)
BDD-A extended (Shen et al, 2020)	1103 × 10s	dash-cam	✗	✗	✓	✗	Human gaze, human desire for an explanation score
Brain4Cars (Jain et al, 2016)	1180 miles	Road + cabin cameras	✗	✗	✓	✗	✗
nuScenes (Caesar et al, 2020)	1000 × 20s	6 cameras	✓	✓	✓	✓	2D/3D bounding boxes, tracking, maps
ApolloScape (Huang et al, 2018)	100 hours	6 cameras	✓	✓	✓	✓	fitted 3D models of vehicles, pixel-level
Lyft L5 (Houston et al, 2020)	1K hours	7 cameras	✓	✓	✓	✗	2D aerial boxes, HD maps
Waymo OpenDataset (Sun et al, 2020)	1150 × 20s	5 cameras	✓	✗	✗	✗	2D/3D bounding boxes, tracking
ArgoVerse (Chang et al, 2019a)	300K × 5s	360° + stereo cameras	✓	✗	✓	✗	2D/3D bounding boxes, tracking, maps
DoTA (Yao et al, 2020)	4677 videos	dash-cam	✗	✗	✗	✗	Temporal and spatial (tracking) anomaly detection
Road Scene Graph (Tian et al, 2020)	506 videos	6 cameras	✓	✓	✓	✓	Relationships
CTA (You and Han, 2020)	1935 videos	dash-cam	✗	✗	✗	✗	Accidents labeled with cause and effects and temporal segmentation

Table 2: Summary of driving datasets. Most used driving datasets for training learning-based driving models are presented in Section 3.1.4; in addition datasets that specifically provide explanation information are presented throughout Section 5.2.1.

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 and is considered as a *black-box*. 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



Explainability hurdles	- Thousands of driving sessions - Various biases - Under/over represented situations	- Spurious correlations - Underfit or overfit some situations - Misspectified objective	- Black-box - Million of parameters - Highly non-linear - Robustness issues - Prone to adversarial attacks
Explainability questions	- Is situation like "X" encountered in the dataset?	- Did the model correctly learn on situations that rarely occur? - Did the model learn to make decisions for the good reasons?	- How will the model behave in a new scenario? - Can the model generalize to unseen situations? - Is the model robust to slightly perturbated inputs?

Fig. 3: Explainability hurdles and questions for autonomous driving models, as seen from a machine learning point of view.

not underfit on some situations and overfit on others. Besides, the model may learn to ground its decisions on spurious correlations during training instead of leveraging causal signals (Codevilla et al, 2019; de Haan et al, 2019). We aim at finding answers to questions like *Which factors* caused *this decision to be taken?*

These known issues related to training deep models apply beyond autonomous driving applications. There is a strong research trend trying to tackle these problems through the prism of explainability, to characterize the problems, and to try to mitigate them. In Section 4 and Section 5, we review selected works that link to the self-driving literature.

3.2.2 Autonomous vehicles are 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 vehicle inevitably requires understanding how each of these intermediate steps is carried and how it interacts with others, as illustrated in Figure 4. 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, some recent end-to-end neural networks blur the lines and perform these simultaneously (Bojarski et al, 2016). 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, dis-

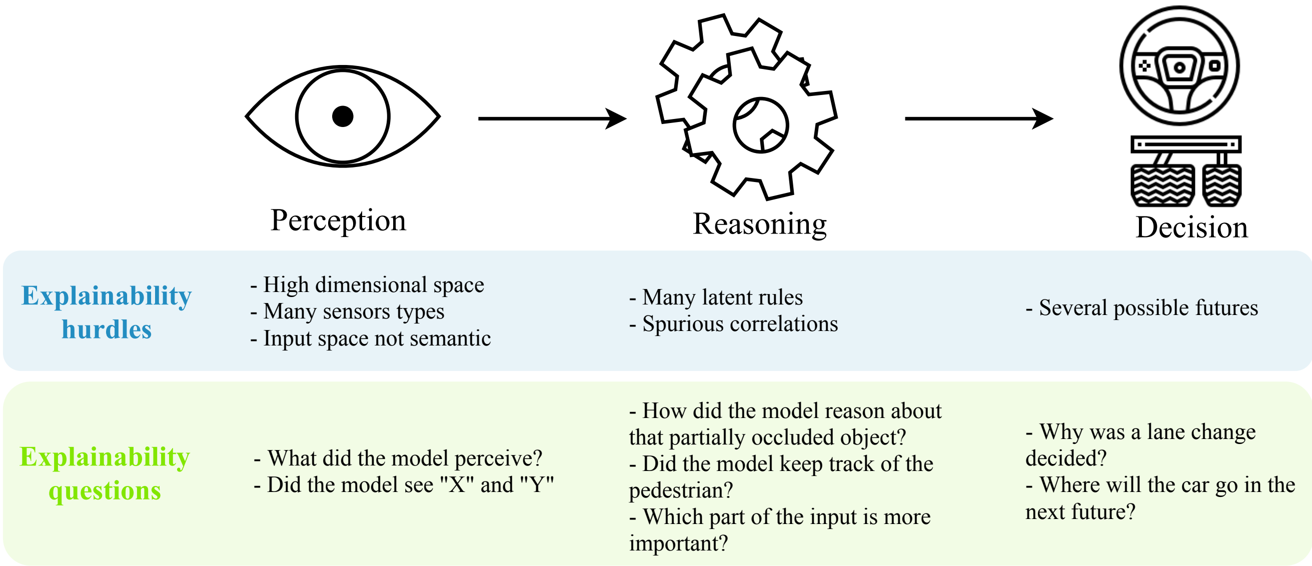


Fig. 4: Explainability hurdles and questions for autonomous driving models, as seen from an autonomous driving point of view.

entangling perception, reasoning, and decision in neural driving systems constitutes a non-trivial challenge.

3.2.3 Organization of the rest of the survey

As explained in this previous section, there are many aspects to be explained in a self-driving model. Several orthogonal dimensions can be identified to organize the X-AI literature, regarding for example whether or not the explanation is provided in a post-hoc fashion, whether it globally explains the model or just a specific instance, depending on the type of input/output/model. At this point, we want to emphasize the fact that the intention of our article is not to exhaustively review the literature on X-AI, which was comprehensively covered in many surveys (Gilpin et al, 2018; Adadi and Berrada, 2018; Xie et al, 2020; Vilone and Longo, 2020; Moraffah et al, 2020; Beaudouin et al, 2020), but to cover existing work at the intersection of explainability and driving systems. For the sake of simplicity and with autonomous driving research in mind, we classify the methods into two main categories. Methods that belong to the first category (Section 4) are applied to an already-trained deep network and are designed to provide *post-hoc* explanations The second category (Section 5) contains intrinsically explainable systems, where the model is designed to provide upfront some degree of interpretability of its processing. This organization choice is close to the one made in (Gilpin et al, 2018; Xie et al, 2020).

