Knowledge Augmented Machine Learning with Applications in Autonomous Driving: A Survey

Julian Wörmann[7], Daniel Bogdoll[9], Etienne Bührle[9], Han Chen[2], Evaristus Fuh Chuo[2], Kostadin Cvejoski[8], Ludger van Elst[4], Tobias Gleißner[8], Philip Gottschall[8], Stefan Griesche[10], Christian Hellert[3], Christian Hesels[8], Sebastian Houben[8], Tim Joseph[9], Niklas Keil[1], Johann Kelsch[5], Hendrik Königshof[9], Erwin Kraft[3], Leonie Kreuser[1], Kevin Krone[8], Tobias Latka[6], Denny Mattern[8], Stefan Matthes[7], Mohsin Munir[4], Moritz Nekolla[9], Adrian Paschke[8], Maximilian Alexander Pintz[8], Tianming Qiu[7], Faraz Qureishi[11], Syed Tahseen Raza Rizvi[4], Jörg Reichardt[3], Laura von Rueden[8], Stefan Rudolph[6], Alexander Sagel[7], Gerhard Schunk[11], Hao Shen[7], Hendrik Stapelbroek[2], Vera Stehr[11], Gurucharan Srinivas[5], Anh Tuan Tran[10], Abhishek Vivekanandan[9], Ya Wang[8], Florian Wasserrab[1], Tino Werner[5], Christian Wirth[3], and Stefan Zwicklbauer[3]

[1]Alexander Thamm GmbH
[2]Capgemini Engineering
[3]Continental AG
[4]Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI)
[5]Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)
[6]Elektronische Fahrwerksysteme GmbH
[7]fortiss GmbH
[8]Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V. (FOKUS & IAIS)
[9]FZI Forschungszentrum Informatik
[10]Robert Bosch GmbH
[11]Valeo Schalter und Sensoren GmbH

Abstract: The existence of representative datasets is a prerequisite of many successful artificial intelligence and machine learning models. However, the subsequent application of these models often involves scenarios that are inadequately represented in the data used for training. The reasons for this are manifold and range from time and cost constraints to ethical considerations. As a consequence, the reliable use of these models, especially in safety-critical applications, is a huge challenge. Leveraging additional, already existing sources of knowledge is key to overcome the limitations of purely data-driven approaches, and eventually to increase the generalization capability of these models. Furthermore, predictions that conform with knowledge are crucial for making trustworthy and safe decisions even in underrepresented scenarios. This work provides an overview of existing techniques and methods in the literature that combine data-based models with existing knowledge. The identified approaches are structured according to the categories integration, extraction and conformity. Special attention is given to applications in the field of autonomous driving.

──────────── ◆ ────────────

CONTENTS

1   Introduction   3

2   Overview use case domains   3
    2.1   Perception   3
    2.2   Situation Interpretation   5
    2.3   Planning   6

3   Knowledge Representations   7
    3.1   Symbolic Representations and Knowledge Crafting   7
    3.2   Knowledge Representation Learning   9

4   Knowledge Integration   11
    4.1   Auxiliary Losses and Constraints   11
    4.2   Neural-symbolic Integration   15
    4.3   Attention Mechanism   17
    4.4   Data Augmentation   19
    4.5   State Space Models   21
    4.6   Reinforcement Learning   25
    4.7   Deep-Learning with Prior Knowledge Maps   28

5   Knowledge Transfer   29
    5.1   Transfer Learning   29
    5.2   Continual Learning   31
    5.3   Meta Learning   33
    5.4   Active Learning   35

6   Knowledge Extraction - Symbolic Explanations   39
    6.1   Rule Extraction and Rule Learning   40
    6.2   Structured Output Prediction   44
    6.3   Natural Language Processing for Legal Domain   45
    6.4   Question Answering   46

7   Knowledge Extraction - Visual Explanations   46
    7.1   Visual Analytics   47
    7.2   Saliency Maps   48
    7.3   Interpretable Feature Learning   50

8   Knowledge Conformity   51
    8.1   Uncertainty Estimation   52
    8.2   Causal Reasoning   56
    8.3   Rule Conformity   59

List of Abbreviations   61

References   62

1 INTRODUCTION

Data-driven learning, first and foremost deep learning, has become a key paradigm in the vast majority of current Artificial Intelligence (AI) and Machine Learning (ML) applications. The excellent performance of many models learned in a supervised manner can be predominantly attributed to the availability of huge amounts of labeled data. Prominent examples are image classification and object detection, sequential data processing as well as decision making. On the downside, the unprecedented performance comes at the cost of lacking interpretability and transparency leading to so called black box models that do not allow for easy and straightforward human inspection.

Transferring data-driven approaches to safety critical applications becomes thus a major challenge. Usually, in these scenarios labeled data is scarce due to high acquisition costs or, not least, for ethical reasons. Furthermore, both developers and users postulate the requirement to be able to understand the decisions made by the deployed model. In order to tackle both problems, the exploitation of *knowledge* sources in form of, e.g., basic laws of physics, logical databases of facts, common behaviour in certain scenarios, or simply counterexamples is key to evolve purely data-driven models towards robustness against perturbations, better generalization to unseen samples, and conformity to existing principles of safe and reliable behaviour.

This survey provides a collection of different methods that are suitable in order to augment data driven models with knowledge, to extract informative concepts and patterns out of given models and to compare observed outputs and representations to existing basic assumptions and common understanding about safe, reliable and intuitive behaviour. Eventually, this overview on the integration of knowledge and data will pave the way to trustworthy ML approaches that can be used safely in critical applications.

This review of the state of the art is structured as follows. In the following Chapter 2, we introduce three major tasks that autonomous agents encounter during interaction with their environment. Starting from different perspective to represent knowledge and to make it machine readable discussed in Chapter 3, subsequently, different general methods eligible to combine knowledge with data-driven approaches, as well as more specific methods tailored to the autonomous driving use case, are reviewed in Chapter 4. Furthermore Chapter 5 introduces learning paradigms in the context of knowledge transfer.

Besides integration of knowledge, current approaches focusing on the extraction of concepts and structures are outlined in the subsequent chapters. While Chapter 6 summarizes methods that provide symbolic, partly natural language explanations, Chapter 7 puts emphasize on procedures that allow for visual inspection of the decision process. We conclude our survey in Chapter 8 with an overview of techniques that consider conformity to already existing as well as newly discovered knowledge components, which eventually completes the pipeline of knowledge empowered artificial intelligence.

2 OVERVIEW USE CASE DOMAINS

The task of automated driving may be sub categorized into the following categories: perception, situation interpretation, planning and control [306]. The foremost task in the autonomous driving is to understand and perceive the environment around the vehicle. Section 2.1 provides an introduction to the *perception* module with a special focus on image-based pedestrian detection. Once the objects are detected and segmented, the second task in the autonomous driving is to understand the environment along with the road users. In order to perform safe maneuvers, the *situation interpretation* is a decisive step. In this module, the goal is to answer important questions related to object's states and actions, like what an object could do next. An overview is given in section Section 2.2. After figuring out these situational scenarios, next task in autonomous driving is to plan the motion of ego vehicle. The *planning* module described in Section 2.3 utilizes the output of the previous two modules and takes high level routing and trajectory planning decisions.

2.1 Perception

Authors: Syed Tahseen Raza Rizvi, Mohsin Munir, Ludger van Elst

2.1.1 Perception in the AD Stack

Perception plays a crucial role in attaining the goal of autonomous driving. An ego-vehicle is generally equipped with a variety of sensors including cameras, lidar and radar. These sensors serve as the senses of an ego-vehicle and therefore enable the capability of perceiving the environment around the ego-vehicle in different spectrums. Object detection, and in particular pedestrian detection, has significant importance in the perception spectra as it serves as a critical piece of information for the downstream tasks associated with the autonomous driving pipeline.

2.1.2 Task Formulation

Autonomous driving systems highly rely on object detection models to identify all the traffic participants. Pedestrians are usually the most common and abundantly found traffic participant. Therefore, the detection of a pedestrian is more prominent and crucial for the perception of an autonomous driving system.

Pedestrian detection deals with the identification of pedestrians in the environment around an ego-vehicle. There exist approaches in the literature which perform pedestrian detection only using lidar sensors [438]. However, such approaches are usually not popular in the community due to fact that the features obtained from camera images are significantly richer as compared to the ones obtained from lidar or radar. On the other hand, [223] uses lidar to incorporate depth information into the image data for the pedestrian detection task. Therefore, the approaches to perform pedestrian detection mainly using camera images are generally widely adopted. The images from the mounted cameras serve as an input from which individual pedestrians are identified and are enclosed in a bounding box. A variety of solutions have been proposed to effectively identify individual pedestrians in the surrounding environment.

The neural network based object detection solutions can be divided into two main categories: One-stage and Two-stage approaches. One-stage approaches are generally based on a fully convolutional architecture and consider the object detection problem as a simple regression problem [693]. For a given input image, the One-stage detectors learn class probabilities and the coordinates of a bounding box encompassing an object. On the other hand, Two-stage approaches are more sophisticated where each stage specializes in a sub-task which eventually contributes to the final output of the system. The first stage is responsible for identifying the region of interest and the second stage is responsible for the object classification and bounding box regression. Both types of approaches have certain pros and cons. Most notably, Two-stage approaches yield better detection accuracy than One-stage approaches as they have specialized stages where the output of the second stage is built on top of the output of the first stage. However, One-stage approaches are much faster than Two-stage approaches as they do not have an additional stage with supplementary computational overhead.

Single Shot MultiBox Detector (SSD) [447], You Only Look Once (YOLO) [589, 591, 590], RetinaNet [437] and Fully Convolutional One-Stage object detector [720] are the most prominent One-stage object detectors. Generally, these approaches divide the image into a grid followed by predicting the probability of a class object in each grid box along with its bounding box coordinates. However, some of these approaches are slightly different as they employ a unique focal loss or pixel-wise classification to achieve a higher detection accuracy in real-time. On the other hand, Fast Regions with CNN (R-CNN) [249], Faster R-CNN [596], Mask R-CNN [297], MimicDet [459] are the most common examples of a Two-stage object detector. Generally the first stage in these Two-stage object detection model consists of a Region Proposal Network (RPN), where in the second stage the candidate region proposals are classified based on the feature maps. Approaches like Mask R-CNN have a mask branch which is a small Fully Convolutional Network (FCN) [144] applied to each Region of Interest (ROI), predicting a pixel-wise segmentation mask. Additionally, Feature Pyramid Network (FPN) [436] is generally used in combination with RPN and Faster R-CNN to make bounding box proposal more robust especially for small objects.

Pedestrian detection is applied in various vision-based applications ranging from surveillance to autonomous driving. Despite their good performance, it is still unknown how the detection performs on unseen data. Hasan et al. [288] presented a study in quest of generalization capabilities of pedestrian detectors. In their cross-dataset evaluation, they have tested several backbones with their baseline detector (Cascade R-CNN [88] on famous autonomous driving data sets including Caltech [175], CityPersons [854], ECP [77], CrowdHuman [666], and Wider Pedestrian [124]). Cross-dataset evaluation is an effective way of evaluating a method on unseen data and checking its generalization capability, otherwise, a method may overfit on a single dataset. The analysis presented in the paper is very interesting. The authors have demonstrated that the existing pedestrian detection methods perform poorly when compared with general object detection methods given larger and diverse

datasets. A carefully trained state-of-the-art general-purpose object detector can outperform pedestrian-specific detection methods. The trick lies in the training pipeline and the dataset. In this study, the authors used large datasets that contain more persons per image. These general purpose datasets, generally collected by crawling the web and through surveillance cameras, are likely to have more human poses, appearances, and occlusion cases as compared to pedestrian-specific datasets. It is also shown in this study that by progressively fine-tuning the models from largest (general purpose) to smallest (close to target domain), performance can be improved. The generalization ability of pedestrian detectors has been compromised due to the lack of diversity and density of the pedestrian benchmarks. However, benchmarks such as WiderPerson [856], Wider Pedestrian [124], and CrowdHuman [666] provide much higher diversity and density.

Pedestrian detection has improved a lot in recent years, however, it is still challenging to detect occluded pedestrians. The pedestrian appearance varies in different scenarios and depends on a wide range of occlusion patterns. To address this issue, Zhang et al. [855] proposed an architecture for pedestrian detection based on the Faster R-CNN. In contrast to ensemble models for most frequent occlusion patterns, the authors leverage different attention mechanisms to guide the detector in paying more attention to the visible body parts. The authors proposed to employ channel-wise attention in a convolution network that allows the network to learn more representative features for different occluded body parts in one model. The observation that many Convolutional Neural Network (CNN) channels in a pedestrian CNN are localizable, strongly motivates them to perform re-weighting of channel features to guide the detector to pay more attention to the visible body parts. In order to generate the attention vector, different realizations of attention networks are examined. The attention vector is trained end-to-end for all of the attention networks either through self-attention or guided by some additional external information like convolution features, visible bounding boxes, or part detection heatmaps. Eventually, the features are passed to the classification network for category prediction and bounding box regression. The experimental results are shown on the CityPersons [854], Caltech [175], and ETH [192] datasets. The results show improvements over the baseline Faster R-CNN detector.

Vulnerable road user detection is another major challenge in pedestrian detection. The safety of road users is and should be the utmost priority in the domain of autonomous driving. In addition to detect occluded pedestrians, another key challenge is to detect pedestrians at long range. When a pedestrian is detected at long range, it increases the security of the pedestrian and driver at the same time, also, it leads to a comfortable driving experience. Fürst et al. [223] introduced an approach that targets long range 3D pedestrians detection. Their approach leverages the density of Red Green Blue (RGB) images and precision of lidar. The symmetrical fusion of RGB and lidar helps them outperform current state-of-the-art for long range 3D pedestrian detection.

2.1.3   Goals and Requirements

Perception plays a pivotal role in autonomous driving. It enables the ego vehicle to analyze and understand the traffic scene and surrounding circumstances. Detection of traffic participants, i.e., pedestrians, vehicles, cyclists, etc serves as the core of perception involved in autonomous driving. Additionally, traffic circumstances like road, weather, and light conditions are also important factors in a traffic scenario. For instance, rainy weather results in a wet road which consequently has a direct impact on the decisions like breaking distance. This is due to that braking distance is significantly more in wet weather than that of the normal dry road. Therefore, traffic participants and their surrounding circumstances collectively provide a basis for planning and executing decisions taken by an ego vehicle. The significance of the perception can be understood by the fact that it directly contributes towards use cases like collision avoidance, trajectory planning, etc.

With the rise of deep learning for solving a universe of different tasks, object detection has also benefited from deep learning One- and Two-stage models to achieve higher detection performance. The effectiveness of an object detection approaches heavily relies on the efficacy of the trained object detection model. In other words, it can be said as, provided an effective object detection model, the quality of perception can be ensured. In order to train an effective object detection model, it requires a large amount of high-quality data. For this purpose, several real-life public datasets are available, i.e., Caltech [175], CityPersons [854], ECP [77], etc. However, certain scenarios are possibly scarce or outright not feasible in such pedestrian detection datasets. For example, it is infeasible to find a dataset that contains a traffic scenario where the ego vehicle is about to collide with another traffic participant. Such a scenario can be helpful to evaluate the performance of an object detection model to detect and evade collision in such a hazardous environment. For this purpose, datasets with simulated custom scenarios can be generated to fill this gap in real-life datasets. Ultimately, a combination of real and simulated data is the key thus enabling the object detection model to effectively perform under several unseen or rarely occurring traffic scenarios.

2.1.4   Necessity of Knowledge Integration

Computer vision methods and in general ML methods have significantly improved over the last years. Different methods are able to accurately interpret a situation presented in an image or video. Even with such advancements, there are scenarios where ML methods react differently as humans. The main reason of this gap is the absence of the background knowledge from the learned model. The ML methods only account for patterns present in the training data, whereas humans have implicit knowledge that could help them to interpret a critical situation more robustly. In the context of autonomous driving, and in general too, it is not possible to train a model for every possible scenario that could happen on road. To provide a safer environment for pedestrians and autonomous vehicles, it is important to incorporate knowledge in the module that is responsible for taking important decisions.

2.2   Situation Interpretation

Authors: Daniel Bogdoll, Abhishek Vivekanandan, Faraz Qureishi, Gerhard Schunk

2.2.1   Situation Interpretation in the AD Stack

Situation interpretation is typically a follow-up module of the perception stage as shown in Section 2.1. Accordingly, this module is aware of objects, their states, and classifications within the surrounding environment. Its main objective is to interpret the situation, which includes questions such as "What is an object doing next?", "Is there an implicit meaning of an object's action?" or "Is a rule exception applicable right now?".

2.2.2   Task Formulation

Automated driving relies on accurate perception of the environment. We follow the concept of Gerwien et al. [244], who describe *situation interpretation* as a module which provides a "situation-aware environment model", that expands an environment model, which is typically the results of the perception stage, by *situation recognition* and *situation prediction*. They classify these three modules as Situation Awareness (SA) levels 1-3. The output of the perception layer can be represented in various forms, for instance with object lists or probabilistic maps. Independent of the structure, the output is critical for the functioning of subsequent Autonomous Driving (AD) layers, which are tasked with situation interpretation, path planning – as shown in Section 2.3 – and vehicle control.

Nevertheless, sometimes raw data in addition to the outputs of the perception layer is relevant to detect intentions or meanings which are typically not addressed by perception systems. Two examples are direction of view [287] and hand gestures [781].

Situation interpretation works in tandem with perception, planning and control. A typical example of situation interpretation may involve cut in scenarios during automated driving using adaptive cruise control [559]. In a cut in scenario, the situation interpretation system shall be able to detect if a collision is imminent (using perception and planning output) and employ mitigation measures (braking in this case) in due time, ensuring the safety of the ego vehicle and its occupants.

In the aforementioned example, the collision detection and avoidance can be designed by using vehicle motion models and traffic rules. In complex situations, however, the task of situation interpretation may not be accomplished by only using a predefined set of rules. Especially for urban scenarios where the number of interactions between the ego vehicle and the objects in the scene are significantly higher. Additionally, there might be situations where a particular rule needs to be violated in order to ensure safety of human life.

2.2.3   Goals and Requirements

To be consistent with the previously defined SA levels, level 2 takes in raw data and adds semantic meaning to it in the form of semantic data models. Many works, especially [244], have defined the operational context in regard to adding more semantic structure to identify situations of interest.

As with SA level 3 defined by [244], motion prediction forms the abstract layer for situation understanding, which

comprises different actors in the ego space. It plays a crucial role in determining safety critical applications for the autonomous driving stack by providing the service of estimating the future positions of an object. For instance, when driving in a highway scenario, assuming that a lead vehicle suddenly merges or cuts-in to the ego lane; the primary goal of this layer is to mitigate the collision by anticipating the intention of the lead vehicle(s). The crash avoidance maneuver should have safety properties such that the maneuver itself should not cause an additional collision, e.g., while hard braking could prevent the crash it could lead to a rear ended collision with other vehicles. This requires not only a prediction module but also a system that checks for the validity of the planned decision based on dynamic safety reasoning methodologies which could influence the Time-To-Collision (TTC), such as including weather constraints.

Most of the existing behavior prediction approaches perform simultaneous tracking and forecasting with the use of Kalman Filters or in the form of rule based approaches, as can be seen from the previous works [418]. Although variants of Kalman filters are good for short term predictions, their performance degrades for long term motion problems as they fail to make use of the situation or environmental knowledge [132] which could be obtained via vectored maps. As a result, prediction modules should make use of domain knowledge to forecast reliable predictions [76].

In a typical AD stack, motion prediction is a separate module which does prediction based on the outputs from the previous perception layer. For example, the object detection outputs bounding box coordinates of an object along with the probability score of a class it belongs to such as truck, car, or construction cone. When this is used as an input to the motion prediction, a failure to propagate uncertainty happens due to the softmax outputs [224]. To alleviate those shortcomings, end-to-end networks, which take raw inputs such as lidar point clouds and camera fusion to produce motion predictions directly [797, 174] should be considered. Additionally, knowledge about one's own path planning can be integrated into the prediction component [29].

2.2.4 Necessity of Knowledge Integration

Vehicles equipped with a level 4 or 5 driving automation system are expected to master a wide variety of situations within their Operational Design Domain (ODD) [628]. Since many situations do not occur frequently in real life, ML based systems are struggling to extrapolate from their trained domain. Therefore, hybrid approaches that integrate rule- and knowledge based algorithms and insights into ML systems have the potential to combine the best of two worlds – great general performance and improved handling of rare situations, such as corner cases.

2.3 Planning

Authors: Etienne Bührle, Hendrik Königshof, Abhishek Vivekanandan, Moritz Nekolla

2.3.1 Motion Planning in the AD Stack

The planning module uses the outputs of the perception and prediction modules to plan a trajectory for the vehicle, which is subsequently handed down to the vehicle controls to be executed. This plan considers high-level routing decisions, and follows the rules of the road as well as basic principles of safe and comfortable driving.

A wide range of methods has been developed to tackle the trajectory tracking control problem, and we refer to [535] for an overview. However, the motion planning problem, especially in highly complex and dynamic environments like road traffic, remains largely unsolved and constitutes an area of ongoing research.

2.3.2 Task Formulation

Formally, the solution to the trajectory planning problem is a function that assigns every point in time a position in configuration space (typically, planar coordinates and heading). Classical approaches include variational methods (which represent the path as a function of continuously adjustable parameters), graph-search methods (which discretize the configuration space), and incremental search methods (which improve upon graph-search methods by using iterative refinement procedures). An excellent overview is given in [535].

The mentioned approaches are usually modular and interpretable. However, as hand-engineered solutions to difficult problems, they tend to be brittle and require extensive manual fine-tuning. Additionally, isolated changes to parts of the system might reduce or break the overall system performance, requiring careful re-tuning [840].

These drawbacks motivate the use of deep learning based approaches, which have proven more robust to variations and can be trained in an end-to-end fashion. The current applications of deep learning to autonomous driving can roughly be classified into two groups. Full end-to-end approaches that map raw sensory input directly to vehicle commands (steering, acceleration), and methods that produce or work on intermediate representations. An overview can be found in [709].

2.3.3 Goals and Requirements

The motion planning system is in charge of ensuring behavioral safety of the self-driving vehicle [515, 516]. This includes taking the correct behavior and driving decisions, based on the knowledge of traffic rules and the behavior of other traffic participants, as well as the ability to safely navigate expected and unexpected scenarios.

The U.S. Department of Transportation (DOT) has recommended that Level 3, Level 4, and Level 5 self-driving vehicles should be able to demonstrate at least 28 core competencies adapted from research by California Partners for Advanced Transportation Technology (PATH) at the Institute of Transportation Studies at University of California, Berkeley. These basic behavioral competencies include, amongst others, keeping the vehicle in lane, obeying traffic laws, following road etiquette, responding to other vehicles, and responding to hazards [516].

While the majority of these behavioral competencies cover normal driving, i.e., regularly encountered situations, a self-driving vehicle is also responsible for Object and Event Detection and Response (OEDR), which includes detecting unusual circumstances (emergency vehicles, work zones, ...) as well as planning an appropriate reaction, which typically takes place in the behavior and planning components. Above

all, the planning system is responsible for crash avoidance, and should be able to handle control loss, crossing-path crashes, lane changes/merges, head-on/opposite-direction travel, rear-end, road departure, and low speed situations (backing, parking). At any time, the system should be able to execute a fallback action that brings the vehicle to a minimal risk condition. According to [515], "a minimal risk condition will vary according to the type and extent of a given failure, but may include automatically bringing the vehicle to a safe stop, preferably outside of an active lane of traffic."

Finally, the motion planner not only interacts with other traffic participants, but also to a great extent with its passengers. In particular, it must be able to communicate proper function, malfunction, as well as an eventual takeover request to a human driver, who must be able to take over in time.

2.3.4 Necessity of Knowledge Integration

Level 5 self-driving vehicles are expected to function in a wide variety of operational design domains (we refer to [81] for a taxonomy). While the basic principles of safe and comfortable driving remain unchanged, the concrete implementations at the level of traffic laws, customary behavior, and scene structure might be subject to change. We argue that the inclusion of knowledge into a motion planning system will make it easier to handle these situations by increasing traceability (e.g., in the case of crash reconstructions) and reliability. Furthermore, a transparent decision process based on a common understanding between humans and machines will increase interpretability and trust. Finally, we expect the emergence of alternatives to extensive simulation testing, which is at the core of present validation concepts [25, 450, 465].

Emphasizing the advantages of Knowledge Integration, [116] demonstrates many of the aspects mentioned above. Fan Chen et al. integrate rules, in the form of social norms, by extending the agents reward function, e.g., passing objects with a minimum distance. Violating these rules results in a reward penalty. According to their results, agents with such restrictions exhibit behavior more similar to a human level. Therefore, when integrating knowledge into the machine learning pipeline, models become more interpretable and confidential not solely for experts but for ordinary people since these constraints occur in everyday life. Furthermore, their extension of the agent´s knowledge reduces learning effort which accelerates training and enables them to out-perform their benchmark algorithm in most cases. Despite those promising benefits, integrating knowledge typically narrows down the broad variety of possible solutions while consuming human work force for hand engineering. This shrinks the original, holistic approach of machine learning. Therefore, the trade-off between knowledge integration and self-learning needs to be chosen carefully [116].

3 KNOWLEDGE REPRESENTATIONS

The symbolic and the sub-symbolic methods represent two ends of the AI spectrum. The former is more driven by the knowledge and the latter by the data. A plethora of ongoing research can be found in the literature to develop hybrid-AI systems which exploit the strengths of one another.

However, there still exists a core challenge in representation of knowledge used in symbolic space to integrate or augment within the data-driven sub-symbolic/statistical world. An overview of formalism and languages for representing symbolic knowledge which exists in the form of facts, rules and structured information is reviewed in Section 3.1. Furthermore, in Section 3.2 a survey on knowledge embedding is presented, which focuses on transforming prior knowledge from the symbolic space to a real-vector space, i.e., embeddings. These embeddings can be leveraged to improve the sub-symbolic methods (Neural Network (NN), Deep Learning (DL)) for effective training, inference and improved reasoning. In addition to it, methods and approaches dealing with injection of hard and soft rules together with embeddings are discussed in Section 3.2.3. Each of the sections in this chapter dealing with different mechanisms in representing knowledge is concluded with an outlook that is more tailored to the field of autonomous driving. Mapping perceived information to semantic concepts and reasoning using symbolic models provides improved understanding of driving situation. Furthermore, formalized traffic rules and legal concepts are used to derive possible driving actions conditioned on their legal consequences Section 3.1.3.

3.1 Symbolic Representations and Knowledge Crafting

Authors: Denny Mattern, Tobias Gleißner

In contrast to numerical representations, e.g., vector embeddings, symbolic representations use symbols to represent things (cars, motorcycles, traffic signs), people (pedestrians, driver, police), abstract concepts (overtake, brake, slow down) or non-physical things (website, blog, god) as well as their relations. Symbolic knowledge representations comprise all kinds of logical formalism, as well as structural knowledge representing entities with their attributes, class hierarchies and relations.

3.1.1 Logic Formalism

Logic formalisms are used to express knowledge (mostly facts and rules) as formal logical terms. Logic formalisms or logic systems differ in expressivity, complexity and decidability. The choice of the right formalism depends on the concrete problem to model. The most simplistic (and decidable) logic formalism is propositional logic. It consists of a set of symbols representing the individual propositions and a set of junctions that define the relation between propositions or modify the value of a proposition. The value of a proposition can either be `true` or `false`.

P: The car drives carefully. Q: The car is in good condition. R: The car does not cause an accident.

Logical connectors ($\neg, \wedge, \vee, \longrightarrow, \longleftrightarrow$) are used to build compound propositions that again can either be true or false. E.g., $P \wedge Q \longrightarrow R$: The car does not cause an accident if it drives carefully and is in good condition.

In order to make logic statements that apply to many objects, predicate logic (also known as first-order-logic (FOL)) extends propositional logic with truth-valued functions, predicates, constants, variables and quantifiers ($\forall, \exists$). Predicate logic is more expressive than propositional logic but not always decidable, meaning that the truth value of a statement cannot be inferred in every case.

$Destroyable(X)$: X is destroyable. $Car(X)$: X is a car. $\forall X : Destroyable(Car(X))$

We can deduce from the set of axioms that $Car(Model\ T) \longrightarrow Destroyable(Model\ T)$: If Model T is a car (which is true), then it is destroyable. Note that the truth value of the proposition made by a predicate depends on the actual variable.

Although predicate logic is more expressive than propositional logic, both share the property of being truth-valued. I.e., both are binary and both are dealing with statements of ontological nature (statements of being). In order to model legal norms – as we are aiming for – we need logic formalisms that are concerned with the concepts of obligation and permission and would not be binary anymore.

According to scholars of legal theory, norms can be expressed with rules of the structure *IF $A_1,A_2,...,A_n$ THEN C*, where $A_1,A_2,...,A_n$ are the pre-conditions of a norm, *C* is the normative effect and $IF - THEN$ is the normative conditional [26]. In contrast to implications in propositional logic ($\longrightarrow$), the normative conditional is generally defeasible, meaning that even if all pre-conditions hold the normative effect *C* is only concluded presumingly but not necessarily, which allows for reasoning in the face of contradictions. Defeasibility is the property that a conclusion is open in principle to revision in case more evidence to the contrary is provided [26].

Logic formalisms concerning with statements of obligation, permission, prohibition, right, etc. are a type of *modal logic* and known as *deontic logic*. Propositions are augmented with deontic modal operators $[OBL], [PERM], [FOR]$ meaning obligation, permission and prohibition/forbidden. The operators qualify the content of the augmented proposition. E.g., $[OBL]p$ means that the content of p is considered obligatory. The deontic operators relate to another as follows (see [26] for more details):

- $[OBL]p \equiv \neg[PERM]\neg p$: if p is obligatory, then its opposite, $\neg p$, is not permitted.
- $[FOR]p \equiv [OBL]\neg p$: if p is forbidden, then its opposite is obligatory.
- $[PERM]p \equiv \neg[OBL]p \wedge \neg[FOR]p$: p is neither obligatory nor forbidden.

Computer interpretable formalization of legal norms is topic of active research in the field of legal informatics. There are multiple logic formalisms for formalizing legal rules and norms, e.g., Standard Deontic Logic (SDL) ([143, 478]), Reified Input-Output Logic ([609, 610]) or (non-deontic) Temporal Logic ([471, 193]). However, there is still no consensus on the "best" logic formalisms. In order to keep the formalized legal rules agnostic to possible (deontic) logic systems, an intermediate formal representation of the legal norms can be used. LegalRuleML ([537, 27, 26]) aims to provide such an interchange format for legal rules, supporting deontic operators and defeasiblity among other features for formalizing legal norms. As open as the question of the "best" logic formalism for norm representation is the question of a good interface for legal experts who want to represent legal norms computer understandable. A recent work proposes a dedicated editor allowing for intuitive formalization of legal texts and featuring consistency checks as well [433]. Another approach proposes an agile and repetitive process [42].

3.1.2 Relational Knowledge

Knowledge concerning entities, concepts, their hierarchies and properties as well as their relations to another is naturally represented by graph structures. Prominent examples for graph structured representations of structural knowledge are *Taxonomies*, *Ontologies* and *Knowledge Graphs*.

Taxonomies categorize entities into a hierarchy of classes and sub-classes represented as a directed acyclic graph with nodes representing the entities, classes and sub-classes, and edges representing the relations. Taxonomies categorize objects regarding one specific aspect and commonly use only one type of relation – the "is-a" relation. E.g., a car is a vehicle, which is a machine.

An *Ontology* is a formal, explicit specification of a shared conceptualization [699]. This means an *Ontology* is an abstract model of explicitly defined, relevant concepts of the specific domain of discourse and their relations which is constructed in a computer understandable manner. The definitions of the meaning of the relevant concepts and relations reflect the common sense of domain experts. The given definition of what an *Ontology* actually is, implies that the development of a specific *Ontology* is a process which involves different persons (e.g the knowledge engineer, the domain experts, maybe also the users) and that it takes a certain communication effort to develop a shared understanding of the concepts, the formalizations of those concepts as well as the usability of the *Ontology* for the user. Hence, *Ontology* building is ideally an iterative and repetitive design process for which multiple process patterns had been developed [525, 156, 237].

Concrete *Ontologies* consist of classes and sub-classes which refer to domain concepts as well as the properties and relations between those, which is referred to as terminological knowledge. Additionally to the class definitions, relations and constraints for concrete instances of classes are also defined in an *Ontology* and referred to as assertional knowledge. These definitions and constraints are expressed in description logic which is a decidable fragment of predicate logic, where the terminology *TBox* and *ABox* are often used instead of *terminological knowledge* and *assertional knowledge*. The logic is commonly represented in the *Web Ontology Language (OWL)*, which is a computational language based on description logic that allows for formalizing complex knowledge such that it can be exploited by computer programs [534]. An *Ontology* can be interpreted as a meta-schema for domain-specific data, that not only specifies the relational structure and semantics of the data but also allows, e.g., to verify the consistency of that knowledge or to infer implicit knowledge through its strong logical foundation. *Ontologies* have been developed for a wide range of domains and applications.

In literature *Knowledge Graphs* and *Ontologies* had often been used as synonyms until [187] proposed the following definition: "A knowledge graph acquires and integrates information into an *Ontology* and applies a reasoner to derive new knowledge." In a *Knowledge Graph* data from heterogeneous data sources is integrated, linked, enriched with contextual information and meta-data (e.g., information about provenience or versioning) and semantically described with an *Ontology*. Through their linked structure *Knowledge Graphs* are prominently used in semantic search applications and recommender systems but also allow for logical reasoning when featuring a formal meta-schema in form of an

Ontology. Surveys on *Knowledge Graphs* and their general applications are provided by [349, 884] and *Knowledge Graphs* for recommender systems specifically by [267].

3.1.3 Applications

Symbolic representations improve scene understanding by mapping detected objects to a formal semantic representation of the current traffic scene (e.g., as a scene graph ([6, 102])). To integrate knowledge into machine learning algorithms, a representation of this knowledge is essential. While this knowledge is in form of embeddings, a symbolic representation allows traceability and makes it understandable for humans.

Given a sound formalization of traffic rules and a semantic representation of the entities, actions and legal concepts in traffic scenes (analogue to the legal ontology modeling the concepts of privacy proposed by [538]), we can derive the current legal state of an AD vehicle. An example where knowledge graphs are used as embeddings is [528]. In this case a knowledge graph is build upon a road scene ontology to recognize similar situations that are visually different. Using this technique to integrate legal knowledge and derive the legal state of different situations is a possible approach.

Analogue to the application of symbolic representations for situation understanding we make use of formal representations of traffic rules and legal concepts as well as symbolic scene descriptions for planning tasks by ranking possible alternative trajectories and actions, e.g., according to their legal consequences.

3.2 Knowledge Representation Learning

Author: Stefan Zwicklbauer

Complementary strength and weaknesses of data-driven and knowledge-driven AI systems have led to a plethora of research works that focus on combining both symbolic (e.g., Knowledge Graphs (KGs)) and statistical (e.g., NNs) methods [148]. One promising approach is the conversion of symbolic knowledge into embeddings, i.e., dense, real-vector representations of prior knowledge, that can be naturally processed by NNs. Typical examples of symbolic knowledge are textual descriptions, graph-based definitions or propositional logical rules. The research area of Knowledge Representation Learning (KRL) aims to represent prior knowledge, e.g., entities, relations or rules into embeddings that can be used to improve or solve inference or reasoning tasks ([439], [424]). Most existing literature narrows down the problem by defining KRL as converting prior knowledge from KGs only [439]. Thus, our focus in this survey also lies on knowledge modeled in graph-based structures.

3.2.1 Textual Embeddings

With the development and advances in DL, Natural Language Representation Learning has become a hot topic over the last couple of years. Natural Language Models, such as proposed in [164], [563], [78] are capable of directly converting natural language text, e.g., common sense text like Wikipedia articles or textual rules like road traffic regulations into embeddings that implicitly represent the syntactic or semantic features of the language [576]. Those embeddings are mostly used for specific downstream tasks like Question Answering (QA) [875], Neural Machine Translation (NMT) [817] or Common Sense Reasoning [697], but probably lack power of expressiveness when it comes to representing specific rules and logic. As a consequence, most research works extract entities, relations and rules from sentences first and model them in a more expressive representation format, e.g., KGs, afterwards. In the following, we do not further elaborate literature regarding Natural Language Representation Learning but refer to the respective surveys ([576], [131]) and assume that knowledge has already been converted to an expressive format like KGs or another logical system.

3.2.2 Knowledge Graph Embeddings

Many research works described how to create dense-vector representation for either homogeneous (i.e., graphs with a single type of edge) and heterogeneous (i.e., graph with multiple types of edges) graphs [145]. Graphs with auxiliary information ([519], [270]) and graphs constructed from non-relational data [845] are out of scope in this survey. For homogeneous graphs, the authors of [562] made a significant progress in KRL. They created a node corpus by randomly walking over the graph and applied Word2Vec [485] to generate node embeddings. The authors of [885] further improved and used this approach for heterogeneous graphs. Tang et al. [713] and especially Grover at al. [262] proposed state-of-the-art works which intelligently explore the specific and varying neighborhoods of nodes and consider the respective node order to create their embeddings. Most research works however, focus on heterogeneous graphs since they are best suited for rule and relation modeling. We first focus on pure node (entity) and edge (relation) representation learning, also called Triplet Fact-based Representation Learning Models. Hereby, we further distinguish between Translation-Based Models, Tensor Factorization-Based Models and NN-Based Models.

Starting with Translation-Based Models, the first influential work proposed *TransE* [73], a framework to create embeddings for heterogeneous graphs. Given a triple (h, r, t), with h and t denoting the head and tail entity and r denoting the respective relation, the idea is to embed each component h, r and t into a low-dimensional space $\mathbf{h}, \mathbf{r}, \mathbf{t}$ in a way that $\mathbf{h}$ and $\mathbf{r}$ translate to $\mathbf{t}$: $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. The authenticity of the respective triplet is defined via a specific scoring function, which is the distance under either ℓ_1 or ℓ_2 norm:

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_p \qquad (1)$$

with $p = 1$ or $p = 2$. This objective function is minimized with a margin-based hinge ranking loss function over the training process. Since *TransE* came up with several limitations, such as not being able to model one-to-many, many-to-one and many-to-many relations, various authors addressed these shortcomings by using *TransE* as foundation for their works. For instance, the authors of *TransH* [773] introduced relation-related projection vectors where the entities are projected onto relation-related hyperplanes. *TransH* enables different embeddings based on the underlying relation. All entities and relations are still represented in the same feature space. In *TransR* [440], the entities h and t are projected from their initial entity vector space in to the relation space of the connecting relation r. This allows us to render entities that

are similar to the head or tail entity in the entity space as distinct in the relation space. Further improvements can be found in the *TransD* [348] model, which has fewer parameters and replaces matrix-vector multiplication by vector-vector multiplication for an entity-relation pair, which is more scalable and can be applied to large-scale graphs. Another problem of existing approaches is the non-consideration of crossover-interactions, bi-directional effects between entities and relations including interactions from relations to entities and interactions from entities to relations [858]. To provide an example, predicting a specific relation between two entities typically relies on the entities' relevant topic in form of their connecting entities/relations. Not all connected entities and relations belong to the topic of the relation to be found. This is modeled in *CrossE* [858], which simulates crossover interactions between entities and relations by learning an interaction matrix to generate multiple specific interaction embeddings. Another state-of-the-art approach *Hake* [862] is capable of modeling a) entities at a different level in the semantic hierarchy, and b) entities on the same level of the semantic hierarchy. This is achieved by mapping the entities in the polar coordinate system. Entities on a different hierarchy level are modeled with a modulus approach, whereas the phase part aims to model the entities at the same level of the semantic hierarchy.

Regarding Tensor Factorization-Based Models, *RESCAL* [517] represents the foundational work for most follow-up works. *RESCAL* uses a tensor representation to model the structure of KGs. More specifically, a rank-d factorization is used to obtain the latent semantics: $\mathbf{X}_k \approx \mathbf{A}\mathbf{R}_k\mathbf{A}^T, for \ k = 1, 2, ..., m$, with $\mathbf{A} \in \mathbb{R}^{nxd}$ being a matrix that captures the latent semantic representation of entities and $\mathbf{R}_k \in \mathbb{R}^{dxd}$ being a matrix that models the pairwise interactions in the k-th relation. Based on this principle, the scoring function is defined as $f_r(h,t) = \mathbf{h}^T\mathbf{M}_r\mathbf{t}$, where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ denote the entity embeddings and the matrix $\mathbf{M}_r \in \mathbb{R}^{dxd}$ represents the pairwise interactions in the k-relation ([517], [145]). The work *DistMult* [813] improves *RESCAL* in terms of algorithmic complexity and embedding accuracy by restricting $\mathbf{M}_r$ to be diagonal matrices. To overcome the problem of *DistMult* that head and tail entities are symmetric for each relation symmetry, the works *Complex* [728] and *QuatRE* [513] satisfy the key desiderata of relational representation learning, i.e., modeling symmetry, anti-symmetry and inversion. Both approaches leverage complex-value embeddings to support asymmetric relations. More recently proposed state-of-the-art models use special tensor factorization methods. For instance, *SimplE* [372] leverages an adapted and simpler version of Canonical Polyadic Decomposition to allow head and tail entities to have embeddings that are dependent on each other, which would be impossible with the original model. Similar, *TuckER* [36] is based on the Tucker-Decomposition on a binary entity-relation-entity matrix.

Due to their success in the last decade, NN-Based Models became also a hot topic for KRL. The first shallow NN approaches comprise standard feed-forward networks [73] (with linear layers) and neural tensor networks [692] (with bi-linear tensor layers). Over time deeper variants such as *NAM* [446] have established to provide more flexibility when it comes to train a network towards the underlying training goal. More recently, graph neural networks

[872] were introduced which strive to explicitly model the peculiarities of (knowledge) graphs. In particular, graph convolutional networks for multi-relational graphs [384] generalize nonvolutional neural networks to non-euclidean data and gather information from the entity's neighborhood and all neighbors contribute equally in the information passing. Graph convolutional networks are mostly built on top of the message passing neural networks framework [248] for node aggregation. Many works are limited to create embeddings for knowledge entities only ([637], [664]), but recent approaches tried to overcome this limitation ([163], [738], [826], [739]). A neighborhood attention operation in graph attention networks [742] can enhance the representation power of graph neural networks [808]. Similar to natural language models, these approaches apply a multi-head self attention mechanism [740] to focus on specific neighbor interactions when aggregating messages ([808], [4], [467]). Many authors incorporated mechanisms to improve the overall quality of entity and relation embeddings. For instance, the idea of negative sampling is to intelligently sample specific wrong samples that are needed for margin-based loss functions. Recent methods employed Generative Adversarial Networks (GANs) [253] in which the generator is trained to generate negative samples ([761], [86]). Another work *ATransN* suggested to improve existing embeddings by leveraging GANs to correctly align the embeddings with those from teacher KGs [756].

In this section, we mostly concentrated on methods that exclusively generated their embeddings on relational data. However, some approaches consider additional information, such es textual (entity) descriptions (e.g., [221], [774], [273]), path-based information (e.g., [521], [266]) and even hierarchies (e.g., [862], [863]) as available in ontologies.

3.2.3 Knowledge Graph Embeddings with Rule Injection

So far, we have discussed approaches to embed knowledge that is formalized within KGs. These methods create representations that purely reflect the items' graph-based modeling (e.g., triples). In addition to this, specific rules (soft or hard rules) can be derived from KGs, which is also known as rule learning (e.g., [312], [859], [529]), or be leveraged in the embeddings learning process, also known as rule injection. In the following, we focus on the former works, how to additionally integrate pre-defined or mined rules into embeddings. The authors of *RUGE* [269] presented a novel paradigm to leverage horn soft rules mined from the underlying KG in addition to the existing triples. Their iterative training procedure improves the transfer of the knowledge contained in logic rules into the learned embeddings. The framework *SLRE* [268] also presents an option to leverage horn-based soft rules with confidence scores to improve the accuracy of down-stream tasks. These rules are directly integrated as regularization terms in the training mechanism for relation embeddings. The authors of [521] additionally enriched the horn-based rules with path information to improve the state of the art. A related work [762] mines inference, transitivity and anti-symmetry rules from the given KG first and converts them into first-order logic rules in the second step. Finally, the proposed rule-enhanced embedding method can be integrated in any translation-based KG embedding model.

Apart from rules directly mined from the underlying knowledge graph, other approaches exist that try to apply more extrinsic rules. For instance, the authors of [171] try to improve the embeddings' capability of modeling rules by using non-negativity and approximate entailment constraints to learn compact entity representations. The former naturally induce sparsity and embedding interpretability, and the latter can encode regularities of logical entailment between relations in their distributed representations. Other works propose to encode knowledge items into geometric regions. For instance, [274] encodes relations into convex regions, which is a natural way to take into account prior knowledge about dependencies between different relations. *Query2box* [594] encodes entities (and queries) into hyper-rectangles, also called box embeddings to overcome the problem of point queries, i.e., a complex query represents a potentially large set of its answer entities, but it is unclear how such a set can be represented as a single point. Box Embeddings have also been used to model the hierarchical nature of ontology concepts with uncertainty [429].

Most approaches described above rely on common-sense knowledge bases like DBPedia [28] or Freebase [70] and leverage their developed embedding approaches for knowledge base link prediction or inference / reasoning tasks. However, we believe that existing models and algorithms can be similarly applied to special domain knowledge bases, e.g., knowledge bases with data for AD [787].

3.2.4 *Applications*

The application of KGs in the AD domain has not received too much attention at the current point of time, albeit it can be an effective way to help situation or scene understanding [732]. For instance, the authors of [283] built a specific ontology to represent all core concepts that are essential to model the driving concept. The built KG *CoSi* models information about driver, vehicle, road infrastructure, driving situation and interacting traffic participants [283]. To classify the underlying traffic situation with a NN, a relational graph convolutional network [637] is used to convert the underlying KG into embeddings first. Similar, the work by Buechel et al. [82] presented a framework for driving scene representation and incorporated traffic regulations. Wickramarachchi et al. [787] focused on embedding AD data and investigated the quality of the trained embeddings given various degrees of AD scene details in the KG. Moreover, the authors evaluated the created embeddings on two relevant use cases, namely *Scene Distinction* and *Scene Similarity*.

4 KNOWLEDGE INTEGRATION

A plethora of methods and approaches have been proposed in literature that focus on augmenting data driven models and algorithms with additional prior knowledge. Among the most prominent approaches are the modification of the training objective via customized cost functions, especially knowledge affected constraints and penalties. An overview of *auxiliary losses and constraints* that take into account physical and domain knowledge in various peculiarity is presented in Section 4.1. Often these approaches are accompanied by problem-specific designs of the architecture, leading to hybrid models that leverage symbolic knowledge in form

of logical expressions or knowledge graphs. The merging of symbolic and sub-symbolic methods, also referred to as *neural-symbolic integration* is focus in Section 4.2.

Besides external input, recent methods rely on preferably internal representations in order to focus *attention* on distinct features and concepts within a network itself. Key weighting and guidance approaches are discussed in Section 4.3. Last but not least, *data augmentation* techniques form the backbone to integrate additional domain knowledge into the data and thus indirectly into the model. Approaches starting from data transformations to augmentations in feature space up to simulations are discussed in Section 4.4.

In addition to these prevalent general approaches, this chapter concludes with methods and paradigms that are more tailored to the field of autonomous driving, considering multiple agents that interact with specific environments typical for the application under investigation. Especially inferring and predicting the state of an agent plays an essential role in the considered *state space models* in Section 4.5 and *reinforcement learning* in Section 4.6. The involvement of positional as well as semantic information is essential part of the *information fusion* approach outlined in Section 4.7.

4.1 Auxiliary Losses and Constraints

Authors: Tino Werner, Maximilian Alexander Pintz, Laura von Rueden, Vera Stehr

The usual Empirical Risk Minimization (ERM) principle in machine learning amounts to replacing the minimization of an intractable risk, i.e., an expected loss over a ground-truth data distribution, by the minimization of the empirical risk. A mismatch between the expected loss and its empirical approximation causes ERM to result in models that do not generalize well to unseen data. This manifests either in overfitting, where the model represents the training data too closely and fails to capture the overall data distribution, or underfitting, where the model fails to capture the underlying structure of the data. Regularization schemes have been proposed to mitigate the problem of overfitting. The Structural Risk Minimization (SRM) principle [275, 736] extends the ERM principle for regularization. SRM seeks to find models with the best tradeoff between the empirical risk and model complexity as measured by the Vapnik-Chervonenkis dimension or Rademacher complexity. In practice, this encompasses minimizing an empirical risk with an added regularization term. This technique has successfully entered variable selection as done in the path-breaking work of [721] who introduced the Lasso. Regularization in general proved to be indispensable in high-dimensional regression [186, 882, 89, 835, 685], classification [544, 735], clustering [791], ranking [410] and sparse covariance or precision matrix estimation [38, 220, 87].

As for knowledge-infusion into AI, a natural strategy is to similarly use regularization terms (so-called auxiliary losses) that correspond to formalized knowledge. However, constraints may also appear in terms of hard constraints, for example, if some logic rule must not be violated so that integrating it in a soft manner via auxiliary losses would not be appropriate, as dependencies or as regularization priors. This section is structured as follows: After describing techniques that integrate physical knowledge or other

domain knowledge via auxiliary losses, we review ideas to incorporate constraints into the AI training and the AI architecture, followed by works that propose uncertainty quantification for knowledge-infused networks. At the end, we review applications of knowledge-infused networks for perception and planning in the automotive context.

Let us first point out the advantages of such techniques, besides the stronger adaptation of the model to the knowledge. For example, the authors in [370] highlighted that the knowledge-based regularization term does generally not require labeled inputs, which enables data augmentation with unlabeled instances, saving a large amount of time and money that would be required to generate a large labeled data set. The approach in [230] does not even require any labeled instance. Moreover, a common result is better generalizability of the model, paralleling the improved generalization ability of models trained with complexity regularization. Each improvement in explainability and interpretability of deep models is especially relevant for autonomous driving in order to increase the public acceptance of self-driving vehicles.

4.1.1 Knowledge Integration via Auxiliary Losses

One has to distinguish between common penalty terms that regularize the complexity of the model as outlined in the previous paragraph and knowledge-based penalty terms that integrate formalized knowledge into the model. Surveys on knowledge integration, including auxiliary loss functions, are given by [696, 159, 580, 788, 623, 74]. An important side effect from knowledge integration, apart from a better generalization performance, is that it increases explainability and interpretability of the model by at least partially explaining the predictions by the knowledge. This is especially true for deep models which are usually black boxes.

Regarding the success of regularization in machine learning, knowledge-based regularization terms have the potential to significantly improve machine learning models by encouraging them to respect existent knowledge without wasting computational effort for learning this knowledge again from scratch during training but more efficiently. It is important to note that knowledge integration via losses and penalties is also possible if the knowledge is not present in the data (for example, if it is related to rare cases) or if it cannot easily be derived from the data. Work in this direction has been done by [696], but note that [392] already had the idea of physics-based regularization when solving inverse problems. The authors in [696] consider the applications of predicting a height curve when throwing an object (constraint: it is a parabola), the location of a walking person (constraint: the velocity should remain constant) and casual relationships (video game) by adding a suitable regularization term to the loss function.

Physical Knowledge: Real-world environments are constrained by physical laws, which need to be considered for realistic modeling. Several approaches have been proposed for infusing such physical knowledge into neural networks. In [583, 582, 581] physics-informed NNs are introduced to reliably solve partial differential equations (i.e., enforcing the solution to respect physical laws) like the Schrödinger equation or discrete time models like Runge-Kutta models. The authors in [458] describe how to impose soft boundary conditions for Partial Differential Equations (PDEs) via auxiliary loss functions. The authors in [669] use physics-informed CNNs in order to predict physical fields. The authors in [307] propose physics-guided NNs that solve PDEs while satisfying thermodynamical constraints. Further applications in differential equation and dynamics modeling are given for example in [45, 147, 586, 587, 475, 828, 853, 765, 766, 230, 346, 345, 807].

Several works on physics-informed NNs consider the problem of temperature modeling (e.g., of lakes or sea surfaces), such as [370, 59] or [504], who try to encourage a monotonicity constraint by an auxiliary loss term (e.g., the water density increases monotonically with depth). The authors in [350] use physics-guided recurrent graph networks to model the flow and the temperature in rivers and enforce the model to respect local patterns via physics-guided regularization. In [351], an energy conservation constraint is integrated while [734] apply physical regularization in fuel consumption modeling.

Domain Knowledge: The authors in [747] incorporate domain knowledge (here: sentiment dictionary/ontology, linguistic patterns) into DL in the context of sentiment analysis. Medical domain knowledge in terms of priors on abdominal organ sizes is integrated into Deep Neural Network (DNN) models in [874] for the task of segmenting organs on Computerised Tomography (CT) scans. The authors in [101] propose knowledge-guided GANs that are trained using image data and additional textual descriptions of (potentially unseen) input images (types of flowers). They train two generators, one for generating images of seen categories and one for unseen categories, and use an auxiliary loss to transfer knowledge between the generators. In [815] the contribution of domain knowledge is quantified by approximating the Shapley value (see also Section 7.2.2) of a particular knowledge constraint.

Imposing general logical rules (equations, inequalities, orderings) on network outputs is also considered for incorporating domain knowledge in the literature. The authors in [804] construct a loss function, such that the output of neural network satisfies certain first-order logic sentences upon minimization of the loss. In [213] are more general framework is proposed that turns general first-order sentences into differentiable loss functions using max or logit operators. The training of the constrained NN is simply done using standard optimization techniques like Stochastic Gradient Descent (SGD).

4.1.2 Integration of other Constraints

Adding a knowledge-based regularization term to a loss function typically enforces constraints in a soft manner. However, in many cases we would like to ensure that constraints are perfectly satisfied, i.e., enforce hard constraints that correspond to the limit case of auxiliary regularization terms with infinite regularization parameters. In the following, several approaches are introduced that aim at incorporating hard constraints. Besides using auxiliary losses, these often employ other approaches for constraint incorporation such as a change in architecture or use different optimization

schemes such as projected gradient descent or conditional gradients [588].

Hard constraints: Methods to train NNs with hard constraints on the output layer are explored in [476], but due to the large dimensionality it is infeasible to apply standard Lagrangian techniques and even worse, if the constraints are incompatible, one will face numerical instabilities. In order to solve the linear system imposed by the Karush Kuhn Tucker (KKT) conditions, they use the Krylov subspace method which iteratively solves linear equations. The required products of Jacobians and vectors are computed using the Pearlmutter trick. It is further discussed how the Krylov method can be improved to cope with ill-posed constraints, how a constrained Adam looks like and how to reduce the number of constraints during learning. The latter is achieved by randomly selecting active constraints on the unlabeled data as SGD selects instances for labeled data. They suggest that randomly choosing them may be replaced by using the ones for which the constraint violation is largest, i.e., it is some kind of active learning approach for the constraints. The authors in [506] show how to perform deep learning with hard constraints by converting hard label constraints to soft logic constraints over distributions. They point out that the work of [170] is similar to theirs but does not use a full Lagrangian for respecting the constraints (logical formulas) but modify the loss functions so that hard constraints cannot be handled. Equality constraints are formulated using the two corresponding inequality constraints. Using the Hinge loss, the constraints can be equivalently written as equality constraints ([376] call it ReLU Lagrangian) which reduces the number of constraints and allows any constraints as long as they are differentiable. Training is done via subgradient descent. The authors in [212] consider inequality constraints on DNNs and formulate it very similarly to [506] using the Hinge loss, but they consider a primal-dual formalization as [110] for solving the problem. The authors in [376] propose log-barrier extensions to approximate the Lagrangian optimization of constrained CNNs with a sequence of unconstrained losses with an initial feasible set of parameters. The main idea is to first compute any feasible point of the constrained problem with an inequality constraint and to approximate the original problem with the unconstrained problem but where the inequalities enter as penalty terms with a log-barrier function which approximates the Hinge loss. They provide a continuous and twice differentiable log-barrier extension which is no longer restricted to feasible points and therefore does not require to find a feasible initial point. The authors in [549] consider the training of matrix inequality constrained (semidefinitely constrained) NNs that are used for enforcing Lipschitz continuity or stability. Training robust, i.e., Lipschitz NNs has already been considered in [550] who solve the Lipschitz-regularized optimization problem using an Alternating Direction Method of Multipliers (ADMM) scheme. In order to capture even nonlinear matrix inequality constraints, [549] propose to transform the constrained problem into an unconstrained problem using log-det barrier functions. The framework of [213] enforces besides the logic-based soft constraints, also convex hard constraints via projected gradient descent (projecting gradients back into the convex constraint region). Besides enforcing soft boundary conditions of physics-based

models, the authors in [458] also propose a NN architecture for encoding hard constraints.

Constraint incorporation via layers: Other techniques address the problem which knowledge constraints to integrate when and to which extent (e.g., how the regularization parameters have to be chosen). The authors in [403] criticize that many existing approaches incorporate the knowledge before or after the learning process by feature extraction or validation and therefore propose a method how to incorporate it within the hidden layers themselves, i.e., by infusing the knowledge between the layers. In order to decide whether knowledge should be incorporated between particular hidden layers and how the latent representation and the knowledge representation merge, they propose two loss functions. As for the knowledge representation, they build knowledge graphs. The knowledge infusion is realized by a knowledge-infusion layer which minimizes the gap between the learned representation and the knowledge representation (called differential knowledge) using a knowledge-aware loss function, i.e., a relative entropy loss quantifying the information gain from the knowledge representation. Finally, a weight matrix based on the differential knowledge is learned and the AI is trained using Backpropagation (BP).

The authors in [15] propose OptNet that uses special DNN layers for solving optimization problems which encode constraints as well as complex dependencies among the hidden nodes. They concentrate on quadratic problems and the solution becomes the output of the respective layer. To enable BP through these layers, the derivatives of the solutions (i.e., of the argmin operator) have to be computed, which is done by differentiating the KKT conditions. They prove that the OptNet layers are subdifferentiable everywhere and that they can approximate any piecewise linear function but, however, point out that OptNet layers are costly. In [30], a linear complementarity problem for equality- and inequality-constrained reinforcement learning (see Section 4.6) is formulated which, using the results from [15], allows for gradient computation while keeping the BP solution scheme. Their approach can be interpreted as adding a physics-based layer to the network.

There are also efforts to incorporate logic constraints directly into the network architecture. The authors in [427] consider logic rules on the activations of DNNs. To enforce such rules, the pre-activations of the network are augmented with terms that increase when given logical statements are satisfied. A differentiable logic layer for trajectory prediction that can incorporate symbolic priors and temporal logic formulae is proposed in [430]. Since this requires much less labeled data, trajectory predictors can serve as trajectory generators. The parameter adjustment to the rules is done in the BP step. Furthermore, the layer can check whether rules are satisfied/violated. The idea is to define a robustness function based on signal temporal logic formulae so that they are satisfied if and only if the robustness function is greater than zero. Minimum and maximum operators are smoothly approximated. Training is done by BP where the gradients of this robustness function are used. In [329], DNNs are combined with declarative first-order logic rules. This is done by enforcing the NN to predict the outputs of a logic-rule-based teacher and updating both NNs iteratively. For a classification task, the softmax output that the student

network assigns to an instance is projected onto the rule subspace where the constraints are satisfied, leading to the softmax output of the teacher network. The parameters of the student network are iteratively updated while the teacher network is trained so that it satisfies the first-order constraints by minimizing the Kullback-Leibler (KL)-divergence.

Posterior regularization: The authors in [479] propose robust RegBayes which does not incorporate knowledge via the priors but by posterior regularization w.r.t. first-order logic rules (see **??**). The idea builds upon regularized Bayesian inference (RegBayes) from [877]. Robust RegBayes takes the uncertainty about the domain knowledge into account and outputs parameters that reflect the importance of each logic constraint which will be low in cases of large uncertainties about the knowledge. In [328], the method of [329] is generalized by jointly learning both the regularized DNN models as well as the structured knowledge. More precisely, the task is to learn the regularization parameters in the penalized objective function as well as dependency structures of the knowledge constraints. Their technique can be interpreted as regularized Bayes with generalized posterior [877]. The authors in [850] also propose posterior regularization (see [229]) for prior knowledge integration in order to handle multiple overlapping prior knowledge sources in the context of neural machine translation. They penalize the likelihood by the KL-divergence of the resulting model and a distribution that encodes prior knowledge. In [558], discrete constraints and regularization priors for CNNs are proposed, leading to discrete-valued regularization terms. The optimization problem is re-formulated as Augmented Lagrangian Method (ALM) and solved using an ADMM scheme.

4.1.3 Uncertainty Quantification of Knowledge-based DNNs

The authors in [154] combine the physics-guided architectures with Monte Carlo (MC) dropout (c.f. Section 8.1) for uncertainty quantification and show that the physics-guided NN approach still yield black-box models and that the random dropping of weights again leads to physically inconsistent predictions. They remedy this issue by introducing physically-informed connections and physical intermediate variables which grant certain neurons a physical interpretation. They consider a monotonicity-preserving Long Short-Term Memory (LSTM) which extracts temporal features and predicts an intermediate physical quantity (water density) such that the monotonicity is satisfied for this quantity by hard-coding it into the architecture. Then, an Multi-Layer Perceptron (MLP) combines these predictions with the inputs to get the predicted responses. The perturbations injected by MC Dropout do not destroy the consistency with the physical knowledge. In [847], a dropout variant for uncertainty estimation (both approximation and parameter uncertainty) in physics-guided NNs is suggested for the context of forward and inverse stochastic problems by invoking polynomial chaos and MC dropout. The authors in [821] propose a latent-variable-based adversarial inference procedure for uncertainty quantification of physics-based NNs. In [362, 363], uncertainty quantification for physics-guided NNs in dynamical systems is done by a coarse-graining process which again results in a Bayesian-type approach where an evidence lower bound is maximized.

4.1.4 Applications

Knowledge integration has touched upon several perception tasks. As for object detection, [463] integrate prior knowledge about the size of the bounding boxes of vehicles into the model by imposing size constraints for the boxes. The authors in [546] consider equivariance constraints in weakly supervised segmentation in order to cope with affine image transformations. As CNNs are not equivariant in general, they impose an equivariance-preserving loss and extend this technique for shared information between multiple networks. The authors in [827] propose a knowledge-based attentive Recurrent Neural Network (RNN) (see also Section 4.3) for traffic sign detection, motivated by the fact that small objects are not yet detected reliably by DNNs. The idea is to impose a prior distribution on the location of the traffic signs that represents the domain knowledge that the driver's attention is the bias of the center and the intuitive knowledge that human's attention follows a Gaussian distribution. The former emerged from [130] who automatically learn priors that respect that issue, i.e., it learns the bias from eye fixations. As for semantic segmentation, [375] impose constraints such that each bounding box at least has to contain a foreground pixel (to prevent excessive shrinking) and no background pixel (background emptiness constraint). To solve the resulting problem, they employ log-barrier extensions and optimize the corresponding Lagrangian function directly via SGD as proposed in [376]. The authors in [757] propose a bounding box tightness prior for weakly supervised image segmentation by applying a smooth maximum approximation instead of posing it directly as constraints as in [375]. In [547], constrained CNNs are proposed to incorporate weak supervision into the learning procedure. The idea is to define linear constraints on the output layer that enforce the output being near the latent distribution from weak supervision. Their formulation covers for example bounds for the expected number of foreground and background pixel labels in a scene, suppression of a label in a scene (object is not allowed to appear) or size constraints. They concretize the problem using a KL-divergence-based loss function which can be solved using SGD on the dual. A related approach is presented by [527] who impose anatomical constraints on a CNN. The authors in [763] propose virtual adversarial training for anatomically-plausible image segmentation, i.e., they generate adversarial samples that violate the topological constraints and let the network learn to avoid such predictions. They point out that additional losses that correspond to some constraint violation may not exist or may not be differentiable. Even worse, if the constraints are complex relationships, the NN may never violate them during training so that the constraint will always lead to a gradient of zero. They optimize a regularized cross-entropy loss where the context-aware regularizer is the maximum of a KL-divergence, penalized by a constraint loss which encourages adversarial samples. The authors in [558] impose discrete constraints which may be lower and upper bounds for the foreground size and the regularization prior can be a measure of the similarity of the intensity or color of neighboring pixels. [557] experimentally derive that existing Active Learning (AL) methods work poorly for lane detection due to label noise (maybe due to occlusion or

unclear lane markings) and due to the fact that the entropy criterion leads to selecting images with no or only few lanes. They propose to train a student model using the same loss as for the teacher model, regularized with a distillation loss. As for mitigating the label noise that may be the reason for large discrepancies of the teacher and the student, they train another student without knowledge distillation. They select samples where the discrepancy of the student's predictions are large but where the discrepancy of the teacher and the distilled student are low (teacher may be erroneous here) or where the latter discrepancy is large and those between the students is low (knowledge is difficult to learn). Experiments are conducted on the LLAMAS and the CULane data set.

In [476], human pose estimation with hard symmetry constraints is considered while [327] impose a consistency constraint that encourages the body parts of the generated images match the respective parts in the real images. As for image classification, [480] include hierarchical domain knowledge into classification tasks, i.e., that all parts belonging to a certain vehicle or all vehicles that contain a given part are considered. The authors in [815] incorporate symbolic knowledge in classification, i.e., they consider super-classes that provide information about the potential actual sub-classes. Knowledge can also be integrated into tracking and trajectory prediction. In [261], the Yaw loss, an auxiliary differentiable heading loss that penalized angle differences between the optimal and the predicted headings, is proposed, where the case of road intersections is also respected. The authors in [518] propose an off-road loss for improving the movement prediction of traffic participants. This loss is the mean Euclidean distance between each predicted waypoint and the corresponding nearest feasible (drivable) point. In [76], this approach is extended by using a pre-trained model (according to off-road loss) and by combining it with models like CoverNet from [566] that respect dynamic constraints and that make multimodal probabilistic trajectory predictions or by the method from [140] who predict kinematically feasible trajectories using a kinematic layer. The authors in [696] enhance pedestrian tracking models by the world knowledge that the walking speed is constant. The idea in [33] is to add residuals to knowledge-driven trajectories in order to better reflect the stochastic behavior, to make it more realistic and to let the prediction effectively account for other agent's behaviors. They also consider social rules (world knowledge) concerning the movements of pedestrians. They show that their approach can also be used for multimodal prediction and combined with the kinematic layer from [140]. The authors in [864] propose STINet for joint pedestrian detection and trajectory prediction. The idea is to model temporal information for each pedestrian so that current and past states are predicted. They also model the interaction of the pedestrians with an interaction graph. A temporal-region proposal network is applied in order to make object proposals in terms of past and current boxes, supervised by the ground truth boxes. In [356], the interaction-aware Kalman NN for predicting interaction-aware trajectories is proposed.

As for planning, knowledge-infused models for semantic segmentation, object recognition and trajectory prediction outlined in the perception subsection can potentially be used for planning the ego-trajectory since they improve the quality of the observed and predicted states respectively. Especially approaches like STINet [858] that incorporate social interactions are candidates since the interactions with the ego-vehicle can be included. The authors in [626] add different regularization terms corresponding to speed limits, dynamics or lane changes. In [141], the ellipse loss is proposed which penalizes the bounding box regression and orientation loss with an off-road loss computed by a non-drivable region mask which is added to the computed Gaussian raster. Position and heading of the agent enter as regularization terms in the ChauffeurNet of [40]. The authors in [94] consider penalties, for example for trajectory curvature, lateral acceleration and off-road driving.

The authors in [69] propose to use physics-guided NNs for inversion-based feed-forward control applied to linear motors. Two physics-guided NNs are considered, one in which the inputs are transformed according to the feed-forward controller (i.e., physics-guided input transformation) and one in which a physics-guided layer is used where the output is transformed according to the physical model (maybe enhanced with a physics-guided input transformation). Their model is applied to tracking tasks.

4.2 Neural-symbolic Integration

Authors: Tobias Gleißner, Philip Gottschall, Christian Hesels, Gurucharan Srinivas

Machine learning and deep learning techniques (so-called sub-symbolic AI techniques) have proven to be able to achieve great performance in pattern recognition tasks of numerous kinds: image recognition, language translation, medical diagnosis, speech recognition, recommender systems and many more. While the accuracy in performing those tasks which require dealing with large and noisy input is often on par with human abilities or even beyond that, they come with certain drawbacks: They usually offer no justification for their output, require (too) much data and computational power to be trained, are susceptible to adversarial attacks and are often criticized to generalize weakly beyond their training distribution. On the other hand, "classic" so-called symbolic AI systems such as reasoning engines can provide output that is explainable but performs badly when it comes to handling large or noisy input.

Merging methods from the fields of symbolic and sub-symbolic AI is the purpose of neural-symbolic integration. Its goal is to remedy the drawbacks of both approaches and combine their advantages by integrating methods of both fields. A first taxonomy for the types of those integrated systems was proposed by Henry Kautz at AAAI 2020 [371] and provides a quick survey on the kinds of systems in neural-symbolic integration:

- Neural networks that create symbolic output from symbolic input, e.g., machine translation.
- Neural pattern recognition subroutines within a symbolic problem solver, e.g., the Monte-Carlo search in the core neural network of AlphaGo [682].
- Systems in which the neural and symbolic are plugged together and utilize the output of the other system(s), e.g., the neuro-symbolic concept learner [474] or a reinforcement agent working together with symbolic planners [338].

- Neural networks that have knowledge compiled into the network, e.g., if-then rules [234].
- Symbolic logic rules embedded into a neural network that acts as a regularizer.
- Neural networks that are capable of symbolic reasoning such as theorem proving.

4.2.1 Methodological Overview

Neural-symbolic methods for reasoning. In [407] multiple approaches were presented to integrate symbolic systems in Graph Neural Networks (GNNs). GNNs allow for two major advantages in solving reasoning tasks. They apply an inductive bias directly through their architecture and offer permutation invariance because of their update and aggregation functions. Permutation invariance simplifies the representation of literals and clauses. Therefore the order of logical symbols does not impact the learning and understanding of such clauses. For example the GNN handles the logical expression $(x_1 \lor \neg x_2 \lor x_3)$ semantically the same as the expression $(x_1 \lor x_3 \lor \neg x_2)$. GNNs enable visual scene understanding and reasoning superior to Convolutional Neural Networks as shown in [633].

Tensorization of first-order logic is another approach for solving reasoning tasks utilizing Deep Learning in combination with neural-symbolic integration. Logic Tensor Networks (LTNs) as presented in [657] are able to use full first-order logic with function symbols by embedding these logic symbols into real-valued tensors. They propose a neural-symbolic formalism called Real Logic in addition to the computational model that is designed for defining logical expressions suited for tensorization in LTNs.

Real Logic is a many-valued, end-to-end differentiable first-order logic. It consists of sets of constant, functional, relational and variable symbols. Formulas build from these symbols can be partially true and therefore Real Logic includes fuzzy semantics. Constants, functions and predicates can also be of different types represented by domain symbols. The logic also includes connectives $\diamond \in \{\neg\}, \circ \in \{\land, \lor, \rightarrow, \leftrightarrow\}$ and quantifiers $Q \in \{\forall, \exists\}$. Semantically Real Logic interprets every constant, variable and term as a tensor of real values and every function and predicate as a real function or tensor operation. Therefore Logic Tensor Networks are able to efficiently compute an approximate satisfiability by mapping logical expressions to real-valued tensors.

Moreover, [32] presents multiple related approaches that integrate logical reasoning and deep learning while being end-to-end differentiable:

- Logical Neural Networks [603] use a logical language to define their architecture. By applying a weighted Real Logic a tree-structured neural network is built with different logical operators represented by different activation functions.
- DeepProbLog [473] is a probabilistic logic programming language that implements a Neural Network capable of solving reasoning tasks by applying logical inference.

Neural-symbolic architectures for context understanding. In [528] two applications for neural-symbolism are demonstrated and evaluated. The first application focuses on autonomous driving and uses Knowledge Graph Embedding Algorithms to translate Knowledge Graphs into a vector space. The Knowledge Graph is generated from the NuScenes dataset and consists of the given Scene Ontology with a formal definition of a scene and a subset of Features-of-Interests and events defined within a taxonomy. By creating the Knowledge Graph and the use of Knowledge Graph Embeddings it is possible to calculate the distances of scenes and to find similar situations that are visually different. Presented methods to create Knowledge Graph Embeddings are TransE, RESCAL and HoIE, where TransE shows the most consistent performance on the quantitative Knowledge Graph Embeddings-quality metrics.

The second application is "Neural Question-Answering" with knowledge integration using attention-based injection. The presented method uses knowledge from ConceptNet and ATOMIC and injects it into an Option Comparison Network by fusing the commonsense knowledge into BERT's output. It is evaluated with the CommonsenseQA dataset and the analysis suggests, that attention-based injection is preferable for knowledge injection.

Neural-Symbolic Program Search for Autonomous Driving Decision Module Design. In [700] Neural Architecture Search (NAS) framework is proposed, which automatically synthesizes the Neuro-Symbolic Decision Program (NSDP) to improve the autonomous driving system design. Neuro-Symbolic Program Search (NSPS) synthesizes end-to-end differentiable Neuro-Symbolic Programs (NSPs) by amalgamating neural-symbolic reasoning with representation learning. Symbolic representations of driving decisions are described with Domain-Specific Language (DSL) for autonomous driving. DSL contains both basic primitives for parts with driving, along with conditional statements to enforce higher-level priors. The design of DSL is allowed to specify all the behaviors for autonomous driving in a differentiable neural-symbolic behavior paradigm. Further NSPS is formulated as a stochastic optimization problem allowing to efficiently search for program architecture that integrates the neural-symbolic operations ensuring end-to-end learning possibilities. The NSPS in Figure 1 is combined with Generative Adversarial Imitation Learning (GAIL) to learn in an end-to-end fashion to generate neural-symbolic decision programs to output specific instructions (e.g., target waypoint index, target velocity) to the motion planner and controller.

Integrating Prior Knowledge into Deep Learning. In [170] a Semantic Based Regularization (SBR) [169] framework is used to express prior knowledge as set of First order Logic (FOL) clauses. SBR is a statistical relational learning framework, holding the ability to learn from examples and logic rules. Partial definition of the mapping from input to output is provided through expressed FOL clauses. Statistical relational learning is employed to inject logical knowledge into learning. It transforms logic knowledge into continuous constraints which are integrated with cost functions as a regularizer. Experimental analysis of the proposed work is focused on image classification problems. A subset of ImageNet dataset [624] is used for the classification task.
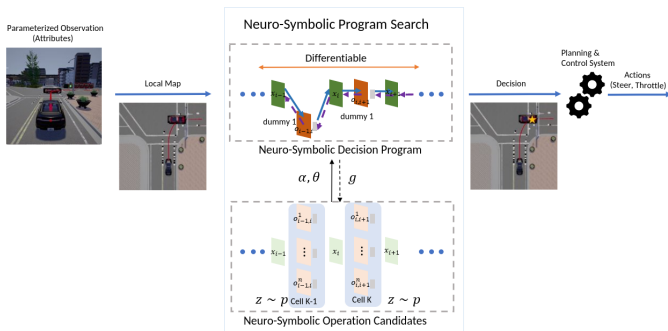
Fig. 1: The NSPS searches The NSDP, through back-propagation g to update α and β. The decision policy generated by NSDP is further used for planning and control. Semi-transparent operators shows the candidates for parent operations used in the search from NSPS [700].

4.2.2 Applications in Perception

Reasoning. A neural-symbolic reasoning engine could employ world knowledge or common sense knowledge to make sense of the scenery of a perception module or a combination of such modules. It could work as a regularizer (semantic loss function) during the training of a perception neural network which describes the scenery and penalizes implausible combinations of the recognized entities and their attributes or relations. Secondly, the same principle could be applicable during inference in which the perception module(s) outputs multiple possible scenery descriptions and the reasoning engine checks for contradictory elements in the scenery thus assessing the plausibility of the output.

Scene Consistency. As in [528], concepts in labeled driving data-sets along with domain experts can be utilized to model SceneOnotology representing spatial relational knowledge among the scene concepts. However, the ontology in [528] is used to generate triplets or facts exhibited in the scene. Vector representation of these facts is performed with Knowledge Graph Embeddings (KGEs). The KGE framework allows translating the triplets to latent vector space for vertical integration of spatial knowledge with deep learning methods for downstream scene understanding task. Qualitative intrinsic evaluation of KGEs for complete encoding of all knowledge facts is challenging and is subject to the different score functions utilized in KGEs [528]. Therefore, one could use SBR [170] a statistical relational learning framework that can represent these facts as FOL clauses and transforms these clauses into continuous constraints. These constraints could be integrated into the cost functions as a regularizer term, allowing to find optimal parameters for the learning algorithm by restricting the solution space. The downstream deep learning task such as object detection could use this relational knowledge expressed with SBR framework to improve the precision and accuracy by integrating the knowledge of the object's spatial-relational existence.

4.2.3 Applications in Situation Interpretation

Neural-symbolic architectures for context understanding. Because the application of Knowledge Graph Embeddings in [528] is already used in the autonomous driving context, it is applicable as it is. Discovering similar situations that are visually different helps to understand certain scenarios and could be useful as an input to help classifying dangerous situations.

The attention-based injection approach can be useful for reasoning in certain situations, for example by injecting the Straßenverkehrsordnung (StVO) into a QA network to encode new sentences in BERT.

Behavior prediction of target vehicle. As presented in [700], one could make use of DSL for describing driving maneuvers in symbolic space. DSL could be designed to contain StVO logical rules along with primitive attributes (e.g, velocities, accelerations, pose, associate lane type, lane attributes, road typologies) to enforce higher-level priors for maneuvers.

The NSPS framework proposed in [700] is a stochastic optimization problem. Joined with the mechanism of NAS it can be employed to generate decision policy for down-stream motion control and planning tasks. Similarly, Neural-Symbolic Behavior Programs (NSBPs) are generated using the NSPS framework to reason about the target vehicles' behaviors to support cooperative planning in the scene. The synthesized NSBP shall be an operation involving Neural-Symbolic operations (numerical operation and logical operations), rather than plain neural networks. The NSPS framework together with GAIL could be used to learn NSBPs in an end-to-end fashion.

4.2.4 Applications in Planning

Reasoning. The creation of formalized knowledge requires a methodology capable of validating the resulting formalization. One such method is querying the formalization against test cases, e.g., check if the current formalization of the StVO entails undesired properties such as it is possible to infer that endangering pedestrians in order to make way for an ambulance is ok. Those queries are also formalized statements and answered by a neural-symbolic reasoning engine that employs the formalized knowledge.

The formalization of legal knowledge is a prerequisite for checking compliance of an already taken or planned action for a certain traffic situation with regulations such as the StVO. Neural-symbolic reasoners could perform such compliance checks enhancing two applications in the autonomous driving domain: Firstly, a planner could use the compliance check to assess several courses of action. Secondly, a compliance check could be employed as a regularizer during the training phase of a planner, forcing the model to prefer legally compliant solutions over non-compliant solutions.

4.3 Attention Mechanism

Author: Tianming Qiu

Human beings can focus on a specific area in fields of view or recent memories to avoid over-consuming energies. Inspired from the visual attention of human beings, an algorithmic attention mechanism becomes a popular concept in deep

learning. NMT [34], a classical Natural Language Processing (NLP) task, is one of the earliest successful attempts which apply attention mechanism. Traditional NMT approaches are based on a sequential encoder-decoder architecture which uses RNN. The encoder maps source sentences word by word to hidden states and the decoder predicts target sentences. One of the drawbacks is that the longer the input sentence is, the more severe forgetting of previous words. The attention mechanism gives specific words (or tokens) more emphasis to avoid long distance forgettings. Similar to NLP's attention concept, many machine learning tasks also require efficient focus on specific data or information. Such specific focus comes from prior knowledge or experience which is very helpful for the objective task. Furthermore, this attentive information is usually intuitive for human understanding and it provides useful interpretability. For example, image captioning tasks look for heatmaps on input images which indicates where caption words refer to [805]. If the attention mechanism is considered as a form of human knowledge, learning such semantic knowledge is expected to benefit networks performance.

In Computer Vision (CV) tasks, the attention mechanisms are categorized into three different modeling approaches: *spatial attention*, *channel-wise attention*, and *self-attention.*

4.3.1 Spatial attention

Spatial attention attempts to imitate how human beings are attracted by significant objects or features visually. Technically, it emphasizes spatial areas in input images with highlighted heatmaps. The common spatial mechanism is written as

$$\alpha_i = f_{att}(\mathbf{v_i}),$$
$$\mathbf{v'_i} = \alpha_\mathbf{i} \odot \mathbf{v_i}, \tag{2}$$

where $\mathbf{v_i}$ represents a certain feature map of an input image, f_{att} is a nonlinear mapping and $\odot$ represents Hadamard product, namely an element-wise product. A tiny two- or three-layer neural network is used to describe a nonlinear mapping of f_{att}, whose parameters are updated during training. The mask assigns different weights on the original feature map $\mathbf{v_i}$ by using Hadamard product so that it emphasizes information beneficial for following classification tasks and weaken less important features. These weighted masks on feature maps are scaled up to the original input image size and visualized by heatmaps to illustrate semantic image pixel-level attention.

The key point is to learn a nice attention function f_{att} which generates a semantic attention heatmap. Such an attention heatmap is integrated again into the neural network for improving final performances and provides semantic meaningful visualizations. Similar to machine translation tasks, the attention on input original sentence words now switch to input image areas. Similar works are seen in HydraPlus-Net [448], which develops a complicated and huge neural network by duplicating Inception networks several times. HydraPlus-Net is designed for pedestrian re-identification so it should be capable to detect detailed features on pedestrians. All the above papers learn attention functions only by standard loss functions which only contain predicting losses for bounding box class and localization, but no predicting loss for attention heatmaps. They design

special structures but do not provide extra information for attention function training. The only 'guide' for attention learning comes from the loss functions. Another approach to learn attention is to add an extra auxiliary loss function specifically for f_{att} training [541, 855]. In object detection tasks, datasets provide segmentation ground truth which is used to evaluate attention as well. The loss function that measures overlaps between attention heatmap and ground-truth segmentation is used as a very strong guide to learn attention function [541]. Another approach that leverages additional information to train attention networks is to use pre-trained attention layers from other tasks. In a pedestrian detection task, such an additional dataset like MPII Pose Dataset [18] which provides precise predictions of 14 human body key points demonstrates a good attention result on the primitive task [855]. Spatial attention is seen as a special feature representation. It learns the spatial knowledge from input images that different spatial areas have different impacts on the final outputs of neural networks.

4.3.2 Channel-wise attention

In computer vision tasks, channel-wise attention weighs channels of convolutional layers' outputs differently. Similar to the aforementioned spatial attention, channel-wise attention is still a probability mask. It assigns various weight values for each channel of output separately with the supervision of classification or detection outputs. Convolution layers are considered to be able to show the hierarchical nature of features [838]. Each convolutional kernel is assumed to represent a different feature extraction ability. Hence, channels of the output feature map behave differently to various image patterns. Each channel may contain different features which might affect the final output. Channel-wise attention was first used to aggregate information from the entire receptive field for involving more global information than local spatial information [325]. In pedestrian detection tasks, [855] interprets CNN channel features of a pedestrian detector visually and indicates that different channels activate response for different body parts respectively. An attention mechanism across channels is employed to represent various body parts. By emphasizing detected human body parts, occluded pedestrian detection results are improved.

4.3.3 Self-attention

Self-attention is widely used in NLP because it is good at extracting the correlations between words. The relationship between each word plays a significant role of text understanding. Self-attention in CV analyses the correlations between pixels and are formulated as a formal function of query $\mathbf{q}$, value $\mathbf{k}$ and key $\mathbf{v}$:

$$\mathbb{R}^{d_k \times n_q} \times \mathbb{R}^{d_k \times n_k} \times \mathbb{R}^{d_v \times n_k} \to \mathbb{R}^{d_v \times n_q},$$
$$\mathbf{q}, \mathbf{k}, \mathbf{v} \mapsto \text{Attention}(\mathbf{q}, \mathbf{k}, \mathbf{v}). \tag{3}$$

Query $\mathbf{q}$, value $\mathbf{v}$ and key $\mathbf{k}$ concepts come from retrieval systems, where the best matched 'value' should be returned according to a certain 'query'. Usually, query is first converted to keys that are connected to values. Here query and key refer to the projected outputs of the decoder and encoder. Sometimes key and value are the same. Attention

computes for each query $\mathbf{q}$ an attention vector $\mathbf{a}_i$ by returning a weighted sum of all values, i.e.,

$$\mathbf{a}_i = \sum_{j=0}^{n_k} \alpha_{i,j} \mathbf{v_j}. \qquad (4)$$

The weights are determined from some measurements of similarity between the queries and keys. Transformer architecture uses word relevance to improve translation performance [740]. Self-attention represents the image block or pixel relevance in computer vision tasks. Apart from local features within each block, self-attention provides more global features [768, 585]. Alternatively, each pixel in an image is seen as the query $\mathbf{q}$. Self-attention of each query pixel is calculated on the other pixels in an image. Compared with convolution layers, self-attention is also able to extract different levels of features at different layers. Furthermore, due to its ability to extract global features, self-attention is able to achieve better performance than convolution in many tasks [129].

4.3.4 Applications

Attention mechanism in CV is widely used in autonomous driving perception tasks such as pedestrian detection. In Zhang's work [855], attention is integrated into the network to enhance the potential ability to find more occluded pedestrians. Similarly, integrating attention heatmap to the existing detector backbone improves the detection results as well [541]. Attention mechanism isn't used for planning directly, but it is used for interpretabilities of planning or decision making. Works [380, 379] from Berkeley Deep Drive use attention heatmap to explain why vehicle takes a certain controller behavior and textual explanations would be generated. Attention is updated during training, meanwhile, it also affects the training results in the end. For scene understanding, it is not considered as a feasible method.

4.4 Data Augmentation

Authors: Stefan Matthes, Tobias Latka

Data augmentation comprises a number of techniques that increase the amount of data for little additional cost. It provides a way to integrate knowledge about how concrete changes in the input signal affect the model's target output, such as invariance to small perturbations. Training with the additional data usually improves the generalization of the model and can be especially helpful when data is scarce or imbalanced.

Which technique can be used depends on the format of the input data (e.g., image, audio, point clouds) and the machine learning task. It is essential that the applied algorithm preserves task-relevant information. For example, color space distortions can be helpful in image-based license plate recognition (by making the model more robust to color changes), but can reduce performance in bird species classification, since color is an important distinguishing feature for many species. For some tasks, such as density estimation, it is inherently difficult to define appropriate data augmentations. On the other hand, data augmentation is even an integral part of some unsupervised models, for example in contrastive learning [113].

In the following, we discuss different approaches to data augmentation. Data transformation (or manipulation) techniques modify single instances, whereas in data synthesis parts of two or more instances are recombined. Furthermore, we distinguish between augmentations in data space and feature space and whether a trained model, such as a GAN or Auto Encoder (AE), is used to generate additional samples. Some authors do not consider simulation as data augmentation, but since simulation is a useful tool for knowledge integration and plays a crucial role in autonomous driving, we will discuss it here as well.

4.4.1 Data Transformation

Many classical approaches using DNNs add random noise to the training data [592][61]. Bishop [61] showed that applying small perturbations to the inputs during training leads to a smoothed target function and is equivalent to optimizing with an additional regularization term or constraining the weight updates (see also [592] and [523]). However, it is unknown what the optimal noise distribution is.

A related technique is random erasing, for example, graying out pixels [167] and dropping words in text [783]. This is similar to dropout [695] where instead network weights are masked with some probability in each optimization step.

Furthermore, the effect of many input-specific transformations is well studied. Typical image manipulations include geometric transformations, such as cropping, translations, rotations, reflections, and projections; kernel filters, e.g., sharpening and blurring; and color space transformations, such as random grayscale and color jitter [673][523]. While for image classification these transformations generally have no effect on the output, for object detection bounding boxes have to be modified equivariant to the geometric transformations used.

Audio datasets are often enhanced by using scale changes (pitch shifting and time stretching), compression, quantization, equalizing, filtering, reverberation and background noise injection [483]. Moreover, several of these elementary transformations can be combined in arbitrary order.

Adversarial examples are a special type of perturbation. These are slightly distorted inputs that lead to incorrect and usually overconfident predictions, but can often not be distinguished from the original by humans [254]. Adversarial training, i.e., feeding these examples back into the model, leads to more robust predictions [254]. Miyato et al. [492] extend this procedure to the semi-supervised setting by computing the adversarial examples using the model's predictions instead of ground truth labels.

4.4.2 Data Synthesis

There are many ways to synthesize new instances from existing data. One of the early approaches is SMOTE [106]. It was developed for imbalanced datasets and can be used to oversample underpopulated classes by interpolating between nearest neighbors from the same class. Mixup [849] and SamplePairing [341] explore the same technique for image data. The former also interpolates the labels accordingly and uses soft labels, which however cannot be used in the semi-supervised setting.

Another common technique for image datasets is to cut and paste patches from different images

[183][182][199][243][836]. To avoid that the model cheats by detecting artifacts at the boundary of the inserted patches, various blending techniques and distractors (patches that do not contain any of the relevant objects) can be used [183]. Instead of inserting objects randomly, several techniques were proposed for more realistic object placements, such as using a visual context model [182], depth and semantic information [243], and a heat map for appearance consistency [199]. YOLOv4 [65], a 2D object detector, additionally uses mosaic data augmentation that concatenates multiple images before cropping. This improves the detection of smaller objects.

4.4.3 Data Augmentation in Feature Space

The methods described so far directly modify the raw data, but it is also possible to augment data in the feature space. In the latter, the input data is fed through the first layers of the DNN and then the intermediate representations are manipulated before being passed through the remaining layers.

New instances can be synthesized either by interpolation, extrapolation or simply by adding noise [166]. Similar to Mixup [849], Manifold Mixup [743] additionally interpolates between points from different classes by also interpolating the labels accordingly, which can therefore be considered its natural extension. Alternatively, an AE can be used to transform the modified features back into the input space [166], but unlike the other approaches, this already requires a trained decoder. These methods have the advantage of being domain agnostic. However, experiments by Wong et al. [792] suggest that augmentations in the data space, when applicable, are preferable to data augmentation in feature space alone.

4.4.4 Data Generation

A more elaborate approach to create additional data is to first train a generative model with the given data and then sample from it. Neural Style Transfer [239][482] is a technique that can be used to change the appearance of an image while leaving the content unaffected. It has mainly artistic applications, but can also be used to render images with the appearance of different seasons, times of day, and different weather conditions [482]. A major drawback is that these models already require large amounts of training data and may take a long time to sample.

4.4.5 Automatic Data Augmentation

Towards automating the machine learning pipeline, Cubuk et al. [139] applied a reinforcement learning approach to search the space of augmentations. The learned policies specify the order and strength of predefined operations, including geometric transformations, photometric transformations, kernel filters, as well as Cutout [167] and SamplePairing [341]. Since then, several variations and extensions have been developed [673]. In contrast to the previous approaches, Benton et al. [54] directly optimize distributions over augmentations with respect to the training loss.

4.4.6 Simulation

Simulations provide another way to generate large amounts of data at low cost, which is especially useful for data hungry models like DNNs. They enable the generation of more data for interesting situations or rarely occurring events, which is often difficult or infeasible in the real world for financial or moral reasons. Additionally, they offer the possibility to evaluate safety-critical systems in specific test scenarios.

For the development of machine learning models, simulation results are predominantly used in the natural sciences, e.g., thermodynamics, material sciences, and autonomous driving [622]. There exists a plethora of open source and commercial simulators for the development and benchmarking of Self-Driving Vehicles (SVDs) [177][602][56]. A recent overview of their configuration options and available sensors can be found in [615]. Besides data generation, there are other ways to combine simulations and machine learning models. We refer the interested reader to a recent overview [621].

One of the biggest hurdles in transferring the trained models to the real world is the domain shift (distribution change) between simulation and reality. Even highly accurate models in simulation can perform poorly on real data if the data distributions differ too much. Therefore, the models often have to be additionally fine-tuned on real data. Apart from developing more accurate models, there are two approaches to bridge the gap, domain randomization and domain adaptation.

The idea of domain randomization is that given enough variability in the virtual environment, the model may interpret the real world as just another variation. For example, in the context of grasping experiments with a robotic arm, Tobin et al. [723] demonstrated that randomizing the rendering of images improves the transferability from simulation to hardware. Interestingly, they found that designing the simulation to be as realistic as possible was less effective than varying the styles.

Domain adaptation is a type of transfer learning that leverages labeled data in one or more related source domains for prediction in a target domain. Two recent surveys with applications in computer vision can be found in [138] and [759]. The latter has a stronger focus on deep learning models.

4.4.7 Structural Causal Models for Data Augmentation

Structural Causal Models (SCMs) encode knowledge about an environment [555]. In that respect, they can be thought of as the data generating process. Mathematically, an SCM is nothing else than a Directed Acyclic Graph (DAG) equipped with both a set of functions and a distribution on the DAG's root vertices. While the DAG's vertices correspond to the variables of the environment, its directed edges represent independent causal mechanisms between variables. In particular, the causal mechanisms describe how variables affect one another in a deterministic manner. Thus, every SCM naturally defines a joint distribution on its variables. Thereby, its shape is determined by the set of functions and by the distribution of the SCM's root variables. Moreover, distribution shifts can be modeled in the SCM-framework as interventions, for instance, exchanging one function for another one.

As discussed, for instance, in [640], SCMs are perfectly suited to generate valid and consistent samples at will in the sense that they are consistent with causal relations encoded in the SCM. In this way, SCMs serve as a kind of lightweight simulator of the underlying environment leading to different

interventional distributions depending on the concrete set of interventions. To be more precise, an arbitrary training set can be thought of as being composed of individual distributions. These individual distributions can either represent the distribution defined by the unmodified SCM or originate from different interventions applied to the original SCM at the time data is being generated. Hence, interventions effectively modify the environment and cover reasonable variations of the environment. In this way, sampling data from different joint and interventional distributions (as constructed from the original SCM) naturally increases the diversity of the overall training distribution and can be interpreted as some kind of data augmentation.

4.4.8 Applications

Data augmentation can be used in all stages of autonomous driving. Especially in stack-based architectures and end-to-end approaches that compute interpretable intermediate representations, data augmentation can be used in a variety of ways.

The early end-to-end approach from Bojarski et al. [67] learns steering commands directly from monocular image data and emulates the SVD at various displacements from the center of the lane and angles to the direction of the road. They extend the dataset by transforming the viewpoint of the images using two additional forward-facing side cameras and adjusting the steering angle accordingly. This results in a more robust driving model that is better able to recover from adverse situations. Photometric transformations, kernel filters, noise injection, and various other augmentation techniques that do not affect the control commands can also be applied at this stage [125].

Many of the techniques used in image-based object detection were already discussed above. For 3D object detection from point clouds, it is common to randomly shift, rotate, flip and scale each ground truth bounding box and its associated points [873][670][811]. Except for translations, these can also be applied to the point cloud as a whole. Yan et al. [811] additionally synthesize new point clouds by inserting points belonging to bounding boxes from different scans. Implausible outcomes are avoided by performing collision tests.