4 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 pretrained 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 (Section 4.1), and *global* methods that seek to explain the model in its entirety (Section 4.2), i.e. by gaining a finer understanding on learned representations and activations. Besides, we also make a connection with the system validation literature which aims at automatically making a stratified evaluation of deep models on various scenarios and discovering failure situations in Section 4.3. Selected references from this section are reported in Table 3.

4.1 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 three types of approaches: saliency methods which determine regions of image x influencing the most the decision (Sec-

Approach	Explanation type	Section	Selected references
Local	Saliency map	4.1.1	VisualBackprop (Bojarski et al, 2018, 2017) Causal filtering (Kim and Canny, 2017) Grad-CAM (Sauer et al, 2018) Meaningful Perturbations (Liu et al, 2020)
	Local approximation	4.1.2	∅
	Counterfactual interventions	4.1.3	Shifting objects (Bojarski et al, 2017) Removing objects (Li et al, 2020c) Causal factor identification (Bansal et al, 2019)
Global	Model translation	4.2.1	∅
	Representations	4.2.2	Neuron coverage (Tian et al, 2018)
	Prototypes and Criticisms	4.2.3	∅
Evaluation		4.3	Specific test cases (Bansal et al, 2019) Subset filtering (Hecker et al, 2020) Automatic finding of corner cases (Tian et al, 2018)

Table 3: Key references aiming at explaining a learning-based driving model.

tion 4.1.1), local approximations which approach the behavior of the black-box model f locally around the instance x (Section 4.1.2) and counterfactual analysis which aims to find the cause in x that made the model predict $f(x)$ (Section 4.1.3).

4.1.1 Saliency methods

A *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 two main lines of methods to obtain a saliency map for a trained network, namely *back-propagation methods* and *perturbation-based 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 its generalized version (Simonyan et al, 2014), Guided Backprop (Mahendran and Vedaldi, 2016), 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.

In the autonomous driving literature, saliency methods have been employed to highlight image regions that influence the most driving decisions. By doing so, these methods mostly explain the perception part of the driving architectures. The first 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. 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 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 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 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 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).

While 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) 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 saliency methods is to use additional annotations provided by humans, which can be costly to acquire, to be matched with the produced saliency map (Fong and Vedaldi, 2017). Second, Adebayo et al (2018) indicate that the generated heat maps may be misleading as some saliency methods are independent both of the model and the data. Indeed, they show that some saliency methods behave like edge-detectors even when they are applied to a randomly initialized model. Besides, 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.

4.1.2 Local approximation methods

The idea of a local approximation method is to approach the behavior of the black-box 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. Such methods include the Local Interpretable Model-agnostic Explanations (LIME) approach (Ribeiro et al, 2016), which learns an interpretable-by-design input/output mapping, mimicking the behavior of the main model in the neighborhood of an input. In practice, such mapping can be instantiated by a decision tree or a linear model. To constitute a dataset to learn the surrogate model, data points are sampled around the input of interest and corresponding predictions are computed by the black-box model. This forms the training set on which the interpretable model learns. Note that in the case of LIME, the interpretable student model does not necessarily use the raw instance data but rather an interpretable input, such as a binary vector indicating the presence or absence of a superpixel in an image. The SHapley Additive exPlanations (SHAP) approach (Lundberg and Lee, 2017) has later been introduced to generalize LIME, as well as other additive feature attribution methods, and provides more consistent results. In (Ribeiro et al, 2018), anchors are introduced to provide local explanations of complex black-box models. They consist of high-precision if-then rules, which constitute sufficient conditions for prediction. Similarly to LIME, perturbations are applied to the example of interest to create a local dataset. Anchors are then found from this local distribution, consisting of input chunks which, when present, almost surely preserve the prediction made by the model.

In the autonomous driving literature, we are not aware of any work that aims to explain a self-driving model by locally approximating it with an interpretable model. Some relevant work though is the one of Ponn et al (2020), which leverages the SHAP approach to investigate performances of object detection algorithms in the context of autonomous driving. The fact that almost no paper explains self-driving models with local approximation methods is likely due to the cost of local approximation strategies, as a set of perturbed inputs are sampled and forwarded in the main model to collect their corresponding labels. For example, in the case of SHAP, the number of forward passes required to explain the model is exponential in the number of features, which is prohibitive when it comes to explaining computer vision models with input pixels. Sampling strategies need to be carefully designed to reduce the complexity of these explanation models. Besides, those methods operate on a simplified input representation instead of the raw input. This interpretable semantic basis should be chosen wisely, as it constitutes the vocabulary that can be used by the explanation system. Finally, these techniques were shown to be highly sensitive to hyper-parameter choices (Bansal et al, 2020).

4.1.3 Counterfactual explanation

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?" 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 *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. Besides, 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. Besides, 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.

Regarding the autonomous driving literature, there only exists a limited 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

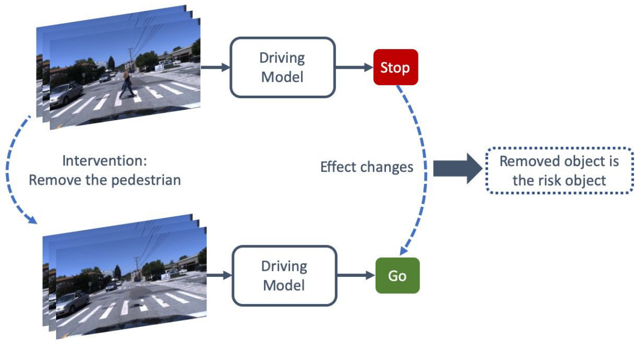


Fig. 5: Removing a pedestrian induces a change in the driver's decision from *Stop* to *Go*, which indicates that the pedestrian is a risk-object. Credits to (Li et al, 2020c).

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 Figure 5). 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.

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 Section 5.1.2. Interestingly, as more and more neural driving systems rely on semantic representations (see Section 3.1.2), 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.

4.2 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. We cover three families of methods to provide global explanations: *model translation* techniques, which aim at transforming an opaque neural network into a more interpretable model (Section 4.2.1), *representations explanation* to analyze the knowledge contained in the data structures of the model (Section 4.2.2), and *prototypes-based* methods, which provide global explanations by selecting and aggregating multiple local explanations (Section 4.2.3).

4.2.1 Model translation

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 Section 4.1.2, model translation methods are different as they should approximate the main function *globally* across the data distribution. In the work of Zhang et al (2018) 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.

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.

4.2.2 Explaining representations

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 (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).