In multimodal object detection, additional care must be taken to ensure that augmentations do not cause inconsistencies between data streams such as pasting objects at implausible locations. By performing occlusion and collision tests the cut and paste augmentation can be extended to image data for multi-modal object detection [860].

Many recent approaches to object detection, semantic segmentation, and related tasks incorporate additional synthetic data from virtual environments, such as from the game Grand Theft Auto V [602][796] and the SYNTHIA dataset [613]. Integrating synthetic data generally leads to better performance, but the gains level off above a certain ratio. In addition, photorealism plays a smaller role than realistic modeling of sensor distortions and environmental distribution.

Several approaches advocate a bird's eye view, also known as plan view, as an intermediate representation for subsequent motion prediction, trajectory planning, and control [40][753][109]. A common approach is to render detected objects and information about the environment into a multi-channel image, which is then processed by a CNN. The bird's eye view image can be augmented with geometric transformations such as random translations and rotations [109].

As yet another example of data augmentation in the context of autonomous driving, imagine, for instance, an SCM that describes the vehicle trajectories (i.e., the physical laws connecting vehicle states and actions to new states). New trajectories of the vehicle's motion can be generated from existing ones by means of such a vehicle SCM following the subsequent steps. First, values of the external (and usually) unobserved random variables are inferred from existing trajectories (known as the abduction step) effectively reconstructing the situation the vehicle was in when the observed trajectory was recorded. Second, an intervention or sequence of interventions (actions) are applied to the vehicle SCM, while sustaining the updated distribution from the abduction step. Third, the (intervened upon) SCM predicts a new trajectory that is grounded in the observed trajectory. The procedure just described returns a so-called counterfactual trajectory that could have evolved, if another sequence of actions had been taken. In this sense, this technique transforms an observed trajectory into a counterfactual one, while complying with the rest of the SCM that was not intervened upon (data transformation). Thus, the technique allows to augment data in a way that it is still anchored in an observed trajectory, but can, at the same time, be used to generate more data, especially covering hazardous and underrepresented scenarios. Moreover, this technique was shown to be useful for explaining the causes of Machine Learning (ML)-model decisions, as discussed, for example, in [158] and thus assists in situation understanding.

Bansal et al. [40] increase the diversity of vehicle trajectories by adding random perturbations. The perturbations are chosen such that the vehicle is brought back to its original trajectory after a perturbation. However, too strong distortions degrade performance as the model learns bad behavior.

They also employ an augmentation technique called past motion dropout [40]. Because the model is provided with past ego-motion from expert demonstrations, it can learn to exploit cues in the motion history rather than learning the underlying causes of such behavior, such as stopping at a stop sign because it sees a deceleration. Randomly dropping the past motion forces the network to look for other signals to explain the future trajectory.

TrafficSim [702] learns to simulate multi-agent behaviors which can be used as effective data augmentation for training better motion planner. The learned driving model is fed into other vehicles in the simulation to produce more realistic behavior.

4.5 State Space Models

Author: Jörg Reichardt

Driving is an inherently sequential dynamic activity. Sensory information is acquired through a stream of observations that exhibits causal dependencies and correlations across a very wide range of time scales.

As a passive observer, our ability to understand any dynamic phenomenon depends on our ability to predict future observations $\mathbf{o}_{t..t+\Delta t}$ from past observations $\mathbf{o}_{0..t}$ *in probability* [663]. We stress *in probability* to highlight that possible inherent randomness of a dynamic process can only be understood and modeled in terms of distributions, not individual outcomes [137]. Our central object of interest is thus the distribution over future observations given past observations

$$p(\mathbf{o}_{t+\Delta t}|\mathbf{o}_{0..t}). \tag{5}$$

We denote all time points from start time 0 to time $t > 0$ with the indices $0..t$ and further assume $\Delta t > 0$. If we are taking an active role in the dynamics of the system, then future observations will naturally also depend on our past and future actions $\mathbf{a}_{0..t+\Delta t}$ and our central object of interest becomes

$$p(\mathbf{o}_{t+\Delta t}|\mathbf{a}_{0..t+\Delta t}, \mathbf{o}_{0..t}). \tag{6}$$

Equipped with this distribution, we are able to optimize/plan our actions to increase the chances of a desired future outcome and observation.

These two distributions are intimately related and State Space Models (SSMs) are the most commonly applied tool for their modeling. Firmly rooted in probability theory, state space models are uniquely suited for the study of the steady stream of information that originates from traffic phenomena. They permeate all aspects of driving from perception to situation understanding and planning [717].

In the context of knowledge integration, SSMs represent an *algorithmic prior* [355]. They provide a scaffold of probability distributions and their corresponding conditional independence structure. Data driven methods can then be used to learn parameterizations for these probability distributions. Ideally, optimization and learning can then be achieved end to end.

We will next review the fundamental terminology and assumptions of state space models that are central to their understanding in the modeling of dynamical systems and control. With this terminology clarified, we will then focus on the peculiarities of applying SSMs to traffic and autonomous driving and review the possibilities of creating hybrid learning systems within the framework provided by SSMs.

State space models introduce a latent dynamical random variable $\mathbf{x}_t$ called the system's *state*. The system's state governs the system dynamics, but is in general not directly accessible to an observer, i.e., state information has to be inferred. The system state has two defining qualities: First, it gives rise to the system dynamics via a Markov Process:

$$p(\mathbf{x}_{t+\Delta t}|\mathbf{a}_{0..t+\Delta t}, \mathbf{x}_{0..t}) = p(\mathbf{x}_{t+\Delta t}|\mathbf{a}_{t..t+\Delta t}, \mathbf{x}_t), \tag{7}$$

i.e., only the most recent state and future actions matter for the future evolution of the system. The relation $p(\mathbf{x}_{t+\Delta t}|\mathbf{a}_{t..t+\Delta t}, \mathbf{x}_t)$ is causal and is called the *motion model* of the system. All prior knowledge about the system dynamics may enter in this model.

Second, the current state and only the current state gives rise to observations via measurements:

$$p(\mathbf{o}_t|\mathbf{x}_t). \tag{8}$$

This relation is causal [553] and is called *observation model* or *measurement model*. Measurements do not change the state.
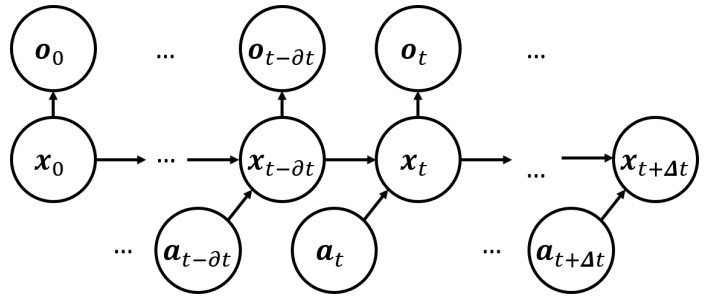

Fig. 2: Conditional independence relations represented as graphical model

Measurements can only reduce our uncertainty about the state through Bayes' theorem in which the observation model plays the role of the *observation likelihood*. All knowledge about the measurement process such as measurement noise or sensor transfer functions are captured in the observation model.

From these two qualities, it follows that the state renders past and future observations and actions conditionally independent through d-separation [553]:

$$p(\mathbf{o}_{t+\Delta t}|\mathbf{a}_{0..t+\Delta t}, \mathbf{x}_t, \mathbf{o}_{0..t}) = p(\mathbf{o}_{t+\Delta t}|\mathbf{a}_{t..t+\Delta t}, \mathbf{x}_t). \tag{9}$$

This means the state provides as much information about future observations as all past observations and past actions combined. If we know the state, we can forget about all past observations and actions taken. Note that the state is not meant to provide a good reconstruction of past observations and actions - it only extracts the information from past observations that is necessary to predict future observations. The state is all we need to make optimal predictions about the future and to plan our actions [663]. Figure 2 illustrates the conditional independence relations discussed above in the form of a graphical model.

With the state having such formidable qualities, the central object of interest for state space models becomes the *posterior state density*

$$p(\mathbf{x}_t|\mathbf{a}_{0..t}, \mathbf{o}_{0..t}), \tag{10}$$

which can be obtained through the application of Bayes' Theorem in a recursive manner from earlier state density, as given in Eq. (11), where we have used that only the state $\mathbf{x}_t$ can give rise to observations at time t and the defining quantity of the state as the sole generator of system dynamics.

The quantity

$$p(\mathbf{x}_{t+\Delta t}|\mathbf{o}_{0..t}) = \int d\mathbf{x}_t p(\mathbf{x}_{t+\Delta t}|\mathbf{a}_{t..t+\Delta t}, \mathbf{x}_t) p(\mathbf{x}_t|\mathbf{a}_{0..t}, \mathbf{o}_{0..t}), \tag{12}$$

is called the *predictive state density* and we note the role of the motion model in this expression.

The continued update of a state estimate over time is called *tracking* or *Bayesian Filtering*. In the construction of the posterior state distribution the observation model plays the role of the likelihood term and the predictive state distribution that of a prior. The *evidence* term $p(\mathbf{o}_{0..t+\Delta t})$ acts as a normalizing factor for the posterior state density. The Bayes Filter is a generative model for the observations.

$$p(\mathbf{x}_{t+\Delta t}|\mathbf{a}_{0..t+\Delta t}, \mathbf{o}_{0..t+\Delta t}) = \frac{p(\mathbf{o}_{t+\Delta t}|\mathbf{x}_{t+\Delta t})}{p(\mathbf{o}_{0..t+\Delta t})} \int d\mathbf{x}_t p(\mathbf{x}_{t+\Delta t}|\mathbf{a}_{t..t+\Delta t}, \mathbf{x}_t) p(\mathbf{x}_t|\mathbf{a}_{0..t}, \mathbf{o}_{0..t}) \tag{11}$$

It is this sequential process of absorption of current evidence into a latent state representation via a sound probabilistic framework that makes state space models extremely appealing from a conceptual point of view [412, 717]. We will next discuss under what conditions this conceptual appeal can be translated into computationally efficient algorithms and at what point approximations are introduced to obtain computational efficiency.

The first condition is that $p(\mathbf{x}_t|\mathbf{a}_{0..t}, \mathbf{o}_{0..t})$ can be normalized, i.e., we are able to evaluate $p(\mathbf{o}_{0..t})$. For all but the simplest kinds of observation spaces, this is hopeless to do exactly. One option is to resort to a sample / histogram based approach [354]. Alternatively, one can make use of an *assumed density* for $p(\mathbf{x}_t|\mathbf{o}_{o..t})$, i.e., via a parametric distribution for which the normalization constant can be computed from a set of sufficient statistics. The most common example of such an assumed density is the multivariate Normal. This also fixes the state representation $\mathbf{x}_t \in \mathbb{R}^n$ for which we have made no restrictions so far. The canonical example of such a *state vector* is the kinematic state vector describing an object's position, velocity and acceleration [717]. The second condition is that we can perform the integration necessary to produce the predictive state distribution. Variational methods [531] and Monte Carlo methods are a viable option here in particular if they can be executed on highly parallel hardware [355]. If an assumption was already made on the form of the posterior state distribution, then it is natural to make the same assumption on the predictive state distribution. It then depends on the form of the motion model $p(\mathbf{x}_{t+\Delta t}|\mathbf{a}_{t..t+\Delta t}, \mathbf{x}_t)$ whether this is exact. If the motion model is also a multivariate Normal with $\mathbf{x}_{t+\Delta t}$ being a linear function of $\mathbf{x}_t$ and $\mathbf{a}_{t..t+\Delta t}$, then no approximation is introduced at all. Under assumed distributions for the state, it is further desirable that the observation model $p(\mathbf{o}_{t+\Delta t}|\mathbf{x}_{t+\Delta t})$ be conjugate to the predictive state distribution. Then condition, the posterior can be updated in closed form [235, 258].

Depending on the type of approximations used, the update equations for the posterior state density are commonly referred to as *Kalman Filter* for $p(\mathbf{x}_t|\mathbf{o}_{0..t}) \sim \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_t, \boldsymbol{\Sigma}_t)$ with linear Gaussian motion $p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t) \sim \mathcal{N}(\mathbf{x}_{t+\Delta t}; \hat{\mathbf{x}}_{t+\Delta t} = \mathbf{F}(\Delta t)\mathbf{x}_t, \mathbf{Q})$ and observation models $p(\mathbf{o}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{o}_t; \hat{\mathbf{o}}_t = \mathbf{H}\mathbf{x}_t, \mathbf{R})$ [360]. The co-variance matrices $\mathbf{Q}$ and $\mathbf{R}$ herein model the so-called *process noise* and *observation noise*, respectively. They have to account for errors both introduced by the assumption of linearity for dynamics and observation process as well as for actual noise and measurement uncertainty and can be estimated and tuned from data [117, 2]. If these modeling assumptions are fulfilled, the Kalman filter update equations provide an exact closed form solution to the state estimation problem. In case the means of motion model and/or observation model are non-linear functions $\hat{\mathbf{x}}_{t+\Delta t} = \mathbf{F}(\mathbf{x}_t, \Delta t), \mathbf{o}_t = \mathbf{H}(\mathbf{x}_t)$ one can linearize around the mean of $\mathbf{x}_t$ to obtain the *Extended Kalman Filter*. The *Unscented Kalman Filter* instead uses integration by Quadrature, i.e., nonlinear functions $\mathbf{F}$ and $\mathbf{H}$ are evaluated at at set of judiciously chosen *sigma points* and the results are combined into a weighted average to give $\hat{\mathbf{x}}_{t+\Delta t}$ and $\hat{\boldsymbol{\Sigma}}_{t+\Delta t}$ [751, 294]. Giving up on the assumption of a Gaussian state distribution, one models the state distribution as a population of sample/particles and perform prediction and update step in a Monte Carlo fashion. This approach is known as a *Particle Filter* [252].

The above relations and algorithms form the basis for much of the classic model based study of dynamical systems and signal processing. We will next discuss the requirements for their application in an autonomous driving application. We will discover a wide range of opportunities for the application of data driven learning algorithms while still following the general scaffold of Bayesian data assimilation described above.

Fundamental to the specific aspects of perception, situation interpretation and planning is the substrate on which they operate: the state space. Hence, it will be discussed first.

Traffic is a *multi agent* phenomenon [605]. Traffic participants are very diverse - we observe fast cars, slow cars, small cars, large trucks, cable-cars, motorcycles, bicyclists, pedestrians, roller skaters, scooter drivers and - depending on country - horses, cows or moose on the road. Traffic participants are not particles, but decision making goal driven agents that are bound to move according to the laws of physics as well as by the rules of the road - though the latter being obeyed to a lesser extent in general. Goals and intentions of traffic participants are not measurable. They are either actively signaled by an agent in an explicit manner, e.g., through turn signals, or must be inferred from the agents' dynamical behavior. Traffic participants interact with each other to avoid collisions and operate in a structured dynamic environment, i.e., they follow roads and lanes or sidewalks or react to stop lights.

Hence, a state variable must be able to represent a *varying number* of *diverse* traffic participants at any given moment in time. If the state is to be the sole generator of system dynamics, it must include components for signaled/inferred *goals and intentions* of traffic participants. It must further include components that model the *environment* to the extent it is necessary to make predictions about their future movements [377, 289, 462, 752]. The multi agent nature of traffic further requires that distributions over states are invariant under a permutation of objects/traffic participants [837].

In order to cope with these requirements, two classical approaches exist. Multi-Object tracking algorithms [464] abandon the representation of state as a single vector and instead use a random finite set of state vectors for individual objects [748, 258]. These representations are again fully probabilistic and even treat the number of objects in the set as a random variable. Alternatively, one can keep a fixed size state vector by rendering state information into a fixed size grid using the always available positional information of objects for positional encoding, i.e., to specify the grid cells [40, 161]. All available features are then stored in a dimension perpendicular to the grid. In essence, this amounts to saving

state information in a multi-channel image and unlocks the ability to use (convolutional) neural network architectures on the state variables at the expense of a vastly increased state space dimension.

4.5.1 Applications in Perception

The perception stage of an autonomous driving stack enters SSM through the generative observation model or observation likelihood $p(\mathbf{o}_t|\mathbf{x}_t)$. In principle, the observations $\mathbf{o}_t$ could be raw sensory input such as camera images or lidar points, but that is currently not computationally feasible without strong approximations [387]. One alternative is to drop the generative nature of the observation model and learn to estimate the posterior state density $p(x_t|\mathbf{a}_{0..t}, \mathbf{o}_{0..t})$ directly, this approach is know as a discriminative Filter [277]. Note, however, that this approach is practical only for fixed history lengths and thus sacrifices the ability of the standard formulation to represent correlations in time of arbitrary length.

The second alternative is to pre-process raw sensor readings via detection algorithms to yield observations that correspond to object-level data [832]. This corresponds to the "tracking from detection" paradigm. One can further differentiate if an object can give rise to at most one detection ("point detection algorithms") or multiple detections. If the latter is not an artifact, but a result of multiple sensor readings on an object's physical extension, so-called "extended object tracking" algorithms result [257, 802]. If detections are available from every available sensor modality, the observation likelihood is a natural place to perform sensor fusion. Alternatively, sensor fusion is performed prior to the application of a detection algorithm on the raw sensor level.

In correspondence to the set of objects we observe in traffic, detection algorithms will return a set of detections. Detection algorithms, however, are not perfect. There can be false detections, so called clutter, that do not arise from actual objects. There can also be objects that do not give rise to detections at the current moment due to occlusions or failures of the detection algorithm. Further, detection are not necessarily labeled, i.e., there is no known correspondence between a tracked object and its detection. From this arises the so called *data association problem* of multi object tracking: the need find this very correspondence of the elements of the set of observations with the elements of the set of objects tracked. Standard Algorithms exist to solve it [153, 488]. Once this correspondence is established and the posterior state estimates of the tracked objects have been updated, any errors made in updating the state vector of an object with the wrong detections cannot be recovered. In order to mitigate this problem, so-called multi-hypothesis tracking algorithms [593] maintain several plausible potential data association hypotheses until possible uncertainties in data association are resolved by additional evidence [235].

Two more aspects of SSMs pertain to the perception module that allow for knowledge integration. The first is the so-called *birth model*, i.e., the prior distribution for the state vector for new objects $p(\mathbf{x}_0)$ that is needed for the state estimation from an object's first detection via $p(\mathbf{o}_0|\mathbf{x}_0)$. The birth model can represent the sensitivities of the sensors as well as prior knowledge about where and how objects will enter the sensor range of an autonomous vehicle. The second

are the probabilities of detection $P_D(\mathbf{x})$, survival $P_S(\mathbf{x})$ and the clutter intensity $P_C(\mathbf{o})$ that can add further specify the observation model [235]. With $P_D(\mathbf{x})$ we are able to specify that an object, though present, currently cannot be detected, e.g. due to occlusions [257]. With $P_S(\mathbf{x})$ we can express our prior knowledge about how objects leave the sensor range. For example, we can forget about oncoming traffic immediately once it has left the sensor range, while objects that have left the sensor range in the forward direction have a much higher probability to be re-encountered. Finally, with $P_C(\mathbf{x})$ we can model prior knowledge about the accuracy of the detection algorithms used.

4.5.2 Applications in Situation Interpretation

In the previous section, we have decidedly spoken about objects in order to address the handling of both traffic participants and elements of the environment such as roads or intersections. It is part of the appeal of SSMs that they can be used to model both the moving traffic participants as well as the static environment as seen from a moving sensor.

Situation interpretation primarily consists of two key tasks, the problem of estimation and tracking of the *current* state of the system $\mathbf{x}_t$ from past observations $\mathbf{o}_{0..t}$ and actions $\mathbf{a}_{0..t}$, i.e., the state tracking and filtering problem. Specifically, this entails the mapping and localization problems, i.e., modeling the static environment from observations and referencing the ego vehicle and other traffic participants in this environment [752]. It is important to note that the state update equations are evaluated at the rate at which new observations are available, typically $\Delta t \leq 50$ms and thus the motion model is used on very short prediction horizons. Hence, one generally uses simple kinematic motion models [646]. Model uncertainty can then be adequately modeled as random noise and data can be used to tune the noise distribution [117, 2]. In particular, in multi object tracking, one may assume independence between the motion of individual agents and neglect the interactions with both the environment and other traffic participants. This simplification results in growing errors as the time frame without new observations, e.g., due to occlusion, grows.

The second task of situation interpretation is to extrapolate this estimate into the future and enable anticipatory planning which is essential for safe and comfortable driving. Now the situation becomes markedly different as we are predicting the evolution of the traffic state over time scales Δt typical for driving maneuvers, i.e., several seconds. Consider the situation illustrated in Figure 3 depicting two vehicles driving on a highway on-off-ramp.

Given only the observable kinematic information for the current point in time and the environmental structures, we see that we have two possible future trajectories for each vehicle. Assuming independence of future motion for individual vehicles i, i.e.

$$p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t) = \prod_i p(\mathbf{x}_{t+\Delta t}^i|\mathbf{x}_t^i), \qquad (13)$$

we would need to consider 4 different futures for the traffic scene, three of which contain potential conflicts that a planning algorithm may have to deal with as can be seen in Figure 3b. Now consider the same situation in which two different plausible past trajectories are given as shown in

(a) Two cars driving on a highway on-off ramp.

(b) Four scenarios of future trajectories.
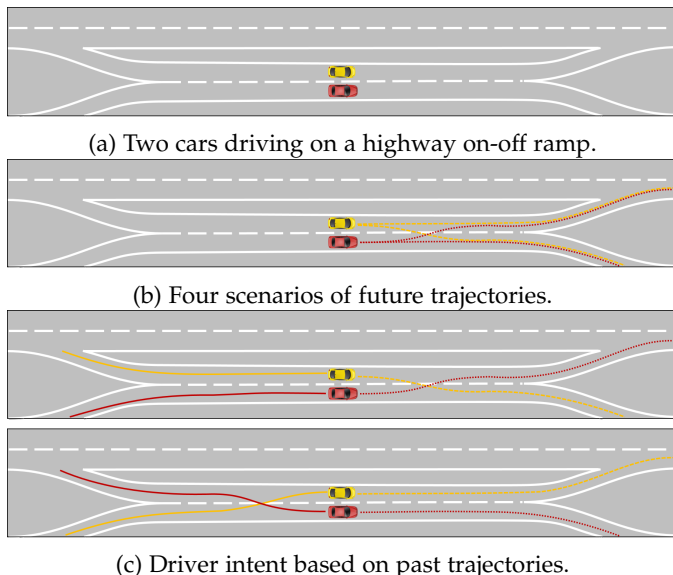

(c) Driver intent based on past trajectories.

Fig. 3: Illustration of potentially conflicting trajectory planning of two vehicles on a highway on-off ramp.

Figure 3c. These past trajectories provide information about the probable intent of the drivers. Now, there is only one plausible future trajectory for each vehicle and even the uncertainty with respect to the future evolution of the scene has been reduced. This underscores the necessity to model a traffic participant's intent.

Driver intent is often modeled as an unobservable discrete state variable indicating one of several possible maneuvers classes, such as lane change left, turn right, follow road [643, 642]. These classes must be mutually exclusive and collectively exhaustive and the concrete class has to be inferred from observations. Often, specialized motion models are associated with a maneuver class leading to so-called multiple model filters for such maneuvering targets [477].

The complete reduction of uncertainty about future trajectories is not always possible and more than one possible option for future trajectories has to be represented. This implies that $p(\mathbf{x}_{t+\Delta t}|\mathbf{a}_{t..t+\Delta t}, \mathbf{x}_t)$ should be multi modal both for the motion of individual traffic participants as well as for the entire set of traffic participants in a scene. Under the factorization assumption, this can lead to a combinatorial explosion of possible futures for the entire traffic scene including many future scenarios with conflicting trajectories. This problem can be dealt with by pruning the conflicting scenarios with corresponding computational expense [790]. More desirable would be a motion model for the entire traffic scene that produces conflict free scenarios from the start.

4.5.3 Applications in Planning

Planning happens in state space. Given the current state of the traffic situation $\mathbf{x}_t$ including the state vector of the ego vehicle $\mathbf{x}_t^e$, the predicted trajectories of all traffic participants and dynamic aspects of the environment, the planning algorithm must find a sequence of actions that bring it closer to its destination while maintaining safety and comfort. It must do so taking into account the uncertainty in the future behavior of the traffic participants [335].

For this, generally model predictive algorithms are employed that optimize the expected cost of a target function $C(\mathbf{x}_t, \mathbf{a}_t)$ over a constant planning horizon T under a set of feasibility constraints [509, 247]:

$$\underset{\mathbf{a}_{t..t+T}}{\operatorname{argmin}} \int_0^T C(\mathbf{x}_{t+\Delta t}, \mathbf{a}_{t+\Delta t}) p(\mathbf{x}_{t+\Delta t}|\mathbf{a}_{0..t}, \mathbf{o}_{0..t}) d\Delta t$$

$$\text{subject to } f_i(\mathbf{x}_{t..T}, \mathbf{a}_{t..T}) \geq 0 \ \ \forall i.$$

$$(14)$$

The feasibility constraints allow to include environmental and safety constraints. The optimization problem is discretized in time as a Sequential Quadratic Program (SQM) with initial condition set to the current kinematic state of the ego vehicle and continuity constraints between the individual stages of the SQM [66]. In order to perform the optimization, the expected kinematic state of every other traffic participant has to be known at every stage of the optimization. Hence, it is required that the future trajectories of other traffic participants $p(\mathbf{x}_{t..t+\Delta t}|\mathbf{x}_t)$ can be evaluated efficiently. Ideally, this prediction model is *interaction aware*, i.e., it takes the influence of the ego-vehicle's actions on the expected behavior of other traffic participants into account. This is a difficult problem, especially if the motion model is non-linear and thus the long term evolution is likely very susceptible to uncertainties in the initial state estimate. A possible remedy is to learn a state space representation and observation model under the constraint of a *linear* motion model [779, 369]. This will shift complexity and computational expense to the observation model which may however be less critical as no temporal extrapolation is needed in the observation model. Once an optimal sequence of controls for the ego vehicle over the entire planning horizon is found, the ego vehicle applies only the first step of this sequence and the process repeats with new observations, an updated state estimate of the traffic scene, updated predictions of future trajectories.

4.6 Reinforcement Learning

Authors: Stefan Rudolph, Daniel Bogdoll, Tim Joseph

Reinforcement Learning (RL) is a set of techniques where agents optimize their behavior given a reward signal over a period of time. A detailed introduction to the field is given in [705]. Here, we stick to a brief description of the so-called RL problem, which is depicted in Figure 4. An *agent* interacts with an *environment* by executing an *action* in each time step. It decides which action to choose based on the current *state* and its estimated evaluation from past experiences. This mapping from the state to an action is called the *policy* of the agent. Subsequently, the agent receives a *reward* in each time step which reflects the notion of a local evaluation of the agents actions. But, often, this immediate reward alone is not sufficient to judge how good an action is since a larger reward will only be given after a beneficial sequence of actions, i.e., the agent faces a sequential decision making problem. For instance, the agent moves in the right direction over multiple time steps to reach a defined goal. That is why RL algorithms commonly aim to find a policy that maximizes the expected cumulative reward instead of the immediate reward. Over the last years, deep RL has become the dominant form of RL, where deep learning is used to realize an RL agent. An introduction to these modern approaches is given in [5]. They
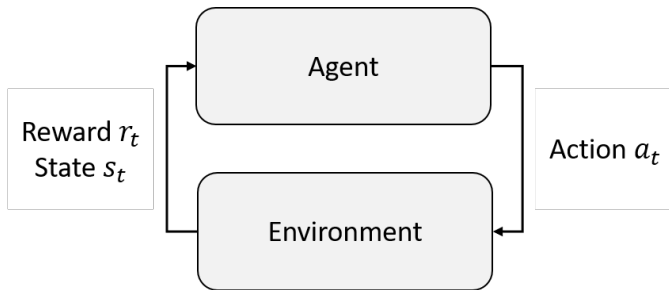
Fig. 4: The basic reinforcement learning setting.

can be roughly divided in two categories, namely model-free and model-based algorithms. The model-based algorithms make use of an explicit model of the environment which is either given beforehand or learned from experience. Model-free algorithms on the other hand do not use such a model and always act directly in the environment. An overview regarding current approaches is given in Section 4.6.1 and Section 4.6.2. Another common classification is the distinction between on-policy and off-policy algorithms. The former can only improve the value estimation for the policy the agent is currently carrying out. In contrast, the latter can improve the estimation of the value of the best policy independently from the actions taken by the agent. In the following, we provide an introduction to recent developments in the area of multi-agent reinforcement learning, where multiple agents operate in the same environment and might interfere (c.f. Section 4.6.3). Therefore, the agents have to take each other's actions into account. This type of reinforcement learning is especially interesting since we face a multi-agent system in the autonomous driving domain. Another current approach is the idea of inverse RL, where one aims to learn a reward function given examples of interactions with the environments, briefly outlined in Section 4.6.4. Finally, we give a brief overview regarding the recent works on the integration of knowledge into RL algorithms (c.f. Section 4.6.5) and the current state of the art regarding RL in the automotive domain (c.f. Section 4.6.6).

4.6.1 Model-free Reinforcement Learning

Model-free Reinforcement Learning (MFRL) algorithms learn a policy directly from real experiences in an environment without the need for a model. MFRL can be divided into methods that derive a policy from state-value estimates and methods that directly optimize a policy (policy gradient-based methods). Q-learning [778] has been one of the most popular methods based on state-(action)-value estimates. While early work in the tabular setting and with linear function approximation provided proofs on different convergence properties [729, 35] and showed some first results [715], more recent work showed the potential to solve complex tasks when Q-learning is combined with deep neural networks, called Deep Q Learning (DQN) [493]. Various improvements have followed [636, 776, 46], including extending DQN for partially observable Markov decision processes [292], mitigating value overestimation [290] and a combination of all previously mentioned improvements called Rainbow-DQN [309].

In contrast to Q-learning, policy gradient-based methods directly maximize the expected future sum of rewards [705]. A major advantage is the ability to learn in environments with continuous action spaces, something that is not possible with standard Q-learning. An early policy gradient method is the REINFORCE algorithm [789]. It uses the policy gradient in its most basic form. However, when the score function estimator is used as in REINFORCE the policy gradient is known to be of high variance and thus, learning is slow. Sutton et al. [703] show that the variance can be reduced when a baseline function (e.g., an advantage function) is incorporated into training. Another approach is to use Deterministic Policy Gradient (DPG) [681, 434] which are advantageous in environments with a high number of action dimensions. Another class of on-policy policy gradient algorithms constrains the policy change at each update step to allow for multiple updates with the same batch of data (note that the policy gradient is only valid in expectation with respect to data collected by the current policy). The most prominent members of this class are Trust Region Policy Optimization (TRPO) [648] and Proximal Policy Optimization (PPO) [647]. In contrast, a recent state-of-the-art off-policy algorithm is Soft Actor Critic (SAC) [278]. SAC adds an entropy bonus to the policy learning to enable better exploration and uses the re-parameterization trick [598, 383] instead of the score function estimator which makes it more stable than methods based on DPG. One of the major disadvantages of pure model-free reinforcement learning agents is that they often need a very high number of environment steps (e.g., a common setting for ATARI [47] based benchmarks is 200 millions steps) to converge to a good policy. Furthermore, the black box characteristic of most existing MFRL algorithms makes it hard to analyze and interpret actions or future behavior. Thus, training a MFRL in the real-world is prohibitively expensive.

4.6.2 Model-based Reinforcement Learning

Model-based Reinforcement Learning (MBRL) algorithms make use of an existing or learned model of the world to either provide imagined experiences [704] to train a policy, to provide better gradients for policy training [300] or to plan at inference time. The two main challenges of MBRL are to learn a model (if none exists already) and to use it effectively. A problem with learning a good model is model bias, i.e., that the policy will exploit regions in the model that deviate from the real environment. One approach that considers this problem is PILCO [157]. The model is implemented as a Gaussian process and used to roll out imagined trajectories from which a policy can be learned with analytic gradients. Gal, McAllister, and Rasmussen [228] improve PILCO with an ensemble of deep neural networks instead of Gaussian processes. Both avoid model bias by not just training the policy with a single dynamics model, but with a distribution of possible models. Similar approaches have also been used in further works in combination with TRPO [648] instead of back propagation through the transition dynamics [404], to learn a meta-policy [123] or with terminal Q-functions [121] for better long-term learning. Model bias is mostly addressed when an agent acts in an environment with a low-dimensional observation space.

For high-dimensional observations, such as images from an RGB camera, the focus shifts to higher-capacity models and efficiently predicting future rewards. Wahlström, Schön, and Deisenroth [750] learn a deep dynamics model that consists of an auto-encoder and a latent dynamics model. They then plan with this model with model predictive control. PlaNet [281] learns a sequential latent variable model to account for the stochasticity of the environment and uses Model-predictive Control (MPC) to efficiently plan in latent space. Ke et al. [373] uses a similar approach, but explicitly enforces correct long-term predictions through an auxiliary loss that enforces latent variables to be informative about the future. Ha and Schmidhuber [276] train a policy purely on imagined experiences in a low-dimensional latent space that are generated with a learned world model. Dreamer [280] makes use of the differentiable model of PlaNet [281] to learn a policy by back propagating through the latent transition dynamics.

Finally, there are approaches that use an existing model, most famously AlphaGo [683] and AlphaZero [682]. Both assume that a model that can be queried efficiently for a lot of trajectories does exist. It is then rolled out with Monte-Carlo Tree Search (MCTS) and a policy and value function are learned. However, recently Schrittwieser et al. [644] have presented a variant of AlphaZero called MuZero, that uses a learned model and is able to match or outperform AlphaZero.

4.6.3 Multi-agent Reinforcement Learning

In Multi-Agent Reinforcement Learning (MARL), the basic idea of the interaction of an agent with an environment is extended to a setting where multiple agents interact with the environment and each other at the same time. A full review of current methods and a taxonomy of the algorithms can be found in [308]. Here, we focus on some recent main achievements regarding the application of RL to multi-agent systems, as well as some extensions of standard RL algorithms to the multi-agent case.

The authors in [745] present a method to train an agent capable of playing competitively with top human players in real-time strategy games. Such games can be categorized as two-player zero-sum games and a special emphasis regarding the training of the agent must be put on the fact that an agent should be robust against a variety of counter strategies. To stimulate the learning of such a behavior the authors introduced a so-called league training with the main idea to extend fictitious self-play with three types of special agents (cf. Figure 5). The first type is named *main agents* and uses prioritized fictitious play meaning it selects opponents based on the win rate against the agent. The second is the *main exploiter* type which is playing against current *main agents* only to find weaknesses in their behavior. The third type are the *league exploiter agents* which use a similar strategy as the *main agents* but cannot be targeted by the *main exploiter agents*. Therefore, they have the opportunity to find strategies to exploit the entire league.

A similar challenging environment has been approached in [530]. There, an agent for a Multiplayer Online Battle Arena (MOBA) game has been trained which shows super human performance on a slightly less complex version of the game that, e.g., limits the number of available champions. MOBA games are interesting in the context of MARL because this
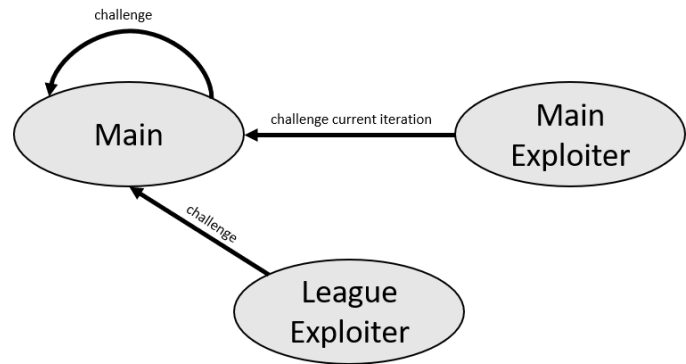

Fig. 5: League training as proposed in [745].

type of game is played five on five, meaning that there are five cooperative agents which face five competitive agents. In this work, the authors used PPO as a basis. As in single-agent reinforcement learning, the agents start the training with a local reward only based on their own benefit from certain actions because a global reward introduced too much variance to the reward function. Later in the training, their mechanism *team spirit* shifts gradually from the local reward to the global reward to encourage the agents to play as a team.

The authors in [823] improved on the results in the MOBA domain by introducing curriculum self-play learning. Here, the limitation regarding the number of champions has been weakened by training individual agents for small sets of champions and later merge them with a multi-teacher policy distillation. Furthermore, they used an off-policy variant of PPO called Dual-clip PPO.

The authors in [343] trained a team of agents for a capture the flag game. Each game consisted of two agents who use the same interface as humans, i.e., a RGB image as input and produces control actions while the in-game statistics are used as reward signal. To address the special circumstances a hierarchical learning mechanism has been installed that on the one hand uses an actor-critic algorithm for the individual agents and on the other hand an evolutionary algorithm that optimizes the reward function of the agents based on the available game points.

Based on MCTS, there is a multi-agent extension that has been applied to a simple grid world, where each agent has to learn to move to one of the defined goals but each tile can only be used by one agent [843]. The method uses MCTS with default and random policies for the rollout and combines it with difference evaluation in the reward function.

4.6.4 Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) is the process of learning a reward function from data-based observations. Arora and Doshi [24] provide a recent survey on IRL, motivating IRL based on its potential to model the performance and preferences of others. They address two core challenges: "Finding a reward function that best explains observations is essentially ill-posed" and "computational costs of solving the problem tend to grow disproportionately with the size of the problem", which is especially relevant in the complex domain of autonomous driving, since existing methods

"do not scale reasonably to beyond a few dozens of states or more than ten possible actions". They cluster existing methods based on four categories. *Max margin methods* try to "maximize the margin between value of observed behavior and the hypothesis", while *max entropy methods* are designed to "maximize the entropy of the distribution over behaviors". *Bayesian learning methods* "learn posterior over hypothesis space using Bayes' rule" and *Classification and regression methods* "learn a prediction model that imitates observed behavior". Additionally, there are many extensions to IRL, which Arora and Doshi again cluster in three categories: "Methods for incomplete and noisy observations, multiple tasks, and incomplete model parameters".

4.6.5 Reinforcement Learning and Knowledge Integration

In the following, we shortly summarize which techniques of knowledge integration have been identified.

Reward Shaping: The most common form of knowledge integration is the shaping of the reward [511, 614]. The idea is to design the reward function in a way that makes it easier for the agent to find an optimal policy, while still optimizing the original target in the limit. This can be especially useful in situations with long time horizons and sparse reward signals [530].

Models: A common way to integrate prior knowledge in an RL algorithm is to use some sort of model of the environment. This method has defined the area of MBRL in the first place. While the trend goes towards models that are learned by the agent during runtime, e.g., in [644], it has been showcased that human-designed models can allow to solve very complex tasks [683] and improve the learning speed by integrating knowledge, e.g., represented by a structural causal model (cf. Section 8.2) [84], in the learning system.

Learning by Demonstration: The idea of learning by demonstration (or apprenticeship learning) is around for some time [635]. It defines a paradigm where humans give a demonstration of a desired behavior of a learning system to speed up the learning process. One common approach to it is to use IRL [1]. In others, there is already a reward signal available [530, 745]

Auxiliary Tasks: A method to integrate prior knowledge into neural networks are auxiliary tasks. The main idea is to share one network over several tasks that force it to create structures that are beneficial for the main task. It has been used with actor-critique methods in a 3D labyrinth environment [344] and a multi-agent capture the flag game [343], for instance.

4.6.6 Applications

Kiran et al. [385] provide a broad overview of deep reinforcement learning within the context of autonomous driving. They see many tasks where RL could be utilized, including path planning, controller optimization and scenario-based policy learning. They provide an overview of common simulation environments and a detailed overview about the topics motion planning and inverse reinforcement learning for behavior cloning of experts. Since bridging the gap from simulation to reality is hard, they discuss many real world challenges, including validation, sample efficiency, and exploration issues.

Since RL is often utilized to train agents end-to-end, perception tasks on an explicit level have not been within the scope of RL in the past. To tackle this issue, a recent method called *Latent Deep Reinforcement Learning* [111] utilizes the latent space to not only create control commands, but also map the sensor input, namely RGB camera data and a Bird's-Eye-View (BEV) lidar pointcloud, to a semantic mask of the environment, including the map and surrounding objects. This way, the model does provide an interpretable environment model, which is a common output of pure perception modules.

Krasowski, Wang, and Althoff, as part of a *Safe Reinforcement Learning* framework [396], predict the occupancies of other traffic participants on a highway scenario. Their predictions are part of a safety layer within their RL framework, which they utilize to only allow for safe actions during the exploration phase. Since these occupancy predictions stem from an external algorithm [391], RL is not utilized for situation interpretation, but situation interpretation is embedded within RL. Further approaches on safe reinforcement learning can be found in [236].

Ye et al. [824] provide an overview of recent methods on RL-based planning methods. They separate methods into end-to-end systems, based on sensor data as input, and motion planning modules as a follow-up module of a perception stage. The type of available actions ranges from strategic maneuvers over lane changes and trajectories to direct control. The utilized algorithms vary widely, including classical RL, DQN, Deep Deterministic Policy Gradient (DDPG) and Asynchronous Advantage Actor-Critic (A3C).

4.7 Deep-Learning with Prior Knowledge Maps

Authors: Evaristus Fuh Chuo, Han Chen, Hendrik Stapelbroek

Object detection and recognition problems are often approached with deep-learning methods. They yet remain a great challenge in aspect of model accuracy, especially for certain circumstances, i.e., objects are occluded, too far away from the sensors or in bad light conditions. Challenges can also be found in improving data efficiency, especially when the data capacity is low. Finding a way to extract and combine information becomes important.

4.7.1 Semantic Segmentation

One possible way to address these issues is to incorporate prior knowledge into data-driven models. In [22], Ardeshir et al. introduced a method of combining RGB image with information extracted from Geographical Information System (GIS) system. The RGB images are firstly segmented as the initial status. GIS databases can provide interesting semantics in 3D coordinates, which are projected onto the 2D plane of the camera. The projected geo-semantics are weighed in terms of projection accuracy before used in improving the image segmentation and projection accuracy in an iterative manner.

In [645], a prior fusion network is introduced, which leverages information from revisited locations during previous traversals to improve image semantic segmentation. A scene prior is defined as images frames that have a high degree of visual coherence to the current scene. In the paper, Schroeder and Alahi experimented with 3 different network

architectures, two of them have access to prior knowledge, the other one is an eight-layer convolutional network that serves as the baseline model. The difference between the two prior networks lies in the integration of the scene prior, one lets the prior images pass through an encoder along with the current frame, while the other one through a decoder. The results suggest that the decoder network which enables the prior to be fused at various resolutions outperforms the others and the improving of the model performance is even more significant for dynamic objects such as pedestrians, cars, cyclists, etc.

4.7.2 Applications

In [812], Yang et al. introduced the HDNET for 3D object detection with High-Definition (HD) map integration. In this approach they exploit the semantic road mask available on HD maps as prior knowledge about the scene and convert road layout information into a binary channel in discretized lidar representation. They also conduct map estimation from a single lidar revolution for where a HD map is not directly available and apply data dropout on the semantic prior, which randomly feeds an empty road mask to the network, to get also good results when HD maps are unavailable or noisy. Another work that aims at improving 3D object detection with HD map integration introduced a more generalized framework that can be used with other 3D object detection algorithms [200]. A 2D Feature Extractor module is implemented to extract high-level features from HD maps and the features are fused to the voxel representation of the input points. For evaluation, the framework was integrated with three major 3D object detection algorithms, SECOND [811], PointPillars [409] and CenterPoint [829] and a general improvement of the mean Average Precision (mAP) on public datasets can be seen for all models.

Instead of integrating HD maps, another approach [114] uses ground plane as prior, as the detected objects should appear on the ground. A set of 3D sampled bounding box candidates on the ground are projected and classified in the image space.

5 KNOWLEDGE TRANSFER

In the previous chapter, knowledge integration has been predominantly addressed by means of proper modeling, e.g., modifications of the architecture and cost function design. This chapter examines the task from a more algorithmic perspective, leading to learning strategies capable of transferring implicit knowledge, often captured and represented by the weights or representations of a neural network, to some target domain of interest. One of the primary goals here is to learn reliable models given only a small amount of data from the target domain. Fine-tuning of selected layers of a pre-trained network can be attributed to *transfer learning* as presented in Section 5.1. While this concept focuses on adaptation to a single target task, in *continual learning* the aim is to steadily adapt to various consecutive targets, without forgetting previous tasks. An overview is given in Section 5.2. A similar idea is pursued in *meta learning* where experience from multiple tasks is gathered in an episodic training process in order to improve the learning strategy with regard

to upcoming tasks. Section 5.3 gives an introduction to this topic. Finally, knowledge transfer is considered form the perspective of purposeful selection and re-labeling of data. Current paradigms of this *active learning* strategy are outlined in Section 5.4.

5.1 Transfer Learning

Authors: Florian Wasserrab, Niklas Keil

DL has gained much momentum in recent years and is now the de facto standard approach in many applications. The use cases range from speech recognition to image processing or AD. Despite the great success, DL algorithms share the serious drawback that they rely on large amounts of well-annotated data to achieve their performance. The collection of such a dataset becomes very expensive and time-consuming in reality. In addition to the procurement of the datasets, the processing of such volumes of data is also a major challenge, as it requires large amounts of computing power and electricity. Transfer learning is a technique that tries to tackle these issues by reducing the amount of training data while maintaining or even improving the performance. Therefore, it has become an integral part of many deep learning models. In simple terms, this technique takes knowledge in the form of intermediate representations learned from previous tasks and applies it to novel tasks [710]. A great advantage of this approach is the possibility for any developer to make use of parameters optimized by data centers or supercomputers of big companies like Google (GoogLeNet) without needing resources for it themselves [707]. As a contemporary example, [337] implemented a lung cancer detection model based on GoogLeNet and showed that other state-of-the-art methods are outperformed on the same dataset while only requiring a fraction of training iterations.

According to a survey [540], transfer learning involves the concept of a domain and a task. A domain $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$ over the feature space, where $X = x_1, x_2, ..., x_n \in \mathcal{X}$. In this context, X is a particular dataset and x_i represents the ith feature vector. If two domains differ, they either have different feature spaces or different marginal distributions. Based on the domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a task is described by a label space $\mathcal{Y}$ and a predictive function $f(\cdot)$ that is also be written as conditional probability function $P(Y|X)$. The target function is not observed directly but can be learned from the data consisting of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathcal{Y}$. Based on a source domain $\mathcal{D}_S$ and a corresponding source task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, the objective of transfer learning is to improve the predictive function $f(\cdot)$ in $\mathcal{D}_T$ with the knowledge gained from $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{T}_S \neq \mathcal{T}_T$ or $\mathcal{D}_S \neq \mathcal{D}_T$.

5.1.1 Network-based Transfer Learning

The authors of [710] classified transfer learning in the context of DL into four different categories: instances-based, mapping-based, adversarial-based and network-based deep transfer learning. The first approach selects specific data points from the source domain and adds them to the target domain adjusted with a weight. The second method combines two domains into one data space, where instances

are similar to each other. The third approach aims to find representations that apply to both the source domain and the target domain by using adversarial technology. Finally, network-based deep transfer learning takes a model which was pre-trained in the source domain and re-use it in the target domain. In the following, we focus on the latter approach, which is currently the most commonly used way to perform transfer learning.