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. Tian et al (2018) use this value to partition the input space: to increase the neuron coverage of the model, they automatically generate corner cases where the self-driving model fails. This approach is presented in more details in Section 4.3. Overall, we encourage researchers to provide more insights on what is learned in intermediate representations of self-driving models through methods explaining representations.

4.2.3 Protoypes/Criticism and submodular picks

A *prototype* is a specific data instance that represents well the data. Prototypes are chosen simultaneously to represent the data distribution in a non-redundant way. Clustering methods, such as partitioning around medoids (Kaufmann, 1987), can be used to automatically find prototypes. As another example, the MMD-critic algorithm (Kim et al, 2016) selects prototypes such that their distribution matches the distribution of the data, as measured with the Maximum Mean Discrepency (MMD) metric. Once prototypes are found, *criticisms* — instances that are not well represented by the set of prototypes — can be chosen where the distribution of the data differs from the one of the prototypes. Despite describing the data, prototypes and criticisms can be used to make a black-box model interpretable. Indeed, by looking at the predictions made on these prototypes, it can provide insight and save time to users who cannot examine a large number of explanations and rather prefer judiciously chosen data instances. Ribeiro et al (2016) propose a similar idea to select representative data instances, which they call *submodular picks*. Using the LIME algorithm (see Section 4.1.2), they provide a local explanation for every instance of the dataset and use the obtained features importance to find the set of examples that best describe the data in terms of diversity and non-redundancy.

This type of approach has not been employed as an explanation strategy in the autonomous driving literature. Indeed, the selection of prototypes and criticisms heavily depends on the kernel used to measure the matching of distributions, which has no trivial design in the case of high-dimensional inputs such as video or LiDAR frames.

4.3 Fine-grain evaluation and stratified performances

System validation is closely connected to the need for model explanation. One of the links between these two fields is made of methods that automatically evaluate deep models on a wide variety of scenarios and that seek rare corner cases where the model fails. Not only are these methods essential for validating models, but they can provide a feedback loop to improve future versions with learned insights. In computer science and embedded systems literature, validation and performance analysis is related to the software and security literature. However, we are dealing here with *learned* models and methods from these fields of research poorly apply. Even if several attempts have been made to formally verify the safety properties of deep models, these techniques do not scale to large-scale networks such as the ones used for self-driving (Huang et al, 2017; Katz et al, 2017). We thus review in this subsection some methods that are used to precisely evaluate the behavior of neural driving systems.

A popular way of analysing and validate self-driving models is stratified evaluation. Bansal et al (2019) present a model ablation test for the ChauffeurNet model, and they specifically evaluate the self-driving model against a variety of scenarios. For example, they define a series of simple test cases such as stopping for stop signs or red lights or lane following, as well as more complex situations. Besides, since their model works on structured semantic inputs, they also evaluate ChauffeurNet against modified inputs where objects can be added or removed as explained in Section 4.1.3. Moreover, Hecker et al (2020) argue that augmenting the input space with semantic maps enables the filtering of a subset of driving scenarios (e.g. sessions with a red light), either for the training or the testing, and thus gaining a finer understanding of the potential performance of the self-driving model, a concept they coin as "performance interpretability". With the idea of detecting erroneous behaviors of deep self-driving models that could lead to potential accidents, Tian et al (2018) develop an automatic testing tool. They partition the input space according to the *neuron coverage* concept from (Pei et al, 2019) 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 car 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. In the same vein, Ponn et al (2020) use a SHAP approach (Lundberg and Lee, 2017) to find that the relative rotation of objects and their position with respect to the camera influence the prediction of the model. Their model can be used to create challenging scenarios by deriving corner cases.

Some limits exist in this branch of the literature as manually creating the system's specifications, to automatically evaluate the performance of deep self-driving models, remains costly and essentially amounts to recreate the logic of a real driver.

5 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), presented in Section 3.1.2, 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 Section 4, but cannot be done directly by observing the model itself.

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* (Section 5.1), which aim at communicating which perceptual information is used by the model. Secondly, we study *intermediate-level explanations* (Section 5.2) that force the network to produce supplementary information as it drives. Then we consider *output-level explanations* (Section 5.3), which seeks to unveil high-level objectives of the driving system. Selected references from this section are reported in Table 4.

5.1 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 (Section 5.1.1) and models that use semantic inputs (Section 5.1.2).

5.1.1 Attention-based models

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.

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

Approach	Explanation type	Section	Selected references
Input interpretability	Attention maps	5.1.1	Visual attention (Kim and Canny, 2017) Object centric (Wang et al, 2019) Attentional Bottleneck (Kim and Bansal, 2020)
	Semantic inputs	5.1.2	DESIRE (Lee et al, 2017) ChauffeurNet (Bansal et al, 2019) MTP (Djuric et al, 2020; Cui et al, 2019)
Intermediate representations	Auxiliary branch	5.2.1	Affordances/action primitives (Mehta et al, 2018) Detection/forecast of vehicles (Zeng et al, 2019) Multiple auxiliary losses (Bansal et al, 2019)
	NLP		Natural language (Kim et al, 2018; Mori et al, 2019)
Output interpretability		5.3	Sequences of points (Lee et al, 2017) Sets of points (Cui et al, 2019) Classes (Phan-Minh et al, 2020) Auto-regressive likelihood map (Srikanth et al, 2019; Bansal et al, 2019) Segmentation of future track in bird-eye-view (Caltagirone et al, 2017) Cost-volume (Zeng et al, 2019)

Table 4: Key references to design an explainable driving model.

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 Figure 6, combining visual attention with information bottleneck results in sparser saliency maps, making them more interpretable.

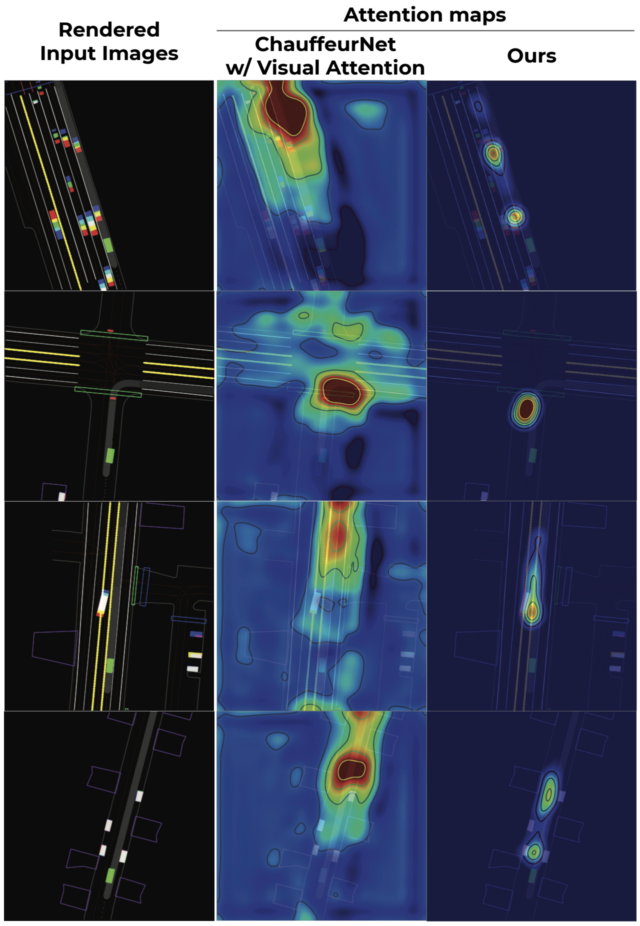


Fig. 6: Comparison of attention maps from classical visual attention and from attention bottleneck. Attention bottleneck seems to provide tighter modes, focused on objects of interest. Credits to (Kim and Bansal, 2020).