The intuition behind network-based transfer learning stems from the nature of neural networks. Typically, first few layers capture low-level features such as edges in an image, while higher layers focus on more complex details. Since low-level features of a pre-trained model are not domain-specific, we simply use these features for a different task and train a new model on them [710]. In practice, this is done by replacing the last layers of the pre-trained network with new ones. We then fix all base layers to ensure that we do not unlearn previously acquired knowledge. In this way, only the new classifier is initially trained for the intended task. Depending on how different the new use case is, the model may additionally benefit from further fine-tuning. This means instead of keeping the pre-trained parameters fixed, they are only used for initialization and included in the training process, such that they also get adjusted to the new problem. However, to ensure that the learned low-level features are still kept, the learning rate for fine-tuning is usually set rather low [142]. In the last years, transfer learning has been particularly popular for computer vision tasks. As one of the first implementations, [532] transferred parameters taken from a classification model pre-trained on ImageNet to other visual tasks with limited data and demonstrated significant improvements over previous results. Nowadays, popular CV models like ResNet [298], VGG [687], Inception [707] or DenseNet [330] are typically pre-trained on a large image dataset to produce universal image features which serve as foundation for the specific task. For example, in [698] the authors used pre-trained models of different domains as foundation for further fine-tuning on historical documents.

5.1.2 Applications

Transfer learning does not refer to a specific application, but is understood as a general approach that optimizes the learning process of complex DL models. This often involves computer vision tasks, such as image classification [532] or object detection [250], which are also applied in the field of autonomous driving. For instance, [317] showed that using a pre-trained CNN significantly improves the task of detecting pedestrians. They applied a large network trained for the ImageNet classification task and fine-tuned it for the pedestrian detection use case. In this context, they demonstrated that using another model pre-trained for a different task, i.e., scene recognition on the Places dataset [871], led to similarly performance. From this they concluded that the specific pre-training task was not crucial, since pre-training is mainly about learning basic universal features. The authors of [667] went a step further by using a pre-trained model from a completely different domain. In order to develop a framework for fault diagnosis of mechanical machines, they first transferred low-level features from a model pre-trained on ImageNet. Then, they converted machine sensor data to images using time-frequency imaging and performed their

target task on it. Using this approach, they were able to improve accuracy and reduce the training time, even though completely different domains were combined. Of course, the idea of transfer learning is not limited to pedestrian detection or sensor data evaluation, but can be applied to various use cases in the area of AD, e.g., road lane detection [883].

In many DL scenarios, collecting data in the real world is expensive and time-consuming. For such use cases, we use simulations to acquire knowledge about the real world by pre-training our model on it. Especially in large-scale projects, such as the development of AD technology, simulations play an essential role, as it massively reduces the training time. For example, Waymo [780] stated that one day of simulation is comparable to 100 years of real-world driving experience. Moreover, with simulations, we focus on the most interesting situations in AD, making the model able to handle rare edge cases more easily. For example, [12] pre-trained their model with data from CARLA [177], an open-source simulator for AD research. In addition to the CNN model, their architecture also consists of an LSTM model. This allowed them to consider temporal information, such as the speed of the vehicle, instead of focusing only on spatial features.

The planning stage of AD deals with finding the optimal trajectory that the ego vehicle follows in certain situations. One of the most relevant examples are lane changing and overtaking scenarios where, among other parts of the environment, traffic of the current and the adjacent lanes must be considered to find potential time slots and trajectories to pass a vehicle or obstacle in front while satisfying kinematic and safety constraints [172]. Most of the recently proposed ways for controlling an AD vehicle use methods like potential fields, cell decomposition, optimal control or model predictive control [536]. As these strategies are real-time optimization problems, i.e., they do not fall in the category of deep learning, the above mentioned network-based deep transfer learning is not applicable here. However, besides them, there are also learning based methods for controlling AD vehicles that incorporate previously learned knowledge. These are mainly (deep) reinforcement learning or (semi-)supervised learning approaches for trajectory planning for which literature shows that transfer learning could indeed be utilized.

As an example, [630] used transfer learning for real-time robot path planning, which is similar to the planning of AD vehicles. The robot recorded sequences of actions (called options) while passing different obstacles which were reused afterwards to learn suitable, collision-free trajectories in more complex situations in the future. The options were stored in a library $\mathcal{L}$ and depending on the fitness score or Jacard Index of the currently perceived obstacle and those of $\mathcal{L}$, a suitable option of $\mathcal{L}$ is retrieved. Subsequently, it is transformed to best align with the new obstacle and finally applied to the current environment. Also for Unmanned Aerial Vehicles (UAV) as provider for emergency communication services, the trajectory design problem was modeled as a deep reinforcement learning process with transfer learning to benefit from previously learned knowledge [861]. In case the base station of a communication system fails (for instance, due to a natural catastrophe), an UAV then served as aerial communication station that needs to adjust its position to dramatic changes of geography, surviving user

distribution and demand. Even though the domain of both applications is different, such approaches also serve as basis for the development of AD technology. In the area of semi-supervised learning, [408] showed that scaling up the pre-training by enabling the usage of unlabeled data considerably outperformed supervised models on state-of-the-art benchmarks. Due to the significantly increased number of images in pre-training, the models benefit even more from transfer learning effects.

Summarizing, for autonomous driving, one of the major present challenges remains to close the simulation-reality gap, where transfer learning plays an important role [385]. As it is the case for nearly all AI problems that rely on data, transfer learning definitely helped solving many AD use cases. However, it should be mentioned that transfer learning can only be applied to trajectory planning, if the vehicle controlling strategy actually learns from data.

5.2 Continual Learning

Authors: Christian Wirth, Stefan Rudolph

Continual learning aims at learning a model for multiple tasks sequentially without performance degradation on preceding tasks. It is closely related to transfer learning, as it is also applicable to reuse already trained models as a starting point for learning a new task. However, in contrast to transfer learning, continual learning aims to keep the important parts of the learned model intact. Therefore, continual learning does not use past task for (only) improving the results for new tasks, but for creating a unified model that for solving all tasks. Technically speaking, continual learning does not allow arbitrary changes to the model, but only such changes that will not introduce a degradation of the performance on past tasks. Therefore, the most important aspect is to identify which parts of the model are relevant for an observed outcome. Most of the continual learning methods work with the assumption, that it is relevant to determine the relevance of models parameters in regard to the prediction. For instance, Synaptic Intelligence (SI) [841] determines the importance of a parameter by its contribution to the loss. An alternative approach is approximating how much a weight can be changed while not reducing the performance on already observed tasks [386]. For these approaches, the scalar parameters are replaced by parametric probability distribution, e.g. a Gaussian, by directly learning the mean and variance of these distributions during training (see Section 8.1.1). It is then assumed, that it is possible to limit the acceptable parameter changes in terms of the Fisher matrix or the Kullback-Leibler (KL) divergence. Another, recent family of methods are the gradient based algorithms. They alleviate catastrophic forgetting during the training by directly manipulating the gradient for the weight update.

In this chapter, we are focusing on Bayesian and gradient based approaches. Bayesian approaches are interesting, due to the desired properties of Bayesian Neural Networks, as explained in Section 8.1. The gradient-based approaches have shown good results in the past years and especially they promise to allow for positive backward transfer, meaning the accuracy on past tasks can be improved while training new tasks.

Known types of Continual Learning are [323]:

- Incremental domain learning: The input and the output space are assumed to be identical but the data distribution differs.
- Incremental class learning: The input space is identical, but the output space differs. Each task covers a subset of classes from the same multi-class set. The data distribution may also differ.
- Incremental task learning: The input space is identical, but the output space differs. Each task covers a distinct set of different prediction targets. The data distribution may also differ.

5.2.1 Bayesian Continual Learning

Nearly all Bayesian Continual Learning approaches follow the same principle:

1) Learn parametric weight distributions on task N
2) Use (parts of) the weight distributions as a preconditioner
3) Add a regularization term for training on task N+1, that penalizes the distance to the preconditioner

Some approaches [722, 539] do not operate in the weight space, but in the function space. However, the differences revolve more around the structure of the (neural network) model and that will be covered in Section 5.2.9. We categorize Bayesian Continual Learning algorithms by their following properties:

- Approximation: Which approximation of the weight distribution is used?
- Preconditioner: How is the approximation captured?
- Regularizer: How are the parametric distributions regularized against the preconditioner?
- Update Mode: Which information from seen tasks has to be preserved?

Furthermore, some approaches do not work on a purely incremental scheme, where explicit information over all history tasks is captured in the last model. These methods merge the task specific models [386, 185], requiring an additional computation step. On the other side, some approaches work fully online, meaning it is possible to discard information from all but the latest model. Additionally, some approaches use a replay buffer [722, 539, 512], meaning parts of the training data are preserved and reused. For an method overview, see Table 1.

5.2.2 Gradient-based Methods for Continual Learning

The gradient-based approaches are a recent development in the continual learning domain. The main idea is to:

1) Save a small set of samples from previous tasks in a memory (similar to a replay buffer)
2) Constraint the gradient calculated for the samples of the current task by the gradient wrt. the replay buffer samples.

This avoids that the results on the previous tasks, which are represented by the replay buffer, degenerates since the loss on samples does not increase.

Such a method has first been proposed in [454] and is called Gradient Episodic Memory (GEM). There, the main idea is to keep the angle between gradient of the samples

TABLE 1: Overview of Bayesian Continual Learning Methods

Method	Year	Space	Preconditioner	Regularizer	Approximation	Replay Buffer (Selector)	Mode
EWC [386]	2017	Weight	Fisher Matrix	L2	mean-field (MF) Laplace	-	Incremental
EWC [607]	2018	Weight	Fisher Matrix	L2	KF Laplace	-	Incremental
IMM [414]	2017	Weight	Fisher Matrix	L2, Custom	MF Laplace	-	Incremental
VCL [512, 706]	2018/19	Weight	Mean & Scale	KL	MF VI	(K-means)	Online
GVCL [453]	2020	Weight	Mean & Scale	Tempered KL	MF VI	-	Incremental
UCL [539]	2020	Weight	Variance	LR	MF VI	-	Online
CBLN [426]	2020	Weight	Mean & Scale	(Expectation)	MF VI	(GMM)	Incremental
BGD [842]	2018	Weight	Mean & Scale	KL	MF Approx.	-	Online
FRCL [722]	2020	Projected Fn.	Cholesky	KL	FC Approx./VI	Reconstruction Error	Incremental
FROMP [539]	2020	Full Fn.	Mean & Scale	Expectation	MF Laplace	Maximal Uncertainty	Incremental

TABLE 2: Overview of Gradient-based Continual Learning Methods

Method	Year	Space	Constraint	Replay Buffer (Selector)	Update Mode
GEM [454]	2017	Gradient	Gradient angle	Random	Incremental
A-GEM [105]	2019	Gradient	Gradient angle	Random	Incremental
MER [604]	2019	Loss	Regularize on gradient angle	Random	Online
MEGA [271]	2020	Gradient	Adaptive loss constraint	Random	Incremental
Adam-NSCL [764]	2021	Projected Gradient	Project update in null space	-	Incremental
GPM [629]	2021	Projected Gradient	Orthogonal task updates	-	Incremental

from the memory and the gradient of the currently new sample below 90 degree to avoid the adaption of the weights in a direction that will likely make the accuracy on a previous task worse. The authors formulate an optimization problem that is solved during the training phase and finds the gradient that does not violate the constrains regarding the memory and is closest to the original gradient of the current samples.

Since the runtime to solve the optimization problem is considerably long Averaged Gradient Episodic Memory (A-GEM) [105] has been introduced. It provides similarly good results with a lower runtime by using only a randomly sampled subset instead of all samples of the memory to constrain the gradient. Furthermore, they make use of task descriptors that allow for improvements regarding zero-shot learning techniques.

In [271], a unified view on the approaches GEM and A-GEM has been found and two new methods based on a Mixed Stochastic Gradient (MEGA) have been introduced. The generalization uses a reformulation of the problem to reveal the fact that it is possible to weight between the current and past tasks via a parameter. Based on this knowledge the new methods utilize information from the loss to balance between these tasks automatically by setting the parameters appropriately, i.e., they put emphasis on the current task if the loss is high and emphasis on the past tasks if the loss is low.

Recently, Gradient Projection Memory (GPM) has been introduced which uses gradient subspaces that are only calculated once after each task with a singular value decomposition and a basis is stored [629]. For the current task, the gradient is then projected in a space that is orthogonal to this projected subspace. A property of this algorithm is that it does not require to hold samples from past tasks in the memory but only the basis of the subspace. This method reduces the required memory and improves privacy. In [764], another idea based on singular value decomposition called Adam-NSCL has been introduced. Here, the authors calculate the feature covariance of past tasks and approximate the null space of it. Subsequently, they calculate a gradient that lies in this null space. Similar to GPM the samples for each tasks do not have to be stored after one task has been learned. See Table 2 for an overview.

5.2.3 Update Mode

Most continual learning algorithms work in an incremental update mode. That is, each new model will be suitable for predicting all already-seen tasks and are regularized against all preconditioners or replay buffer samples from all those older tasks. Hence, these algorithms require to store the preconditioner and replay buffer for all tasks, which is a memory overhead. Different to that, online approaches use only the solution from the last task. A special case is the Progress & Compress algorithm [651], which works by training each task independently (possible initialized by the last solution), but computes a joined model by regularizing against the preconditioners in an online fashion.

5.2.4 Replay Buffer

Some algorithms reuse data from old tasks as a replay buffer. This means, a subset of the training data is used as "keypoints", where it is assumed that retraining (or constraining) with these will ensure the performance on already seen task. Therefore, all data needs to be stored, but only small parts are used for continual trainings. The methods for selecting datapoints for replay try to obtain elements, that are interesting, either in terms of representativeness by using cluster centers [512, 706], or by determining their importance for the prediction quality, in terms of a (model) reconstruction error [722] or uncertainty [539]. Gradient-based continuous learning methods usually simply use a randomly sampled subset of all training data [454, 105].

5.2.5 Parametric Approximation

Bayesian Continual Learning methods usually require to approximate the posterior parameter distribution of the model. As explained in in the Subspace and Parametric Approximation parts of Section 8.1.1, the approximations differ in expressiveness and runtime requirements. In the

following, we will only present a short recap. Please refer to the according chapter for more details.

The most common approximation is the mean-field (MF) approximation which only covers the mean and variance of each parameter distribution independently. A full-covariance (FC) approximation also captures the layer-wise covariances, whereas the Kronecker Factorization (KF) is an approximation of the full covariance matrix. For learning the parametric distributions, usually Variational Inference (VI) (c.f. Section 8.1.3) is used, which learns the mean and (co-)variance by gradient decent. The Laplace approximation can be computed after learning a deterministic model, but is computationally more costly and does not scale well. Only few methods use method specific approaches.

5.2.6 Preconditioner

The preconditioner determines the form in which the information captured by the parametric approximation is stored. This implicitly defines the regularization basis for learning additional tasks, as the regularizer works by penalizing the distance between the preconditioner and the current model in the same space. Therefore, the regularizer employs a given distance function.

A common preconditioner is the Fisher matrix [386, 651, 607, 414], which is an approximation of the second order derivative of the loss near a minimum [545]. Gradients are sufficient for its computation, but the method induces some computational overhead. However, this preconditioner is usually paired with a Laplace approximation of the posterior distribution, which requires an approximation of the Hessian, which is computationally substantially more costly. An alternative is directly using the mean and variance of posterior weight distribution, which can be learned "online" with variational inference. However, in this case common L2 regularizers are ususally not applicable, as described in the regularizer section. A substantial limitation of nearly all used preconditioners is the disregard of the covariance, with the exception of FRCL [722], which uses the lower triangular cholesky matrix. However, this limitation is not induced due to issues with the preconditioner or the reguarlization, but by the scalablility of the parametric approximation.

5.2.7 Regularizer

As explained before, the basic idea is to ensure that parameter values relevant for already observed tasks stay the same. Therefore, it is important to consider which information is contained in the used preconditioner. The Fisher information matrix is a measure of the information contained in a parameter, given a loss function (and examples). Hence, large changes of the values in the Fisher matrix also mean large changes in the information contained within a parameter and therefore it is sufficient to place an l2 regularization on these values. An alternative is the Kullback-Leibler (KL) divergence over the posterior distribution. If the variance of the preconditioner is small, relevant changes to the mean will result in large KL values. However, when changing the mean for values with large variance, the KL loss will stay small. Hence, parameters where it was possible to determine precise values (low variance) stay the same.

5.2.8 Constraints

Regarding the gradient-based methods, we see different possibilities to formulate concrete constraints that are used to avoid an update that makes the network perform worse. Early algorithms [454, 105] simply limit the actually used gradient to one that has a smaller angle than 90 degrees to the gradient from the old tasks, or regularize based on the difference [604]. Modern approaches [764, 629], project the gradients into a subspace that fulfills certain conditions, like orthogonality to previous task(s).

5.2.9 Function Space Regularization

Most continual learning approaches regularize in the weight space, assuming this is sufficient for preserving performance on old tasks. However, function space regularization [722, 706] tries to preserve the function (output) values for old tasks. This is computational infeasible to do directly, as it would require storing (and reusing) all training data. Nevertheless, approximations are possible with the use of kernel neural networks, as explained in Section 8.1.2, since these approaches capture the full output distribution in the model parameters of the kernel. However, using a full kernel matrix is usually computationally too expensive and therefore variational distributions at inducing points are often used. With knowledge over the kernel matrix and possibly the variational distribution from past tasks, it is possible to compute the difference of the output distribution between the model of the new and the old model(s) at a given set of points. This will require to store (a subset) of the training data from past tasks, as described in the replay buffer section.

5.2.10 Applications

Most aforementioned applications of continual learning focus on image classification tasks, like rotated/split MNIST, belonging to the domain of incremental class learning (c.f. Section 5.2). With regard to AD, this bears relation to adding new classes to object detection tasks. Hence, continual learning allows to increase the number of available classes later on.

Furthermore, the incremental domain learning setting is of interested in the area of AD. It allows for incremental learning, in the sense that it is possible to improve already learned predictors with newly available data, without training from scratch. Although, with regard to knowledge integration, we use incremental domain learning for integrating multiple knowledge sources into a joined model. The knowledge sources can relate to physical knowledge, expert knowledge, rules of the road or observed examples. As example, it is possible to train a model that predicts all locations, where it is physical possible that a vehicle can occur or drive to and then use continuous learning approaches for learning with examples where vehicles have been observed. The joined model can then correctly predict all viable positions (with low probability) for vehicles, while still emphasizing positions where this is most likely, based on the observed examples.

5.3 Meta Learning

Authors: Julian Wörmann, Alexander Sagel, Hao Shen

One factor of intelligent behavior is the capability to effortlessly transfer skill and knowledge from one task to a related

one. While humans generally exhibit this ability, e.g., humans can derive concepts about certain object classes from only a few representative examples, data-driven AI approaches usually require a vast amount of samples in order to achieve the intended task. Meta-learning, also referred to as learning-to-learn [718], addresses this issue via training a model that improves its learning performance after encountering multiple tasks. In this context, recognizing images of the same class, or predicting the trajectory of a robot in a specific environment can be considered a task. The goal of meta learning is to update the training algorithm in such a way, that the model easily adapts to new tasks, e.g., inferring the class label from only a few unseen instances in a few-shot scenario or planning the trajectory in an environment with modified conditions. Eventually, this approach allows the algorithm to re-use some structure that is shared across different tasks in order to improve its performance on a similar task, which can hardly be solved by learning entirely from scratch. Guiding the training phase via meta-learning can thus be considered as implicit integration of knowledge into data-driven models. The preparation and organization of the data, as well as the inclusion of the meta-objective is key in order to transfer existing architectures into knowledgeable systems.

5.3.1 Concepts and Algorithms

According to Hosepdales et al. [318], meta-learning can be viewed from several perspectives, namely task oriented, from an optimization point of view, or model-centric. While the task oriented view allows to motivate the overall goal, i.e., to gather knowledge about how to learn a model that generalizes well to unseen tasks, the optimization perspective enables to formalize this concept as an explicit bi-level optimization problem. In general, two intertwined objectives are pursued. Base learning aims at optimizing a model that is capable in solving a base problem, e.g., achieving high classification accuracy. In the associated meta-learning stage, the meta algorithm updates the base algorithm, i.e., the algorithm used to learn the base model. Through updating the learning strategy of the base algorithm, the base model improves an outer or meta-criterion, e.g., fast learning due to effective initialization. The progress is generally measured by means of the outcome of learning episodes. An episode represents the current model as well as its performance on instances from the training tasks. After updating the base algorithm, the performance of the resulting model is assessed again. Via iterating this scheme, the training algorithm itself 'learns how to learn' such that, e.g., generalization to new classes can be achieved from only a few samples. The model-centric view reveals the close relation to general representation learning approaches. Meta-learning an embedding function represented as a feed-forward model is one of the cornerstones to provide solutions which quickly adapt to new tasks. A decisive advantage of these so called *amortized* models is that they can be readily applied to new tasks in form of a simple feedforward pass.

One of the most popular realizations of the meta-learning paradigm is presented by Finn et al. [210], called Model-Agnostic Meta-Learning (MAML). Their framework focuses on acquiring meta-knowledge about how to learn the parameters of a model, e.g., a neural network, such that the learned model can be fine-tuned to a new task in only a few gradient steps with a small amount of training data. To put things more formally, denoting T_i as a task sampled from a distribution of tasks $p(T)$, f_θ the sought model parameterized by θ and $L_T(\cdot)$ the loss according to task T, the base objective is to update the parameters, such that the loss on task T_i is improved. Applying gradient descent, this step reads

$$\theta^\star = \theta - \alpha \nabla L_{T_i}(f_\theta), \tag{15}$$

with α denoting the step size. The outer meta-objective consists in finding the optimal parameters θ across several tasks drawn from $p(T)$, i.e.

$$\min_\theta \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta - \alpha \nabla L_{T_i}(f_\theta)}). \tag{16}$$

Via optimizing the outer criterion (16), the parameters θ are forced to a representation where, according to (15), only one gradient step improves model performance regarding the individual tasks T_i. In this way, a suitable initialization of the weights is provided that allows for fast adaptation.

The gradient based MAML framework has been applied in various domains, e.g., fast adaption to environmental changes in a trajectory estimation problem tackled via reinforcement learning [122], or acquiring meta-knowledge from clustered data in an unsupervised learning setting [322]. Furthermore, extensions exists with regard to Bayesian Meta-Learning [831], which aims at rapid estimation of an approximate posterior that allows to infer uncertainty, as well as Probabilistic Meta-Learning [211], that combines MAML with structured variational inference in order to enable simple and effective sampling of models for new tasks.

Meta-learning or meta-criterion design can also be used to learn other design parameters of the optimization process, e.g., an optimal step size [20, 432], the weighting of regularizers [216] or a suitable update direction [584], leading to an approach that is agnostic of the choice of base optimizer.

Model-centric approaches often involve meta-learning a mapping realized as a feed-forward network, i.e., $\eta = g(S)$, where S denotes the input data, and the function $g(\cdot)$ is approximated by means of a neural network architecture [691]. This approach is also closely related to metric learning, where the high-dimensional input S is mapped to a low-dimensional, even scalar representation η such that the similarity between two instances can be easily compared. Another interesting field of application is to meta-learn the loss used in the base algorithm by means of a simple neural network [256]. In this way, non-differential objectives like, e.g., Area under the ROC curve, or F1 measure, can be approximated and thus, the base model can be tuned towards certain metrics. Meta-knowledge can further be gathered about architectures, where for instance [442] propose a gradient-descent based method that efficiently searches for suitable convolutional or recurrent network architectures. Other approaches in turn focus on curriculum learning [352], which aims at finding a sample strategy such that the meta-learned models require less resources (training data or iterations) and generalize better to unseen data.

Many meta-learning approaches, however, require the meta-objective to be differentiable. The authors in [318]

denote another dimension of meta-learning as the choice of optimization strategy. Besides the presented gradient descent scheme, especially reinforcement learning or closely related evolutionary algorithms are appropriate methods. The design of the overall goal pursued in meta-learning, e.g., generalization across many tasks vs. efficient optimization in single-task scenarios can be considered another axis.

In the remainder of this section, applications of meta-learning to perception, situation interpretation and planning are discussed, while the range of applications goes even beyond, covering, e.g., architecture search, transfer from simulation to real environment, or language processing.

5.3.2 Applications

One of the main application domains of meta-learning is the problem of few-shot learning, which is of high practical relevance in many real world computer vision settings. The main goal of few-shot learning is to learn reliable models from only limited amount of data such that the model generalizes well to unseen or underrepresented instances. Meta-learning is often used to acquire class-agnostic features that contain information shared by many classes. Adaptation is then achieved via modulating these features based on class-specific information. Among the few-shot learning approaches, one-shot or few-shot classification [746, 388] aim to learn a class from only one or a few labeled instances. Few-shot meta-learning usually involves a set of N support and query pairs that constitute a base set, i.e., $\mathcal{B} = \{(\mathbf{S}, \mathbf{Q})_i\}_{i=1}^{N}$, where $\mathbf{S}$ and $\mathbf{Q}$ represent a set of images, respectively. Each of the N pairs can be considered a task, where meta-training consists in updating the model given $\mathbf{S}_i$. The performance on the task is measured by means of $\mathbf{Q}_i$. While updating the model after encountering several tasks, the learning algorithm gathers knowledge about how to learn such that during meta-testing, the same learning algorithm can be used to train a model on the support set of a new, unseen pair $(\mathbf{S}_{test}, \mathbf{Q}_{test})$. The knowledge about how to learn allows the algorithm to learn a model that generalizes to the samples in $\mathbf{Q}_{test}$, even when the number of samples in $\mathbf{S}_{test}$ is small.

As another branch, few-shot *detection* has emerged recently, focusing on recognition and localization of multiple objects in images from only a few annotated samples [769, 198]. Rather than missing class labels, few-shot detection handles cases where only a few bounding box annotations are available. Analogously to the standard object detectors outlined in Section 2.1, also their meta-counterparts can be divided into one-stage and two-stage approaches. Kang et. al propose Meta-YOLO [365] that combines feature map learning with a reweighting module. In the first stage, generalizable meta features are learned from base classes, that allow for simple adaptation to novel classes. The subsequent reweighting scheme transfers class specific feature information from few support images to the meta features of the query image via modulating the feature coefficients. Together with the detection prediction module that outputs class labels and bounding boxes, the whole detector can be trained end-to-end. Another approach is presented in [561], that augments the one-stage CentreNet object detection model with a meta-learned class-specific code generator in order to achieve incremental few-shot detection. A meta variant of the two-stage R-CNN based object detector can be found in [810].

Given the features generated by the original Faster/Masked R-CNN implementation, meta learning is applied to ROI features where background and potential objects are already separated. Similar to [365], feature reweighting in form of class specific channel-wise attention vectors is performed on the ROI features.

Object detection usually requires the definition of anchor points or identification of auspicious regions (via Region Proposal Networks). In order to allow for an end-to-end learning without the need for an intermediate region proposal step, Zhang et al. [848] recently introduced Meta-DETR, which combines meta learning for few-shot object detection with the transformer structure presented in [93].

In order to increase the generalization capabilities of data driven classification or detection models, data augmentation has been used as pre-processing step to improve generalization capabilites (see also Section 4.4). Meta-learning has also been used to automate the augmentation process such that models can efficiently adapt to given tasks or datasets. One of the most famous approaches has been introduced by Cubuk et al. [139]. The overall goal consists in searching for an augmentation policy that maximizes accuracy in the respective downstream task. For this purpose, different augmentation operations, e.g., translation, or rotation with varying parameters are applied to the images. The corresponding search algorithm is implemented as a reinforcement learning problem. The authors in [430] propose to relax the costly policy selection to a differentiable optimization problem with the aim to reduce the computationally complexity on the one hand, and to allow for a joint optimization of the network weights and the augmentation parameters on the other hand.

The concept of meta-learning offers useful advantages also in the context of RL, which has been applied for motion planning in dynamic environments. Especially, meta-knowledge about, e.g., goals, rewards or environments can be used in order to reduce the training time in terms of sample efficient exploration strategies [318].

Gupta et al. [272] propose an exploration strategy based on the MAML approach, that is informed by prior knowledge due to a meta-learned policy and latent space. Quick adaptation of the RL agent to new real world tasks or environments via meta-learning is also investigated in [122]. RL combined with a meta-learned self-supervised interaction loss is proposed in [794]. By aligning the gradients of a supervised and self-supervised loss in the training phase, during inference the agent can use the gathered meta-knowledge about the imitated supervised loss for self-adaptive visual navigation.

Applications to trajectory planning scenarios are presented in [825], where the authors combine meta learning with RL to develop a strategy for decision making for lane change maneuvers on highways. Again, a variant of MAML is used to improve adaptation speed for new scenarios, e.g., traffic density, road geometries, or traffic environments.

5.4 Active Learning

Author: Tino Werner

AL is an iterative learning strategy that incorporates human/expert feedback. The idea is to query the labels of some yet unlabeled instances resp. to create new labeled instances and to update the model taking the new information into

account, alternatingly. The advantage of AL when querying labels is the reduction of the annotation cost while clever sampling of new data points enables to train the model with less data than with uninformed sampling. Due to these benefits and the opportunity to better estimate the model uncertainty (see also Section 8.1), AL is a promising strategy for integrating knowledge into the AI training process.

Querying labels requires the existence of a so-called unlabeled pool and is therefore known as pool-based sampling [420], so one selects an unlabeled instance or a batch of unlabeled instances and asks for its label. Another strategy is to generate new samples which is called selective sampling if such samples can be selected from an existing pool or stream of data so the learner can still decide whether to ask for the label or not without having wasted significant computational time. The latter method is also called membership query synthesis if artificial new instances are generated according to criteria listed below or by just combining valid features from an abstract feature space ([126, 19, 661], see also for example [152, 151]). The key question is which of the unlabeled samples have to be considered for annotation resp. how new samples can be generated since, evidently, random sampling is not efficient and leads to high annotation costs. There are different sampling paradigms like uncertainty/margin-based sampling, see [420, 37], i.e., sampling from regions where the model is uncertain which for classification model means to sample from near the decision boundaries or query-by-committee [662] where one invokes different models and samples from regions where the models disagree most. Another paradigm is density-based sampling, i.e., sampling from representative regions, emerging from the fact that uncertainty sampling is prone to select outliers. Other paradigms consider the informativity/impact of the samples which is quantified by criteria like expected model change, expected error reduction or variance reduction. Furthermore, active learning is not limited to querying the label of one single unlabeled instance per iteration but can also consider whole batches of unlabeled instances. This has been studied for example in [98, 347, 358, 806] who aim at finding the batch of the best instances according to the given paradigm but by simultaneously being aware to minimize the redundancy of the single instances in the batch. The number of samples in a batch does not have to be uniform over the iterations and may have to be appropriately chosen, see, e.g., [548]. AL can be used for knowledge integration in AI training by augmenting the training data (see also Section 4.4) with instances where high losses are suffered, where the model is uncertain or where an expected model change or variance reduction is possible in supervised tasks like object recognition, semantic segmentation, person re-identification or tracking. Furthermore, AL can even enter unsupervised tasks by considering for example the model's entropy. One may invoke AL in knowledge extraction or conformity checking tasks by sampling from regions where the model output diverges from existing knowledge. As for testing a particular trained model, active testing may be applied in order to reduce the annotation costs for the test data.

See [661] for an excellent overview of sampling paradigms for AL. See [595] for a recent overview of deep AL which is a combination of deep learning and AL in order to keep the high predictive quality of deep neural networks, including feature abstraction while reducing the labeling costs.

5.4.1 Different paradigms for active learning

An important result that connects the paradigm of AL with the available sampling budget is given in [279] who theoretically derive that the best AL strategy crucially depends on the available labeling budget. More precisely, for a high budget, one should query uncertain samples while for a low budget, one should focus on "typical" points where they define the typicality as the inverse of the average squared distance to the nearest neighbors. In order to encourage diversity, they first cluster the points and then sample typical points from clusters which not yet contain labeled points.

Uncertainty sampling: For uncertainty sampling, a lot of different criteria have been suggested in literature. The authors in [80] focus on the 1-vs-2-score, i.e., the difference of the class-probabilities for the best and second best class for a given sample, in the context of deep object detection, based on the fact that a small difference between the probability of the most resp. second most probable class indicates uncertainty, more precisely, the sample is located near the intersection of the decision boundaries. The authors in [180] propose DeepFool active learning (DFAL) where the selection of unlabeled data is performed by choosing those which are closest to their adversarial attacks (generated by DeepFool, see [499]) whose corresponding perturbation magnitudes can be seen as approximating the distance to the nearest decision boundary. A Bayesian approach is given by Bayesian active semi-supervised learning for Deep CNNs [617] that iteratively reduces the classification entropy on the unlabeled data which can be combined with MC Dropout [695, 225]. Other Bayesian approaches include Generative Adversarial Active Learning (GAAL) [876] where training instances are iteratively generated by a GAN and task-aware VAAL (variational adversarial active learning), [381] who combine the VAAL strategy from [688] with the loss prediction technique of [830] which predicts the target loss for the yet unlabeled instances. Ensemble methods for uncertainty estimation have been considered in [48] and [401]. In [367], specific uncertainty measures for object detection are proposed, namely localization tightness (how tight the bounding boxes are around the object) and localization stability (how stable are the boxes in the presence of input perturbation). Imbalanced data has been considered in [578] who propose to compute a weighted uncertainty to take the imbalance into account. The authors in [320] take a contrary position to the paradigm to sample from the vicinity of the decision boundaries by performing lowest likelihood sampling where the likelihood is approximated by a Gaussian Mixture Model (GMM) in order to discover yet unseen classes. In [162], an adaptive supervision framework for AL for object detection is proposed. The idea is to avoid querying strong labels, here bounding boxes, but weak labels in the sense of weak supervision in order to optimize the model. Concretely, they use center-clicking from [542] and derive an algorithm that decides whether one has to switch to querying strong labels. [304] propose DEAL (deep evidential AL) for image classification by replacing the softmax output of a CNN by a Dirichlet distribution that allows for uncertainty quantification. [50] consider

the label-dispersion, a measure to quantify how much the label assigned to a specific instance changes during NN training, i.e., over different epochs. A high label-dispersion indicates that there is a large uncertainty. Experiments show a considerable benefit of this method. The method from [8] can be interpreted as a narrower version of label-dispersion. They introduce the alamp uncertainty measure that is given by the difference of the 1-vs-2-scores for a given sample on the current and the previous iteration, divided by the sum of these two quantities.

Diversity/informativity sampling: Diversity of the samples is encouraged using clustering [514, 115] and sampling representative points from each cluster. Methods based on the KL-divergence or the cross-entropy have been suggested in [809, 184, 98, 548]. Adversarial active learning has been considered in [844] who introduce state-relabeling adversarial active learning (SRAAL) where a state discriminator identifies the most informative samples which are those being dissimilar from the already labeled ones. Further adversarial strategies are VAAL, based on a discriminator which considers the difference of the unlabeled instances to the latent representation of the labeled instances in [688] and adversarial active learning for LSTMs [428] where again the similarity of labeled and unlabeled samples is considered. In [727], the most informative sample is detected using a Variational Auto Encoder (VAE). The authors in [759] propose true positive sampling in the context of person re-identification with multi-view data and argue that true positives are informative. Since true positives from the same camera view are often similar, so that a view-specific bias would arise, they compute the distance of two tracklets for an adaptive resampling strategy. Density-based approaches are given in [431], based on an information density measure related to the conditional entropy, and in [816] where uncertain samples that are similar to most of the training samples are selected. The authors in [758] consider informativity based on either entropy, least confidence or margin and iteratively select the K most informative unlabeled samples. Their CEAL algorithm checks for which of these instances the current model makes a high-confidence prediction and adds these pseudo-labeled instances into the labeled set so that the model can be updated. [188] state the problem of selecting informative points to query as a constrained optimization problem and derive that the Lagrangian dual arising by optimizing some loss L subject to some loss L' being bounded by some threshold on the whole training set allows for using the dual variables as an informativity measure. [293] experimentally compare different informativity scores and techniques to ensure the diversity of the selected samples. [242] derive minimax rates for the excess risk in batch AL. They propose to optimize a tradeoff between informativity and diversity in pool-based AL. More precisely, for noisy linear models, their algorithm first considers all points where the current model is not confident and queries the labels for a batch of points whose Mahalanobis distance from the already queried points exceeds an iteratively refined threshold, allowing for varying batch sizes among the iterations. They extend their method to generalized linear models and detail out the logistic case. As for the noise, they point out that by re-training the model at most logarithmically often w.r.t. the size of the available pool, their algorithm adapts to the noise level provided that a low-noise condition is satisfied.

Variance reduction/expected model change/expected model improvement: An approach based on kernel mean matching where two conditional distributions are matched without requiring to estimate the densities has been introduced in [771] for transfer learning (match the conditional distributions of the responses in the training resp. the test domain) with the goal to reduce the predictive variance. Variance reduction has also been the goal in [58, 616, 574] who combine AL with importance sampling where one samples from a so-called proposal distribution in order to reduce the variance compared to sampling from the original distribution. Several works aim at reducing the misclassification risk, either by approximating it with the Shannon entropy in person re-identification tasks [149], by finding wrong detections by exploiting the temporal coherence in object detection in videos [49], via decomposing the expected loss [655], via Bagging to reduce the variance of the error estimate [618] or by approximating the expected error/performance by Bayesian approaches [394, 395, 631]. Analogously, [319] and [800] consider the expected accuracy/utility improvement. The authors in [806] approximate the expected improvement by means of influence curves which assign a real number to each point of the input space, quantifying its infinitesimal impact on the estimator. The authors in [321] consider entropy reduction by maximizing the decrease in the expected posterior entropy in a Bayesian approach. Their objective is termed as BALD objective while [227] also consider a Bayesian setting and aim at optimizing a generic acquisition function. In [800] and [760], one aims at minimizing the total annotation cost by dividing the data into clusters and by determining the optimal number of queried labels per cluster, leading to the algorithms CADU resp. CATS. The expected model change is considered as criterion in [296] (change in the modelled local density around each sample). The authors in [219] estimate the expected changes in a Gaussian process model and [218] generalize the works of [219] and [744] who consider Gaussian process regression and provide the EMOC algorithm. This EMOC algorithm also has been applied in [79, 359, 358]. Expected model change can even enter as AL strategy in reinforcement learning where the sensitivity of the optimal policy towards changes in the rewards resp. transitions are considered [191]. In a tracking and video annotation context, [749] interpolate the trajectory and approximate the expected trajectory label change when querying the annotations for selected frames.

Combined approaches: Several strategies explicitly combine different sampling paradigms. In [680], superpixels of images are selected in segmentation tasks in an uncertainty- and diversity-based manner. Uncertainty and diversity are also simultaneously considered in [189], based on the entropy resp. the ratio of the minimal distance to another unlabeled sample, see also [820]. The authors in [770] address person re-identification with weak labels, i.e., which only indicate whether a particular person appears in a video but not in which frame, and propose a combined criterion based on the cross-entropy and a penalty term that discourages sampling highly correlated videos. The authors in [772] concentrate on informativity and diversity while [347] aim at selecting influential, uncertain and diverse

instances. The authors in [833] apply AL in deep visual tracking using CNNs by considering representativeness and diversity of the queried samples. Diversity is encouraged by a cooperation-based selection method using multiple frames in order to eliminate the background so that it does not affect the selection, while for the representativeness, a nearest neighbor discrimination method is applied in order to detect low-quality samples.

Other applications and paradigms of active learning: Other approaches include concentrating on strong negatives in person re-identification tasks [754], learning a selection strategy for AL using RL [201, 449, 793], AL for ranking [452] or Kriging (Gaussian process regression) [819] or active Support Vector Machine (SVM) [724]. The authors in [399] propose active reinforcement learning where one has to pay for observing the rewards. [390] consider active offline policy selection where a set of candidate policies can only be evaluated on a fixed total budget of rollouts, which is solved by Bayesian optimization. AL is not limited to select single instances resp. batches of instances. For example, [680] sample superpixels instead of whole images and [423] experimentally compute the uncertainties for the estimation in active learning approaches in image segmentation and derive that such approaches where regions instead of whole images are labeled are much more effective and significantly reduce the labeling effort. The authors in [770] consider the case of weak labels for videos which do not provide the frame indices in which a particular person appears but only the binary response whether this person appears in the whole video and [749] propose to query only the annotations in selected frames of videos. [233] propose unsupervised clustering AL (UCAL) for person re-identification. After having applied a clustering algorithm like DBSCAN, they consider centroid pairs of person IDs that are erroneously grouped into the same resp. different clusters. These clusters are re-organized via representative pairs, i.e., splitting clusters is done via an independence-compactness tradeoff while for merging, one decides which pairs to label via the similarity of adjacent centroids. [653] propose fine-grained AL for partially labeled scenes, e.g., for predicting future positions by autonomous vehicles. They criticize that there exist AL approaches for object detection but none for prediction and perception. Sensor observations are represented by HD maps and the labels by bounding boxes. They point out that considering a fixed batch size in AL per iteration assumes that each sample can be labeled for the same cost which is not true. They introduce a cost function so that the different efforts for manually labeling sparse or crowded scenes are reflected and fix a cost bound per iteration. They choose the next scenes to label by the ratio of the informativity score (e.g., the detection or prediction entropy) and the cost. The costs are unknown but can be approximated by the number of detections after non-maximum suppression [205]. They generalize their method to partially labeled scenes where they consider a fine grid of the scene, allowing for a fine-grained selection, again by the ratio of the informativity and the cost. [52] study the combination of self-supervised learning and AL and derive than it is only beneficial for high labeling budgets, marginally improving the cost for self-supervision and considerably for AL. The two paradigms themselves are also compared with the result that the former is way more efficient in reducing the labeling cost than AL.

Limitations: Despite the successes of AL, there are several limitations. The authors in [491] experimentally show that AL without data augmentation for semantic segmentation and image classification may fail (i.e., they are worse than random sampling) if the labeling budget is small and postulate that this may result from biasing the distribution of the annotated samples. Especially, their findings encourage to combine semi-supervised learning with AL. Another study has been executed in [503] who derive that the question whether AL in a DL context provides an improvement heavily depends on the hyperparameters, the paradigms, the data sets etc.. The authors in [457] derive that it is difficult to assess the quality of AL since this would normally require a large number of queried samples. Additionally, the first queried samples may improve the model, but later queried samples may decrease its performance. Moreover, the training set and the model are coupled due to AL which decreases the generalization ability across models and tasks. The authors in [347] point out that human annotations in semantic segmentation are often too expensive while automatic segmentations are not sufficiently accurate. The sampling bias is studied in [570] who derive that methods based on the posterior entropy are robust to sampling biases while [340] show that sampling probabilities proportional to the uncertainty often lead to suboptimal proposals and can perform worse than random sampling. They propose optimal sampling schemes where optimality is meant either w.r.t. the variance of the estimated loss, the mean squared error of the prediction or the expected total loss. The authors in [202] study the sampling bias and propose corrective weights which leads to estimators which are unbiased and convergent as long as the acquisition proposal puts non-zero mass on all training samples. [51] point out that AL may become problematic in the presence of class imbalance due to AL may be biased towards particular regions, see also [58], [202]. [51] correct this bias, due to the true labels not being available, by treating the predicted labels (more precisely, the softmax output) as proxy. Having this softmax matrix P and $z \in \{0,1\}^n$ which represents the selection of samples in AL, the goal is to minimize $||\Omega(c) - P^T z||_1$ for the number $\Omega(c)$ of required samples of class c. This leads to the class balanced AL (CBAL) algorithm which minimizes the entropy regularized with the norm above. They also show how to find a tradeoff between representativeness and class balance. [7] propose an AL strategy that is more focused on the minority class in imbalanced data by assigning more budget to them. They experimentally derive that transfer learning an AL can be combined by using the features extracted from a pre-trained model on a generic data set. [405] experimentally derive that AL indeed does not imply monotonically increasing accuracy but that learned knowledge can be forgotten during the AL iterations. They propose to use knowledge distillation in order to mitigate this but point out that the current model may not be the best teacher due to this issue. They select the teacher by a tradeoff between accuracy and consistency (i.e., making correct predictions over subsequent AL iterations). The authors in [100] study how to select the initial pool in pool-based AL. The initial pool usually contains around 1-

10% of the data set. The goal is to cope with the disadvantage that deep AL seldomly outperforms random sampling AL in the presence of changes in the data (class imbalance) or training procedures (regularization, data augmentation). They propose two strategies. The first is to train a self-supervised model on the raw data and to include the samples on which this model suffers a high loss into the initial pool. The other strategy is cluster sampling, i.e., the data are divided into clusters and one samples from each cluster with equal weight in order to encourage diversity and that the initial pool represents the data set well. A related semi-supervised strategy is given in [684] who point out that semi-supervised learning concentrates on the most certain samples while AL concentrates on the most uncertain ones. However, [231] criticize that the method of [684] is independent from the model training and propose to unify selection and model updates. They point out that uncertainty sampling is often not appropriate for deep learning AL since DNNs tend to be overconfident. They found out that their consistency-based approach outperforms the entropy-based approach and also works sophisticatedly in terms of diversity of the queried samples. A severe issue of their method (including related AL methods) is the cold start failure which means that the initial model trained on the initial pool has a poor performance and leads to significantly wrong decision boundaries, so an entropy-based method would query from near this wrong boundary. Their semi-supervised strategy also suffers from cold start failures. They point out that if one had a validation set, one could track the performance of the model to avoid cold start failures, however, this is often not the case in practice. For the CE-loss, they propose an upper and a lower bound which enable a tracking during training. As for a validation and a test set, [393] introduce active testing, i.e., sample-efficient model evaluation where test data is labeled in an active manner by maximizing the accuracy of the risk estimate. They point out that (naturally) selecting most uncertain points to label overestimates the risk which is even worse for overconfident models. To remedy this, they apply the IS strategy from [202] since a sampling bias is much more harmful for testing than for training.

5.4.2 Application

We point out that AL is not tailored to a specific task but serves as a paradigm for sophisticated iterative (re-)training of the respective AI. Therefore, AL can be used to improve a particular learning algorithm for perception or planning models, but as already outlined above, the concrete sampling or selection paradigms of AL differ.

As for existing work for perception, AL has been applied to object detection/segmentation in [80] who invoke the PASCAL VOC 2012 data set. The authors in [162] additionally run experiments on the PASCAL VOC 2007 data set and point out that it is crucial how the threshold in their strategy is chosen since switching too quickly to strong labels causes much higher annotation costs than staying in the weak labeling case which however is prone to produce noisy labels. The authors in [367] consider the PASCAL VOC 2007, the PASCAL VOC 2012 and the MS COCO data set and derive that their localization tightness metric improves the model significantly provided that the ground truth is known, otherwise the improvement is rather low. Their localization

stability metric improves the model but is more expensive to compute than other metrics. The authors in [423] apply their region sampling method to the GlaS Challenge Contest and the 2016 International Skin Imaging Collaboration data and find out that sampling regions instead of whole images can indeed lead to a better calibrated model and significantly reduce the annotation cost. Evidently, all these approaches and other ones for classification or segmentation tasks, even if not primarily designed for automotive use cases, can directly be applied for automotive data. As for person (re-)identification which is of course directly related to the automotive setting, [759] apply their method to the PRID, the MARS and the Duke-video data set. On all data sets, a significant improvement is observed while keeping the annotation ratio below 3%. The authors in [149] consider the iLIDS-VID data set and [49] consider a synthetic data set as well as the ImageNet-VID data set where they only require to label 10% of the latter data set. The authors in [749] make experiments on the VIRAT data and report that they are able to reduce the annotation costs by 90% compared to a labeling strategy where a fixed rate of instances is to be labeled. In [754], experiments are executed on the CUHK03, the Market-1501 and the VIPeR data set where also the human-out-of-the-loop (HOL) case is considered which means that the number of human annotations is limited, so human feedback becomes eventually unavailable. They show that their AL strategy even leads to competitive performance in the HOL setting. The authors in [770] consider the two weakly labeled data sets WL-MARS and WL-DukeV and even study the setting of missing annotations where for example a person that is visible in a few frames is missed by the annotator so that even weak labels are not available. They show that even in the latter setting, their method can improve the performance of the model. [833] apply their AL method on seven benchmark data sets for tracking and derive that it achieves a comparable performance to methods that require huge labeled training data. [207] apply AL with the uncertainty criterion (measured using MC dropout or ensembles) for lidar 3D object detection. They conduct experiments on the lidar depth and intensity maps from the KITTI data set and derive superior performance in terms of required labeling effort in comparison to a training method where samples are queried randomly.

Active learning also successfully entered trajectory planning and MPC, see, e.g., [481] where the uncertainty is actively learned and [857] where informative trajectories for MPC are generated. In stochastic MPC, [301, 303, 302, 21] apply AL for learning structural uncertainties of the model in order to enable better model discrimination. In [85, 91, 133, 880], AL for Gaussian process dynamics, where either uncertainty minimization or informativity maximization is performed, is considered. Model-based active exploration in MPCs has been suggested in [677, 649]. [324] propose active uncertainty learning for motion planning with stochastic MPC.

6 KNOWLEDGE EXTRACTION - SYMBOLIC EXPLANATIONS

The explanation and interpretation of the functioning of data driven models is an essential prerequisite on the way to trust-

worthy systems. Special focus is put on an understandable representation and formalization of the behavior learned by a neural network. The extraction and consideration of learned decision patterns and concepts not only motivate a final validation of the function but are already equally useful in the development of methods for use in safety-critical applications.

Symbolic interpretations play an important role and have been extensively investigated in recent years. In order to inspect the functionality of a neural net in a more formal way, *rule learning* has established itself as one of the most prominent methods in this context. The approaches introduced in Section 6.1 range from methods that focus on the internal structure and interplay of neurons to methods that try to model the input-output relations in a human interpretable way. The latter is also connected to rule extraction and pattern mining with respect to the input data as discussed in Section 6.2 focusing on *structured output prediction*.

Natural language, as another form of symbolic representations beyond rules, is considered in the two remaining sections. While knowledge extraction in the context of regulations and norms from *legal domain* is the focus in Section 6.3, natural language as accompanying explanation to visual stimuli is the goal in *visual question answering*, presented in Section 6.4.

6.1 Rule Extraction and Rule Learning

Authors: Adrian Paschke, Ya Wang, Kevin Krone, Etienne Bührle, Hendrik Königshof

This section looks at algorithms for rule extraction and rule learning. A distinction is made between extraction approaches that use the internal structure of the trained models (e.g. decompositional rule extraction) and approaches that use the models and extracted concepts as a whole (pedagogical and eclectic rule learning) to derive new (symbolic) knowledge, and inductive learning approaches (e.g. Inductive Logic Programming), which learn rules from training data. The section first considers the extraction approaches, then approaches for inductive logic programming (inductive learning of symbolic rules), and last applies rule learning in the context of planning.

6.1.1 Rule extraction from trained neural netwoks

In recent years DNNs have achieved remarkable performance. However, their extremely complex internal structures make them incomprehensible as a black box, which is not acceptable in some safety crucial application domains, such as medical diagnosis, industrial process control and fully autonomous driving [295]. Many approaches have been proposed to interpret and explain the DNNs in a human acceptable way. One feasible approach is to extract understandable symbolic rules from neural networks. The study on rule extraction dates back to early 90s, which originally aimed to synthesize knowledge for knowledge-based systems and discover the full potential of neural networks [16]. It is basically recognized that acquired knowledge during training is encoded as the network parameters, activation and architecture design. Knowledge could be extracted in form of If-Then rules, M of N rules [725], decision tree, decision table, etc. [658]. Based on whether extracted rules reveal the internal
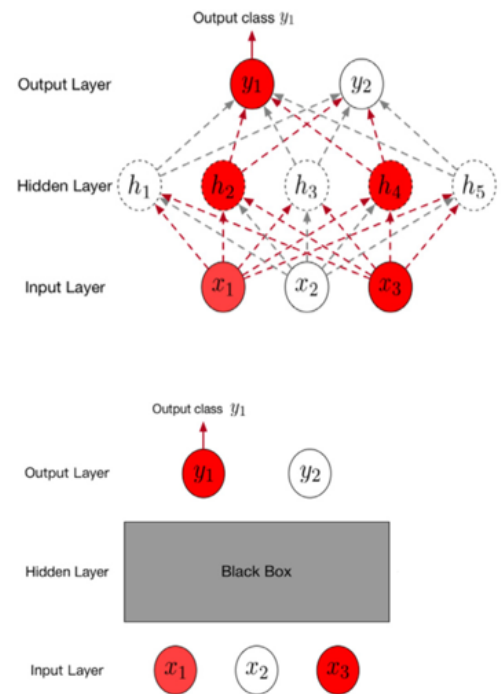

Fig. 6: Illustration of the functionality of decompositional (top) and pedagogical (bottom) approaches [295]. (red color stands for activated state and transparent color for deactivated state)

structure of neural networks, Craven and Shavlik [135] proposed a classification of rule extraction into two basic categories, namely the decompositional and pedagogical approaches. The former [53, 435, 730, 634] extracts the rules by analyzing the activation and parameters of the layers, while the latter [134, 638, 659, 716, 719, 708, 867] extracts rules by mapping the input-output relationships as closely as the neural networks understand without considering internal structure. As illustrated in Figure 6 (top), the decompositional approach extracts the rule $IF \ \sigma(x_1) \land \sigma(\neg x_2) \land \sigma(x_3), THEN \ h_2, h_4$ from the first layer and the rule $IF \ \sigma(\neg h_1) \land \sigma(h_2) \land \sigma(\neg h_3) \land \sigma(h_4) \land \sigma(\neg h_5), THEN \ y_1$ from the second layer. As we aggregate them and eliminate h_2, h_4, the rule mapping between input layer and output layer is extracted as $IF \ \sigma(x_1) \land \sigma(\neg x_2) \land \sigma(x_3), THEN \ y_1$. In contrast, the pedagogical approach (Figure 6, bottom) treats the whole network as a black box and learns the rule mapping directly by observing the inputs and outputs. Besides that, a third class could be the eclectic approach [725, 660], which is a combination of both decompositional and pedagogical approaches. Since our goal is to understand NNs by analyzing their internal structure, we will only focus on the decompositional approach in this subsection.

Though Neural Networks have been widely studied and applied, there is still a lack of research on rule extraction from DNNs [282, 879]. Most proposed algorithms targeted shallow neural networks. The reasons is attributed to two aspects. First, the number of extracted rules is proportional to
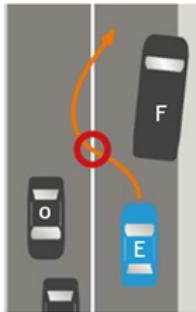
Fig. 7: A controlled rule exception scenario

Fig. 8: Comparison of two different approaches for rule extraction from FCs in neural network models (top: DeepRED, bottom: ICDT)

the depth of the network. [879] applied their algorithm to a four-layer NN with structure of *784-10-5-2* and extracted over 10^5 rule terms and consumed large amount of computation time and memory space. Nowadays, even the first version of DNNs AlexNet consists of 8 layers, state-of-the art NNs are often designed over 100 layers which makes decompositional rule extraction intractable without pruning. Secondly, the interpretability of the rules will be lost if the number of rules is too large. Many End-to-End NNs are applied directly on perceptual level which take the raw sensory data as the input. Although Guido Bologna's [71] algorithm extracts rules from a simple convolutional neural network for image classification task, the relationship between individual input pixel and the output class does not explain the decision process very well. Based on these considerations, we argue that extracted rules will be more understandable if we apply the decompositional approach to the top layers of DNNs. Instead of mapping from raw data, the algorithm takes the features, which have significant semantic meanings, from intermediated layers as the input.

The decompositional approach helps to understand the classification process of DNN. In the context of autonomous driving, the reason for identification of a rule exception can be extracted as human understandable knowledge. We consider the following specific driving scene as shown in Figure 7: an ego-vehicle E (blue marked) drives along a lane which is partially blocked by an obstacle F (black marked) and limited to the right by a solid line. A rule exception of crossing solid line is allowed if it is not foreseeable when the obstacle F will disappear. Before path planning and legal rule integration, the class of the situation should be correctly predicted. To explain rationales of the prediction, we identified two existing algorithms that are potentially applicable to this problem. The first one called DeepRED [879] was proposed in by Jan Ruben Zilke *et al.* in 2016, which was claimed as the first algorithm that has explicitly been tested on the task of extracting rules from DNNs. The algorithm was developed based on CRED [634]. It uses decision trees to describe a NN's performance based on the units in its hidden layer. Then, it builds up new decision trees to describe the split points of the first decision trees. At last, the rules are merged to describe a NN's behaviour. We refer to [879] for more detail. Formally, we define the rule extraction task as follows: given a well-trained NN and an example as input, the goal of the task is to find rule sets $R_{i \to o}^v$ that describe the relationship between the input layer i and the output layer o for each class v as closely as possible.
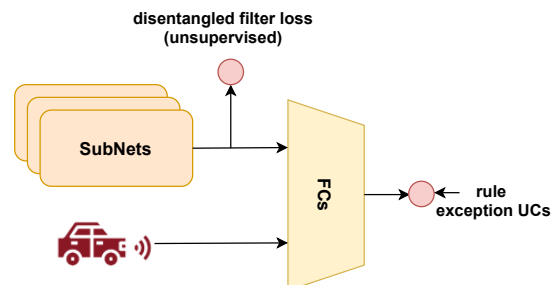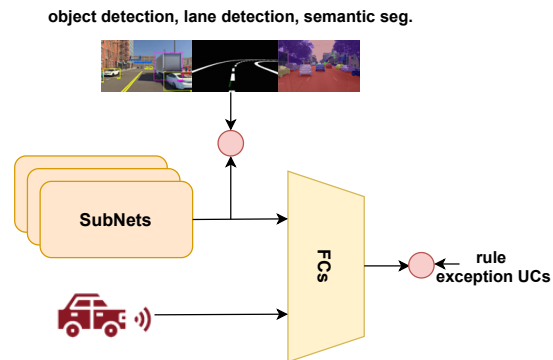
As mentioned before, when applying rule extraction, the input layer must not be the example itself but the features of intermediated layers in the NN. For classification of rule exception scenarios problem, the input can be the result of different detectors and sensory data, and the output can be the class of one or multiple defined use cases.

As shown in Figure 8 (top), the FCs (fully connected layers) are built on the top of one or multiple subNets (e.g., CNN models). The supervisory signals for different tasks, such as for object detection, lane detection assign semantic meanings to the input of FCs. When applying DeepRED to the FCs, the following rule set is expected to be extracted:

$$IF \ isSolid(Line) = True,$$
$$AND \ Speed(VehicleF) = 0,$$
$$AND \ LocX(Line) - LocX(VehicleF) > 3,$$
$$AND \ LocY(VehicleO) - LocY(VehicleE) < 0,$$
$$THEN \ classRuleexception1.$$

In other words, the rule set says a rule exception case is identified, when a left ego-line is solid, an upfront vehicle is stopping and it is more than 3 meter right from the solid line and all oncoming vehicles are already behind the ego-vehicle. In this case, an overtaking of the upfront vehicle is allowed. These rationales explain the decision very well, however, we still need ground truth data to supervise different detection tasks. To overcome this problem, an approach proposed by Zhang *et al.* [852] is potentially transformable to this task.
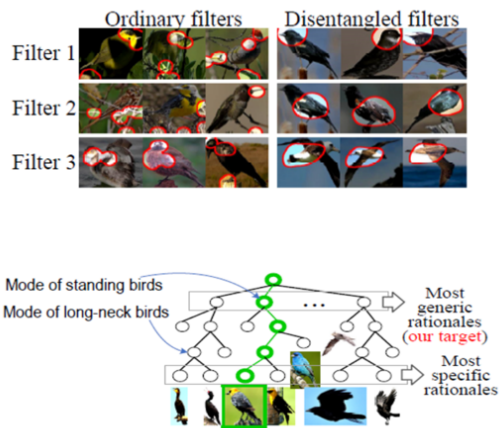
Fig. 9: top: Comparisons between ordinary CNN feature maps and disentangled feature maps; bottom: Decision tree that encodes all potential decision modes of the CNN in a general-to-specific manner [852]

Instead of using additional supervisory signal, the author designed specific filter loss [851] (Figure 8, bottom) that disentangles the feature representations in the top layers. It is commonly recognized that, an ordinary filter in the top conv-layers usually represent the pattern that is the mixture of parts and textures. Using such disentangled filters we make each filter represent a specific object part (Figure 9, top). Different from the first approach, the author designed a clustering algorithm on the FCs to derive a decision tree to explain the decision process (Figure 9, bottom). Despite its advantages of no need for additional supervisory signal for object parts, it requires adaption of the NN structure and finetuning of FCs. In the case of lane detection, disentangled filters may not work, since the texture of the lane lines plays key role in distinguishing. Another shortcoming as claimed by the author is that it is not compatible with the Nets that have skip connections, such as ResNet. Though the first approach requires additional annotations for different tasks, the rule sets extracted by DeepRED are more expressive and easier to understand.

6.1.2 Inductive Logic Programming

While modern ML approaches based on deep learning have been extremely successful in recent years, they still often fail to generalize on seemingly simple tasks. For example, given enough training data a standard deep learning model learns the less-than relation on handwritten numbers, but it will fail to generalize it to previously unseen pairs of digits. Inductive Logic Programming (ILP) is a set of techniques for learning logic programs from given examples. Instead of relying on statistical mechanisms, ILP is based on logical inference while incorporating ideas from automated reasoning and knowledge representation. This way ILP programs are often able to generalize from very small data sets. Another key advantage of ILP is the fact that its solutions are comprehensible and verifiable.

While different ILP frameworks employ different strategies to solve a given task, most of them agree on the following

definition of an ILP problem. An **ILP problem** $(\mathcal{B}, E^+, E^-)$ consists of the following data:

1) $\mathcal{B}$ is a set of background assumptions,
2) E^+ is a set of positive examples,
3) E^- is a set of negative examples.

The goal of ILP is to construct a logic program (a hypothesis) that, together with given background assumptions, entails the positive examples and rejects the negative examples.

Over the last decades many significant improvements to ILP have been made. Today, it is possible for ILP systems to learn recursive programs and to invent auxiliary predicates from background assumptions [136]. However, there are a few weaknesses most ILP frameworks have in common. Since ILP represents data as logic programs this requires the user to choose from a range of available languages for this task. For example, Prolog is a Turing-complete logic programming language that is commonly used in ILP. Datalog is a subset of Prolog, which trades Turing-completeness for efficiency and decidability. Similarly to feature selection in classical ML approaches, choosing the appropriate background assumptions is crucial for the success of ILP.

One of the main disadvantages of traditional ILP frameworks is their inability to handle mislabeled data. Recently, there have been attempts to combine the strengths of deep neural networks with the benefits of ILP by replacing absolute logical reasoning with continuous values reflecting the confidence of the conclusion. Richard Evans and Edward Grefenstette [194] presented a reimplementation of ILP in an end-to-end differentiable architecture called ∂ILP. According to the authors ∂ILP is robust to mislabeled data (up to 20%), while also being able to handle ambiguous and fuzzy data. However, the search space needs to be constrained via handcrafted rule templates and all logic programs are restricted to definite Datalog clauses, disallowing function symbols. Furthermore, ∂ILP only allows for two atoms per rule and predicates of arity of at most two. Shindo et al. [671] improve on ∂ILP by introducing several new algorithms that deal with more complex programs including function symbols. For both frameworks scalability remains an issue. Another approach by Ali Payani and Faramarz Fekri [552] implements a differentiable version of forward chaining. According to the authors, first-order rules are learned without prior specification of a rule template, while maintaining desirable ILP features such as being able to learn recursive predicates as well as predicate invention. A second architecture avoiding the use of rule templates is named Neural Logic Machines (NLMs) and was proposed by Matthieu Zimmer et al. [881]. It combines reinforcement learning and ILP in a differentiable framework, while retaining full-interpretable solutions. However, only a restricted space of first-order programs is learned and predicate invention is only possible at the loss of explainability of the solutions.

6.1.3 Applications

In the context of autonomous driving, extracted symbolic rules reveals the internal functionality of the neural networks, which can explain the result of a scene classification task by

identifying key features and their importance. Furthermore, ILP can be employed to derive unwritten legal or behavioral rules from a given formalization of traffic rules. This extracted knowledge can afterwards be incorporated into the body of legal or behavioral knowledge.

6.1.4 Rule Learning for Planning

The goal of rule learning is to infer the constraints that govern expert behavior based on observed demonstrations. A difficulty arises from the fact that the rule inference problem is ill-defined – the rules are not unique. Furthermore, human expert demonstrations are noisy, i.e., suboptimal, and might even occasionally violate rules.

The following sections highlight methods to extract rules from expert data, with applications to traffic rule extraction. The extracted rules can be used as knowledge bases for subsequent algorithms. The presented methods leverage an explicit formalization of the rules, which, in the case of traffic rules, is still an area of ongoing research [193].

Parameter Learning: The task of rule learning can be framed as learning an objective function which includes penalty terms for rule violations. In fact, Ng and Russell argue that "the reward function is the most succinct, robust, and transferable definition of the task" [511]. Under this interpretation, methods from behavioral cloning and inverse reinforcement learning can be used.

Behavioral Cloning. Behavioral Cloning aims at replicating the observed expert behavior in a supervised learning setting. In the context of autonomous driving, the method was notably used by Pomerleau in his seminal work [567]. A large body of work has used this approach to learn end-to-end driving policies, which, given sensory information, reproduce the observed system inputs [40, 632, 342]. More recently, Generative Adversarial Imitation Learning (GAIL) has been shown to require less training data, while yielding more robust policies [311].

Behavioral Cloning can be used to tune parameteres of explicitly stated rules. In [839, 444], a Markov Random Field (MRF) is used to jointly reason over actor trajectories. The MRF encodes rules as potential functions that penalize violating settings of nodes, which represent the trajectory choices of the actors present in the scene. To find the combination of trajectory choices with the lowest potential, Loopy Belief Propagation is used, which is a differentiable message-passing procedure. Thus, the weights and parameters of the rules can be adjusted by backpropagation of a loss function. The authors use a standard classification loss with expert data as the ground truth.

In [430], a Differentiable Logic Layer (DLL) is presented that can be used as an additional layer on top of any network, and takes a set of Signal Temporal Logic (STL) rules as inputs. Temporal Logics are a popular way to express constraints on trajectories, and have been used to express traffic rules [471, 193, 23, 120, 119]. The STL semantics have the added advantage of being differentiable. The DLL can thus act as a corrector in the forward pass, increasing rule conformity. In the backward pass, gradients can be propagated to both the underlying network and the STL rule parameters.

Inverse Reinforcement Learning. The goal of Inverse Reinforcement Learning (IRL) is to estimate the objective function governing expert behavior. It was first posed by

Kalman [361] in the context of Inverse Optimal Control, and later by Ng and Russell [511]. Typically, the structure of the objective function is assumed to be known, reducing the problem to a parameter estimation problem. A recent overview is given in [24].

Maximum Margin IRL. The IRL problem is ill-posed, because multiple objective functions can explain a given expert behavior, including the "always zero" reward. Maximum Margin IRL overcomes this difficulty by requiring that expert trajectories be optimal by some margin [24]. This requirement can be encoded as a loss function.

In [840, 782], the max-margin loss is used to train a numerical scene-dependent planning cost volume end-to-end. In [626], it is used to learn the parameters of an objective function that is subsequently used for behavior (coarse-scale) and trajectory (fine-scale) planning. In [627], it is used to learn the weights of an objective function that uses a semantic occupancy grid map to score trajectory candidates. A similar idea is presented in [94], which uses the max-margin loss to learn penalties for a small set of illegal behaviors, including off-road and opposite-direction driving, and running traffic lights.

Maximum Entropy IRL. Maximum Margin IRL suffers from a problem known as label bias, which arises from the fact that the algorithm has to commit to what it deems to be the optimal objective function with a margin. Maximum Entropy IRL overcomes this by maximizing the entropy of the trajectory distribution [878]. It can be shown that the distributions with highest entropy belong to the exponential family [24]. The principle can be extended to deep architectures [801].

Structure Learning: The previous section presented methods for parameter learning, where the learning problem was framed as an optimization problem over a fixed class of functions with tunable parameters (e.g., the weights of a neural network). However, the structure of rules is often hard to express succinctly and needs to be discovered, requiring a joint optimization over the (typically) continuous parameter space and the (typically) discrete rule space. The following paragraphs highlight methods that bridge this gap.

Maximum-Likelihood Constraint Learning. The maximum entropy principle yields a distribution over trajectories. This can be used to find constraints with maximum likelihood, given an objective function and system model. In [652], the case of of an Markov Decision Process (MDP) with markovian constraints is studied. The authors present an algorithm that iteratively maximizes the probability mass of forbidden trajectories, while keeping observed trajectories legal. To prevent overfitting, the process is stopped once the decrease of the KL divergence between the observed and inferred trajectory distributions crosses a lower threshold. A similar method is presented in [741], which uses a surrogate likelihood criterion and leverages inclusion relations to efficiently search the space of constraint candidates.

Neural Architecture Search. Neural Architecture Search attempts to find the optimal structure of network layers for a given task. This problem has many similarities with rule structure learning. A multitude of methods, including Reinforcement Learning, Evolutionary Algorithms, and Gradient Descent can be applied, which are surveied in [190].

A related field is Neural Program Synthesis, which

attempts to infer programs based on input-output examples. Successful approaches include the use of sequence-to-sequence models and beam search to generate valid programs in domain-specific languages [543, 165]. An overview of Neural Program Synthesis is given in [366].

In [700], a gradient descent-based method from neural architecture search is used to learn a symbolic decision program for autonomous vehicles in a domain-specific language. To this end, the space of programs is expressed as a supergraph, whose edge weights define the inferred computation graph. The authors use the GAIL objective to learn the computation graph from expert demonstrations.

In [577], a recurrent neural network is used to generate rule candidates for knowledge graph completion. A survey on knowledge graphs is given in [349].

6.2 Structured Output Prediction

Author: Christian Hesels

Pattern Mining is used to discover unexpected and useful patterns in a database. The resulting patterns are human interpretable and thus useful for data understanding and decision making. Pattern Mining first was introduced to create association rules for market basket analysis [10]. The subsequent algorithms built up on this idea and are generally summarized as Frequent Pattern Mining. Algorithms like association rules only consider the occurence of itemsets and they are applied to non sequential databases. This is called Frequent Itemset Mining. Another approach in Frequent Pattern Mining is Sequential Pattern Mining, where not only the amount of occurences is taken into account, but also the order determined by a timestamp in a given database. A slightly different approach but yet very close to the mentioned ones is Time Series Pattern Mining. While Frequent Itemset Mining and Sequential Pattern Mining receive a symbolic representation as input data, Time Series Pattern Mining is used to find patterns in high dimensional data [222]. Generally, sorted episodes or sequences are defined as serial, not sorted ones are called parallel and if both is possible epsiodes are named general.

6.2.1 Pattern Mining

Frequent Itemset Mining algorithms like Association Rule Mining searches for if.. then.. statements for Items. Which in the context of Basket Case Analysis means, if a customer buys one item then he buys another one with a certain confidence. This rules are not limited to single cardinality, but also work for more combinations. To find these rules, Association Rule Mining calculates three metrics: Support, Confidence and Lift. The support is calculated from the item count N and the frequency of two items $freq(A, B)$ occuring together. Confidence gives information about how often the items A and B occur together. Lift is calculated from the support of A, B and the global support.
Apriori is an algorithm to optimize this principle. Because with this Association Rule Mining approach thousands of rules would be generated without pruning the input, it takes a lot of space and iterations to mine those rules. Apriori is based on the concept, that a subset of a frequent itemset must also be a frequent itemset. Therefore when a subset does not fullfil a given minimum support every itemset

in this subset might as well be removed from the dataset [11].

For Sequential Pattern Mining the definitions are the following: an event is defined by a set of items i with the length k, i.e., $(i_1, i_2, ..., i_k)$, often in combination with a timestamp. A sequence is a set of events and a sequence with k-items is called a k-sequence. The events are taken from a database D. The support (or frequency) is given by $\sigma(\alpha, D)$. When a sequence pops up more often than the user defined minimum support value, it is considered a frequent sequence. Often a user defined Time Window is given, in which range the patterns need to appear in. Otherwise frequent patterns from huge sequences would be mined, that may not have any correlation.

Time Series Mining has two different approaches to generate a suitable data representation. One is to reduce the dimension of the input data with methods like basic sampling or critical point model [41]. Yet the goal remains to reduce the dimensions and calculate the distances between multiple time series to cluster them. The second approach is to discretize the data to receive a symbolic representation. On this symbolic representation, algorithms from Sequential Pattern Mining are applied. Measuring the similarity of time series is done in two ways. Whole sequence matching and subsequence matching. To calculate the distance between the time series the nearest pairs need to be saved and the distance measurement has to be done with all possible offsets [222]. One way to do it for whole sequence matching is to minimize the dimension as mentionend before (for example with DFT coefficients [9]) and calculate the Euclidean distance on it. In subsequence matching, a query is given and the goal is to find it in the longer time series. The subsequence therefore needs to be compared to every offset in the longer time series. There are a lot of ways to do this like DualMatch [822] or GeneralMatch [498]. The final pattern mining is mostly done with clustering. An initial cluster center is defined (random or from a sequence). As a parameter the amount of clusters is given (static or variable). Afterwards the distance of each input datapoint to the cluster is calculated and the closest cluster wins and updates the cluster center. The algorithm either converges or finishes when the maximum iteration number is reached [222].

6.2.2 Applications in Situation Interpretation

Because the input is a high level symbolic representation, Time Series Pattern Mining is not suitable for this approach (except sensor data is used, in which case a suitable sampling method is needed to reduce the complexity). Assuming that in traffic a sequence is of importance, for example stopping the car before a pedestrian is in front of the car, Sequential Pattern Mining is reasonable. State of the Art algorithms like CM-Spade are used to extract the patterns [215]. These extracted Patterns (complex events) are used to train a transparent classifier for action/event prediction. For this approach the training data is created in a way where a certain time window is given and every time the events to predict occur, it is defined as a label and the previous events (depending on the time window) as training data.

6.3 Natural Language Processing for Legal Domain

Authors: Stefan Griesche, Anh Tuan Tran

Previous sections about knowledge representation learning and knowledge crafting highlighted knowledge graphs and ontologies as a feasible approach to represent knowledge. These concepts rely on annotated data. For smaller data sets and proof-of-concepts this can be achieved by a human annotator or crafting by hand. However, legal knowledge in form of norms, traffic regulations and laws sum up to a large amount of unstructured data in different languages. Natural Language Understanding (NLU) and Natural Language Processing (NLP) provide methods to tackle this large amount of data. However, for the legal domain additional challenges have to be considered. As mentioned by [571], computational models which deal with legal texts and traffic regulations have to take exceptions, rule conflicts, open texture and vagueness, rule change, and the need for commonsense knowledge into account. Therefore, existing NLP concepts and models have to be tailored or trained to the domain and domain specific solutions have to be developed [611].

6.3.1 NLP tasks

The adaptation to the legal domain addresses already different NLP tasks. This includes legal entity recognition [419], text similarity [489], [505], legal question answering [869], legal summarization [364] and pretrained legal language models [99]. Zhong et al. [868] provides a good summary on the benefits of AI in understanding legal text. An additional task that is particularly well-suited to legal NLP is argument mining [441], in which the arguments are extracted from unstructured documents in a hierarchical form: in the lower level exists linguistic form such as argumentative sentences, argumentation boundary, etc. while in the higher level, the arguments are formed from argumentative utterances and evidence. Further tasks in legal NLP are also of recent research interest [128], including legal case retrieval, legal case entailment, judicial prediction. Sleimi et al. [689] used NLP to query legal requirements and support the legal requirement handling.

6.3.2 Datasets and legal language models

There is a increasing proliferation of new corpora specific to legal domains such as [569], [419], [305], [733]. Most of these corpora are designed to facilitate a specific task. Recently, there are attempts to provide the data for multiple task training (and with it, a pre-trained model that can be customized and fine-tuned to further downstream tasks), for example CaseHold [866], Edgar [72]. Most pre-trained models rely on transformer architectures and provide a lightweight variant, fine-tuning from larger models such as BERT or GPT, namely Legal-BERT [99] and Legal-GPT [72]. The Natural Legal Language Processing (NLLP) community [524] provides a good amount of references to these corpora.

6.3.3 Software

There is an increasing number of open-source and proprietary software system for legal NLP. ICLR&D (Incorporated Council of Law Reporting for England and Wales) has open-sourced Blackstone [64], which is a legal NLP pipeline built on top of Spacy [196]. Blackstone NLP models consider the nature of legal texts. The named-entity recognizer is for instance able to detect legal concepts such as provisions, citations, the text categorizer annotates axioms, issues, conclusions. LexNLP [421] offers segmentation, tokenization, pre-trained word embedding, classifiers, fact extraction specific for legal texts. FlairNLP also provides easy to access API for working with deep learning models via HuggingFace and PyTorch. QuantLaw [579] has recently open-sourced their toolkits for analyzing and detecting legal references from text.

6.3.4 Research projects

There have been two European projects in the past years of interest in the legal domain which considered NLP techniques and made an essential contribution to the state-of-the-art. The Mirel project [490] developed tools for mining and reasoning with legal text. The project aimed to close the gap between the community working on legal ontologies and NLP parsers and the community working on reasoning methods and formal logic. The project provided first legal ontologies [43], [168] and used NLP techniques for legal reasoning [178]. The lynx project [466] developed a legal knowledge graph for information retrieval services. Moreover, the lynx project developed an annotation service, published legal datasets and trained language models to the legal domain [419].

6.3.5 Applications

The state-of-the-art shows an increasing activity of research on NLP in the legal domain. For automated driving, traffic rules and parameters written in traffic regulation and norms are of utmost concern. However, to the authors' best knowledge, NLP for traffic regulations and norms seems currently largely unexplored. Even though there are first NLP applications which focus on analyzing requirements and technical documentation (test specifications) in software and system development [641], [209], the potential of NLP is not fully exploited. NLP can play an essential role to analyze legal requirements and find similarities and conflicts among traffic regulations and norms of different countries. Furthermore, NLP can deliver the metadata to enrich requirements with different sources such as legal commentaries to interpret fuzzy traffic rules. This can be further processed in knowledge graphs or reasoning engines and support the software development process to design for instance legal conformity planner. NLP methods also have the potential to support editors such as [433] to semi-automate the process of formalizing traffic rules [193]. Ontologies for traffic regulations introduced by [75] or [82] provide first formal descriptions. Using this as an input, NLP methods can be seen as a part of a pre processing to accelerate / to automate the process of formalization legal statements into logical statements by supporting the understanding of legal texts through a semantic annotation. These logical statements can be further processed and integrated into hybrid AI models for situation understanding or planning as described in Chapter 4.

6.4 Question Answering

Author: Tianming Qiu

Question Answering (QA) is a NLP task of answering questions based on a given certain question and the corresponding context. As a popular task towards general artificial intelligence, the algorithm is designed to understand the context and grasp some human knowledge. A step further, Visual Question Answering (VQA) extends question answering by combining it with computer vision tasks. The VQA task aims at answering a given question based on images. Hence, VQA approaches are required to possess not only language understanding but also image reasoning ability. The answers predicted by the algorithm are seen as the knowledge in the visual scene. The more questions are answered, the more knowledge the algorithm extracts from the images.

6.4.1 VQA: a combination of NLP and CV

Current VQA tasks are solved by supervised learning: the image and the language feature extractors are applied to the input, that contains one picture question pair. The next step is to forward the image-language features via another network to get the final output answer. By comparing the output answer with the ground-truth label, the whole network which consists of the three aforementioned parts is updated. Therefore, various VQA algorithms investigate different modifications on the following perspectives:

- Image or language feature extraction;
- Fusion of image and language features;
- Network model for final answer. prediction.

Since VQA is extended from QA, a language model like RNN or LSTM model is used as both the language feature extractor and the final language output. All the latest developments of these NLP models are further deployed in VQA models. For the image part, different convolutional neural network backbones are widely used. A key issue in the VQA problem is how to fuse image features together with the RNN or LSTM language representation. Two-modal feature fusion approaches are summarized in a comprehensive survey work [798]. A basic method is the joint embedding approach which embed image features and language features to a common space. Other approaches apply the attention mechanism to find the most relevant part for final answers. [798] keeps the attention mechanism from NLP. The algorithm pays attention to different areas in the image feature map instead of tokens in previous NLP tasks. The attention mechanism is believed to reinforce the performance by re-weighting the image features differently according to their spatial positions.

Since Transformer [740] has been widely used in NLP and even CV communities, the VQA community is also looking for breakthroughs with the help of this powerful framework [326]. The original Transformer [740] is a sequence-to-sequence NLP model which uses a self-attention mechanism. For a sentence, the Transformer encoder calculates the relevance of each token in this input sentence by the inner product. Such a relevance matrix is called a self-attention matrix. In the VQA task, the Transformer structure replaces the traditional NLP model such as RNN or LSTM. And the self-attention between language tokens and image feature units provides stronger semantic meanings due to the inner product.

Another popular task called Weakly Supervised Object Localization (WSOL) [846] is also related to VQA. Weakly supervised learning refers to a training process when only partial information regarding the task (e.g., class label or bounding box) on a small dataset is available. WSOL refers to the object detection task with insufficient ground-truth bounding boxes. Without bounding box labeling, categorical token or language query is helpful for object localization [232]. Token Semantic Coupled Attention Map (TS-CAM) [232] first splits an image into a sequence of patch tokens for spatial embedding and then re-allocates category-related semantics for patch tokens, enabling each of them to be aware of object categories. As a multi-modal task, these language-query-based WSOL problems have a strong connection with the VQA tasks.

6.4.2 Applications

If we take language-based WSOL also as VQA problems, then it is related to the perception topic. Otherwise, VQA is usually not relevant to perception. VQA tries to understand the image and question at the same time, hence, from this perspective, all the papers mentioned above are related to situation understanding. Regarding the autonomous driving scenario, an autonomous driving VQA dataset called ISVQA [39] queries a set of images instead of one single image for answers. The set of images are taken by six cameras mounted at different positions of a vehicle. Traditional VQA methods are deployed on this ISVQA dataset, while Transformer architecture achieves better performance. VQA methods provide a tool to extract more knowledge from a given autonomous driving scene.

7 KNOWLEDGE EXTRACTION - VISUAL EXPLANATIONS

Visualization is a great way to represent abstract and complex knowledge in impressive form. Therefore, it can also serve as an effective tool to demonstrate knowledge extracted from machine learning models. Unlike text, formulas, or other symbolic interpretations, visualizations can provide humans with an intuitive impression of some complex information. Better understandings and even algorithm improvements can be inspired by the visualization.

Much research has focused on the visual interpretation of machine learning models. One of them, called *visual analytics*, aims to use the human visual system and human knowledge to identify or hypothesize patterns that are often hidden in large datasets. Different methodologies of visual analytics are discussed in Section 7.1. *Saliency map* methods in Section 7.2 emphasize how to generate heatmaps on input images for computer vision tasks. In Section 7.3, *interpretable feature learning* demonstrates how visualization results can be used to generate ad-hoc explanations or numerical evidence. All these visual explanation offer semantic level knowledge extraction approach for better interpretability.

7.1 Visual Analytics

Authors: Kostadin Cvejoski, Johann Kelsch

Visual Analytics (VA) is a field of research that defines and investigates processes that use the human visual system and human knowledge for identifying or hypothesizing about usually hidden patterns in big datasets. While this encompasses a number of techniques that have long been used in data visualization and therefore applied statistics and data science, the need for thorough examination of complex machine learning models like neural networks has driven the development of VA methods in this field as well.

The actual deployment of VA in the ML lifecycle is flexible as is the basis and the goal of its investigation. In the scope of machine learning, VA investigates and develops both the methodology and organizational processes, i.e., feedback loops among different stages of the ML development lifecycle. For the scope of this concise introduction, we refer to the CRISP-ML Lifecycle and focus on the stages Data Preparation, which includes acquisition and labelling of data, Model Engineering, which includes architecture search and training regimen, and Quality Assurance. First and most straightforward, VA can be used as quality check for the Data Preparation stage. Investigations focus on comparing data points and checking consistency of related labels in similar or equal situations. At this point, VA is also often used to perform dataset analysis, e.g., they investigate and quantify imbalanced distributions and recognize missing cases which is, hence, fed back to the data acquisition process. Insights can directly be reported and taken into account within Data Preparation stage itself. Second, by use of a trained model, VA allows for identifying merits and shortcoming of ML models and cluster semantic situations in which they succeed or fail systematically. These findings can subsequently be fed back to the Model Engineering stage, specifically to the model definition or model training process. However, if it becomes apparent that a desired analysis cannot be performed due to lack of input data or insufficient annotations, a corresponding request can be issued to the Data Preparation stage. For example, VA can support the labelling process in suggesting meta data for future analysis and straightforward search and selection operations. This touches upon the field of active learning (c.f. Section 5.4) in which an interplay of a human operator and a trained machine learning models drives the selection of new data points and its labels. Likewise, intuitive visualization techniques bear overlap with the field of Explainable AI (c.f. Section 7.3) which as one part aims to derive clear visual representations for hard-to-interpret ML models.

Shneiderman [672] summarizes the levels of VA process in his mantra: Overview first, zoom and filter, then details-on-demand. This defines the main components of any VA approach: means for visualization, selection and search. For this brief overview, we restrict ourselves to forms of VA that go in line with knowledge extraction or plausibility checking. These usually includes data-driven machine learning but we explicitly do not cover visualization methods as they have long been established in data science. Instead we focus on use cases relevant for automated driving which, by the nature of the field, puts its main emphasis on environment perception, in particular computer vision tasks.

7.1.1 Surveys of Visual Analytics

We cite Sacha et al. [625] who provide an overview and a fundamental categorization of known VA workflows. Hohman et al. [315] provide an extensive survey over VA techniques for the most common use case of deep learning. As one prime example for VA in the field of safety in automated driving, albeit without a special focus on machine-learning-driven vehicle functionality, we refer to the toolset SafetyLens by Narechania et al. [507]. Chatzimparmpas et al. [104], on the other hand, focus on machine learning approaches but open the field of application to safety-critical problems in general. Yuan et al. [834] provide a survey of VA Techniques for ML where the first-level categories in their proposed taxonomy are: techniques before model building, techniques during model building, and techniques after model building. Further, for each category they provide example analysis task and the most recent works.

7.1.2 Visual Analytics in Environment Perception

In particular computer vision but also 3d lidar or radar perception are prime examples for the use of VA in robotics and automated driving. Gou et al. [255] develop and investigate a VA system for their sophisticated traffic light classification model. Computer vision models in automated driving are investigated by Bojarski et al. [68] who present a real-time capable visualization method for CNNs and verify its potential on several well-known datasets from automated driving. Similarly, Wang et al. [775] present their CNN Explainer which aims explicitly at teaching the functionality and stepwise transformations of convolutional neural networks. Liu et al. [445] target the susceptibility of deep neural networks to adversarial attacks and allow for visualizing the set of neurons in the network that were fooled by the attack. In doing so, they enable the human operator to find prevalent adversarial attacks to a network and work out techniques to build up robustness against them. Ma et al. [468] address the problem of domain shift and aim to reveal the knowledge that two models share due to transfer learning.

7.1.3 Deriving Symbolic Knowledge

Due to their need for straightforward and low-dimensional data representations, VA is often deployed to derive human-understandable rules in datasets or model behaviour. Cao and Brown [90] developed a system that combines machine-learning-based rule deduction with a visualization of the corresponding data and the degree of satisfaction of automatically derived rules. This enables the human operator to identify promising, semantically meaningful relations without being an expert in machine learning. Xie et al. [803] present a system for the causal analysis of predictions with a small number of semantically meaningful features. Their system is able to incorporate and convey uncertainty of the decision making process and aims to help decision-makers with multi-causal data. Andrienko et al. [17] provides theoretical definition for the concept of a pattern as a combination of multiple interrelated elements of two or more data components that can be represented as whole object. This definition raises a range of interactive analytical operation for discovering knowledge. Ge et al. [241] propose

the visual reasoning explanation framework (VRX) that gives causal understanding of a model's inference step. The method has the main contributions: (i) understanding to what an NN pays attention to using high level concept graphs and their relationships; (ii) to explain the NN's reasoning process they use GNN-based graph reasoning network. In their paper they also empirically show that their proposed framework can answer questions like "why" and "why not".

7.1.4 Applications

VA is a field where visual techniques are developed to better understand the data, the building of the models and the decision the models make. In a safety critical domain like autonomous driving, transparency and accountability of the models is of great importance. The increased utilization of DNN in the autonomous driving domain requires developing techniques that will give answers to the questions "Why" and "Why not" the model came to certain decision. Using VA techniques in all stages of the ML pipeline can help experts gain better understanding of the strengths and weaknesses of the object detection models they build. VA systems opens a door for building human-in-the-loop AI systems which will help to develop better models for perception. It will give us visual explanation for example "why an object was detected as person" and why not as a dog. By using techniques from VA we will be able to build better dataset for perception, situation interpretation or planning tasks ([44], [55], [107], [357]).

7.2 Saliency Maps

Author: Christian Hellert

Methods for heat mapping, pixel attribution or saliency maps are methods to obtain interpretable visual representations. These methods are generating heatmaps that indicate the relevance of the input pixels for a certain input sample or a set of input samples towards the output. Figure 10 shows two examples of saliency maps. Thereby, these maps can be directly used for manual interpretation or debugging. According to [650], there are two kinds of methods regarding locality, which are local and global locality in the context of Explainable Artificial Intelligence (XAI). Local methods explain a ML model only on a subset of samples or for a specific behavior on a restricted input data space. Saliency map methods are belonging to the local locality category and often local methods are used on very similar samples to evaluate if a model uses the same clues for a prediction. For example, when classifying pedestrians, specific regions like the head or the body are assumed to be important for the classification result. Global XAI methods try to explain the behavior of a complete ML model. As an example, for the task of image classification, for a class it is expected that one class consists of multiple parts. Each of these parts are treated as concepts and the global method will reveal the shared concept between all samples of a class. The concepts itself can be semantically interpretable and in the case of the class vehicle concepts can be tires, windows, or lights. Section 7.3 provides more details on global XAI methods.

Furthermore, saliency map methods can be model-specific or model-agnostic [496]. Model-specific methods can only be used for a specific kind of model, e.g., a CNN, while
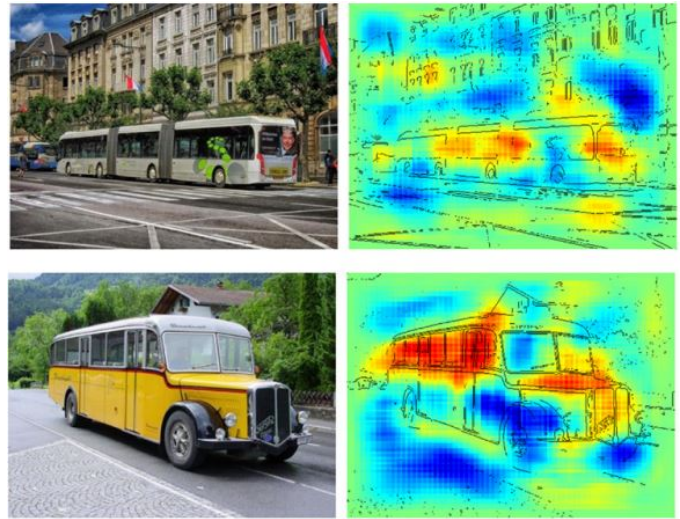

Fig. 10: Examples of saliency maps for the task of image classification [31]. Note that reddish and bluish pixels represents positive and negative attribution respectively.

model-agnostic methods treat the ML model as a blackbox. Typically, model-specific methods are using gradient and model-agnostic methods are using perturbation approaches. Table 3 gives an overview of the saliency map methods along with the type and the task, which will be discussed in the following sections.

7.2.1 Model-specific Methods

One of the first saliency map methods was introduced by Simonyan et al. [686] and called image-specific class saliency or often called vanilla gradients. The method generates the saliency map for classification tasks by computing the derivative of the class score with respect to the input image. The interpretation is that the method seeks for the input pixels that needs to be changed the least and affects the class score the most, which is equal to the magnitude of the class score derivative. One problem of the image-specific class saliency method is the saturation problem [676], which was solved by [838], by redefining backpropagation for ReLU layers, so that only positive error signals are backpropagated. In addition, Springenberg et al. [694] combined both approaches, resulting into a method called Guided Backpropagation.

Another method was introduced by Selvaraju et al. [654] and is called Gradient-weighted Class Activation Map (Grad-CAM) and is able to generate saliency maps for different tasks as long as a convolutional layer is involved in the architecture. Typical tasks involve image classification, visual question answering and image captioning. The method starts by performing a forward step following by setting the class gradient of interest (layer before soft-max) to one and all others to zero. Then, the gradients are backpropagated till the first occurring convolutional layer. The gradients are then globally averaged pooled and afterwards a ReLU is applied to obtain a class saliency map. Grad-CAM can also be used for object detection without modification, but for very small objects that are very close nearby, it will not be possible to distinguish between them, since Grad-CAM stops at the

TABLE 3: Overview of sailency map methods

Method	Type	Task	Reference
Image-Specific Class Saliency	specific	Classification	[686]
DeconvNets	specific	Classification	[838]
Guided Backpropagation	specific	Classification	[694]
CAM	specific	Classification	[870]
Grad-CAM	specific	Classification	[654]
SIDU	specific	Classification	[501]
Grad-CAM++	specific	Classification	[103]
Score-CAM++	specific	Classification	[755]
SmoothGrad	specific	Classification	[690]
Integrated gradients	specific	Classification	[701]
XRAI	specific	Classification	[368]
DeepLIFT	specific	Classification	[676]
LRP	specific	Detection	[31]
Deep Taylor Decomposition	specific	Classification	[497]
CRP	specific	Detection	[731]
LIME	agnostic	Classification	[599]
RISE	agnostic	Classification	[564]
D-RISE	agnostic	Detection	[565]
Anchors	agnostic	Classification	[600]
SHAP	agnostic	Classification	[461]
Explain to Fix	agnostic	Detection	[263]

last CNN-layer, which has a considerably lower resolution compared to the input data or image. Therefore, some of the spatial information gets lost. Note that Grad-CAM is the generalization of CAM [870]. There are also some improved versions of Grad-CAM: Chattopadhay et al. [103] propose a method that gives better object localization and improved heatmaps, when multiple object instances occur and in [501] a method with a more precise heatmap is described using a similarity difference mask, which is combined with a heatmap to generate the final heatmap. In addition, Wang et al. [755] introduced a class activation mapping method, that removed the dependency on the gradients by using a linear combination of weights and activation maps.

For all gradient-based methods, an extension is described by Shrikumar et al. [690], where noise is added to the input image. The resulting batch of saliency maps is then averaged to obtain the final heatmap. The intuition behind this is that gradients in DNNs are fluctuating on slight changes in the input. Hence, this method smoothes the gradients and is therefore also called SmoothGrad.