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. Besides, 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.

5.1.2 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 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. Second, counterfactual analysis is simplified as the input can be manipulated more easily without the risk of generating meaningless imperceptible perturbations, akin to adversarial attacks.

Using semantic inputs. Besides 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-eye-view corresponds to a 10cm×10cm square of the environment. While this input representation provides information about the presence or absence of an obstacle at a certain location, 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-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 Figure 7.

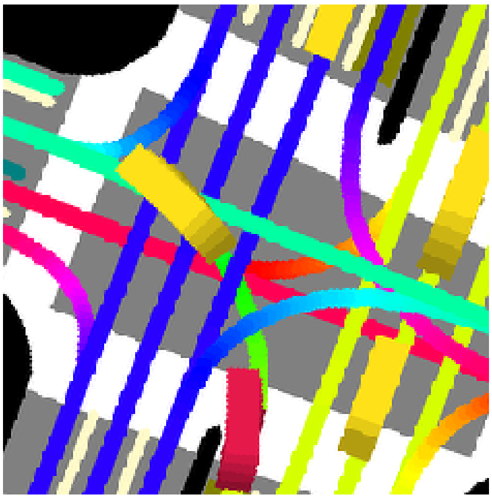


Fig. 7: RGB image of the perceived environment in bird-eye-view, that will be used as an input to the CNN. Credits to (Djuric et al, 2020).

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. First, this can make the input space more interpretable by having dimensions we can play on. Importantly, having such a feature would nicely synergies with many of the post-hoc explainability methods presented in Section 4. For example, to learn counterfactual examples without producing 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 been 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, 2019b; 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). Manipulations could be done by playing on attributes (Lample et al, 2017), by inserting virtual objects in real scenes (Alhaija et al, 2018), 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).

5.2 Intermediate representations

A neural network makes its decisions by automatically constructing intermediate representations of the data. 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. A first class of methods, presented in Section 5.2.1, uses *supervised learning* to specify the content of those representation spaces. 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. In the second class of methods, detailed in Section 5.2.2, this representation space is constrained in an *unsupervised* fashion, where a structure can be enforced so that the features automatically recognize and differentiate high-level latent concepts.

5.2.1 Supervising intermediate representations

As was stated in (Zhou et al, 2019), sensorimotor agents benefit from predicting explicit intermediate scene representations in parallel to their main task. But besides this objective of model accuracy, predicting scene elements may give some insights about the information contained in the intermediate features. In (Mehta et al, 2018), a neural network learns to predict control outputs from input images. Its training is helped with auxiliary tasks that aim at recognizing high-level action primitives (e.g. “stop”, “slow down”, “turn left”, *etc.*) and visual affordances (see Section 3.1.2) 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.

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.

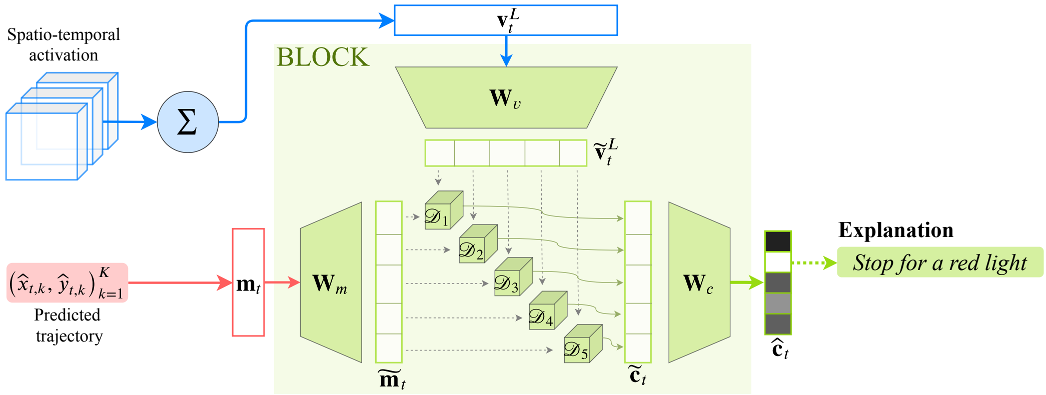


Fig. 8: Explanations for driving decisions are expressed as a fusion between the predicted trajectory and perceptual features. Credits to (Ben-Younes et al, 2020).

The fusion, depicted in Figure 8, 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 Section 6.1).

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.

5.2.2 Unsupervised learning

Over the last years, models have been developed to learn and discover *disentangled* latent variables in an unsupervised fashion. Such representations capture underlying salient data factors and each individual variable represents a single salient attribute: allocating separate dimensions for each attribute thus offers interpretability (Bengio et al, 2013; Chen et al, 2016). For example on a human face dataset, these latent variables include the hairstyle, the face orientation, or the person gender (Pu et al, 2016). These models promise that the learned low-dimensional space provides a rich vocabulary for explanations, which is thus better suited than high-dimensional input spaces. The family of unsupervised models that learn disentangled representations encompass the Variational Auto-Encoder (VAE) (Kingma and Welling, 2014; Higgins et al, 2017) and the Generative Adversarial Networks (GAN) (Goodfellow et al, 2014) (more specifically, the infoGAN variant (Chen et al, 2016)). Yet, in the self-driving literature, we are not aware of any works producing interpretable or disentangled intermediate representations without using external supervision. The dimensions discovered by an unsupervised algorithm may not align with interpretable features such as the one a human driver would use, or the widely accepted visual affordances (see Section 3.1.2). Overall, obtaining disentangled representations in an unsupervised way is not trivial with such high dimensional input data (video streams, LiDAR point-clouds, *etc.*). In the general case, learning disentangled representations is known to be fundamentally impossible without any inductive biases in the models and the data (Locatello et al, 2019), and identifying well-disentangling models requires some supervision.