There are also methods that propagate completely back from output to input. One of these methods is called Layer-wise Relevance Propagation (LRP) [31]. LRP performs a decomposition of the classifier output into sums of feature and pixel relevance scores. The relevance scores are propagated from output (classification results) through the layers (features) to the input, i.e., pixel level. Thereby, the relevance score is obtained by a set of constraints resulting into several relevance functions. Another way of obtaining the relevance scores is proposed by Montavon et al. [497] and called Deep Taylor Decomposition, where the outputs of the neurons are rewritten in first-order Taylor expansions. This allows the computation of partial relevance scores for the neurons in previous layers. LRP was also adapted for object detection for the special case of a SSD network architecture in [731].

Many gradient-based saliency map methods have the problem, that sometimes the resulting heatmap looks like noise. One of the underlying problems is that well-trained models are flat in the vicinity of an input. This problem

is addressed in [701], where a method called Integrated Gradients (IG) is proposed. IG introduces a baseline or reference, which can be in terms of an image classification task, a black image. The baseline is a kind of reference point where basically no information exists. From this point along a straight line to the input image, the gradients are calculated and integrated. Practically, the integration is replaced by stepwise approximation. An adaptation of IG is described in [368] and is called XRAI. XRAI uses two baselines, a black and a white image, to compute the attributions for a provided image. In addition, the image is segmented into regions and the final saliency map is obtained by adding regions in dependency of the summed attributions. Another method that uses a reference is DeepLIFT [676]. In contrast to IG the contribution of a feature towards the output is computed by the difference to the reference using multipliers, which are basically a modified variant of partial derivatives. In addition, some rules are proposed to assign the contribution scores for different layer types.

7.2.2 Model-agnostic Methods

Apart from gradient-based methods, there are also methods that perturb the input in order to estimate the importance of the input to the output. Ribeiro et al. [599] introduced a method called Local interpretable model-agnostic explanations (LIME), which learns a surrogate model that can explain a blackbox model locally. Thereby, an input sample is perturbed resulting into multiple samples in the proximity of the original input sample. Afterwards the predictions from the blackbox model are obtained and finally an interpretable ML model (e.g., a decision tree) is learned. LIME is also applicable for image data, by perturbing the image via superpixels. A superpixel is a region in the image that belongs together by similar color values and can be blinded out by setting the superpixel to a defined color value (e.g., gray or black). Another method that also perturbs the input, but does not learn a surrogate model, is proposed by Petsiuk et al. [564] and is called Randomized Input Sampling for Explanation (RISE). The method perturbs input

images by randomly generated masks, which blind out input pixels, resulting into multiple images, where different pixels are blacked out. Then, the predictions are obtained from the image classification model and a weighted sum multiplied by the respective mask generates the saliency map. This method was also adapted for object detection in [565]. Furthermore, another perturbation-based method was published in [600] and is called Anchors. Like RISE, the input is perturbed, but the predictions are then used to generate rules, the anchors, containing model features along with a coverage and precision. The method can be applied for image classification, but unfortunately not for object detection.

An alternative to perturbation-based methods are methods that uses Shapley values [668]. Shapley values are inspired by game theory and measure the contribution of each feature or a feature group to a respective result. The method SHapley Additive exPlanations (SHAP) [461] uses Shapley values to let features be present or absent for calculating the contributions, by random sampling. Afterwards the predictions are obtained from the original model along with the weights and then a linear model is trained to finally obtain coefficients of the linear model. A further extension of SHAP is introduced by Gudovskiy et al. [263] for object detection tasks.

7.2.3 Applications

Almost all methods listed in Table 3 can be used to gain local explanations for *perception* components. The majority of the methods can be applied to models solving classifications tasks, but some of them are already, or can be, adapted for object detection, which is the primary task to solve for perception components in automated vehicles. Often, the local explanations are used to debug or improve the ML models´ performance, but the results from those methods can also give hints about robustness of the perception component, when the input is perturbed (e.g., dirt on the camera lens). Of course, there is in general one major disadvantage of all the methods: they can only give local explanations to specific inputs and it is not possible to generalize the observations. But there are also attempts to obtain global information from local saliency maps [411]. Furthermore, especially for perception also the computational costs for the saliency map methods are considerable. Model-agnostic methods, like RISE or LIME, require several forward evaluations of the ML model, while structure-based or gradient-based methods, e.g., LRP, only require one backpropagation of the outputs. In addition, smoothing approaches as IG or SmoothGrad, require again some forward and backward runs to ensure stability.

Saliency maps are not directly applicable for *situation interpretation*. On the one hand, saliency maps can be used to gain interconnections between objects, where certain features contribute to multiple object instances. On the other hand, a heatmap can also serve as an additional source of uncertainty. Of course, the drawbacks are additional computational costs. For *planning* components, saliency map methods are also applicable, if they can handle tabular data, which is true for some of them. Therefore, it is possible to estimate the importance of object instance with respect to the predicted trajectory of the automated vehicle. This can give insides

into the behavior of the automated vehicle and other road participants. Formally, it could be possible to perform sanity checks if the predicted trajectory is compliant with a set of rules. Furthermore, saliency map or explainability methods can be used to reveal, which objects influenced the trajectory.

7.3 Interpretable Feature Learning

Authors: Erwin Kraft, Leonie Kreuser

The performance of vision based machine learning models strongly depends on their feature representations. Early works in computer vision focused on the development of hand-crafted features such as Histograms of Oriented Gradients (HOG) [146]. With the advent of deep learning, feature representations were automatically learned from the training data and did no longer require manual tuning. Automatically learned features have been shown to outperform classical hand-designed representations [204], [146] by very large margins, especially for image classification [298], [711] and object detection tasks [65], [712]. However, feature representations found in state-of-the-art deep learning models are often difficult to understand and interpret by humans [378], [799]. This is usually caused by their huge dimensionality and complexity. Explainability and interpretability are very important safety requirements as one has to be able to understand the limitations of a safety-critical system before deploying it to a real-world application. The opaque black-box nature of deep learning models therefore makes it difficult to meet such demands. This applies in particular for applications such as AI-enabled automated driving, where any prediction error could have severe consequences.

To alleviate the problem, researchers have introduced a variety of methods to explain the feature representations of deep learning models [378], [799], [245]. According to Wu et al. [799], these can be roughly grouped into local explanations, based on individual data samples, and global ones, which try to derive decision patterns for the samples of entire classes. In the following section, we give a brief overview of current state-of-the-art developments with an emphasis on methods that provide global explanations.

7.3.1 Post-hoc Explanation Methods

To interpret the internal states of pre-trained deep neural networks, Kim et al. [378] introduced Concept Activation Vectors (CAVs), which are used to test whether the features of an image classification model are sensitive to a specific visual concept. A high-level concept is defined by a set of images, for example images showing striped patterns. CAVs are derived by comparing the feature activations obtained from concept images to those of random images. To do this, a linear binary classifier is trained to separate the concept activations from the non-concept activations by constructing a hyperplane. A CAV is then defined as a normalized perpendicular vector to this plane. Using CAVs in combination with directional derivatives, it is possible to measure the presence or absence of a visual concept for a specific category. For example, the method can be used to verify if the visual concept of striped patterns was learned by an image classification model to distinguish

zebras from horses. CAVs are also used by Graziani et al. [260], who investigate their application in medical domains. They extend the method to also include non-binary concepts since these are more suitable to explain the model behaviour to physicians.

One of the drawbacks of CAVs is that the visual concepts have to be manually selected. This makes it difficult to deploy the method in an automated setting. To address this limitation, Ghorbani et al. [245] introduced Automated Concept-based Explanation (ACE), which can be seen as an automated version of CAVs, as it does not need human supervision. ACE starts with segmenting a given set of images from the same class into groups of pixels. In the next step, these segements are passed through a CNN and similar segments are identified by using the Euclidean distance in the activation space of the final layer. After removing some outliers, an importance score is calculated for the extracted concepts, resulting in a set of automatically generated visual concepts.

Another interesting post-hoc analysis method is presented by Wu et al. [799], who utilize adversarial patterns to occlude specific feature representations in the network and thus are able to measure their overall importance for classifying images. It has been shown that small image perturbations (adversarial patterns) can be used to fool classification models into producing wrong predictions [60]. Wu et al. [799] use these patterns to build global feature occluders, which supress the activations of specific features in the network for a given category. Thus, they are able to compute feature importance scores. This allows to investigate how important a feature set is for classifying an image.

7.3.2 Disentangled Representations

An approach to learn interpretable features in neural networks is based on disentangled representations. The idea of disentangled representations is to train neural networks in a way to identify meaningful input properties that do not influence each other. Thus, if one input property changes, all other properties remain largely unaffected. For example, Zhang et al. [851] train an interpretable CNN by forcing each filter in the network to learn a disentangled representation. A similar idea is also investigated by Chen et al. [118] who replace the batch normalization layers of image classification models with concept whitening modules. The purpose of theses modules is to disentangle the latent feature space and align it with pre-defined concepts. This is done by learning a whitening matrix that decorrelates and standardizes the data and maximizes the activations of known concepts along the latent space axes. In general, there exist different ways to achieve disentangled representations [484]. Unsupervised methods extract factors of variation directly from the data while supervised disentanglement uses data with the desired semantic properties. Locatello et al. [451] point out some shortcomings of unsupervised methods and argue that well-disentangled models cannot be learned without supervision.

7.3.3 Interpretable Model Design

It is possible to use a set of pre-defined high-level concepts in the network design. Koh et al. [389] introduced Concept Bottleneck Models (CBMs). Unlike end-to-end models that go directly from raw input to a prediction, CBMs first learn a set of concepts and thereafter use these concepts to make a prediction. The final prediction module has only access to these pre-defined concepts (information bottleneck). Therefore, it is possible to assess to what degree a concept contributed to the decision and if the decision made by the model follows human reasoning. It is also possible to create a concept bottleneck using variational autoencoders. For example, Mijolla et al. [484] develop an explainability framework based on an encoder-decoder architecture, where the encoder module aims to reduce raw input features of high-dimensional data to compact, semantically meaningful and thus more interpretable latent features.

Interpretable models are also learned by Chen et al. [108] who introduce prototypical part networks (ProtoPNet). The idea is to classify images by identifying object parts that can be matched to prototype parts of object categories. For example, if the input image shows a bird, the model identifies parts of the bird (e.g., the head) that are matched to the parts of a prototypical bird. A prediction is then made based on the weighted combinations of the part similarity scores. Since the prototypical parts are learned by a clustering method, it is possible to inspect the clusters and assign them to high-level concepts. In this way, a model is created that does achieve comparable results with some of the best-performing deep learning models in terms of accuracy and is also more interpretable. However, some shortcomings of ProtoPNet are demonstrated by Hoffmann et al. [314] who claim that image patches that look similar to a ProtoPNet might not necessarily look similar to a human. In their experiments, they show that the models do not perform well in the presence of compression noise.

7.3.4 Applications

Interpretability is a very important prerequisite for object detection tasks in autonomous driving. Detection (2D or 3D) usually resembles the backbone of camera based scene perception. As autonomous driving is a safety critical application, the need for interpretable object detection arises. One of the challenges is that many methods in the literature focus exclusively on image classification tasks. It would be therefore interesting to investigate how these methods could be extended or improved for object detection tasks.

8 KNOWLEDGE CONFORMITY

Autonomous vehicles are safety-critical systems, which means that their malfunction potentially has severe consequences, e.g., when a pedestrian is overlooked by the detection system. We thus must ensure that they operate safe and reliable. In particular, they should conform with existing safety principles and knowledge. One such principle is the identification and handling of *uncertainties*, i.e., factors that potentially cause the system to behave in unpredictable ways. We provide an overview over existing concepts and methods for the estimation and assessment of uncertainties in Section 8.1. Another principle is interpretability, i.e., humans should ideally be able to understand why a system made a particular decision. To improve in that regard, the decision making process of a DL system should be more aligned with human decision-making of which *causal reasoning* is a central component. Therefore, we discuss methods for

infusing causal reasoning into DL systems in Section 8.2. Another aspect is conformity with existing knowledge about the environment of the autonomous vehicle. In particular, the environment is subject certain *rules*, e.g., traffic regulations, physical laws or human common sense, and Section 8.3 discusses concepts and approaches regarding rule conformity.

8.1 Uncertainty Estimation

Authors: Maximilian Alexander Pintz, Christian Wirth, Sebastian Houben

Conventional machine learning algorithms, like neural networks, are only able to predict a single best estimate $y = f(x, \omega)$ (with ω as the network parameters). By default, they do not provide any probability of this single estimate being correct neither do they provide a probability for all of the possible estimates of being correct, i.e., a distribution over the output space. Therefore, neither (reliable) probabilities of correctness for classification tasks, nor a measure of spread of the prediction for regression tasks are available. However, these pieces of information are highly relevant for safety-critical applications like autonomous driving. A measure of correctness or spread allows us to determine if and to what extent it is safe to use a prediction for downstream tasks. Additionally, a system might recognize when the input deviates from the ODD and can therefore likely not be handled correctly. Furthermore, depending on the use case, it is possible to use numeric limits to "err on the side of caution" or to "err on the side of being optimistic". As an example, when performing visual object detection, we should be conservative about "free parking" signs (e.g., only predict if sure) but should predict pedestrians even if we are not sure. Especially in urban scenarios, we also have to factor in the uncertainty over the exact position of a road user and keep some distance, based on the possible localization error. When considering trajectory predictions for other road users, we also need to be able to react to highly different possibilities (like turn vs. drive straight), which requires multiple estimates with according probability of correctness. Furthermore, several advanced machine learning methods, like active learning (c.f. Section 5.4) or continuous learning (c.f. Section 5.2), explicitly require some form of confidence or uncertainty estimates.

In the literature [336], two different types of uncertainty are usually considered: aleatoric and epistemic uncertainty. Aleatoric uncertainty relates to the uncertainty that is inherent to the data, meaning it can not be "explained away" with better models or more data (e.g., observation noise or information that is lacking to solve the problem optimally). Epistemic uncertainty is induced by the limitations of our machine learning model(s) and their training and can theoretically be eradicated with more data or better approaches.

8.1.1 General Methods

In uncertainty estimation, we seek to find distributions that describe the uncertainty in our machine learning models. The distributions are typically in the space of model parameters (for describing epistemic uncertainty) or in the output space (describing aleatoric or combinations of aleatoric and epistemic uncertainty). Uncertainty in the output space is modeled by replacing scalar output values $y = f(x, \omega)$

with a distribution $p(y|x, \omega)$ (see likelihood optimization in Section 8.1.2). For learning distributions in the parameter space, we usually compute the posterior probabilities of parameter values, given the (training) data by applying Bayes theorem

$$p(\omega|\mathcal{D}) = \frac{p(\mathcal{D}|\omega)p(\omega)}{p(\mathcal{D})}, \tag{17}$$

with $p(\mathcal{D}|\omega)$ being the probability of the (training) data $\mathcal{D}$ given the parameter values (the joint likelihood), $p(\omega)$ is a prior over the parameters and $p(\mathcal{D})$ is the probability of observing the data. The prior $p(\omega)$ captures an initial belief about the model parameters before any training data is observed and is commonly chosen to be uniform or Gaussian. We can then compute the predictive distribution

$$p(y|x, \mathcal{D}) = \int_{\omega} p(y|x, \omega)p(\omega|\mathcal{D})d\omega, \tag{18}$$

to describe the uncertainty for new datapoints x. However, computing the posterior or the predictive distribution of neural networks directly is computationally intractable and approximations are required. It is possible to empirically approximate the posterior (see MCMC-based methods in Section 8.1.3), but the number of required samples is usually too high for modern deep learning architectures.

Parametric Approximations: The requirement for an empirical approximation of $p(\omega|\mathcal{D})$ can be circumvented by the assumption that it is possible to approximate the posterior distribution with a parametric distribution $p(\omega|\mathcal{D}) \approx q(\omega, \theta)$, like a Gaussian or Laplace distribution. This allows us to only store the distribution parameters θ, instead of the empirical samples and directly sample from the approximation of $p(\omega|\mathcal{D})$ for inference (see Section 8.1.3). This procedure reduces the amount of samples required for a sufficient approximation of the predictive distribution $p(y|x, \mathcal{D})$ (Equation (18)). In some cases (see Section 8.1.2), it is even possible to compute the predictive distribution analytically, without any sampling.

Aleatoric uncertainty is commonly modeled via parametric distributions $p(y|\theta(x))$ that replace the (deterministic) outputs of a network, e.g.,

$$\theta = (\mu, \sigma) = f(x, \omega). \tag{19}$$

Similarly, epistemic uncertainty is often modeled by replacing the network parameters ω with a parametric distribution. Parametric approximations commonly use loc-and-scale family distributions like Gaussian, Laplace or Cauchy distributions. In these cases, the scalar value we replace is the location μ of the distribution while the scale σ allows us to capture the "range" of possible values (e.g., the displacement error for object detection). In the aleatoric case, it is possible to either also use a model output for defining the scale parameter (as in Equation (19)), or to consider it a free parameter, which does not depend on the inputs. The first setting is called heteroscedastic uncertainty, whereas the second variant relates to homoscedastic uncertainty. Homoscedastic uncertainty is input-independent observation uncertainty, e.g., identical for every datapoint, whereas heteroscedastic uncertainty captures data dependence.

Multimodality: By replacing the weights (epistemic) or outputs (aleatoric) with a unimodal distribution, we can capture the variance but not multimodalities. Within the epistemic setting, this is a less severe problem as it is still possible to obtain multimodal predictions due to the model-inherent nonlinearities. There are efforts to employ multimodal parameter distributions, like Gaussian mixtures (c.f. [62]), but this is computationally difficult due to mode collapse and the computational cost of high dimensional distributions. Therefore, independent distributions are often used, see Ensembles in Section 8.1.2. In the aleatoric case, the computational costs are less severe, as the mixture only scales with the number of outputs, not the number of model parameters.

Covariance Matrix Approximation: In the case of parametric approximations of the posterior or multidimensional outputs, fully specifying the multidimensional distribution goes along with storing and learning the full covariance matrix. This can induce scalability issues, especially for networks with many parameters or high dimensional outputs, as the matrix scales quadratically. Therefore, it is often assumed that all distributions are independent, disregarding the covariances. For epistemic uncertainty, this is also known as the mean-field approximation. However, it can be empirically shown that it is highly relevant to also capture the dependence between the distributions [214, 127], meaning approximations of the covariance matrix are of relevance. The lower Cholesky triangular approximation reduces the size of the covariance matrix by a factor of 2, exploiting that the co-variance matrix is symmetric positive definite but still scales quadratically. Therefore, many state-of-the-art approaches use a vector decomposition of the covariance matrix as an approximation [784, 127].

Subspace Approximation: As mentioned, full covariance matrix approximations are usually computationally very costly. This is especially problematic for epistemic uncertainty estimation due the large number of parameters encountered in conventional neural networks. However, even vector approximations (c.f. previous paragraph) are often not sufficient as approximation [398]. As an alternative, one delineates neural networks into two distinct parts: A feature projector which only projects into a latent space and the classifier/regressor itself. In case of computer vision approaches, the feature projector is the backbone network whereas the classifier is only the last layers, the head. This allows us to capture only the uncertainty in the classifier/regressor head, substantially reducing the number of parameters that need to be approximated by a parametric distribution. This approach is in line with conventional machine learning, like Gaussian Processes, which are not using any learned feature projectors. Due to the parameter reduction, it is then possible to approximate the full covariance matrix, which has shown empirically better results than an all-layer vector or mean-field approximation [398]. Additionally, this allows us to build on uncertainty estimation methods used in conventional machine learning, like replacing the last layer with a Gaussian Process [443]. Another area of research tries to use non-predefined sub-spaces, by selecting the model parts where the probabilistic treatment is most important. As example, [155] determines relevant model parameters by considering the induced Wasserstein distance to the full probabilistic solution.

8.1.2 Deterministic Inference

Deterministic approaches do not require Monte Carlo sampling for inference and provide uncertainty estimates with a single forward pass. These approaches avoid the inherent cost of sampling, but may substantially increase the memory footprint and operation count of a network. Therefore, these are often either computationally costly [334] or only work under certain model assumptions, like specific activation functions [795]. In the following, we present general areas of deterministic inference, that are not restricted by severe model assumptions.

Likelihood Optimization: Among the simplest deterministic approaches is letting the network directly compute the parameters of a predictive distribution $p(y|\theta(x))$ (Equation (19)) and using maximum likelihood or a-posteriori optimization for learning the parameters. Such approaches do not require the computation of (costly) posterior distributions $p(\omega|\mathcal{D})$ and thus enable training and inference without any need of sampling, but come at the cost of restricting the predictive distribution to a simple parametric form, such as a Gaussian in regression or categorical softmax-based distributions in case of classification. In the latter case, the uncertainty is characterized by the probability of the predicted class or by the entropy of the estimated distribution. However, softmax-based class probabilities are often poorly calibrated [238] and thus require recalibration [265, 400] or more advanced training procedures besides likelihood optimization.

A natural extension of this approach is to consider the distribution parameter θ to be uncertain as well. Evidential deep learning [14, 656] considers a parametric distribution on θ, i.e., $p(\theta|\eta(x))$ that takes the form of a Dirichlet distribution (classification) or a Normal-Inverse-Gamma distribution (regression). This enables new ways of measuring uncertainty and taking other types of uncertainties into account, such as epistemic uncertainty. These measures include the variance or entropy of θ or the mutual information between y and θ. The distribution parameters are learned by optimizing the marginal likelihood $\sum_i p(y_i|\eta(x_i))$, a procedure also known as Type II maximum likelihood optimization [63]. A related approach are prior networks, with the main difference lying in the optimization procedure. Prior networks [472] optimize a KL divergence between $p(\theta|\eta(x))$ and hand-crafted distributions centered on the ground-truth. Besides the given training inputs, the model is also explicitly trained on chosen out-of-data inputs (e.g., corrupted training inputs) to encourage the variance of θ to increase for inputs that are atypical relative to the training inputs.

Laplace Approximation: The Laplace approximation [469] approximates optimal Gaussian scale parameters given a mean. Meaning, we compute a point estimate ω, minimizing $\text{NLL}(\omega) = -\sum_i \log p(y_i|f(x_i, \omega))$, as in conventional deep learning, but with an L_2-regularization (or other prior-based terms) added to the loss. The resulting parameters ω are Maximum-a-posteriori (MAP) estimates that serve as the mean of a Gaussian approximation to the posterior $p(\omega|\mathcal{D}) \approx q(\omega, \theta)$ (c.f. Parametric Approximations in Section 8.1.1). The variances can then be estimated by computing the Hessian at the MAP. However, this method is

computationally very costly due to the approximation of the Hessian. This applies to the computational cost in terms of operations as it does to memory consumption. Deterministic inference can than be performed using probit scaling or the Taylor approximation [63], using the stored Hessian. Different methods for approximating the Hessian have been suggested [606]. It should also be noted, that it is not required to use deterministic inference approaches, but one can directly sample from the obtained posterior, comparable to the inference step typically employed in Variational Inference (VI) (c.f. variational inference in Subsec. Section 8.1.3).

Ensembles: Ensembles subsume all approaches that combine the outputs of several models to obtain several samples that represent a distribution over the output space. Usually this involves using techniques for obtaining independent outputs from all models or techniques for properly addressing the dependency among model outputs. In this sense, ensembles constitute the frequentist's approach for providing a distribution over the output space. Ensembles have been and continue to be deployed for better performance and increased robustness but bear a straightforward solution for uncertainty estimation. Special cases comprise bagging, i.e., training the same model type with different parts of the training data, and boosting [217, 291], i.e., selecting and training the next member of the ensemble with the aim of addressing shortcomings of the current ensemble.

Ensembles of neural networks with the same architecture trained with different random initialization are called Deep Ensembles [406]. Such ensembles capture multimodality using the fact that neural networks usually converge to different solutions, based on the initial parameter weights and the optimizer settings. They have been trained and tested with adversarial training which allows for a smoother prediction and a better uncertainty estimation. Hyper-deep Ensembles [786] extend the random initialization of weights to a selection of hyperparameters. However, ensemble techniques are costly in terms of parameters as it is required to store (and train) the parameters for each ensemble member independently. Batch Ensembles [784] try to alleviate this problem by approximating the joint weight matrix over all models by a vector decomposition. The weights of each ensemble member are defined by a shared parameter set, multiplied with a row and column vector, specific for each model. This method is also applicable for parametric approximations, by replacing the row and column vectors with parametric distributions [181].

Kernel-based Methods: An alternative way is the use of Kernel methods, capturing the uncertainty in the space of functions. This allows us to build on uncertainty estimation methods used in conventional machine learning, by replacing the last layer (cf. Section 8.1.1) with a kernel function. Known approaches use RBF Layers [13] or Gaussian Processes [443].

However, it should be mentioned that exact kernel methods, are not applicable in this setting due the high number of training datapoints and the resulting size of the kernel matrix. Therefore, it is not possible to use all training datapoints as inducing points. The authors in [13] suggest using a single prototype per class, whereas [443] uses a random feature approximation of the inducing point space. When using multiple inducing points, it is required to ensure that the deterministic feature projector (everything before the last layer), is preserving the distance between datapoints. Without this constraint, it is possible that the feature projector maps all datapoints of the same class to the same feature vector. This can be achieved by spectral normalization, bounding the Lipschitz constant [443]. Other commonly considered approximations besides the random feature approximation of the inducing point space, are variational Gaussian processes [726].

8.1.3 Sampling-based methods

The ability of deterministic approaches to provide uncertainty estimates with a single forward pass typically goes along with restrictions on the uncertainty distribution (e.g., via imposing parametric models). Sampling-based methods allow to approximate more flexible and complex distributions. A universal tool for this purpose is MC integration. For example, it can be used to approximate an arbitrary complex predictive distribution $p(y|x, \mathcal{D}) \approx 1/N \sum_i p(y|x, \omega^{(i)})$, given N samples $\omega^{(i)}$ from the posterior $p(\omega|\mathcal{D})$. This idea gives rise to multiple approaches that seek to obtain samples from the posterior, that are discussed in the following. In addition, there are also non-Bayesian approaches that employ MC integration, which we discuss at the end of this section.

MCMC-based: A common method for computing an empirical approximation of the posterior distribution is Hamiltonian Monte Carlo (HMC) [179, 508] and its extension No-U-Turn Sampling (NUTS) [313]. The predictive distribution is then obtained by directly inferring a single prediction for every posterior sample. However, these methods are not applicable to batched training and can therefore not be applied to common deep learning tasks. Batch-enabled variants exist, like Stochastic Gradient Langevin Dynamics (SGLD) [425] or Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) [112], but the resulting posterior distributions $p(\omega|\mathcal{D})$ are usually very complex and a large amount of samples is required for an sufficiently good approximation. Obtaining these samples, and/or inferring predictions, is usually computationally to costly for modern neural networks. Note that implementations of HMC, NUTS and SGLD are available in Tensorflow probability and similar libraries.

Variational Inference: Another approach is Variational Inference (VI) [63, 310, 259], that seeks to approximate the parameter posterior using a (simpler) parametric distribution $q(\omega, \theta)$, the so-called variational distribution. This is done by optimizing the (empirical) Evidence Lower Bound (ELBO)

$$\hat{V}(\theta) = 1/N \sum_i \log p(Y|X, \omega^{(i)}) - \text{KL}[q||p(\omega)], \quad (20)$$

with $\omega^{(i)} \sim q(\omega, \theta)$, which effectively minimizes the KL divergence between the posterior and q. For neural networks, computing gradients of the ELBO is possible via the reparametrization trick [382], which sets the sampled parameters $\omega^{(i)} = t(\theta, \varepsilon^{(i)})$ to the result of a deterministic function t and a sample $\varepsilon^{(i)}$ from a distribution which does not depend on any parameters. A common case is that q is a Gaussian with diagonal covariance, which allows for the reparametrization $\omega = \mu + \text{diag}[\sigma]\varepsilon$ with $\varepsilon \sim \mathcal{N}[0, I]$, where we can compute gradients with respect to the variational parameters μ and σ. Applying a VI-based

method to an existing neural network requires changing the loss function to the ELBO and modifying every layer due to the reparameterization. The methods scale to large NNs under the choice of small number of samples (typically one chooses $N = 1$) and feasible q and priors. Several variants of this procedure have been proposed that typically aim at reducing computational costs, faster convergence (eg. Flipout [785]) and enabling the use of more expressive priors and variational posteriors (e.g., Horseshoe priors [246], MVG [456], normalizing flows [597]).

Dropout: Another approach to Bayesian VI in neural networks is MC dropout [225]. This approach utilizes the dropout mechanism [695], which randomly omits units from the neural network in each forward pass. In practice, this is done by multiplying the activations in each layer with a binary mask, whose entries are sampled from a Bernoulli distribution with "drop probability" p. Gal et. al [225] showed that training such a network with a standard L_2-regularized Mean Squared Error (MSE) approximately optimizes an ELBO between a deep Gaussian process model and a variational posterior on network weights, giving further theoretical motivation for the approach. The mean and variance of the corresponding predictive distribution can be obtained by simply computing the mean and variance (plus a hyperparameter-dependent offset) of multiple dropout forward passes. MC dropout has become a popular approach for uncertainty estimation, especially among applied practitioners, as it requires only a small change in network architecture, has low computational complexity and is directly applicable to several different architectures, such as CNNs or RNNs. A drawback of MC dropout is that it employs a quite restrictive variational posterior, that utilizes heavy independence assumptions. In addition, it assumes a fixed-variance Gaussian likelihood, thus allowing it only to describe homoscedastic aleatoric uncertainty. To mitigate this drawback, the authors in [374] proposed to combine MC dropout with parametric likelihood methods, to incorporate heteroscedastic aleatoric uncertainty as well. This method is also the basis for many uncertainty-aware object detection models [285]. Sicking et al. [679, 678] introduced Wasserstein dropout, which directly optimizes the dropout sub-networks to model (heteroscedastic) aleatoric uncertainty. Several other proposed variants include dropout with trainable drop probabilities p [226], different distributions besides Bernoulli [382] and sampling-free variants [568].

Distributional Output: The authors in [127] take a non-Bayesian approach to uncertainty estimation in classification tasks. Classification networks can generally be thought of as generating a utility value u_c for each class c. The network assigns class c to a given input if the corresponding utility u_c is larger than the utility for all other classes. For uncertainty estimation, the framework considers the utility values to be corrupted by random noise ε_c, which allows to assign a probability to each class for having maximum utility. The probabilities reduce to standard softmax probabilities, if the noise is homoscedastic and follows a standard Gumbel distribution for each class. However, the framework allows to generalize this to more realistic types of noise, including noise that is not identically distributed across classes or heteroscedastic noise with full covariance. Due to the higher complexity, the class probabilities cannot be computed in closed form anymore and are thus approximated via MC integration. Regardless, the approach is one of the few highly scalable approaches that incorporate full covariance uncertainty distributions.

8.1.4 Quality Measures

Considering ways to express a prediction uncertainty introduces a number of additional quality measures apart from the original performance metrics of the machine learning model, like classification rate, root-of-mean-squared-error, mean average precision and others. The quality of uncertainty estimates can be evaluated with proper scoring rules [251], which quantify the alignment of the predicted distributions with given ground-truth data points. They attain their optimum when the predicted distribution coincides with the data distribution. Commonly used proper scoring rules include the Negative Log-Likelihood (NLL), brier score (for classification) or the Continuous Ranked Probability Score (CRPS). An alternate way of assessing uncertainty estimates provides the framework of calibration. In general, calibration measures whether class probability or confidence interval estimates conform to the actual model error. A calibrated classifier satisfies that the fraction of correct prediction is p among all class predictions with estimated class probability p. This calibration property can be evaluated quantitatively with the Expected Calibration Error (ECE). It should be noted, that ECE is subject to several pathologies, that should be considered explicitly [522]. A qualitatively assessment is possible via reliability diagrams that compare class probabilities against classification accuracy [265]. Similar notions of calibration also exist for regression models (based on how ground-truth outputs conform with the predicted confidence intervals [400]) and for certain application domains such as object detection models [402]. For a more strict assessment, we might want the calibration property to hold not only across the whole (evaluation) dataset, but also on local regions of the dataset, which is captured by adversarial group calibration [865]. We can also measure the performance of uncertainty estimates in auxiliary tasks, such as out-of-data-detection. Separating true from false detections or in- from out-of-data inputs is a binary classification task, where the classes are determined by thresholding uncertainty estimates. Thus, we can employ standard evaluation methods for binary threshold classifiers, including plotting precision-recall or receiver operating curves for qualitative assessment or computing average precision or the area under the Receiver Operating Characteristic (ROC) curve. Another approach is to compare histograms of uncertainty estimates (e.g., variance or predictive entropy) on inputs from each class (e.g., in-data or out-of-data), as done in [533]. In technical applications the computational complexity, i.e., the overhead for computing the uncertainty estimate when compared to the single-point-estimate, and the latency, i.e., the part from the computational complexity that cannot be efficiently parallelized or pipelined and, thus, result in additional inference time, are taken into account for assessment as well.

8.1.5 Applications

Many of the approaches for uncertainty estimation outlined before apply to a wide range of different models and are

used in numerous different application scenarios. Main challenges of uncertainty estimation in specific application scenarios include a principled treatment of specialized layers, incorporating uncertainty in pre-/post-processing steps, dealing with unusual data structures, leveraging task-specific assumptions and efficiency. In the following, we discuss several works that employ uncertainty estimation in perception tasks. A larger-scoped review of applications of uncertainty estimation can be found in the surveys [3] and [240].

Object detection: A wide bandwidth of models employing uncertainty estimation for object detection tasks have been presented in the literature and we refer to [206] for a survey and comparative study with a focus on object detection in autonomous driving. The approaches cover both 2d and 3d [460] localization of objects as well as different input types such as camera images [487], lidar scans [208] or radar [176]. The main techniques employed by such approaches are typically direct optimization of parametric distributions (e.g., [208, 299]), MC dropout (e.g., [487]) or, based on the work of [374], a combination of both (e.g., [397, 285]). Other works consider ensembling [486] or a spatial clustering of redundant detections [413]. Approaches often differ in the way uncertainty estimates are integrated into non-maximum suppression, a post-processing step to filter redundant detections, or whether anchor-based or anchor-free [417] methods are employed.

Trajectory Prediction: For path planing, it is important to correctly predict the future trajectories of other traffic participants. Naturally this is not possible in a deterministic manner as there are usually multiple options and we can only derive the prediction from the participants past behavior. Therefore, nearly all approaches use multimodal predictions [97, 566, 737, 173] (c.f. Multimodality in Section 8.1.1), usually by reducing the continuous distribution over trajectories (or keypoints) to a categorical distribution over distinct trajectories. These methods may be enhanced by also allowing to predict displacement of the expected trajectory wrt. the predicted trajectory from the categorical distribution. These displacements can be modeled using likelihood optimization (see Section 8.1.2), usually using bivariate Gaussians [97, 737, 632]. This allows to model the aleatoric uncertainty, but not the epistemic uncertainty. Due to the realtime requirements usually involved in trajectory prediction, full Bayesian approaches are uncommon. However, some methods employ Conditional Variational Auto Encoder (CVAE) conditional, variational auto encoder (CVAE) [632] for learning a latent space. CVAEs are in fact a kind of variational inference (Variational Inference in Subsec. Section 8.1.3 method, but only applied to a very limited subspace of the network (see Subspace Approximation in Section 8.1.1).

Other: Parametric distributions and dropout are also utilized in other computer vision-related tasks such as semantic segmentation for single image inputs [502] and video [333], as well as camouflaged or salient object detection [422, 814], which respectively refer to the tasks of segmenting objects that are hidden in the surrounding or stand out for humans on an image. Similar uncertainty estimation techniques are also considered for human pose estimation [57, 264]. To better represent the uncertainty of the rotational part of pose estimates, several works on pose estimation or visual odometry consider employing distributions from directional statistics, such as the von Mises-Fisher [572] or Bingham distributions [160, 560]. Approaches for complete monocular visual odometry also exist that provide uncertainty estimates of trajectories (sequence of position and rotation) based on image sequences. The work of [714] considers object detection based on a sequence of images from multiple camera positions. The approach leverages dropout and Bayesian updating to maintain a posterior class distribution based on multiple images. There is also a similar method [203] that explicitly incorporates localization information into the estimation.

The authors in [238] propose a general framework for probabilistic deep learning that employs Assumed Density Filtering (ADF), an alternative uncertainty estimation method to pure parametric output distributions or dropout. ADF is an efficient approximate form of expectation propagation, where we seek to find the output distribution of the network that arises from stochastic inputs (that, e.g., follow a Gaussian distribution) when all other network components are deterministic. The authors apply their framework to large CNNs to find the optical flow in given video sequences. Dropout-based uncertainty estimation in optical flow tasks is also considered in [416] or [777]. Building on the ADF-based framework, the authors in [455] combine assumed density filtering with dropout to further integrate epistemic uncertainty into the framework. They demonstrate their framework on tasks such as future motion prediction and steering angle prediction.

8.2 Causal Reasoning

Author: Tobias Latka

8.2.1 Background

"*Would I have arrived at my destination earlier, had I taken route A instead of route B?*" Such a query looks familiar to humans and is an instance of human reasoning. The above example can be attributed to the notion of *Causal Reasoning (CR)* [554], a subcategory of *Causal Inference (CI)* [665] that also deals with *Causal Discovery (CD)*, i.e., inferring cause-effect-relationships from data and encoding these in form of a *Causal Model*. As *Causal Discovery* is beyond the scope of this section, we refrain from giving an overview on the topic of *Causal Discovery* here.

There are two common frameworks for the description of *Causal Models* that are mathematically equivalent:

- the *Rubin Causal Model* [316, 619] (also known as the *Neyman-Rubin Causal Model* or the *potential outcome framework of Neyman* [510] *and Rubin* [620])
- the *Structural Causal Model (SCM)*[1] [284].

While the *potential outcome framework of Neyman and Rubin* solely resorts to counterfactual notation, SCMs make explicit use of a graphical representation in terms of a *DAG* which counterfactual expressions can be inferred from (see Figure 11 for an illustration of an SCM). The key characteristic of SCMs

---

1. "Graphical Models serve as a language for representing what we know about the world, counterfactuals help us to articulate what we want to know, while structural equations serve to tie the two together in a solid semantics." [555]
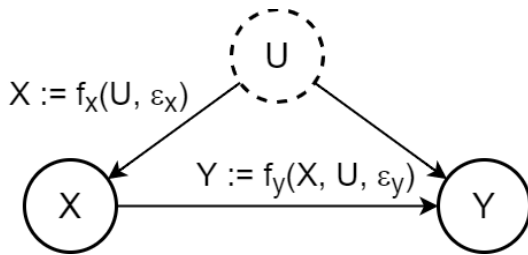
Fig. 11: An SCM consists of both a set of causal mechanisms (f_X and f_Y) and a DAG modeling the flow of causation between variables (directed arrows). The shown SCM is composed of a cause (X) and its effect (Y), which are both confounded by a latent variable (U). Nodes associated with the exogenous noise variables ϵ_X and ϵ_Y are omitted for the sake of clarity.

is that they represent each variable as a deterministic function (f_X and f_Y in Figure 11) of its direct causes together with latent exogenous noise variables (ϵ_X, ϵ_Y and U in Figure 11), whose causes lie outside the SCM.

By definition, *Causal Reasoning* is the process of drawing conclusions from a *Causal Model*, similar to the way probability theory reasons about the outcomes of random experiments. However, since *Causal Models* are thought of as the data-generating process, they contain more information than probabilistic ones and are thus more powerful, as they allow to analyze the effects of interventions or distribution changes on target variables [665].

Questions posed to the *Causal Model* are phrased as *Causal Queries*. They are at the heart of *Causal Reasoning* and can be differentiated into three levels of causation of increasing complexity [554]:

- *Associations* (correlations),
- *Interventions* (actively changing causal mechanisms),
- *Counterfactuals* (retrospective reasoning).

The just described *Causal Hierarchy* is also often referred to as *Pearl's Ladder of Causation* (see Table 4).

On top of that, there are different types of *Causal Reasoning* [554]:

- *Prediction* (reasoning forward in time),
- *Abduction* (reasoning from evidence to explanation),
- *Transduction* (reasoning through common causes),
- *Induction* (from experience to causal knowledge).

Abduction, for instance, is the first step in the three-step process in counterfactual reasoning (abduction, intervention, prediction), as the following example illustrates:

The introductory counterfactual query can be mapped to the graph in Figure 11 and formulated mathematically as $\mathbb{E}(Y_{X=A}|Y', X' = B)$. Here, $\mathbb{E}$ is the expected value, Y is the time of arrival at the desired destination, X stands for the route taken by the driver and X' and Y' are the evidences. The background factor (e.g., the traffic situation) influencing both the driver's decision and the time of arrival is signified by U, which is inferred during the abduction step such that it complies with the evidences. With this information, the expression, $\mathbb{E}(Y_{X=A}|Y', X' = B)$ reads: "*I would have arrived*

at my desired destination at Y instead of Y' o'clock, if I had taken route A instead of route B."

One of the strengths of *Causal Reasoning*, that is closely linked to the above example, is its capability to assess the causal effect of one variable on another from observational data alone, without the need of collecting experimental data. A causal query that is (uniquely) evaluable in this way is said to be *identifiable*, but whether it is *identifiable* highly depends on the causal structure used for answering it. There is a rich literature on theoretical work regarding *Causal Reasoning*, e.g., Do-calculus [332], c-components [353] and identification of joint [331, 339, 674] and conditional interventional distributions [675] even under partial observability [415] for the sake of commonsensical decision making and policy evaluation [554].

8.2.2 Connection to Machine Learning Insufficiencies

Although *CI* and *ML* arose separately, there is, now, an increasing interest in both fields to benefit from the advances of the other [640]. Despite, many ML-methodologies still do not consider causality, but keep modeling correlations between variables, although the maxim "*Correlation does not imply causation*" proves itself true on every occasion. Indeed, the open problems of today's ML-systems like insufficient *robustness*, *generalizability* and *explainability* are closely related to the lack of causality considerations. See [640] for an excellent review on this topic. Further discussions from various domains on why causality is linked to ML-deficiencies and on how to equip ML with *Causal Inference* can be found in [573, 95, 639, 601, 555, 556].

We will next discuss already existing applications of *Causal Reasoning* in the field of *Machine Learning*. For this purpose, we highlight a couple of publications from various domains that take aspects of *Causal Reasoning* explicitly into account for the sake of overcoming today's ML-deficiencies. Eventually, we get on to its potential applications in the context of *AD*.

8.2.3 Applications

Policy Search / Decision Making: As for policy search in *MBRL* (see Section 4.6 for more information on RL), Buesing et al. [84] present an algorithm called *Counterfactually-Guided Policy Search (CF-GPS)* that utilizes counterfactual reasoning explicitly for improving an agent's policy. Thus, it effectively combines RL-concepts with those from Causal Inference to enhance the sampling efficiency on required experiences. Moreover, CF-GPS is thought of as an extension of *Model-based Policy Search (MB-PS)* algorithms. In CF-GPS, the environment is modeled as a *Partially Observable Markov Decision Process (POMDP)*, but cast into the SCM-framework, where its exogenous variables summarize all aspects of the environment that cannot be influenced by the agent and where its causal mechanisms connect state-action-pairs to new states. The central idea of CF-GPS is that, instead of running an agent on scenarios sampled from scratch from a model, one infers these scenarios in hindsight from given off-policy data (real experience, see Section 4.6 for more information on off-policy RL). Then, the agent's policy is evaluated and improved on model predictions of alternate outcomes consistent with both the inferred scenarios and the given or learned causal mechanisms,

TABLE 4: The *Causal Hierarchy*. Questions at level i can be answered only if information from level i or higher is available. Reproduced from [555].

Level (Symbol)	Typical Activity	Typical Questions	Examples
1. Association $p(y\|x)$	Seeing	What is? How would seeing X change my belief in Y?	What does a symptom tell me about a disease? What does a survey tell us about the election results?
2. Intervention $p(y\|do(x),z)$	Doing, Intervening	What if? What if I do X?	What if I take aspirin, will my headache be cured? What if we ban cigarettes?
3. Counterfactuals $p(y_x\|x',y')$	Imagining, Retrospection	Why? Was it X that caused Y? What if I had acted differently?	Was it the aspirin that stopped my headache? Would Kennedy be alive had Oswald not shot him? What if I had not been smoking the past two years?

but under counterfactual actions (i.e., actions that had not actually been taken), while keeping everything else the same. Concretely, while MB-PS algorithms usually sample from a prior trajectory distribution for model rollout, CF-GPS samples trajectories from a posterior distribution under a certain model that is anchored in off-policy data from another policy.

Based on these ideas, Hart and Knoll [286] propose a *Counterfactual Policy Evaluation (CPE)* algorithm with which the authors assess the safety of an ego vehicle's policy prior to its execution in the actual world using counterfactual worlds. In a counterfactual world, the policy of at least one vehicle near the ego vehicle is replaced to estimate the impact another vehicle's policy may have on the ego vehicle's decision. To validate their method, the authors learn a policy using a SAC [277] RL-approach in a lane merging scenario. In the application-phase, the ego vehicle's policy will only be executed after CPE had been performed and if the policy is found to be safe. Here, safety means that the expected collision rate of the current policy does not exceed a predefined threshold. Otherwise, the ego vehicle follows a simple lane-following policy.

Unlike Buesing et al. [84], who use counterfactuals to approximate draws from the interventional distribution, Oberst & Sontag [526] treat the counterfactual distribution as the primary object of interest. The authors point out that evaluating an RL-policy on observational (off-policy) trajectories is challenging, as it is often prone to issues such as confounding and a lack of introspection. To overcome the latter, they propose to pair observed with counterfactual trajectories, so that a domain-expert can "sanity-check" a proposed policy, especially in high-risk settings, e.g., healthcare. In particular, Oberst and Sontag introduce an off-policy evaluation procedure based on counterfactual distributions in POMDPs with discrete states and actions and stochastic state transitions. There, they try to answer counterfactual questions such as: "*How would a specific trajectory have changed, had the policy been different and had all other variables (including noise) been the same?*" The main idea behind their approach is that if the counterfactual trajectory is unreasonable given an observed trajectory (for instance, an individual patient's full health record), the policy might be flawed. The authors finally demonstrate their method on a synthetic environment of sepsis management. From comparison of divergent counterfactual and observed trajectories, they reveal dangerous failure modes of the learned policy, even when off-policy evaluation is overly optimistic.

Richens, Lee, and Johri [601] compare the accuracy of associative diagnostic algorithms to their counterfactual counterparts for the purpose of clinical decision making. The authors argue that the inability of associative algorithms to disentangle correlation from causation can result in sub-optimal or even dangerous diagnoses. Reformulating diagnosis as a counterfactual inference task, they show that counterfactual analysis, in contrast to standard associational approaches, yields expert clinical accuracy on electronic health records, especially for rare and very-rare diseases. In their analysis, they compare the diagnostic accuracy of ranking diseases using traditional associative posterior probabilities to the results obtained from counterfactual inference. To be more precise, the authors set up a disease model in terms of a three-layer Bayesian Network with a Noisy-Or-operation connecting diseases to symptoms. A team of doctors and epidemiologists parameterize this model using prior knowledge, while prior and conditional probabilities of disease and risk factors are attained from multiple independent medical sources and doctors.

Explainability / Interpretability: In an attempt to find human-understandable explanations for the incentives driving the behavior of *MFRL*-agents, Délétang et al. [158] illustrate a methodology for investigating the underlying causal mechanisms. In particular, they show that each question cannot be addressed by pure observation (Level 1 in the *Causal Hierarchy*) alone, but instead requires conducting experiments with systematically chosen interventions (Level 2 in the *Causal Hierarchy*) in order to get the correct causal evidences. For this purpose, an analyst formulates a causal hypothesis (SCM), conducts experiments with carefully chosen interventions and confirms the predictions made by the resulting causal model. Similarly, [92, 195] use structural causal incentive diagrams, a hybrid of influence diagrams and SCMs, to reason about the causal pathways of decisions in the service of maximizing certain utility functions.

In another approach, Madumal et al. [470] use an SCM to derive causal explanations of the behavior of Model-free Reinforcement Learning-agents. In particular, they propose a method that learns an SCM during RL, where the causal structure is given, but where the SCM's structural equations are learned as multivariate regression models. Subsequently, the learned SCM generates explanations of behavior from counterfactual analysis.

A more direct approach of using SCMs in DL is presented by Pawlowski, Coelho de Castro, and Glocker [551]. The authors propose a framework capable of learning SCMs with *DNNs* as causal mechanisms. In particular, they focus on Markovian causal models with mutually independent exogenous variables and with no latent confounders. Additionally, the authors present an approach capable of inferring

exogenous noise (potentially up to approximation) using a tractable evaluation of the abduction step via (amortized) variational inference or normalizing (gradient) flows. To that end, Pawlowski et al. assume that each structural function in the SCM can be expressed as a functional composition of an invertible (low-level) function (e.g., a cumulative distribution function as used in the reparameterization trick) and a non-invertible (high-level) branch (e.g., a CNN or a probabilistic decoder). The proposed method shows good performance on both real (brain MRI scans) and simulated data (Morpho-MNIST).

In contrast to the formal approaches, Dasgupta et al. [150] recently showed that *Causal Reasoning* can also emerge via Meta-Reinforcement Learning (see Section 5.3 for more information on Meta-Learning) when no knowledge of causality is explicitly provided to an agent. The agent is trained with Model-free Reinforcement Learning to solve a range of problems backed by (latent) causal structures, which themselves are represented as *Causal Bayesian Networks (CBNs)*. They split the experimental setup into an *information phase* and a *quiz phase*. During the *information phase* the agent is allowed to collect information by observing or interacting with the environment (CBN) to acquire causal knowledge of the environment. In the subsequent *quiz phase*, the agent is asked to answer causal queries along all three rungs of the *Ladder of Causation*. The authors conclude that the trained agent is able to perform *Causal Reasoning* implicitly from observational data, even in novel situations, and can even make counterfactual predictions without having access to a full causal inference machinery.

For a more comprehensive survey of both traditional and frontier methods in learning causality and relations along with the connections between causality and ML, see [267]. Another survey [500] discusses interpretable / explainable models from a causal perspective and contrasts these with traditional interpretability approaches in ML.

Autonomous Driving: As advertised, we next examine the potential applicability of *Causal Reasoning* in the domain of AD. In particular, we will explore the three main subtasks in AD: *Planning*, *Situation Interpretation* and *Perception*.

According to Pearl [554], a *plan* is an ordered sequence of value assignments to the control variables (actions). *Planning* itself is an optimization procedure of a sequence of concurrent or sequential actions, where each action may be influenced by its predecessors in the plan, with a view of achieving a specified goal (like arriving at a desired destination or preventing accidents). Therefore, *Planning* is closely connected to causality. *Causal Reasoning* can be employed not only for planning the ego vehicle's trajectory, but also for predicting physically reasonable trajectories of other vehicles in its immediate vicinity. A prerequisite for this approach is that existing knowledge about the vehicles' motion (governed by physical laws) is modeled by an SCM in both diagrammatic and algebraic form. As the resulting SCM serves as a lightweight simulator of the vehicle dynamics, it can be leveraged to generate physically reasonable trajectories by repeatedly calling the SCM on sequential state-action tuples, one for each time step. As for planning, actions in the plan can be either dictated by the ego vehicle's learned policy (a mapping from state to action) or enforced by hard interventions on the SCM (setting the action to a fixed value irrespective of the policy's output). The obvious advantage of using an SCM modeling the vehicle dynamics is that an agent does not have to interact with the environment at all, but can rather *act* (in the sense of *interventions*) in an imagined space that is created by the SCM's predictions. For instance, the CF-GPS method presented in [83] could potentially be transferred to the AD domain, but must first be adapted for application in continuous action spaces.

Besides *Planning*, *Causal Reasoning* naturally lends itself to *Situation Interpretation*, insofar as that a causal analysis of a particular plan assesses an action's impact on the trajectory's subsequent evolution. There are basically two forms of causal analyses that matter here: The *counterfactual* and the *interventional* analysis. For instance, the *counterfactual* analysis of an existing trajectory, that is recorded under a certain policy, can unveil explanations / causes (like specific actions taken at a certain time step) for events / effects (such as colliding vehicles after an elapsed time interval). The latter only works as a *post hoc* method, *after* an event has occurred, while the *interventional* analysis (an *ex ante* approach) does not make use of a recorded trajectory. As opposed to the counterfactual analysis, the interventional analysis assesses the importance of certain actions for future events in specific situations and is solely based on the assumptions encoded in the SCM. On the whole, *Causal Reasoning* can give answers to questions like *"Which actions have led or might lead to an accident in a certain situation?"*

As for perception, things are not that easy, since *Causal Reasoning* struggles to derive high-level concepts (like objects in images) from low-level features (e.g., image pixels). However, applications of *Causal Reasoning* to vision tasks can be found in, e.g., [767, 575, 818]. In Wang et al. [767], for instance, high-level features are extracted by an R-CNN, while the high-level features are post-processed and used to make these plausible by tools of *Causal Reasoning*, which are also part of the learning objective.

8.3 Rule Conformity

Authors: Abhishek Vivekanandan, Etienne Bührle, Hendrik Königshof

8.3.1 *What is Rule conformity?*

According to ISO 9000:2000 norms, conformity is defined as fulfilling certain requirements. Rules are a set of formal codes derived from local laws and socio-behavioural structures while effectively considering the Operational Design Domain (ODD) of the algorithm. Conforming to rules is a crucial factor while attempting to categorize the risk an action could lead to, making the Autonomous Vehicles (AVs) safe not only for the passengers but also for other traffic participants like pedestrians and cyclists etc.

Formalizing a subset of traffic rules and associating priorities to the vehicles not only achieves compliant behavior but also maintains the vehicle in a safe state, as can be seen from the works [608].

8.3.2 Rules and Safety

From the literature works [494, 520], one can define mathematical rules to formalize common-sense notions/behaviors such as maintaining a safety gap between vehicles, safe behavior to merge and as such to avoid maneuvers which could lead to possible collisions. Depending upon the ODD one could broadly categorize the behaviors based on formal rules; which are stated by law and could also be derived from the environment, these include stop signs, appropriate speed markings, solid double or dotted lane markings. Each of these briefly define laws which demand an integration into the learning process of the model. In contrast to rules which are explicitly stated by law, informal rules, on the other hand, are difficult to formalize but expected to be followed to avoid collision or (and) increase the comfortability of the passengers. For example, to leave way for an emergency vehicle, it is acceptable to cross the solid line or pass a red light at an intersection paving a gap for the ambulance to pass through. In this situation if there exists a maneuver where breaking the rule does not endanger both the passenger and the Vulnerable Road Users (VRUs) then under defined priority this maneuver could be executed with strict adherence to safety.

8.3.3 Always be compliant, but have common sense

When to follow or to relax the rules is a complex decision process for AVs to comply with, since recognizing the situation itself which lead to such behavior requires *General Intelligence* which could be seen in humans due to the implicit notion of common sense. Let us consider the example of achieving a collision free maneuver in case of an unanticipated merge by an adjacent vehicle driving on the right onto our lane. To generate a compliant behavior, the algorithm should respect the properties iterated below. This hierarchy is defined from the perspective of Ego Vehicle, following a simple bicycle model.

1) Detects and tracks the surrounding objects via forecasting networks, which considers the uncertainties based on the object class. For example, a cyclist could emit different states when compared with pedestrians whose degrees of movement are higher.
2) Calculate the safe lateral and longitudinal gaps defined via Formal rules.
3) Based on 2; a continuous trajectory path which complies not only with the traffic rules but provides safety and comfortability assurances for the passengers and VRUs involved.

   a) If no such valid paths exist, priority should be given to the next best plan, which could be switched at optimal cost.
   b) Backup plans could be pre-calculated (via multi-modal trajectory forecasting) by ensuring that a delicate balance is maintained between relaxation of rule compliance versus safety of the passengers. For example, if it means that lives could be saved by crossing over a double solid lane, then it should be taken.

c) Bringing the vehicle to a Minimal Risk Condition (MRC) should also be evaluated to bring the vehicle to a safe state.

Within the current scope of this work, we do not consider the effects of uncertainty and the ethics involved about human lives. Normally, these plans should be calculated under uncertainty to reduce the risk posed by the real world.

8.3.4 Constraints and Conformity

Embedding constraints on the learning functions are a direct way to ensure conformance of a neural network to a certain objective by defining upper and lower bounds to the output space. Modern AVs utilize machine learning model which are trained with algorithms based on Stochastic Gradient Descent via regularizers and optimizers; placing *soft constraints* in the subspace. Here, regularizers are a set of rules to fine-tune the expected behavior. On the other hand, when hard constraints are a part of the neural network structure [370, 495] the networks should be attentive to the physics involved, and their predictions should not deviate from the said constraints which are imposed. With which we expect the outputs from the networks to be restricted to the physical subspace, ensuring a compliance to the bounds placed. When compared with their Neural Network counterparts trained via soft constraints, physics guided nets respect the bounds implicitly by never deviating from them.

Conformance to rules could be rephrased as a constraint satisfaction problem by reasoning about the temporal properties of the system. Ensuring the safety of the vehicles during planning operation is a crucial task such that possible collisions could be mitigated. As a result, expressing rules and satisfying those specifications in the form of temporal logics [197] and verifying the system properties using Signal Temporal Logic (STL) [612], where rules are checked and verified for their formal conformance. Although, works from Censi et al. take a different approach by defining rules in the form of relational hierarchical structures represented via Linear Temporal Logic (LTL). This allows one to have the flexibility to describe the relative priorities between rules which result in a control command to the planner deemed compliant according to the specification. These scalable "rule book" could explicitly capture the common-sense and informal rules to some extent in the context of automated driving.

LIST OF ABBREVIATIONS

A3C	Asynchronous Advantage Actor-Critic
ACE	Automated Concept-based Explanation
AD	Autonomous Driving
ADF	Assumed Density Filtering
ADMM	Alternating Direction Method of Multipliers
AE	Auto Encoder
A-GEM	Averaged Gradient Episodic Memory
AI	Artificial Intelligence
AL	Active Learning
ALM	Augmented Lagrangian Method
AV	Autonomous Vehicle
BEV	Bird's-Eye-View
BP	Backpropagation
CAV	Concept Activation Vector
CBM	Concept Bottleneck Model
CBN	Causal Bayesian Network
CD	Causal Discovery
CF-GPS	Counterfactually-Guided Policy Search
CI	Causal Inference
CNN	Convolutional Neural Network
CPE	Counterfactual Policy Evaluation
CR	Causal Reasoning
CRPS	Continuous Ranked Probability Score
CT	Computerised Tomography
CV	Computer Vision
CVAE	Conditional Variational Auto Encoder
DAG	Directed Acyclic Graph
DDPG	Deep Deterministic Policy Gradient
DL	Deep Learning
DLL	Differentiable Logic Layer
DNN	Deep Neural Network
DOT	U.S. Department of Transportation
DPG	Deterministic Policy Gradient
DSL	Domain-Specific Language
DQN	Deep Q Learning
ECE	Expected Calibration Error
ELBO	(empirical) Evidence Lower Bound
ERM	Empirical Risk Minimization
FC	full-covariance
FCN	Fully Convolutional Network
FOL	First order Logic
FPN	Feature Pyramid Network
GAAL	Generative Adversarial Active Learning
GAIL	Generative Adversarial Imitation Learning
GAN	Generative Adversarial Network
GEM	Gradient Episodic Memory
GIS	Geographical Information System
GMM	Gaussian Mixture Model
GNN	Graph Neural Network
GPM	Gradient Projection Memory
HD	High-Definition
HMC	Hamiltonian Monte Carlo
HOG	Histograms of Oriented Gradients
IG	Integrated Gradients
ILP	Inductive Logic Programming
IRL	Inverse Reinforcement Learning
KF	Kronecker Factorization
KG	Knowledge Graph
KGE	Knowledge Graph Embedding
KKT	Karush Kuhn Tucker
KL	Kullback-Leibler
KRL	Knowledge Representation Learning
LIME	Local interpretable model-agnostic explanations
LRP	Layer-wise Relevance Propagation
LSTM	Long Short-Term Memory
LTL	Linear Temporal Logic
LTN	Logic Tensor Network
MAML	Model-Agnostic Meta-Learning
mAP	mean Average Precision
MAP	Maximum-a-posteriori
MARL	Multi-Agent Reinforcement Learning
MB-PS	Model-based Policy Search
MBRL	Model-based Reinforcement Learning
MC	Monte Carlo
MCTS	Monte-Carlo Tree Search
MDP	Markov Decision Process
MF	mean-field
MFRL	Model-free Reinforcement Learning
ML	Machine Learning
MLP	Multi-Layer Perceptron
MOBA	Multiplayer Online Battle Arena
MPC	Model-predictive Control
MRC	Minimal Risk Condition
MRF	Markov Random Field
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
NAS	Neural Architecture Search
NLL	Negative Log-Likelihood
NLM	Neural Logic Machines
NLP	Natural Language Processing
NLU	Natural Language Understanding
NMT	Neural Machine Translation
NN	Neural Network
NSBP	Neural-Symbolic Behavior Program
NSDP	Neuro-Symbolic Decision Program
NSP	Neuro-Symbolic Program
NSPS	Neuro-Symbolic Program Search
NUTS	No-U-Turn Sampling
OEDR	Object and Event Detection and Response
ODD	Operational Design Domain
PATH	California Partners for Advanced Transportation Technology
PDE	Partial Differential Equation
POMDP	Partially Observable Markov Decision Process
PPO	Proximal Policy Optimization
QA	Question Answering
R-CNN	Region-based Convolutional Neural Network
R-CNN	Regions with CNN
RGB	Red Green Blue
RISE	Randomized Input Sampling for Explanation
RL	Reinforcement Learning
RPN	Region Proposal Network
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SA	Situation Awareness
SAC	Soft Actor Critic
SBR	Semantic Based Regularization
SCM	Structural Causal Model
SVD	Self-Driving Vehicle

SGD	Stochastic Gradient Descent
SGHMC	Stochastic Gradient Hamiltonian Monte Carlo
SGLD	Stochastic Gradient Langevin Dynamics
SHAP	SHapley Additive exPlanations
SI	Synaptic Intelligence
SRM	Structural Risk Minimization
SSD	Single Shot MultiBox Detector
SSM	State Space Model
STL	Signal Temporal Logic
StVO	Straßenverkehrsordnung
SQM	Sequential Quadratic Program
SVM	Support Vector Machine
TRPO	Trust Region Policy Optimization
TTC	Time-To-Collision
UAV	Unmanned Aerial Vehicles
VA	Visual Analytics
VAE	Variational Auto Encoder
VI	Variational Inference
VQA	Visual Question Answering
VRU	Vulnerable Road User
WSOL	Weakly Supervised Object Localization
XAI	Explainable Artificial Intelligence

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the twenty-first international conference on Machine learning*. ICML '04. New York, NY, USA: Association for Computing Machinery, July 2004, p. 1. (Visited on 07/07/2021).

[2] Pieter Abbeel et al. "Discriminative Training of Kalman Filters." In: *Robotics: Science and systems*. Vol. 2. 2005, p. 1.

[3] Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information Fusion* 76 (2021), pp. 243–297.

[4] Sami Abu-El-Haija et al. "Watch Your Step: Learning Node Embeddings via Graph Attention". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. event-place: Montréal, Canada. Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 9198–9208.

[5] Joshua Achiam. *Spinning Up in Deep Reinforcement Learning*. Mar. 2020. URL: https://spinningup.openai.com/en/latest/index.html (visited on 07/05/2021).

[6] Aniket Agarwal, Ayush Mangal, and Vipul. "Visual Relationship Detection using Scene Graphs: A Survey". In: *arXiv:2005.08045 [cs]* (May 2020). arXiv: 2005.08045. (Visited on 07/02/2021).

[7] Umang Aggarwal, Adrian Popescu, and Céline Hudelot. "Minority Class Oriented Active Learning for Imbalanced Datasets". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9920–9927.

[8] Umang Aggarwal, Adrian Popescu, and Céline Hudelot. "Optimizing Active Learning for Low Annotation Budgets". In: *arXiv preprint arXiv:2201.07200* (2022).

[9] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. "Efficient similarity search in sequence databases". In: *Foundations of Data Organization and Algorithms*. Ed. by Gerhard Goos, Juris Hartmanis, and David B. Lomet. Vol. 730. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 69–84. (Visited on 06/25/2021).

[10] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. "Mining association rules between sets of items in large databases". en. In: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93*. Washington, D.C., United States: ACM Press, 1993, pp. 207–216. (Visited on 06/25/2021).

[11] Rakesh Agrawal, Ramakrishnan Srikant, et al. "Fast algorithms for mining association rules". In: *Proc. 20th int. conf. very large data bases, VLDB*. Vol. 1215. Citeseer, 1994, pp. 487–499.

[12] Shivam Akhauri et al. "Improving Generalization of Transfer Learning Across Domains Using Spatio-Temporal Features in Autonomous Driving". In: *arXiv preprint arXiv:2103.08116* (2021).

[13] Joost van Amersfoort et al. "Uncertainty Estimation Using a Single Deep Deterministic Neural Network". In: *International Conference on Machine Learning* (2020).

[14] Alexander Amini et al. "Deep evidential regression". en. In: *Advances in neural information processing systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 14927–14937.

[15] Brandon Amos and J Zico Kolter. "Optnet: Differentiable optimization as a layer in neural networks". In: *International Conference on Machine Learning*. PMLR, 2017, pp. 136–145.

[16] Robert Andrews, Joachim Diederich, and Alan B. Tickle. "Survey and critique of techniques for extracting rules from trained artificial neural networks". en. In: *Knowledge-Based Systems* 8.6 (Dec. 1995), pp. 373–389. (Visited on 06/22/2021).

[17] Natalia Andrienko et al. "A theoretical model for pattern discovery in visual analytics". en. In: *Visual Informatics* 5.1 (2021). Publisher: Elsevier, pp. 23–42.

[18] Mykhaylo Andriluka et al. "2d human pose estimation: New benchmark and state of the art analysis". In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*. 2014, pp. 3686–3693.

[19] Dana Angluin. "Queries revisited". In: *International Conference on Algorithmic Learning Theory*. Springer, 2001, pp. 12–31.

[20] Antreas Antoniou, Harrison Edwards, and Amos Storkey. "How to train your MAML". In: *International Conference on Learning Representations*. 2019.

[21] Elena Arcari et al. "Dual Stochastic MPC for Systems with Parametric and Structural Uncertainty". In: *Learning for Dynamics and Control*. PMLR, 2020, pp. 894–903.

[22] Shervin Ardeshir, Kofi Malcolm Collins-Sibley, and Mubarak Shah. "Geo-Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[23] N. Aréchiga. "Specifying Safety of Autonomous Vehicles in Signal Temporal Logic". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. ISSN: 2642-7214. June 2019, pp. 58–63.

[24] Saurabh Arora and Prashant Doshi. "A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress". In: *arXiv:1806.06877 [cs, stat]* (Nov. 2020). arXiv: 1806.06877. (Visited on 06/19/2021).

[25] Uber ATG. *Uber ATG Safety Report 2020*. 2020.

[26] Tara Athan et al. "LegalRuleML: Design Principles and Foundations". en. In: *Reasoning Web. Web Logic Rules: 11th International Summer School 2015, Berlin, Germany, July 31- August 4, 2015, Tutorial Lectures*. Ed. by Wolfgang Faber and Adrian Paschke. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 151–188. (Visited on 03/19/2021).

[27] Tara Athan et al. "LegalRuleML: From Metamodel to Use Cases". en. In: *Theory, Practice, and Applications of Rules on the Web*. Ed. by Leora Morgenstern et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, pp. 13–18.

[28] Sören Auer et al. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web*. Ed. by David Hutchison et al. Vol. 4825. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735. (Visited on 07/09/2021).

[29] Aurora. *CVPR 2021: Forecasting for Motion Planning*. June 2021. URL: https://www.youtube.com/watch?v=bnmrXt1g3aQ (visited on 01/28/2022).

[30] Filipe de Avila Belbute-Peres et al. "End-to-end differentiable physics for learning and control". In: *Advances in Neural Information Processing Systems*. 2018, pp. 7178–7189.

[31] Sebastian Bach et al. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 10.7 (2015). Publisher: Public Library of Science, pp. 1–46.

[32] Samy Badreddine et al. "Logic Tensor Networks". In: *arXiv:2012.13635 [cs]* (Jan. 2021). arXiv: 2012.13635. (Visited on 06/30/2021).

[33] Mohammadhossein Bahari, Ismail Nejjar, and Alexandre Alahi. "Injecting knowledge in data-driven vehicle trajectory predictors". In: *Transportation research part C: emerging technologies* 128 (2021). Publisher: Elsevier, p. 103010.

[34] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *International Conference on Learning Representations (ICLR)*. 2015.

[35] L. Baird. "Residual Algorithms: Reinforcement Learning with Function Approximation". In: *ICML*. 1995.

[36] Ivana Balazevic, Carl Allen, and Timothy Hospedales. "TuckER: Tensor Factorization for Knowledge Graph Completion". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5185–5194.

[37] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. "Margin based active learning". In: *International Conference on Computational Learning Theory*. Springer, 2007, pp. 35–50.

[38] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data". In: *Journal of Machine learning Research* 9.Mar (2008), pp. 485–516.

[39] Ankan Bansal, Yuting Zhang, and Rama Chellappa. "Visual Question Answering on Image Sets". In: *European Conference on Computer Vision*. Springer, 2020, pp. 51–67.

[40] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. "ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst". In: *arXiv:1812.03079 [cs]* (Dec. 2018). arXiv: 1812.03079. (Visited on 06/28/2021).

[41] Depei Bao and Zehong Yang. "Intelligent stock trading system by turning point confirming and probabilistic reasoning". en. In: *Expert Systems with Applications* 34.1 (Jan. 2008), pp. 620–627. (Visited on 06/25/2021).

[42] Cesare Bartolini, Gabriele Lenzini, and Cristiana Santos. "An Agile Approach to Validate a Formal Representation of the GDPR". en. In: *New Frontiers*

in Artificial Intelligence. Ed. by Kazuhiro Kojima et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 160–176.

[43] Sotirios Batsakis, George Baryannis, and Ilias Tachmazidis. *D3.1 Ontology-based access to normative knowledge.* 2017. URL: https://www.mirelproject.eu/publications/D3.1.pdf (visited on 07/06/2021).

[44] Alex Bäuerle, Heiko Neumann, and Timo Ropinski. "Classifier-guided visual correction of noisy labels for image classification tasks". In: *Computer graphics forum.* Vol. 39. Number: 3 tex.organization: Wiley Online Library. 2020, pp. 195–205.

[45] Yared W Bekele. "Physics-informed deep learning for flow and deformation in poroelastic media". In: *arXiv preprint arXiv:2010.15426* (2020).

[46] Marc G. Bellemare, Will Dabney, and Rémi Munos. "A Distributional Perspective on Reinforcement Learning". In: *Proceedings of the 34th International Conference on Machine Learning.* Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 449–458.

[47] Marc G. Bellemare et al. "The Arcade Learning Environment: An Evaluation Platform for General Agents". In: *J. Artif. Int. Res.* 47.1 (May 2013). Place: El Segundo, CA, USA Publisher: AI Access Foundation, pp. 253–279.

[48] William H Beluch et al. "The power of ensembles for active learning in image classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018, pp. 9368–9377.

[49] Javad Bengar et al. "Temporal coherence for active learning in videos". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.* 2019, pp. 0–0.

[50] Javad Zolfaghari Bengar, Bogdan Raducanu, and Joost van de Weijer. "When deep learners change their mind: Learning dynamics for active learning". In: *International Conference on Computer Analysis of Images and Patterns.* Springer, 2021, pp. 403–413.

[51] Javad Zolfaghari Bengar et al. "Class-Balanced Active Learning for Image Classification". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2022, pp. 1536–1545.

[52] Javad Zolfaghari Bengar et al. "Reducing label effort: Self-supervised meets active learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2021, pp. 1631–1639.

[53] J.M. Benitez, J.L. Castro, and I. Requena. "Are artificial neural networks black boxes?" en. In: *IEEE Transactions on Neural Networks* 8.5 (Sept. 1997), pp. 1156–1164. (Visited on 06/22/2021).

[54] Gregory Benton et al. "Learning invariances in neural networks". In: *arXiv preprint arXiv:2010.11882* (2020).

[55] Jürgen Bernard et al. "Comparing visual-interactive labeling with active learning: An experimental study". In: *IEEE transactions on visualization and computer graphics* 24.1 (2017). Publisher: IEEE, pp. 298–308.

[56] Julian Bernhard et al. "BARK: Open behavior benchmarking in multi-agent environments". In: *arXiv preprint arXiv:2003.02604* (2020).

[57] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. "Monoloco: Monocular 3d pedestrian localization and uncertainty estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2019, pp. 6861–6871.

[58] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. "Importance weighted active learning". In: *Proceedings of the 26th annual international conference on machine learning.* 2009, pp. 49–56.

[59] Emmanuel de Bezenac, Arthur Pajot, and Patrick Gallinari. "Deep learning for physical processes: Incorporating prior scientific knowledge". In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12 (2019). Publisher: IOP Publishing, p. 124009.

[60] Battista Biggio and Fabio Roli. "Wild patterns: Ten years after the rise of adversarial machine learning". In: *Pattern Recognition* 84 (2018). Publisher: Elsevier, pp. 317–331.

[61] Chris M Bishop. "Training with noise is equivalent to Tikhonov regularization". In: *Neural computation* 7.1 (1995), pp. 108–116.

[62] Christopher M. Bishop. *Mixture density networks.* Tech. rep. 1994.

[63] Christopher M. Bishop. *Pattern recognition and machine learning.* Springer, 2006.

[64] *Blackstone.* URL: https://github.com/ICLRandD/Blackstone.

[65] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection". In: *arXiv preprint arXiv:2004.10934* (2020).

[66] Paul T. Boggs and Jon W. Tolle. "Sequential Quadratic Programming". en. In: *Acta Numerica* 4 (Jan. 1995), pp. 1–51. (Visited on 06/24/2021).

[67] Mariusz Bojarski et al. "End to End Learning for Self-Driving Cars". In: *ArXiv* abs/1604.07316 (2016).

[68] Mariusz Bojarski et al. "Visualbackprop: Visualizing CNNs for Autonomous Driving". In: *arXiv preprint arXiv:1611.05418* 2 (2016).

[69] Max Bolderman, Mircea Lazar, and Hans Butler. "Physics–Guided Neural Networks for Inversion–based Feedforward Control applied to Linear Motors". In: *2021 IEEE Conference on Control Technology and Applications (CCTA).* IEEE, 2021, pp. 1115–1120.

[70] Kurt Bollacker et al. "Freebase: a collaboratively created graph database for structuring human knowledge". en. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08.* Vancouver, Canada: ACM Press, 2008, p. 1247. (Visited on 07/09/2021).

[71] Guido Bologna. "A Simple Convolutional Neural Network with Rule Extraction". en. In: *Applied Sciences* 9.12 (June 2019), p. 2411. (Visited on 06/22/2021).

[72] Łukasz Borchmann et al. "Contract Discovery: Dataset and a Few-Shot Semantic Retrieval Challenge with Competitive Baselines". In: *arXiv:1911.03911 [cs]* (Oct. 2020). arXiv: 1911.03911. (Visited on 07/06/2021).

[73] Antoine Bordes et al. "A Semantic Matching Energy Function for Learning with Multi-Relational Data". In: *Mach. Learn.* 94.2 (Feb. 2014). Place: USA Publisher: Kluwer Academic Publishers, pp. 233–259.

[74] Andrea Borghesi, Federico Baldo, and Michela Milano. "Improving deep learning models via constraint-based domain knowledge: a brief survey". In: *arXiv preprint arXiv:2005.10691* (2020).

[75] Savong Bou et al. "Ontology-Style Relation Annotation: A Case Study". In: *Proceedings of the 12th Language Resources and Evaluation Conference*. 12th Conference on Language Resources and Evaluation: European Language Resources Association, 2020, pp. 4867–4876.

[76] Freddy A. Boulton, Elena Corina Grigore, and Eric M. Wolff. "Motion Prediction using Trajectory Sets and Self-Driving Domain Knowledge". In: *arXiv:2006.04767 [cs, stat]* (Jan. 2021). arXiv: 2006.04767. (Visited on 07/20/2021).

[77] Markus Braun et al. "EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), pp. 1–1.

[78] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *CoRR abs/2005.14165* (2020). _eprint: 2005.14165.

[79] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. "Active and incremental learning with weak supervision". In: *KI-Künstliche Intelligenz* (2020). Publisher: Springer, pp. 1–16.

[80] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. "Active learning for deep object detection". In: *arXiv preprint arXiv:1809.09875* (2018).

[81] BSI. *Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) - Specification*. ISBN 978 0 539 06735 4. Aug. 2020. URL: https://www.bsigroup.com/globalassets/localfiles/en-gb/cav/pas1883.pdf (visited on 06/20/2021).

[82] Martin Buechel et al. "Ontology-based traffic scene modeling, traffic regulations dependent situational awareness and decision-making for automated vehicles". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los Angeles, CA, USA: IEEE, June 2017, pp. 1471–1476. (Visited on 07/01/2021).

[83] Lars Buesing et al. "Learning and Querying Fast Generative Models for Reinforcement Learning". In: *CoRR abs/1802.03006* (2018).

[84] Lars Buesing et al. "Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search". In: *arXiv:1811.06272 [cs, stat]* (Nov. 2018). arXiv: 1811.06272. (Visited on 07/07/2021).

[85] Mona Buisson-Fenet, Friedrich Solowjow, and Sebastian Trimpe. "Actively learning Gaussian process dynamics". In: *Learning for Dynamics and Control*. PMLR, 2020, pp. 5–15.

[86] Liwei Cai and William Yang Wang. "KBGAN: Adversarial Learning for Knowledge Graph Embeddings". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1470–1480.

[87] Tony Cai, Weidong Liu, and Xi Luo. "A constrained l_1-minimization approach to sparse precision matrix estimation". In: *Journal of the American Statistical Association* 106.494 (2011). Publisher: Taylor & Francis, pp. 594–607.

[88] Zhaowei Cai and Nuno Vasconcelos. "Cascade r-cnn: High quality object detection and instance segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019). Publisher: IEEE.

[89] Emmanuel J Candès, Michael B Wakin, and Stephen P Boyd. "Enhancing sparsity by reweighted l_1-minimization". In: *Journal of Fourier analysis and applications* 14.5 (2008). Publisher: Springer, pp. 877–905.

[90] Fang Cao and Eli T. Brown. "DRIL: Descriptive Rules by Interactive Learning". In: *2020 IEEE Visualization Conference (VIS)*. 2020, pp. 256–260.

[91] Alexandre Capone et al. "Localized active learning of Gaussian process state space models". In: *Learning for Dynamics and Control*. PMLR, 2020, pp. 490–499.

[92] Ryan Carey et al. "The Incentives that Shape Behaviour". In: *arXiv:2001.07118 [cs]* (Mar. 2021). arXiv: 2001.07118. (Visited on 06/10/2021).

[93] Nicolas Carion et al. "End-to-end object detection with transformers". In: *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[94] Sergio Casas, Abbas Sadat, and Raquel Urtasun. "MP3: A Unified Model to Map, Perceive, Predict and Plan". In: *arXiv:2101.06806 [cs]* (Jan. 2021). arXiv: 2101.06806. (Visited on 02/07/2021).

[95] Daniel C. Castro, Ian Walker, and Ben Glocker. "Causality matters in medical imaging". en. In: *Nature Communications* 11.1 (July 2020). Number: 1 Publisher: Nature Publishing Group, p. 3673. (Visited on 06/09/2021).

[96] Andrea Censi et al. "Liability, Ethics, and Culture-Aware Behavior Specification using Rulebooks". In: *arXiv:1902.09355 [cs]* (Mar. 2019). arXiv: 1902.09355. (Visited on 07/20/2021).

[97] Yuning Chai et al. "MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction". In: *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*. Ed. by Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura. Vol. 100. Proceedings of Machine Learning Research. PMLR, 2019, pp. 86–99.

[98] Shayok Chakraborty et al. "Active batch selection via convex relaxations with guaranteed solution bounds". In: *IEEE transactions on pattern analysis and machine intelligence* 37.10 (2015). Publisher: IEEE, pp. 1945–1958.

[99] Ilias Chalkidis et al. "LEGAL-BERT: The Muppets straight out of Law School". In: *arXiv:2010.02559 [cs]* (Oct. 2020). arXiv: 2010.02559. (Visited on 07/06/2021).

[100] Akshay L Chandra et al. "On initial pools for deep active learning". In: *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*. PMLR, 2021, pp. 14–32.

[101] Che-Han Chang et al. "KG-GAN: Knowledge-guided generative adversarial networks". In: *arXiv preprint arXiv:1905.12261* (2019).

[102] Xiaojun Chang et al. "Scene Graphs: A Survey of Generations and Applications". In: *arXiv:2104.01111*

[cs] (Mar. 2021). arXiv: 2104.01111. (Visited on 07/02/2021).

[103] Aditya Chattopadhay et al. "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (Mar. 2018). ISBN: 9781538648865 Publisher: IEEE.

[104] Angelos Chatzimparmpas et al. "The state of the art in enhancing trust in machine learning models with the use of visualizations". In: *Computer Graphics Forum*. Vol. 39. Issue: 3. Wiley Online Library, 2020, pp. 713–756.

[105] Arslan Chaudhry et al. "Efficient Lifelong Learning with A-GEM". In: *arXiv:1812.00420 [cs, stat]* (Jan. 2019). arXiv: 1812.00420. (Visited on 05/12/2021).

[106] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[107] Changjian Chen et al. "Oodanalyzer: Interactive analysis of out-of-distribution samples". In: *IEEE transactions on visualization and computer graphics* 27.7 (2020). Publisher: IEEE, pp. 3335–3349.

[108] Chaofan Chen et al. "This Looks Like That: Deep Learning for Interpretable Image Recognition". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[109] Dian Chen et al. "Learning by cheating". In: *Conference on robot learning*. 2020, pp. 66–75.

[110] Jianshu Chen and Li Deng. "A primal-dual method for training recurrent neural networks constrained by the echo-state property". In: *arXiv preprint arXiv:1311.6091* (2013).

[111] Jianyu Chen, Shengbo Eben Li, and Masayoshi Tomizuka. "Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning". In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–11.

[112] Tianqi Chen, Emily Fox, and Carlos Guestrin. "Stochastic Gradient Hamiltonian Monte Carlo". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research. Issue: 2. Bejing, China: PMLR, June 2014, pp. 1683–1691.

[113] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. 2020, pp. 1597–1607.

[114] Xiaozhi Chen et al. "Monocular 3D Object Detection for Autonomous Driving". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[115] Xu Chen and Saratendu Sethi. "Robust active learning for electrocardiographic signal classification". In: *arXiv preprint arXiv:1811.08919* (2018).

[116] Yu Fan Chen et al. "Socially aware motion planning with deep reinforcement learning". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.

[117] Zhaozhong Chen et al. "Kalman Filter Tuning with Bayesian Optimization". In: *arXiv:1912.08601 [cs, eess, math]* (Dec. 2019). arXiv: 1912.08601. (Visited on 06/24/2021).

[118] Zhi Chen, Yijie Bei, and Cynthia Rudin. "Concept whitening for interpretable image recognition". In: *Nature Machine Intelligence* 2.12 (2020). Publisher: Nature Publishing Group, pp. 772–782.

[119] K. Cho et al. "Deep Predictive Autonomous Driving Using Multi-Agent Joint Trajectory Prediction and Traffic Rules". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Nov. 2019, pp. 2076–2081.

[120] Kyunghoon Cho and Songhwai Oh. "Learning-Based Model Predictive Control Under Signal Temporal Logic Specifications". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2018, pp. 7322–7329.

[121] Ignasi Clavera, Yao Fu, and Pieter Abbeel. "Model-Augmented Actor-Critic: Backpropagating through Paths". In: *International Conference on Learning Representations*. 2020.

[122] Ignasi Clavera et al. "Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning". In: *International Conference on Learning Representations*. 2019.

[123] Ignasi Clavera et al. "Model-Based Reinforcement Learning via Meta-Policy Optimization". In: *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*. Vol. 87. Proceedings of Machine Learning Research. PMLR, 2018, pp. 617–629.

[124] *CodaLab - Competition*. URL: https://competitions.codalab.org/competitions/20132#learn_the_details (visited on 07/12/2021).

[125] Felipe Codevilla et al. "End-to-end driving via conditional imitation learning". In: *2018 IEEE international conference on robotics and automation (ICRA)*. 2018, pp. 4693–4700.

[126] David Cohn, Les Atlas, and Richard Ladner. "Improving generalization with active learning". In: *Machine learning* 15.2 (1994). Publisher: Springer, pp. 201–221.

[127] Mark Collier et al. "A simple probabilistic method for deep classification under input-dependent label noise". In: *arXiv preprint arXiv:2003.06778* (2020).

[128] *Competition on Legal Information Extraction/Entailment (COLIEE)*. URL: https://sites.ualberta.ca/~rabelo/COLIEE2021/.

[129] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. "On the relationship between self-attention and convolutional layers". In: *International Conference on Learning Representations (ICLR)*. 2020.

[130] Marcella Cornia et al. "Predicting human eye fixations via an lstm-based saliency attentive model". In: *IEEE Transactions on Image Processing* 27.10 (2018). Publisher: IEEE, pp. 5142–5154.

[131] Alana de Santana Correia and Esther Luna Colombini. "Attention, please! A survey of Neural Attention Models in Deep Learning". In: *CoRR abs/2103.16775* (2021). _eprint: 2103.16775.

[132] Akansel Cosgun et al. "Towards full automated drive in urban environments: A demonstration in GoMen-

tum Station, California". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017, pp. 1811–1818.

[133] Alexander I Cowen-Rivers et al. "Samba: Safe model-based & active reinforcement learning". In: *Machine Learning* (2022). Publisher: Springer, pp. 1–31.

[134] Mark Craven and Jude Shavlik. "Extracting tree-structured representations of trained networks". In: *Advances in neural information processing systems* 8 (1995).

[135] Mark W. Craven and Jude W. Shavlik. "Using Sampling and Queries to Extract Rules from Trained Neural Networks". en. In: *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 37–45. (Visited on 06/22/2021).

[136] Andrew Cropper and Sebastijan Dumančić. "Inductive logic programming at 30: a new introduction". In: *arXiv:2008.07912 [cs]* (Oct. 2020). arXiv: 2008.07912. (Visited on 06/25/2021).

[137] James P. Crutchfield and David P. Feldman. "Regularities unseen, randomness observed: Levels of entropy convergence". en. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 13.1 (Mar. 2003), pp. 25–54. (Visited on 06/25/2021).

[138] Gabriela Csurka. "A comprehensive survey on domain adaptation for visual applications". In: *Domain adaptation in computer vision applications* (2017), pp. 1–35.

[139] Ekin D Cubuk et al. "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 113–123.

[140] Henggang Cui et al. "Deep kinematic models for kinematically feasible vehicle trajectory predictions". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10563–10569.

[141] Henggang Cui et al. "Ellipse loss for scene-compliant motion prediction". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8558–8564.

[142] Yin Cui et al. "Large scale fine-grained categorization and domain-specific transfer learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4109–4118.

[143] Frédéric Cuppens et al. "Norms and Deontic Logic". en. In: *A Guided Tour of Artificial Intelligence Research*. Ed. by Pierre Marquis, Odile Papini, and Henri Prade. Cham: Springer International Publishing, 2020, pp. 253–274. (Visited on 06/29/2021).

[144] Jifeng Dai et al. "R-fcn: Object detection via region-based fully convolutional networks". In: *Advances in neural information processing systems*. 2016, pp. 379–387.

[145] Yuanfei Dai et al. "A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks". In: *Electronics* 9 (2020), p. 750.

[146] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee, 2005, pp. 886–893.

[147] Saumik Dana and Teeratorn Kadeethum. *Physics informed deep learning for coupled flow and poromechanics. Mandel's problem*. 2021.

[148] Adnan Darwiche. "Human-Level Intelligence or Animal-like Abilities?" In: *Commun. ACM* 61.10 (Sept. 2018). Place: New York, NY, USA Publisher: Association for Computing Machinery, pp. 56–67.

[149] Abir Das, Rameswar Panda, and Amit Roy-Chowdhury. "Active image pair selection for continuous person re-identification". In: *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 4263–4267.

[150] Ishita Dasgupta et al. "Causal Reasoning from Meta-reinforcement Learning". In: *arXiv:1901.08162 [cs, stat]* (Jan. 2019). arXiv: 1901.08162 version: 1. (Visited on 08/06/2020).

[151] Sanjoy Dasgupta and Daniel Hsu. "Hierarchical sampling for active learning". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 208–215.

[152] Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. "A general agnostic active learning algorithm". In: *Advances in neural information processing systems* 20 (2007).

[153] Ketan Date and Rakesh Nagi. "GPU-accelerated Hungarian algorithms for the Linear Assignment Problem". en. In: *Parallel Computing* 57 (Sept. 2016), pp. 52–72. (Visited on 06/28/2021).

[154] Arka Daw et al. "Physics-Guided Architecture (PGA) of Neural Networks for Quantifying Uncertainty in Lake Temperature Modeling". In: *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 2020, pp. 532–540.

[155] Erik Daxberger et al. "Bayesian Deep Learning via Subnetwork Inference". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 2510–2521.

[156] Antonio De Nicola, Michele Missikoff, and Roberto Navigli. "A proposal for a unified process for ontology building: UPON". In: *Proceedings of the 16th international conference on Database and Expert Systems Applications*. DEXA'05. Berlin, Heidelberg: Springer-Verlag, Aug. 2005, pp. 655–664. (Visited on 06/30/2021).

[157] MP. Deisenroth and CE. Rasmussen. "PILCO: A Model-Based and Data-Efficient Approach to Policy Search". In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*. Omnipress, 2011, pp. 465–472.

[158] Grégoire Déletang et al. "Causal Analysis of Agent Behavior for AI Safety". In: *arXiv:2103.03938 [cs]* (Mar. 2021). arXiv: 2103.03938. (Visited on 03/10/2021).

[159] Changyu Deng et al. "Integrating Machine Learning with Human Knowledge". In: *Iscience* (2020). Publisher: Elsevier, p. 101656.

[160] Haowen Deng et al. "Deep bingham networks: Dealing with uncertainty and ambiguity in pose estimation". In: *ArXiv* abs/2012.11002 (2020).

[161] Nachiket Deo and Mohan M. Trivedi. "Convolutional Social Pooling for Vehicle Trajectory Prediction". In:

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. June 2018.

[162] Sai Vikas Desai et al. "An adaptive supervision framework for active learning in object detection". In: *arXiv preprint arXiv:1908.02454* (2019).

[163] Tim Dettmers et al. "Convolutional 2D Knowledge Graph Embeddings". In: *Proceedings of the 32th AAAI Conference on Artificial Intelligence*. Feb. 2018, pp. 1811–1818.

[164] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv:1810.04805 [cs]* (May 2019). arXiv: 1810.04805. (Visited on 10/25/2021).

[165] Jacob Devlin et al. "RobustFill: Neural Program Learning under Noisy I/O". In: *arXiv:1703.07469 [cs]* (Mar. 2017). arXiv: 1703.07469. (Visited on 06/20/2021).

[166] Terrance DeVries and Graham W Taylor. "Dataset augmentation in feature space". In: *arXiv preprint arXiv:1702.05538* (2017).

[167] Terrance DeVries and Graham W Taylor. "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552* (2017).

[168] Luigi Di Caro and Serena Villata. *D2.2 Computational ontologies for normative reasoning*. 2017. URL: https://www.mirelproject.eu/publications/D2.2.pdf (visited on 07/06/2021).

[169] Michelangelo Diligenti, Marco Gori, and Claudio Sacca. "Semantic-based regularization for learning and inference". en. In: *Artificial Intelligence* 244 (Mar. 2017), pp. 143–165. (Visited on 07/21/2021).

[170] Michelangelo Diligenti, Soumali Roychowdhury, and Marco Gori. "Integrating prior knowledge into deep learning". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 920–923.

[171] Boyang Ding et al. "Improving Knowledge Graph Embedding Using Simple Constraints". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 110–121.

[172] Shilp Dixit et al. "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects". In: *Annual Reviews in Control* 45 (2018), pp. 76–86.

[173] Nemanja Djuric et al. "MultiXNet: Multiclass Multistage Multimodal Motion Prediction". In: *IEEE Intelligent Vehicles Symposium, IV 2021, Nagoya, Japan, July 11-17, 2021*. IEEE, 2021, pp. 435–442.

[174] Nemanja Djuric et al. "Uncertainty-aware Short-term Motion Prediction of Traffic Actors for Autonomous Driving". In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. ISSN: 2642-9381. Mar. 2020, pp. 2084–2093.

[175] Piotr Dollár et al. "Pedestrian detection: A benchmark". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 304–311.

[176] Xu Dong et al. "Probabilistic oriented object detection in automotive radar". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 102–103.

[177] Alexey Dosovitskiy et al. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[178] Mauro Dragoni et al. "Combining NLP approaches for rule extraction from legal documents". In: *1st Workshop on MIning and REasoning with Legal texts (MIREL 2016)*. 2016.

[179] Simon Duane et al. "Hybrid Monte Carlo". In: *Physics Letters B* 195.2 (Sept. 1987), pp. 216–222.

[180] Melanie Ducoffe and Frederic Precioso. "Adversarial active learning for deep networks: a margin based approach". In: *arXiv preprint arXiv:1802.09841* (2018).

[181] Michael Dusenberry et al. "Efficient and Scalable Bayesian Neural Nets with Rank-1 Factors". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 2782–2792.

[182] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. "Modeling visual context is key to augmenting object detection datasets". In: *Proceedings of the european conference on computer vision (ECCV)*. 2018, pp. 364–380.

[183] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. "Cut, paste and learn: Surprisingly easy synthesis for instance detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1301–1310.

[184] Sayna Ebrahimi et al. "Minimax Active Learning". In: *arXiv preprint arXiv:2012.10467* (2020).

[185] Sayna Ebrahimi et al. "Uncertainty-guided Continual Learning with Bayesian Neural Networks". In: *Proceedings of the 8th International Conference on Learning Representations*. OpenReview.net, 2020.

[186] Bradley Efron et al. "Least angle regression". In: *The Annals of Statistics* 32.2 (2004). Publisher: Institute of Mathematical Statistics, pp. 407–499.

[187] Lisa Ehrlinger and Wolfram Wöß. "Towards a definition of knowledge graphs." In: *SEMANTiCS (Posters, Demos, SuCCESS)* 48.1-4 (2016). Publisher: Citeseer, p. 2.

[188] Juan Elenter, Navid NaderiAlizadeh, and Alejandro Ribeiro. "A Lagrangian Duality Approach to Active Learning". In: *arXiv preprint arXiv:2202.04108* (2022).

[189] Ehsan Elhamifar et al. "A convex optimization framework for active learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 209–216.

[190] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. "Neural Architecture Search: A Survey". In: *arXiv:1808.05377 [cs, stat]* (Apr. 2019). arXiv: 1808.05377. (Visited on 06/20/2021).

[191] Arkady Epshteyn, Adam Vogel, and Gerald DeJong. "Active reinforcement learning". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 296–303.