5.3 Output

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. The first known early attempt to neural control prediction was in (Lecun et al, 2004), where a neural network is trained to predict values of the steering angle actuator. More recently, (Bojarski et al, 2016; Codevilla et al, 2018) revived these approaches by using the progress made by the deep learning community (convolutional networks, training on large datasets, the use of GPUs, *etc.*). However, having a system that directly predicts these command values may not be satisfactory in terms of interpretability, as it may fail to communicate to the end-user local objectives that the vehicle is attempting to attain. Understanding the intermediate near-future goals chosen by the network provides a form of interpretability that command output neural networks do not have.

To this end, other approaches break the command prediction problem into two sub-problems: trajectory planning and control. In these systems, the neural network predicts the future trajectory that the vehicle should take. This predicted trajectory is then passed to a controller that finds the suitable steering, brake and acceleration commands to reach the required position. Often in trajectory planning systems based on machine learning, the controller is considered given and optimal, and the focus is completely cast on learning to predict the correct trajectory. The predicted trajectory can be visualized in the same coordinate system as the input representation, which helps the human user interpret the prediction and infer causal relations between scene elements (road structure, pedestrians, other vehicles, *etc.*) and the decision. 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. For instance, Lee et al (2017) propose DESIRE, a model that learns to predict multiple possible future trajectories for each scene agent. More specifically, recurrent models are trained to sample trajectories as sequences of 2D points in a bird-eye view basis, rank them, and refine them according to perceptual features. In the end, each scene agent is associated to a list of possible future trajectories and their score. In MTP (Cui et al, 2019), multiple future trajectories are predicted for a single agent. Each trajectory consists of a set of 2D points and a confidence score. In practice, a fully-connected layer predicts a vector of size $(2H+1)M$ where H is the temporal horizon and M is the number of modes to predict. CoverNet (Phan-Minh et al, 2020) poses the trajectory prediction problem as a classification one, where each possible class is a predefined trajectory profile. Thus, by taking the k most probable classes according to the model, they can generate multiple trajectory candidates for the near future.

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. One of the main differences with the analytic output family is that virtually any trajectory candidate can be scored according to the model. A downside is that the model does not provide a single clear output trajectory. Finding the *best* prediction requires heuristics such as greedy search or sampling. In INFER (Srikanth et al, 2019), an auto-regressive model is trained to output a likelihood map for the vehicle's next position. At inference time, the most likely position is chosen and a new prediction is computed from there. 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. The loss function used here is a binary cross-entropy, meaning that possible future locations are scored independently from each other. 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 clothoïds) 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.

6 Use case: natural language explanations

As was stated in Section 2.3, 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 Section 6.1, 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 Section 6.2.

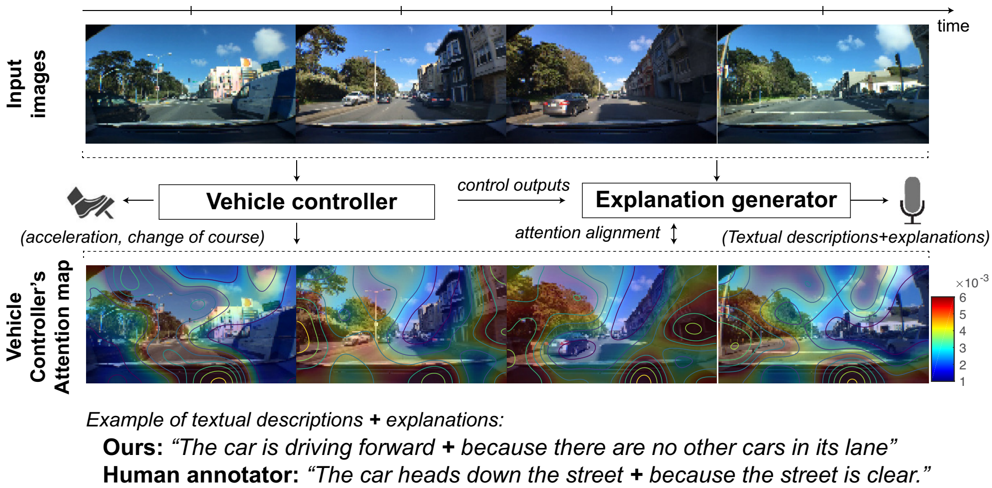


Fig. 9: The vehicle controller predicts scalar values for commands, whereas the explanation generator provides a natural language sentence that describes the scene and explains the driving decision. Credits to (Kim et al, 2018).

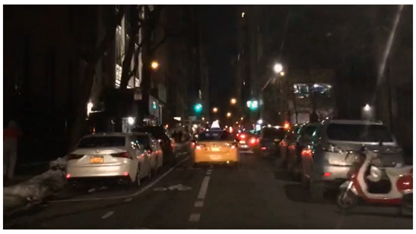
6.1 Generating natural language explanations.

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

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 Figure 9. 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

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

Table 5: **Samples of generated explanations**. GT stands for the the ground-truth (human gold label). Other lines are justifications generated by BEEF, with different runs obtained 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. Credits to (Ben-Younes et al, 2020).

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 5, 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: 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, 2018b; 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).

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.

6.2 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). 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.

7 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 not an easy task. For example, evaluating natural language explanations with a human rating or automated metrics is not satisfying 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. 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 Section 4.1.2. We believe that ways to bridge this gap, detailed in Section 5.1.2, 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 Section 4.1.3, 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.

References

Abbeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: ICML

Abbeel P, Dolgov D, Ng AY, Thrun S (2008) Apprenticeship learning for motion planning with application to parking lot navigation. In: IROS

Abughalieh KM, Alawneh SG (2020) Predicting pedestrian intention to cross the road. IEEE Access

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 IJ, Hardt M, Kim B (2018) Sanity checks for saliency maps. In: NeurIPS

Agrawal A, Lu J, Antol S, Mitchell M, Zitnick CL, Parikh D, Batra D (2017) VQA: visual question answering - www.visualqa.org. IJCV

Agrawal A, Batra D, Parikh D, Kembhavi A (2018) Don't just assume; look and answer: Overcoming priors for visual question answering. In: CVPR

Alhaija HA, Mustikovela SK, Mescheder LM, Geiger A, Rother C (2018) Augmented reality meets computer vision: Efficient data generation for urban driving scenes. IJCV

Alipour K, Schulze JP, 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 TS (2018) Towards robust interpretability with self-explaining neural networks. In: NeurIPS

Amiri AJ, Loo SY, Zhang H (2019) Semi-supervised monocular depth estimation with left-right consistency using deep neural network. In: ROBIO

Anderson JM, Nidhi K, Stanley KD, Sorensen P, Samaras C, Oluwatola OA (2014) Autonomous vehicle technology: A guide for policymakers

Antol S, Agrawal A, Lu J, Mitchell M, Batra D, Zitnick CL, Parikh D (2015) VQA: visual question answering. In: ICCV

Bach S, Binder A, Montavon G, Klauschen F, Müller KR, 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 YM (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 A, Ma S, Ramanan D, Sheikh Y (2018) Recycle-gan: Unsupervised video retargeting. In: ECCV

Bansal M, Krizhevsky A, Ogale AS (2019) Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In: Robotics: Science and Systems

Bansal N, Agarwal C, Nguyen A (2020) SAM: the sensitivity of attribution methods to hyperparameters. In: CVPR

Beaudouin V, Bloch I, Bounie D, Clémençon S, d'Alché-Buc F, Eagan J, Maxwell W, Mozharovskyi P, Parekh J (2020) Flexible and context-specific AI explainability: A multidisciplinary approach. CoRR

Behrendt K, Novak L, Botros R (2017) A deep learning approach to traffic lights: Detection, tracking, and classification. In: ICRA

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 Zablocki, Pérez P, Cord M (2020) Driving behavior explanation with multi-level fusion. Machine Learning for Autonomous Driving Workshop ML4AD@NeurIPS

Bengio Y, Courville AC, Vincent P (2013) Representation learning: A review and new perspectives. TPAMI

Besserve M, Mehrjou A, Sun R, Schölkopf B (2020) Counterfactuals uncover the modular structure of deep generative models. In: ICLR

Bojarski M, Testa DD, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, 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 LD, Muller U (2017) Explaining how a deep neural network trained with end-to-end learning steers a car. CoRR

Bojarski M, Choromanska A, Choromanski K, Firner B, Ackel LJ, Muller U, Yeres P, Zieba K (2018) Visualbackprop: Efficient visualization of cnns for autonomous driving. In: ICRA

Bojarski M, Chen C, Daw J, Degirmenci A, Deri J, Firner B, Flepp B, Gogri S, Hong J, Jackel LD, Jia Z, Lee BJ, Liu B, Liu F, Muller U, Payne S, Prasad NKN, Provodin A, Roach J, Rvachov T, Tadimeti N, van Engelen J, Wen H, Yang E, Yang Z (2020) The NVIDIA pilotnet experiments. 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 RN, Hammers A, Dickie DA, del C Valdés Hernández M, Wardlaw JM, Rueckert D (2018) GAN augmentation: Augmenting training data using generative adversarial networks. CoRR

Bykov K, Höhne MM, 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 AH, Vora S, Liong VE, 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

Casser V, Pirk S, Mahjourian R, Angelova A (2019) Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: AAAI

Castrejón L, Aytar Y, Vondrick C, Pirsiavash 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 JR, Chen YS (2018) Pyramid stereo matching network. In: CVPR

Chang M, Lambert J, Sangkloy P, Singh J, Bak S, Hartnett A, Wang D, Carr P, Lucey S, Ramanan D, Hays J (2019a) Argoverse: 3d tracking and forecasting with rich maps. In: CVPR

Chang Y, Liu ZY, Hsu WH (2019b) Vornet: Spatio-temporally consistent video inpainting for object removal. In: CVPR Workshops

Chen C, Seff A, Kornhauser AL, Xiao J (2015) Deepdriving: Learning affordance for direct perception in autonomous driving. In: ICCV

Chen J, Li SE, Tomizuka M (2020a) Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. CoRR

Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P (2016) Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: NIPS

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

Choi JK, Ji YG (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 AM, 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

Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: CVPR

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 AD (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

Deng Y, Zheng JX, 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 ED (2002) The development of machine vision for road vehicles in the last decade. In: IV

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 MA (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 AK, Yang XJ, Robert Jr LP (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

Espié E, Guionneau C, Wymann B, Dimitrakakis C, Coulom R, Sumner A (2005) Torcs, the open racing car simulator

Fellous JM, Sapiro G, Rossi A, Mayberg HS, Ferrante M (2019) Explainable artificial intelligence for neuroscience: Behavioral neurostimulation. Frontiers in Neuroscience

Feng D, Haase-Schütz C, Rosenbaum L, Hertlein H, Duffhauss F, Gläser C, Wiesbeck W, Dietmayer K (2019) Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. CoRR

Fernandes D, Silva A, Névoa R, Simões C, Gonzalez D, Guevara M, Novais P, Monteiro J, Melo-Pinto P (2021) Point-cloud based 3d object detection and classification methods for self-driving applications: A survey and taxonomy. Information Fusion

Fong RC, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. In: ICCV

Frosst N, Hinton GE (2017) Distilling a neural network into a soft decision tree. In: Workshop on Comprehensibility and Explanation in AI and ML @AI*IA 2017

Fu H, Gong M, Wang C, Batmanghelich K, Tao D (2018) Deep ordinal regression network for monocular depth estimation. In: CVPR

Gao C, Saraf A, Huang J, Kopf J (2020) Flow-edge guided video completion. In: ECCV

Garfinkel S, Matthews J, Shapiro SS, Smith JM (2017) Toward algorithmic transparency and accountability. Commun 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 photorealistic facial expression manipulation. IJCV

Ghorbani A, Abid A, Zou JY (2019) Interpretation of neural networks is fragile. In: AAAI

Gibson JJ (1979) The Ecological Approach to Visual Perception. Psychology Press

Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L (2018) Explaining explanations: An overview of interpretability of machine learning. In: DSSA

Godard C, Mac Aodha O, Brostow GJ (2017) Unsupervised monocular depth estimation with left-right consistency. In: CVPR

Godard C, Aodha OM, Firman M, Brostow GJ (2019) Digging into self-supervised monocular depth estimation. In: ICCV

Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y (2014) Generative adversarial nets. In: NIPS

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 Comput Surv

Guizilini V, Ambrus R, Pillai S, Raventos A, Gaidon A (2020) 3D packing for self-supervised monocular depth estimation. In: CVPR

de Haan P, Jayaraman D, Levine S (2019) Causal confusion in imitation learning. In: NeurIPS

Harradon M, Druce J, Ruttenberg BE (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: Cohn T, He Y, Liu Y (eds) EMNLP (Findings)

Haspiel J, Du N, Meyerson J, Jr LPR, Tilbury DM, Yang XJ, Pradhan AK (2018) Explanations and expectations: Trust building in automated vehicles. In: HRI

Hecker S, Dai D, Liniger A, Gool LV (2020) Learning accurate and human-like driving using semantic maps and attention. CoRR

Hendricks LA, Akata Z, Rohrbach M, Donahue J, Schiele B, Darrell T (2016) Generating visual explanations. In: ECCV

Hendricks LA, 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

Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A (2017) beta-vae: Learning basic visual concepts with a constrained variational framework. In: ICLR

Ho J, Ermon S (2016) Generative adversarial imitation learning. In: NIPS

Houston J, Zuidhof G, Bergamini L, Ye Y, Jain A, Omari S, Iglovikov V, Ondruska P (2020) One thousand and one hours: Self-driving motion prediction dataset. CoRR

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

Huang X, Kwiatkowska M, Wang S, Wu M (2017) Safety verification of deep neural networks. In: CAV

Huang X, Cheng X, Geng Q, Cao B, Zhou D, Wang P, Lin Y, Yang R (2018) The apolloscape dataset for autonomous driving. In: CVPR Workshops

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 A, Koppula HS, Soh S, Raghavan B, Singh A, Saxena A (2016) Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture. CoRR

Jain S, Wallace BC (2019) Attention is not explanation. In: NAACL

Janai J, Güney F, Behl A, Geiger A (2020) Computer vision for autonomous vehicles: Problems, datasets and state of the art. Found Trends Comput Graph Vis

Jaritz M, de Charette R, Wirbel É, Perrotton X, Nashashibi F (2018) Sparse and dense data with CNNs: Depth completion and semantic segmentation. In: 3DV

Johnson J, Hariharan B, van der Maaten L, Fei-Fei L, Zitnick CL, Girshick RB (2017) CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In: CVPR

Katz G, Barrett CW, Dill DL, Julian K, Kochenderfer MJ (2017) Reluplex: An efficient SMT solver for verifying deep neural networks. In: CAV

Kaufmann L (1987) Clustering by means of medoids. In: Proc. Statistical Data Analysis Based on the L1 Norm Conference

Kendall A, Martirosyan H, Dasgupta S, Henry P (2017) End-to-end learning of geometry and context for deep stereo regression. In: ICCV

Kendall A, Hawke J, Janz D, Mazur P, Reda D, Allen J, Lam V, Bewley A, Shah A (2019) Learning to drive in a day. In: ICRA

Kim B, Koyejo O, Khanna R (2016) Examples are not enough, learn to criticize! criticism for interpretability. In: NIPS

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, Bansal M (2020) Attentional bottleneck: Towards an interpretable deep driving network. In: CVPR Workshops

Kim J, Canny JF (2017) Interpretable learning for self-driving cars by visualizing causal attention. In: ICCV

Kim J, Rohrbach A, Darrell T, Canny JF, Akata Z (2018) Textual explanations for self-driving vehicles. In: ECCV

Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: ICLR

Kiran BR, Sobh I, Talpaert V, Mannion P, Sallab AAA, Yogamani SK, 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. IJIDeM

Krähenbühl P (2018) Free supervision from video games. In: CVPR

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: NIPS

Kuderer M, Gulati S, Burgard W (2015) Learning driving styles for autonomous vehicles from demonstration. In: ICRA

Kuznietsov Y, Stückler J, Leibe B (2017) Semi-supervised deep learning for monocular depth map prediction. In: CVPR

Lample G, Zeghidour N, Usunier N, Bordes A, Denoyer L, Ranzato M (2017) Fader networks: Manipulating images by sliding attributes. In: NIPS

Lecun Y, Cosatto E, Ben J, Muller U, Flepp B (2004) Dave: Autonomous off-road vehicle control using end-to-end learning. Snowbird 2004 workshop

LeCun Y, Bengio Y, Hinton GE (2015) Deep learning. Nature

Lee J, Moray N (1992) Trust, control strategies and allocation of function in human-machine systems. Ergonomics

Lee JD, Moray N (1994) Trust, self-confidence, and operators' adaptation to automation. International journal of human-computer studies

Lee JD, See KA (2004) Trust in automation: Designing for appropriate reliance. Human Factors

Lee N, Choi W, Vernaza P, Choy CB, Torr PHS, Chandraker M (2017) DESIRE: distant future prediction in dynamic scenes with interacting agents. In: CVPR

Leonard J, How J, Teller S, Berger M, Campbell S, Fiore G, Fletcher L, Frazzoli E, Huang A, Karaman S, et al (2008) A perception-driven autonomous urban vehicle. Journal of Field Robotics

Li B, Qi X, Lukasiewicz T, Torr PHS (2020a) Manigan: Text-guided image manipulation. In: CVPR

Li B, Qi X, Torr PHS, Lukasiewicz T (2020b) Lightweight generative adversarial networks for text-guided image manipulation. In: NeurIPS

Li C, Chan SH, Chen Y (2020c) Who make drivers stop? towards driver-centric risk assessment: Risk object identification via causal inference. In: IROS

Li P, Qin T, Shen S (2018a) Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In: ECCV

Li Q, Tao Q, Joty SR, Cai J, Luo J (2018b) 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

Li Z, Motoyoshi T, Sasaki K, Ogata T, Sugano S (2018c) Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability. CoRR

Lipton ZC (2018) The mythos of model interpretability. Commun ACM

Liu G, Reda FA, Shih KJ, Wang T, Tao A, Catanzaro B (2018) Image inpainting for irregular holes using partial convolutions. In: ECCV

Liu H, Yin Q, Wang WY (2019) Towards explainable NLP: A generative explanation framework for text classification. In: ACL

Liu Y, Hsieh Y, Chen M, Yang CH, Tegnér J, Tsai YJ (2020) Interpretable self-attention temporal reasoning for driving behavior understanding. In: ICASSP

Locatello F, Bauer S, Lucic M, Rätsch G, Gelly S, Schölkopf B, Bachem O (2019) Challenging common assumptions in the unsupervised learning of disentangled representations. In: ICML

Lu J, Yang J, Batra D, Parikh D (2016) Hierarchical question-image co-attention for visual question answering. In: NIPS

Lundberg SM, Lee S (2017) A unified approach to interpreting model predictions. In: NIPS

Ly AO, Akhloufi MA (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

Mahendran A, Vedaldi A (2016) Salient deconvolutional networks. In: ECCV

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 DK, Heacock L, Moy L, Cho K, Geras KJ (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 UG, Chiroma H, Aljojo N, Abubakar S, Popoola SI, Al-Garadi MA (2020) A survey on deep learning for steering angle prediction in autonomous vehicles. IEEE Access

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

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

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

Ng AY, Russell SJ (2000) Algorithms for inverse reinforcement learning. In: ICML

Nguyen AM, Dosovitskiy A, Yosinski J, Brox T, Clune J (2016) Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: NIPS

Papineni K, Roukos S, Ward T, Zhu W (2002) Bleu: a method for automatic evaluation of machine translation. In: ACL

Park DH, Hendricks LA, Akata Z, Rohrbach A, Schiele B, Darrell T, Rohrbach M (2018) Multimodal explanations: Justifying decisions and pointing to the evidence. In: CVPR

Park J, Joo K, Hu Z, Liu CK, Kweon IS (2020) Non-local spatial propagation network for depth completion. In: ECCV

Patro BN, 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. Commun ACM

Phan-Minh T, Grigore EC, Boulton FA, Beijbom O, Wolff EM (2020) Covernet: Multimodal behavior prediction using trajectory sets. In: CVPR

Pomerleau D (1988) ALVINN: an autonomous land vehicle in a neural network. In: NIPS

Ponn T, Kröger T, Diermeyer F (2020) Identification and explanation of challenging conditions for camera-based object detection of automated vehicles. Sensors

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 NF, 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

Rasouli A, Kotseruba I, Kunic T, Tsotsos JK (2019) PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In: ICCV

Rathi S (2019) Generating counterfactual and contrastive explanations using SHAP. Workshop on Humanizing AI (HAI) @IJCAI

Razavian AS, Azizpour H, Sullivan J, Carlsson S (2014) CNN features off-the-shelf: An astounding baseline for recognition. In: CVPR Workshops

Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: CVPR

Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. CoRR

Redmon J, Divvala SK, Girshick RB, Farhadi A (2016) You only look once: Unified, real-time object detection. In: CVPR

Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS

Ribeiro MT, Singh S, Guestrin C (2016) "why should I trust you?": Explaining the predictions of any classifier. In: SIGKDD

Ribeiro MT, Singh S, Guestrin C (2018) Anchors: High-precision model-agnostic explanations. In: AAAI

Riquelme F, Goyeneche AD, Zhang Y, Niebles JC, Soto A (2020) Explaining VQA predictions using visual grounding and a knowledge base. Image Vis Comput

Romera E, Bergasa LM, Yang K, Alvarez JM, Barea R (2019) Bridging the day and night domain gap for semantic segmentation. In: IV

Ros G, Sellart L, Materzynska J, Vázquez D, López AM (2016) The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR

Rosenfeld A, Richardson A (2019) Explainability in human-agent systems. Auton Agents Multi Agent Syst

Ross S, Gordon GJ, Bagnell D (2011) A reduction of imitation learning and structured prediction to no-regret online learning. In: AISTATS

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 RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2020) Grad-cam: Visual explanations from deep networks via gradient-based localization. Int J Comput Vis

Shariff A, Bonnefon JF, Rahwan I (2017) Psychological roadblocks to the adoption of self-driving vehicles. Nature Human Behaviour

Sharifzadeh S, Chiotellis I, Triebel R, Cremers D (2016) Learning to drive using inverse reinforcement learning and deep q-networks. CoRR

Shen Y, Jiang S, Chen Y, Yang E, Jin X, Fan Y, Campbell KD (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

Srikanth S, Ansari JA, R KR, Sharma S, Murthy JK, Krishna KM (2019) INFER: intermediate representations for future prediction. In: IROS

Sun P, Kretzschmar H, Dotiwalla X, Chouard A, Patnaik V, Tsui P, Guo J, Zhou Y, Chai Y, Caine B, Vasudevan V, Han W, Ngiam J, Zhao H, Timofeev A, Ettinger S, Krivokon M, Gao A, Joshi A, Zhang Y, Shlens J, Chen Z, Anguelov D (2020) Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR

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

Syed U, Schapire RE (2007) A game-theoretic approach to apprenticeship learning. In: NIPS

Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow IJ, Fergus R (2014) Intriguing properties of neural networks. In: ICLR

Tang J, Tian F, Feng W, Li J, Tan P (2019) Learning guided convolutional network for depth completion. CoRR

Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, Fong P, Gale J, Halpenny M, Hoffmann G, et al (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

Tian Y, Carballo A, Li R, Takeda K (2020) Road scene graph: A semantic graph-based scene representation dataset for intelligent vehicles. CoRR

Tjoa E, Guan C (2019) A survey on explainable artificial intelligence (XAI): towards medical XAI. CoRR

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

Torralba A, Efros AA (2011) Unbiased look at dataset bias. In: CVPR

Tran D, Bourdev LD, 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

Ullman S (1980) Against direct perception. Basic books

Urmson C, Anhalt J, Bagnell D, Baker C, Bittner R, Clark M, Dolan J, Duggins D, Galatali T, Geyer C, et al (2008) Autonomous driving in urban environments: Boss and the urban challenge. Journal of Field Robotics

Vedantam R, Zitnick CL, 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 BD, 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

Xia Y, Zhang D, Kim J, Nakayama K, Zipser K, Whitney D (2018) Predicting driver attention in critical situations. In: ACCV

Xiao Y, Codevilla F, Pal C, López AM (2020) Action-based representation learning for autonomous driving. CoRL

Xie M, Trassoudaine L, Alizon J, Thonnat M, Gallice J (1993) Active and intelligent sensing of road obstacles: Application to the european eureka-prometheus project. In: ICCV

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 H, Gao Y, Yu F, Darrell T (2017) End-to-end learning of driving models from large-scale video datasets. In: CVPR

Xu K, Ba J, Kiros R, Cho K, Courville AC, Salakhutdinov R, Zemel RS, Bengio Y (2015) Show, attend and tell: Neural image caption generation with visual attention. In: ICML

Xu Y, Zhu X, Shi J, Zhang G, Bao H, Li H (2019) Depth completion from sparse LiDAR data with depth-normal constraints. In: ICCV

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 AJ (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

Yao Y, Wang X, Xu M, Pu Z, Atkins EM, Crandall DJ (2020) When, where, and what? A new dataset for anomaly detection in driving videos. CoRR

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

Yurtsever E, Lambert J, Carballo A, Takeda K (2020) A survey of autonomous driving: Common practices and emerging technologies. IEEE Access

Zahavy T, Ben-Zrihem N, Mannor S (2016) Graying the black box: Understanding dqns. In: ICML

Zeiler MD, 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 JC, 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 H, Geiger A, Urtasun R (2013) Understanding high-level semantics by modeling traffic patterns. In: ICCV

Zhang Q, Zhu S (2018) Visual interpretability for deep learning: a survey. Frontiers Inf Technol Electron Eng

Zhang Q, Cao R, Shi F, Wu YN, Zhu S (2018) Interpreting CNN knowledge via an explanatory graph. In: AAAI

Zhang Q, Yang XJ, Robert LP (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? Sci Robotics

Zhou T, Brown M, Snavely N, Lowe DG (2017) Unsupervised learning of depth and ego-motion from video. In: CVPR

Ziebart BD, Maas AL, Bagnell JA, Dey AK (2008) Maximum entropy inverse reinforcement learning. In: AAAI

Zilke JR, Mencía EL, Janssen F (2016) Deepred - rule extraction from deep neural networks. In: DS