[192] Andreas Ess, Bastian Leibe, and Luc Van Gool. "Depth and appearance for mobile scene analysis". In: *2007 IEEE 11th international conference on computer vision*. IEEE, 2007, pp. 1–8.

[193] Klemens Esterle, Luis Gressenbuch, and Alois Knoll. "Formalizing Traffic Rules for Machine Interpretabil-

ity". In: *Proceedings of the 3rd IEEE Connected and Automated Vehicles Symposium (CAVS)*. Victoria, B.C. Canada, Oct. 2020.

[194] Richard Evans and Edward Grefenstette. "Learning Explanatory Rules from Noisy Data". In: *arXiv:1711.04574 [cs, math]* (Jan. 2018). arXiv: 1711.04574. (Visited on 06/25/2021).

[195] Tom Everitt et al. "Understanding Agent Incentives using Causal Influence Diagrams. Part I: Single Action Settings". In: *arXiv:1902.09980 [cs]* (Sept. 2019). arXiv: 1902.09980. (Visited on 06/10/2021).

[196] *explosion/spaCy*. original-date: 2014-07-03T15:15:40Z. July 2021. URL: https://github.com/explosion/spaCy (visited on 07/13/2021).

[197] Geogios Fainekos, Hadas Kress-Gazit, and George Pappas. "Temporal Logic Motion Planning for Mobile Robots". In: *Lab Papers (GRASP)* (Apr. 2005).

[198] Qi Fan et al. "Few-shot object detection with attention-RPN and multi-relation detector". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 4013–4022.

[199] Hao-Shu Fang et al. "Instaboost: Boosting instance segmentation via probability map guided copy-pasting". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 682–691.

[200] Jin Fang et al. "MapFusion: A General Framework for 3D Object Detection with HDMaps". In: *CoRR* abs/2103.05929 (2021). arXiv: 2103.05929.

[201] Meng Fang, Yuan Li, and Trevor Cohn. "Learning how to active learn: A deep reinforcement learning approach". In: *arXiv preprint arXiv:1708.02383* (2017).

[202] Sebastian Farquhar, Yarin Gal, and Tom Rainforth. "On statistical bias in active learning: How and when to fix it". In: *arXiv preprint arXiv:2101.11665* (2021).

[203] Yuri Feldman and Vadim Indelman. "Bayesian viewpoint-dependent robust classification under model and localization uncertainty". In: *2018 IEEE international conference on robotics and automation (ICRA)*. 2018, pp. 3221–3228.

[204] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model". In: *2008 IEEE conference on computer vision and pattern recognition*. Ieee, 2008, pp. 1–8.

[205] Pedro F Felzenszwalb et al. "Object detection with discriminatively trained part-based models". In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2009). Publisher: IEEE, pp. 1627–1645.

[206] Di Feng et al. "A review and comparative study on probabilistic object detection in autonomous driving". In: *IEEE Transactions on Intelligent Transportation Systems* (2021). Publisher: IEEE.

[207] Di Feng et al. "Deep active learning for efficient training of a lidar 3d object detector". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 667–674.

[208] Di Feng et al. "Leveraging heteroscedastic aleatoric uncertainties for robust real-time lidar 3d object detection". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1280–1287.

[209] Alessio Ferrari and Andrea Esuli. "An NLP approach for cross-domain ambiguity detection in requirements engineering". en. In: *Automated Software Engineering* 26.3 (Sept. 2019), pp. 559–598. (Visited on 06/28/2021).

[210] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 1126–1135.

[211] Chelsea Finn, Kelvin Xu, and Sergey Levine. "Probabilistic Model-Agnostic Meta-Learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[212] Ferdinando Fioretto et al. "Lagrangian duality for constrained deep learning". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2020, pp. 118–135.

[213] Marc Fischer et al. "Dl2: Training and querying neural networks with logic". In: *International Conference on Machine Learning*. PMLR, 2019, pp. 1931–1941.

[214] Andrew Y. K. Foong et al. "'In-Between' Uncertainty in Bayesian Neural Networks". In: *CoRR* abs/1906.11537 (2019). _eprint: 1906.11537.

[215] Philippe Fournier-Viger et al. "Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by David Hutchison et al. Vol. 8443. Cham: Springer International Publishing, 2014, pp. 40–52. (Visited on 06/25/2021).

[216] Luca Franceschi et al. "Bilevel Programming for Hyperparameter Optimization and Meta-Learning". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 1568–1577.

[217] Yoav Freund and Robert E. Schapire. "A short introduction to boosting". In: *In proceedings of the sixteenth international joint conference on artificial intelligence*. Morgan Kaufmann, 1999, pp. 1401–1406.

[218] Alexander Freytag, Erik Rodner, and Joachim Denzler. "Selecting influential examples: Active learning with expected model output changes". In: *European Conference on Computer Vision*. Springer, 2014, pp. 562–577.

[219] Alexander Freytag et al. "Labeling examples that matter: Relevance-based active learning with gaussian processes". In: *German Conference on Pattern Recognition*. Springer, 2013, pp. 282–291.

[220] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Sparse inverse covariance estimation with the graphical lasso". In: *Biostatistics* 9.3 (2008). Publisher: Oxford University Press, pp. 432–441.

[221] Cong Fu et al. "Collaborative Policy Learning for Open Knowledge Graph Reasoning". In: *EMNLP* (2019).

[222] Tak-chung Fu. "A review on time series data mining". en. In: *Engineering Applications of Artificial Intelligence* 24.1 (Feb. 2011), pp. 164–181. (Visited on 06/25/2021).

[223] Michael Fürst, Oliver Wasenmüller, and Didier Stricker. "LRPD: Long Range 3D Pedestrian Detection Leveraging Specific Strengths of LiDAR and RGB". In: *arXiv:2006.09738 [cs]* (June 2020). arXiv: 2006.09738. (Visited on 07/08/2021).

[224] Yarin Gal. "Uncertainty in Deep Learning". en. PhD. 2016.

[225] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. Ed. by Maria-Florina Balcan and Kilian Q. Weinberger. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 1050–1059.

[226] Yarin Gal, Jiri Hron, and Alex Kendall. "Concrete dropout". In: *Advances in neural information processing systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[227] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. "Deep bayesian active learning with image data". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1183–1192.

[228] Yarin Gal, Rowan McAllister, and Carl E. Rasmussen. "Improving PILCO with Bayesian Neural Network Dynamics Models". In: *Data-Efficient Machine Learning workshop, ICML*. Apr. 2016.

[229] Kuzman Ganchev et al. "Posterior regularization for structured latent variable models". In: *The Journal of Machine Learning Research* 11 (2010). Publisher: JMLR. org, pp. 2001–2049.

[230] Han Gao, Luning Sun, and Jian-Xun Wang. "PhyGeoNet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain". In: *Journal of Computational Physics* 428 (2021). Publisher: Elsevier, p. 110079.

[231] Mingfei Gao et al. "Consistency-based semi-supervised active learning: Towards minimizing labeling cost". In: *European Conference on Computer Vision*. Springer, 2020, pp. 510–526.

[232] Wei Gao et al. "TS-CAM: Token Semantic Coupled Attention Map for Weakly Supervised Object Localization". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2886–2895.

[233] Wenjing Gao and Minxian Li. "Unsupervised Clustering Active Learning for Person Re-identification". In: *arXiv preprint arXiv:2112.13308* (2021).

[234] Artur S. d'Avila Garcez, Lus C. Lamb, and Dov M. Gabbay. *Neural-Symbolic Cognitive Reasoning*. 1st ed. Springer Publishing Company, Incorporated, 2008.

[235] Angel F. Garcia-Fernandez et al. "Poisson Multi-Bernoulli Mixture Filter: Direct Derivation and Implementation". In: *IEEE Transactions on Aerospace and Electronic Systems* 54.4 (Aug. 2018), pp. 1883–1901. (Visited on 06/24/2021).

[236] Javier Garcıa and Fernando Fernández. "A comprehensive survey on safe reinforcement learning". In: *Journal of Machine Learning Research* 16.1 (2015), pp. 1437–1480.

[237] Daniel Garijo and María Poveda-Villalón. "Best Practices for Implementing FAIR Vocabularies and Ontologies on the Web". In: *arXiv:2003.13084 [cs]* (Mar. 2020). arXiv: 2003.13084. (Visited on 06/24/2021).

[238] Jochen Gast and Stefan Roth. "Lightweight Probabilistic Deep Networks". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 3369–3378.

[239] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423.

[240] Jakob Gawlikowski et al. "A survey of uncertainty in deep neural networks". In: *ArXiv* abs/2107.03342 (2021).

[241] Yunhao Ge et al. "A peek into the reasoning of neural networks: Interpreting with structural visual concepts". en. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 2195–2204.

[242] Claudio Gentile, Zhilei Wang, and Tong Zhang. "Achieving Minimax Rates in Pool-Based Batch Active Learning". In: *arXiv preprint arXiv:2202.05448* (2022).

[243] Georgios Georgakis et al. "Synthesizing training data for object detection in indoor scenes". In: *arXiv preprint arXiv:1702.07836* (2017).

[244] Maximilian Gerwien, Alexander Jungmann, and Rick Voßwinkel. "Towards Situation-Aware Decision-Making for Automated Driving". In: *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*. Feb. 2021, pp. 185–189.

[245] Amirata Ghorbani et al. "Towards Automatic Concept-based Explanations". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[246] Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. "Model Selection in Bayesian Neural Networks via Horseshoe Priors." In: *J. Mach. Learn. Res.* 20.182 (2019), pp. 1–46.

[247] Markus Giftthaler et al. "A Family of Iterative Gauss-Newton Shooting Methods for Nonlinear Optimal Control". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, Oct. 2018, pp. 1–9. (Visited on 06/24/2021).

[248] Justin Gilmer et al. "Neural Message Passing for Quantum Chemistry". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 1263–1272.

[249] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[250] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[251] Tilmann Gneiting and Adrian E Raftery. "Strictly proper scoring rules, prediction, and estimation". In:

Journal of the American Statistical Association 102.477 (2007), pp. 359–378.

[252] Simon Godsill. "Particle Filtering: the First 25 Years and beyond". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom: IEEE, May 2019, pp. 7760–7764. (Visited on 06/28/2021).

[253] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014.

[254] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[255] Liang Gou et al. "VATLD: A Visual Analytics System to Assess, Understand and Improve Traffic Light Detection". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 261–271.

[256] Josif Grabocka, Randolf Scholz, and Lars Schmidt-Thieme. *Learning Surrogate Losses*. 2020. URL: https://openreview.net/forum?id=BkePHaVKwS.

[257] Karl Granstrom, Markus Baum, and Stephan Reuter. "Extended Object Tracking: Introduction, Overview and Applications". In: *Journal of Advances in Information Fusion* 12.2 (2016), pp. 139–174.

[258] Karl Granström et al. "Poisson Multi-Bernoulli Mixtures for Sets of Trajectories". In: *arXiv:1912.08718 [cs, eess, stat]* (Dec. 2019). arXiv: 1912.08718. (Visited on 06/24/2021).

[259] Alex Graves. "Practical variational inference for neural networks". In: *Advances in neural information processing systems*. Ed. by J. Shawe-Taylor et al. Vol. 24. Curran Associates, Inc., 2011.

[260] Mara Graziani et al. "Concept attribution: Explaining CNN decisions to physicians". In: *Computers in biology and medicine* 123 (2020). Publisher: Elsevier, p. 103865.

[261] Ross Greer, Nachiket Deo, and Mohan Trivedi. "Trajectory Prediction in Autonomous Driving with a Lane Heading Auxiliary Loss". In: *IEEE Robotics and Automation Letters* 6.3 (2021). Publisher: IEEE, pp. 4907–4914.

[262] Aditya Grover and Jure Leskovec. "node2vec: Scalable Feature Learning for Networks". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 2016, pp. 855–864.

[263] Denis Gudovskiy et al. "Explain to fix: A framework to interpret and correct DNN object detector predictions". In: *arXiv preprint arXiv:1811.08011* (2018).

[264] Nitesh B. Gundavarapu et al. "Structured aleatoric uncertainty in human pose estimation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops*. June 2019.

[265] Chuan Guo et al. "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1321–1330.

[266] Lingbing Guo, Zequn Sun, and Wei Hu. "Learning to exploit long-term relational dependencies in knowledge graphs". In: *International Conference on Machine Learning*. PMLR, 2019, pp. 2505–2514.

[267] Ruocheng Guo et al. "A Survey of Learning Causality with Data: Problems and Methods". In: *ACM Computing Surveys* 53.4 (Sept. 2020). arXiv: 1809.09337, pp. 1–37. (Visited on 05/20/2021).

[268] Shu Guo et al. "Knowledge Graph Embedding Preserving Soft Logical Regularity". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 425–434.

[269] Shu Guo et al. "Knowledge Graph Embedding with Iterative Guidance from Soft Rules". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[270] Shu Guo et al. "SSE: Semantically Smooth Embedding for Knowledge Graphs". In: *IEEE Transactions on Knowledge and Data Engineering* 29.4 (2017), pp. 884–897.

[271] Yunhui Guo et al. "Improved Schemes for Episodic Memory-based Lifelong Learning". In: *Advances in Neural Information Processing Systems* 33 (2020).

[272] Abhishek Gupta et al. "Meta-Reinforcement Learning of Structured Exploration Strategies". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[273] Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. "CaRe: Open Knowledge Graph Embeddings". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 378–388.

[274] Vıctor Gutiérrez-Basulto and Steven Schockaert. "From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules". In: *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning*. 2018.

[275] Isabelle Guyon et al. "Structural risk minimization for character recognition". In: *Advances in neural information processing systems* 4 (1991).

[276] David Ha and Jürgen Schmidhuber. "Recurrent World Models Facilitate Policy Evolution". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[277] Tuomas Haarnoja et al. "Backprop KF: Learning Discriminative Deterministic State Estimators". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. eventplace: Barcelona, Spain. Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 4383–4391.

[278] Tuomas Haarnoja et al. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 1861–1870.

[279] Guy Hacohen, Avihu Dekel, and Daphna Weinshall. "Active Learning on a Budget: Opposite Strategies

Suit High and Low Budgets". In: *arXiv preprint arXiv:2202.02794* (2022).

[280] Danijar Hafner et al. "Dream to Control: Learning Behaviors by Latent Imagination". In: *International Conference on Learning Representations*. 2020.

[281] Danijar Hafner et al. "Learning Latent Dynamics for Planning from Pixels". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 2555–2565.

[282] Tameru Hailesilassie. "Rule extraction algorithm for deep neural networks: A review". In: *arXiv preprint arXiv:1610.05267* (2016).

[283] Lavdim Halilaj et al. "A Knowledge Graph-based Approach for Situation Comprehension in Driving Scenarios". In: *Eighteenth Extended Semantic Web Conference - In-Use Track*. 2021.

[284] Joseph Y. Halpern and Judea Pearl. "Causes and Explanations: A Structural-Model Approach. Part I: Causes". en. In: *The British Journal for the Philosophy of Science* 56.4 (Dec. 2005), pp. 843–887. (Visited on 06/10/2021).

[285] Ali Harakeh, Michael Smart, and Steven L. Waslander. "BayesOD: A bayesian approach for uncertainty estimation in deep object detectors". In: *2020 IEEE international conference on robotics and automation, ICRA 2020, paris, france, may 31 - august 31, 2020*. IEEE, 2020, pp. 87–93.

[286] Patrick Hart and Alois Knoll. "Counterfactual Policy Evaluation for Decision-Making in Autonomous Driving". In: *arXiv:2003.11919 [cs, stat]* (Nov. 2020). arXiv: 2003.11919. (Visited on 04/19/2021).

[287] Irtiza Hasan et al. "Forecasting People Trajectories and Head Poses by Jointly Reasoning on Tracklets and Vislets". In: *arXiv:1901.02000 [cs]* (Oct. 2019). arXiv: 1901.02000. (Visited on 07/06/2021).

[288] Irtiza Hasan et al. "Generalizable pedestrian detection: The elephant in the room". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11328–11337.

[289] Carsten Hasberg et al. "Integrating Spline Curves in Road Constraint Object Tracking". In: *2008 11th International IEEE Conference on Intelligent Transportation Systems*. Beijing, China: IEEE, Oct. 2008, pp. 1009–1014. (Visited on 06/24/2021).

[290] Hado van Hasselt, Arthur Guez, and David Silver. "Deep Reinforcement Learning with Double Q-Learning". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. event-place: Phoenix, Arizona. AAAI Press, 2016, pp. 2094–2100.

[291] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009.

[292] Matthew Hausknecht and Peter Stone. "Deep Recurrent Q-Learning for Partially Observable MDPs". In: *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (AAAI-SDMIA15)*. event-place: Arlington, Virginia, USA. Nov. 2015.

[293] Elmar Haussmann et al. "Scalable active learning for object detection". In: *2020 IEEE intelligent vehicles symposium (iv)*. IEEE, 2020, pp. 1430–1435.

[294] Simon Haykin, ed. *Kalman Filtering and Neural Networks*. en. New York, USA: John Wiley & Sons, Inc., Oct. 2001. (Visited on 06/25/2021).

[295] Congjie He, Meng Ma, and Ping Wang. "Extract interpretability-accuracy balanced rules from artificial neural networks: A review". en. In: *Neurocomputing* 387 (Apr. 2020), pp. 346–358. (Visited on 06/22/2021).

[296] Jingrui He and Jaime Carbonell. "Nearest-neighbor-based active learning for rare category detection". In: *Advances in neural information processing systems* 20 (2007).

[297] K. He et al. "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988.

[298] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[299] Yihui He et al. "Bounding box regression with uncertainty for accurate object detection". In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2019, pp. 2888–2897.

[300] Nicolas Heess et al. "Learning Continuous Control Policies by Stochastic Value Gradients". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015.

[301] Tor Aksel N Heirung and Ali Mesbah. "Stochastic nonlinear model predictive control with active model discrimination: a closed-loop fault diagnosis application". In: *IFAC-PapersOnLine* 50.1 (2017). Publisher: Elsevier, pp. 15934–15939.

[302] Tor Aksel N Heirung, Tito LM Santos, and Ali Mesbah. "Model predictive control with active learning for stochastic systems with structural model uncertainty: Online model discrimination". In: *Computers & Chemical Engineering* 128 (2019). Publisher: Elsevier, pp. 128–140.

[303] Tor Aksel N Heirung et al. "Model predictive control with active learning under model uncertainty: Why, when, and how". In: *AIChE Journal* 64.8 (2018). Publisher: Wiley Online Library, pp. 3071–3081.

[304] Patrick Hemmer, Niklas Kühl, and Jakob Schöffer. "Deal: deep evidential active learning for image classification". In: *Deep Learning Applications, Volume 3*. Springer, 2022, pp. 171–192.

[305] Dan Hendrycks et al. "CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review". In: *arXiv:2103.06268 [cs]* (Mar. 2021). arXiv: 2103.06268. (Visited on 07/06/2021).

[306] Andreas Hermann and Jorg Desel. "Driving situation analysis in automotive environment". en. In: *2008 IEEE International Conference on Vehicular Electronics and Safety*. Columbus, OH: IEEE, Sept. 2008, pp. 216–221. (Visited on 07/06/2021).

[307] Quercus Hernandez et al. "Structure-preserving neural networks". In: *Journal of Computational Physics* 426 (2021). Publisher: Elsevier, p. 109950.

[308] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. "A Survey and Critique of Multiagent Deep Reinforcement Learning". In: *Autonomous Agents and Multi-Agent Systems* 33.6 (Nov. 2019). arXiv: 1810.05587, pp. 750–797. (Visited on 07/06/2021).

[309] Matteo Hessel et al. "Rainbow: Combining Improvements in Deep Reinforcement Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018).

[310] Geoffrey E. Hinton and Drew van Camp. "Keeping the neural networks simple by minimizing the description length of the weights". In: *Proceedings of the sixth annual conference on computational learning theory*. COLT '93. Number of pages: 9 Place: Santa Cruz, California, USA. New York, NY, USA: Association for Computing Machinery, 1993, pp. 5–13.

[311] Jonathan Ho and Stefano Ermon. "Generative Adversarial Imitation Learning". In: *arXiv:1606.03476 [cs]* (June 2016). arXiv: 1606.03476. (Visited on 12/12/2020).

[312] Vinh Thinh Ho et al. "Rule Learning from Knowledge Graphs Guided by Embedding Models". In: *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*. Ed. by Denny Vrandecic et al. Vol. 11136. Lecture Notes in Computer Science. Springer, 2018, pp. 72–90.

[313] Matthew D. Hoffman and Andrew Gelman. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research* 15.47 (2014), pp. 1593–1623.

[314] Adrian Hoffmann et al. "This Looks Like That... Does it? Shortcomings of Latent Space Prototype Interpretability in Deep Networks". In: *arXiv preprint arXiv:2105.02968* (2021).

[315] Fred Hohman et al. "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers". In: *IEEE Transactions on Visualization and Computer Graphics* 25 (2019), pp. 2674–2693.

[316] Paul W. Holland. "Statistics and Causal Inference". en. In: *Journal of the American Statistical Association* 81.396 (Dec. 1986), pp. 945–960. (Visited on 06/10/2021).

[317] Jan Hosang et al. "Taking a deeper look at pedestrians". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4073–4082.

[318] Timothy Hospedales et al. "Meta-learning in neural networks: A survey". In: *arXiv preprint arXiv:2004.05439* (2020).

[319] Timothy M Hospedales, Shaogang Gong, and Tao Xiang. "A unifying theory of active discovery and learning". In: *European conference on computer vision*. Springer, 2012, pp. 453–466.

[320] Timothy M Hospedales, Shaogang Gong, and Tao Xiang. "Finding rare classes: Active learning with generative and discriminative models". In: *IEEE transactions on knowledge and data engineering* 25.2 (2011). Publisher: IEEE, pp. 374–386.

[321] Neil Houlsby et al. "Bayesian active learning for classification and preference learning". In: *arXiv preprint arXiv:1112.5745* (2011).

[322] Kyle Hsu, Sergey Levine, and Chelsea Finn. "Unsupervised Learning via Meta-Learning". In: *International Conference on Learning Representations*. 2019.

[323] Yen-Chang Hsu, Yen-Cheng Liu, and Zsolt Kira. "Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines". In: *CoRR* abs/1810.12488 (2018). _eprint: 1810.12488.

[324] Haimin Hu and Jaime F Fisac. "Active Uncertainty Learning for Human-Robot Interaction: An Implicit Dual Control Approach". In: *arXiv preprint arXiv:2202.07720* (2022).

[325] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-Excitation Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7132–7141.

[326] Ronghang Hu and Amanpreet Singh. "Unit: Multimodal multitask learning with a unified transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1439–1449.

[327] Zhiting Hu et al. "Deep generative models with learnable knowledge constraints". In: *Advances in Neural Information Processing Systems* 31 (2018).

[328] Zhiting Hu et al. "Deep neural networks with massive learned knowledge". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 1670–1679.

[329] Zhiting Hu et al. "Harnessing deep neural networks with logic rules". In: *arXiv preprint arXiv:1603.06318* (2016).

[330] Gao Huang et al. "Densely Connected Convolutional Networks". en. In: *arXiv:1608.06993 [cs]* (Jan. 2018). arXiv: 1608.06993. (Visited on 07/12/2021).

[331] Yimin Huang and Marco Valtorta. "Identifiability in Causal Bayesian Networks: A Sound and Complete Algorithm". en. In: *proceedings of the 21st national conference on Artificial intelligence - Volume 2*. AAAI'06. Boston, Massachusetts: AAAI Press, July 2006, pp. 1149–1154.

[332] Yimin Huang and Marco Valtorta. "Pearl's Calculus of Intervention Is Complete". en. In: *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*. UAI'06. Arlington, Virginia, USA: AUAI Press, July 2006, pp. 217–224.

[333] Po-Yu Huang et al. "Efficient Uncertainty Estimation for Semantic Segmentation in Videos". In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*. Ed. by Vittorio Ferrari et al. Vol. 11205. Lecture Notes in Computer Science. Springer, 2018, pp. 536–552.

[334] Marco F. Huber. "Bayesian Perceptron: Towards fully Bayesian Neural Networks". In: *59th IEEE Conference on Decision and Control, CDC 2020, Jeju Island, South Korea, December 14-18, 2020*. IEEE, 2020, pp. 3179–3186.

[335] Constantin Hubmann et al. "Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction". In: *IEEE Transactions on Intelligent Vehicles* 3.1 (Mar. 2018), pp. 5–17. (Visited on 06/24/2021).

[336] Eyke Hüllermeier and Willem Waegeman. "Aleatoric and epistemic uncertainty in machine learning: an

introduction to concepts and methods". In: *Machine Learning* 110.3 (2021), pp. 457–506.

[337] Muayed Al-Huseiny and Ahmed Sajit. "Transfer learning with GoogLeNet for detection of lung cancer". In: *Indonesian Journal of Electrical Engineering and Computer Science* 22 (2021), p. 1078.

[338] León Illanes et al. "Symbolic Plans as High-Level Instructions for Reinforcement Learning". In: *Proceedings of the 30th International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI Press, 2020, pp. 540–550.

[339] Ilya Shpitser and Judea Pearl. "Identification of Joint Interventional Distributions in Recursive Semi-Markovian Causal Models.pdf". In: *In Proceedings of the Twenty-First National Conference on Artificial Intelligence*. Menlo Park, CA, July 2006, pp. 1219–1226. (Visited on 09/16/2020).

[340] Henrik Imberg, Johan Jonasson, and Marina Axelson-Fisk. "Optimal sampling in unbiased active learning". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 559–569.

[341] Hiroshi Inoue. "Data augmentation by pairing samples for images classification". In: *arXiv preprint arXiv:1801.02929* (2018).

[342] Boris Ivanovic et al. "MATS: An Interpretable Trajectory Forecasting Representation for Planning and Control". In: *arXiv:2009.07517 [cs, eess]* (Sept. 2020). arXiv: 2009.07517. (Visited on 01/14/2021).

[343] Max Jaderberg et al. "Human-level performance in 3D multiplayer games with population-based reinforcement learning". en. In: *Science* 364.6443 (May 2019). Publisher: American Association for the Advancement of Science Section: Research Article, pp. 859–865. (Visited on 07/05/2021).

[344] Max Jaderberg et al. "Reinforcement Learning with Unsupervised Auxiliary Tasks". In: *arXiv:1611.05397 [cs]* (Nov. 2016). arXiv: 1611.05397. (Visited on 07/07/2021).

[345] Ameya D Jagtap and George Em Karniadakis. "Extended Physics-Informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition Based Deep Learning Framework for Nonlinear Partial Differential Equations". In: *Communications in Computational Physics* 28.5 (2020), pp. 2002–2041.

[346] Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. "Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems". In: *Computer Methods in Applied Mechanics and Engineering* 365 (2020). Publisher: Elsevier, p. 113028.

[347] Suyog Dutt Jain and Kristen Grauman. "Active image segmentation propagation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2864–2873.

[348] Guoliang Ji et al. "Knowledge Graph Embedding via Dynamic Mapping Matrix". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 687–696.

[349] Shaoxiong Ji et al. "A Survey on Knowledge Graphs: Representation, Acquisition and Applications". en. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021). arXiv: 2002.00388, pp. 1–21. (Visited on 06/07/2021).

[350] Xiaowei Jia et al. "Physics guided recurrent neural networks for modeling dynamical systems: Application to monitoring water temperature and quality in lakes". In: *arXiv preprint arXiv:1810.02880* (2018).

[351] Xiaowei Jia et al. "Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles". In: *ACM/IMS Transactions on Data Science* 2.3 (2021). Publisher: ACM New York, NY, pp. 1–26.

[352] Lu Jiang et al. "MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 2304–2313.

[353] Jin Tian. "Studies in Causal Reasoning and Learning". PhD thesis. Los Angeles: Department of Computer Science, University of California, 2002. (Visited on 09/04/2020).

[354] Rico Jonschkowski and Oliver Brock. "End-To-End Learnable Histogram Filters". In: *Workshop on Deep Learning for Action and Interaction at NIPS*. Dec. 2016.

[355] Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. "Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors". In: *arXiv:1805.11122 [cs, stat]* (May 2018). arXiv: 1805.11122. (Visited on 06/24/2021).

[356] Ce Ju et al. "Interaction-aware Kalman Neural Networks for Trajectory Prediction". In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. 2020, pp. 1793–1800.

[357] Amílcar Soares Júnior, Chiara Renso, and Stan Matwin. "Analytic: An active learning system for trajectory classification". In: *IEEE computer graphics and applications* 37.5 (2017). Publisher: IEEE, pp. 28–39.

[358] Christoph Käding et al. "Active and continuous exploration with deep neural networks and expected model output changes". In: *arXiv preprint arXiv:1612.06129* (2016).

[359] Christoph Käding et al. "Active Learning for Regression Tasks with Expected Model Output Changes." In: *BMVC*. 2018, p. 103.

[360] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". en. In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. (Visited on 06/28/2021).

[361] R. E. Kalman. "When Is a Linear Control System Optimal?" In: *Journal of Basic Engineering* 86.1 (Mar. 1964), pp. 51–60. (Visited on 06/20/2021).

[362] Sebastian Kaltenbach and Phaedon-Stelios Koutsourelakis. "Incorporating physical constraints in a deep probabilistic machine learning framework for coarse-graining dynamical systems". In: *Journal of Computational Physics* 419 (2020). Publisher: Elsevier, p. 109673.

[363] Sebastian Kaltenbach and Phaedon-Stelios Koutsourelakis. "Physics-aware, deep probabilistic modeling of

multiscale dynamics in the Small Data regime". In: *arXiv preprint arXiv:2102.04269* (2021).

[364] Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. "Text summarization from legal documents: a survey". en. In: *Artificial Intelligence Review* 51.3 (Mar. 2019), pp. 371–402. (Visited on 07/15/2021).

[365] Bingyi Kang et al. "Few-Shot Object Detection via Feature Reweighting". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 8419–8428.

[366] Neel Kant. "Recent Advances in Neural Program Synthesis". In: *arXiv:1802.02353 [cs]* (Feb. 2018). arXiv: 1802.02353. (Visited on 04/29/2021).

[367] Chieh-Chi Kao et al. "Localization-aware active learning for object detection". In: *Asian Conference on Computer Vision*. Springer, 2018, pp. 506–522.

[368] Andrei Kapishnikov et al. "Xrai: Better attributions through regions". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4948–4957.

[369] Maximilian Karl et al. "Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data". In: *arXiv:1605.06432 [cs, stat]* (Mar. 2017). arXiv: 1605.06432. (Visited on 06/24/2021).

[370] Anuj Karpatne et al. "Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling". In: *arXiv:1710.11431 [physics, stat]* (Feb. 2018). arXiv: 1710.11431. (Visited on 07/20/2021).

[371] Henry Kautz. *The Third AI Summer, AAAI Robert S. Engelmore Memorial Lecture, Thirty-fourth AAAI Conference on Artificial Intelligence, New York, NY, February 10, 2020*. Dec. 2020. URL: https://www.cs.rochester.edu/u/kautz/talks/index.html (visited on 06/25/2021).

[372] Seyed Mehran Kazemi and David Poole. "SimplE Embedding for Link Prediction in Knowledge Graphs". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. event-place: Montréal, Canada. Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 4289–4300.

[373] Nan Rosemary Ke et al. "Modeling the Long Term Future in Model-Based Reinforcement Learning". In: *International Conference on Learning Representations*. 2019.

[374] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 5574–5584.

[375] Hoel Kervadec et al. "Bounding boxes for weakly supervised segmentation: Global constraints get close to full supervision". In: *Medical Imaging with Deep Learning*. PMLR, 2020, pp. 365–381.

[376] Hoel Kervadec et al. "Constrained deep networks: Lagrangian optimization via log-barrier extensions". In: *arXiv preprint arXiv:1904.04205* (2019).

[377] D. Khosla. "Adaptive Kalman filter approach for road geometry estimation". In: *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*. Vol. 2. Shanghai, China: IEEE, 2003, pp. 1145–1151. (Visited on 06/24/2021).

[378] Been Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)". In: *International conference on machine learning*. PMLR, 2018, pp. 2668–2677.

[379] Jinkyu Kim and John Canny. "Interpretable learning for self-driving cars by visualizing causal attention". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2942–2950.

[380] Jinkyu Kim et al. "Textual explanations for self-driving vehicles". In: *Proceedings of the European conference on computer vision (ECCV)*. Vol. 11206 LNCS. 2018, pp. 563–578.

[381] Kwanyoung Kim et al. "Task-aware variational adversarial active learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8166–8175.

[382] Diederik P. Kingma, Tim Salimans, and Max Welling. "Variational dropout and the local reparameterization trick". In: *Proceedings of the 28th international conference on neural information processing systems - volume 2*. NIPS'15. Number of pages: 9 Place: Montreal, Canada. Cambridge, MA, USA: MIT Press, 2015, pp. 2575–2583.

[383] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014.

[384] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *Proceedings of the 5th International Conference on Learning Representations*. ICLR '17. event-place: Palais des Congrès Neptune, Toulon, France. 2017.

[385] B. Ravi Kiran et al. "Deep Reinforcement Learning for Autonomous Driving: A Survey". In: *arXiv:2002.00444 [cs]* (Jan. 2021). arXiv: 2002.00444. (Visited on 06/29/2021).

[386] James Kirkpatrick et al. "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the National Academy of Sciences* 114.13 (2017), pp. 3521–3526.

[387] Jonathan Ko and Dieter Fox. "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models". en. In: *Autonomous Robots* 27.1 (July 2009), pp. 75–90. (Visited on 06/28/2021).

[388] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. "Siamese Neural Networks For One-shot Image Recognition". In: *International Conference on Machine Learning*. 2015.

[389] Pang Wei Koh et al. "Concept bottleneck models". In: *International Conference on Machine Learning*. PMLR, 2020, pp. 5338–5348.

[390] Ksenia Konyushova et al. "Active offline policy selection". In: *Advances in Neural Information Processing Systems* 34 (2021).

[391] Markus Koschi and Matthias Althoff. "SPOT: A tool for set-based prediction of traffic participants". en. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los

Angeles, CA, USA: IEEE, June 2017, pp. 1686–1693. (Visited on 07/09/2021).

[392] A Koscianski and JE Souza de Cursi. "Physically constrained neural networks and regularization of inverse problems". In: *Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimisation. Rio de Janeiro, Brazil*. Citeseer, 2005.

[393] Jannik Kossen et al. "Active testing: Sample-efficient model evaluation". In: *International Conference on Machine Learning*. PMLR, 2021, pp. 5753–5763.

[394] Daniel Kottke et al. "Multi-class probabilistic active learning". In: *Proceedings of the Twenty-second European Conference on Artificial Intelligence*. 2016, pp. 586–594.

[395] Daniel Kottke et al. "Toward optimal probabilistic active learning using a Bayesian approach". In: *Machine Learning* 110.6 (2021). Publisher: Springer, pp. 1199–1231.

[396] Hanna Krasowski, Xiao Wang, and Matthias Althoff. "Safe Reinforcement Learning for Autonomous Lane Changing Using Set-Based Prediction". en. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Rhodes, Greece: IEEE, Sept. 2020, pp. 1–7. (Visited on 07/09/2021).

[397] Florian Kraus and Klaus Dietmayer. "Uncertainty estimation in one-stage object detection". In: *2019 IEEE intelligent transportation systems conference (ITSC)*. Number of pages: 8. Auckland, New Zealand: IEEE Press, 2019, pp. 53–60.

[398] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. "Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 5436–5446.

[399] David Krueger et al. "Active reinforcement learning: Observing rewards at a cost". In: *arXiv preprint arXiv:2011.06709* (2020).

[400] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. "Accurate uncertainties for deep learning using calibrated regression". In: *Proceedings of the 35th international conference on machine learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of machine learning research. PMLR, July 2018, pp. 2796–2804.

[401] Weicheng Kuo et al. "Cost-sensitive active learning for intracranial hemorrhage detection". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 715–723.

[402] Fabian Küppers et al. "Multivariate Confidence Calibration for Object Detection". In: *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020.

[403] Ugur Kursuncu, Manas Gaur, and Amit Sheth. "Knowledge infused learning (K-IL): Towards deep incorporation of knowledge in deep learning". In: *arXiv preprint arXiv:1912.00512* (2019).

[404] Thanard Kurutach et al. "Model-Ensemble Trust-Region Policy Optimization". In: *International Conference on Learning Representations*. 2018.

[405] Beong-woo Kwak et al. "TrustAL: Trustworthy Active Learning using Knowledge Distillation". In: *arXiv preprint arXiv:2201.11661* (2022).

[406] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 6402–6413.

[407] Luis C. Lamb et al. "Graph Neural Networks Meet Neural-Symbolic Computing: A Survey and Perspective". In: *arXiv:2003.00330 [cs]* (June 2021). arXiv: 2003.00330. (Visited on 06/30/2021).

[408] Nick Lamm et al. "Vehicle trajectory prediction by transfer learning of semi-supervised models". In: *arXiv preprint arXiv:2007.06781* (2020).

[409] Alex H. Lang et al. "PointPillars: Fast Encoders for Object Detection from Point Clouds". In: *CoRR* abs/1812.05784 (2018). _eprint: 1812.05784.

[410] Léa Laporte et al. "Nonconvex regularizations for feature selection in ranking with sparse SVM". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.6 (2014). Publisher: IEEE, pp. 1118–1130.

[411] Sebastian Lapuschkin et al. "Unmasking Clever Hans Predictors and Assessing What Machines Really Learn". In: *CoRR* abs/1902.10178 (2019). _eprint: 1902.10178.

[412] Kody Law, Andrew Stuart, and Konstantinos Zygalakis. *Data Assimilation: A Mathematical Introduction*. 1st ed. 2015. Texts in Applied Mathematics 62. Cham: Springer International Publishing : Imprint: Springer, 2015.

[413] Michael Truong Le et al. "Uncertainty estimation for deep neural object detectors in safety-critical applications". In: *2018 21st international conference on intelligent transportation systems (ITSC)*. 2018, pp. 3873–3878.

[414] Sang-Woo Lee et al. "Overcoming Catastrophic Forgetting by Incremental Moment Matching". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*. Ed. by Isabelle Guyon et al. 2017, pp. 4652–4662.

[415] Sanghack Lee and Elias Bareinboim. "Causal Effect Identifiability under Partial-Observability". en. In: *International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, Nov. 2020, pp. 5692–5701.

[416] Serin Lee et al. "Fast uncertainty estimation for deep learning based optical flow". In: *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. 2020, pp. 10138–10144.

[417] Youngwan Lee et al. "Localization uncertainty estimation for anchor-free object detection". In: *arXiv preprint arXiv:2006.15607* (2020).

[418] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. "A survey on motion prediction and risk assessment for intelligent vehicles". en. In: *ROBOMECH Journal* 1.1 (Dec. 2014), p. 1. (Visited on 07/07/2021).

[419] Elena Leitner, Georg Rehm, and Julián Moreno-Schneider. "A Dataset of German Legal Documents for Named Entity Recognition". In: *arXiv:2003.13016*

[cs] (Mar. 2020). arXiv: 2003.13016. (Visited on 06/28/2021).

[420] David D Lewis and William A Gale. "A sequential algorithm for training text classifiers". In: *SIGIR'94*. Springer, 1994, pp. 3–12.

[421] *LexNLP*. en-US. URL: https://contraxsuite.com/lexnlp/ (visited on 07/06/2021).

[422] Aixuan Li et al. "Uncertainty-aware joint salient object and camouflaged object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 10071–10081.

[423] Bo Li and Tommy Sonne Alstrøm. "On uncertainty estimation in active learning for image segmentation". In: *arXiv preprint arXiv:2007.06364* (2020).

[424] Chenchen Li et al. "A survey on approaches and applications of knowledge representation learning". In: *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2020, pp. 312–319.

[425] Chunyuan Li et al. "Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. Ed. by Dale Schuurmans and Michael P. Wellman. AAAI Press, 2016, pp. 1788–1794.

[426] Honglin Li et al. "Continual Learning Using Bayesian Neural Networks". In: *CoRR* abs/1910.04112 (2019). _eprint: 1910.04112.

[427] Tao Li and Vivek Srikumar. "Augmenting neural networks with first-order logic". In: *arXiv preprint arXiv:1906.06298* (2019).

[428] Tao Li et al. "Adversarial Active Learning for Named Entity Recognition in Cybersecurity". In: *CMC-COMPUTERS MATERIALS & CONTINUA* 66.1 (2021), pp. 407–420.

[429] Xiang Li et al. "Smoothing the Geometry of Probabilistic Box Embeddings". In: *International Conference on Learning Representations*. 2019.

[430] Xiao Li et al. "Differentiable logic layer for rule guided trajectory prediction". In: *Conf. on Robot Learning*. 2020.

[431] Xin Li and Yuhong Guo. "Adaptive active learning for image classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 859–866.

[432] Zhenguo Li et al. "Meta-sgd: Learning to learn quickly for few-shot learning". In: *arXiv preprint arXiv:1707.09835* (2017).

[433] Tomer Libal and Alexander Steen. "Towards an Executable Methodology for the Formalization of Legal Texts". en. In: *Logic and Argumentation*. Ed. by Mehdi Dastani, Huimin Dong, and Leon van der Torre. Vol. 12061. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 151–165. (Visited on 05/06/2021).

[434] Timothy P. Lillicrap et al. "Continuous control with deep reinforcement learning". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2016.

[435] LiMin Fu. "Rule generation from neural networks". en. In: *IEEE Transactions on Systems, Man, and Cybernetics* 24.8 (Aug. 1994), pp. 1114–1124. (Visited on 06/22/2021).

[436] Tsung-Yi Lin et al. "Feature Pyramid Networks for Object Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 936–944.

[437] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2999–3007.

[438] Tzu-Chieh Lin et al. "Pedestrian detection from lidar data via cooperative deep and hand-crafted features". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1922–1926.

[439] Yankai Lin et al. "Knowledge Representation Learning: A Quantitative Review". In: *CoRR* abs/1812.10901 (2018). _eprint: 1812.10901.

[440] Yankai Lin et al. "Learning Entity and Relation Embeddings for Knowledge Graph Completion". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI'15. event-place: Austin, Texas. AAAI Press, 2015, pp. 2181–2187.

[441] Marco Lippi and Paolo Torroni. "Argumentation Mining: State of the Art and Emerging Trends". In: *ACM Transactions on Internet Technology* 16.2 (Mar. 2016), 10:1–10:25. (Visited on 07/06/2021).

[442] Hanxiao Liu, Karen Simonyan, and Yiming Yang. "DARTS: Differentiable Architecture Search". In: *International Conference on Learning Representations*. 2019.

[443] Jeremiah Z. Liu et al. "Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020.

[444] Jerry Liu et al. "Deep Structured Reactive Planning". In: *arXiv:2101.06832 [cs]* (Jan. 2021). arXiv: 2101.06832. (Visited on 02/07/2021).

[445] Mengchen Liu et al. "Analyzing the Noise Robustness of Deep Neural Networks". In: *IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2018, pp. 60–71.

[446] Quan Liu et al. "Probabilistic Reasoning via Deep Learning: Neural Association Models". In: *CoRR* abs/1603.07704 (2016). _eprint: 1603.07704.

[447] Wei Liu et al. "SSD: Single Shot MultiBox Detector". en. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 21–37.

[448] Xihui Liu et al. "Hydraplus-net: attentive deep features for pedestrian analysis". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* 2017-Octob.c (2017), pp. 350–359.

[449] Zimo Liu et al. "Deep reinforcement active learning for human-in-the-loop person re-identification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6122–6131.

[450] Waymo LLC. *Waymo Safety Report 09-2020*. Sept. 2020.

[451] Francesco Locatello et al. "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *international conference on machine learning*. PMLR, 2019, pp. 4114–4124.

[452] Bo Long et al. "Active learning for ranking through expected loss optimization". In: *IEEE transactions on knowledge and data engineering* 27.5 (2014). Publisher: IEEE, pp. 1180–1191.

[453] Noel Loo, Siddharth Swaroop, and Richard E. Turner. "Generalized Variational Continual Learning". In: *Proceedings of the 9th International Conference on Learning Representations*. OpenReview.net, 2021.

[454] David Lopez-Paz and Marc'Aurelio Ranzato. "Gradient Episodic Memory for Continual Learning". In: *arXiv:1706.08840 [cs]* (Nov. 2017). arXiv: 1706.08840. (Visited on 02/03/2021).

[455] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. "A general framework for uncertainty estimation in deep learning". In: *IEEE Robotics and Automation Letters* PP (Feb. 2020), pp. 1–1.

[456] Christos Louizos and Max Welling. "Structured and efficient variational deep learning with matrix gaussian posteriors". In: *Proceedings of the 33rd international conference on international conference on machine learning - volume 48*. ICML'16. Number of pages: 9. New York, NY, USA: JMLR.org, 2016, pp. 1708–1716.

[457] David Lowell, Zachary C Lipton, and Byron C Wallace. "Practical obstacles to deploying active learning". In: *arXiv preprint arXiv:1807.04801* (2018).

[458] Lu Lu et al. "Physics-informed neural networks with hard constraints for inverse design". In: *SIAM Journal on Scientific Computing* 43.6 (2021). Publisher: SIAM, B1105–B1132.

[459] Xin Lu et al. "MimicDet: Bridging the Gap Between One-Stage and Two-Stage Object Detection". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 541–557.

[460] Yan Lu et al. "Geometry uncertainty projection network for monocular 3D object detection". In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*. Oct. 2021, pp. 3111–3121.

[461] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).

[462] Christian Lundquist and Thomas B. Schön. "Joint ego-motion and road geometry estimation". en. In: *Information Fusion* 12.4 (Oct. 2011), pp. 253–263. (Visited on 06/24/2021).

[463] W. Luo, B. Yang, and R. Urtasun. "Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3569–3577.

[464] Wenhan Luo et al. "Multiple object tracking: A literature review". en. In: *Artificial Intelligence* 293 (Apr. 2021), p. 103448. (Visited on 06/28/2021).

[465] Inc. Lyft. *Lyft Safety Report 2020*. 2020. URL: https://self-driving.lyft.com/wp-content/uploads/2020/06/Safety_Report_2020.pdf (visited on 06/20/2021).

[466] *Lynx - Legal Knowledge Graph for Smart Compliance Services in Multilingual Europe*. URL: https://lynx-project.eu/ (visited on 07/02/2021).

[467] Liheng Ma, Reihaneh Rabbany, and Adriana Romero-Soriano. "Graph Attention Networks with Positional Embeddings". In: *CoRR* abs/2105.04037 (2021). _eprint: 2105.04037.

[468] Yuxin Ma et al. "A Visual Analytics Framework for Explaining and Diagnosing Transfer Learning Processes". In: *arXiv preprint arXiv:2009.06876* (2020).

[469] David J. C. MacKay. "A Practical Bayesian Framework for Backpropagation Networks". In: *Neural Comput.* 4.3 (May 1992). Place: Cambridge, MA, USA Publisher: MIT Press, pp. 448–472.

[470] Prashan Madumal et al. "Explainable Reinforcement Learning through a Causal Lens". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.03 (Apr. 2020). Number: 03, pp. 2493–2500. (Visited on 01/07/2021).

[471] Maierhofer et al. "Formalization of Interstate Traffic Rules in Temporal Logic". In: *Proc. of the IEEE Intelligent Vehicles Symposium* (2020).

[472] Andrey Malinin and Mark Gales. "Predictive uncertainty estimation via prior networks". In: *Proceedings of the 32nd international conference on neural information processing systems*. NIPS'18. Number of pages: 12 Place: Montréal, Canada. Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 7047–7058.

[473] Robin Manhaeve et al. "DeepProbLog: Neural Probabilistic Logic Programming". In: *arXiv:1805.10872 [cs]* (Dec. 2018). arXiv: 1805.10872. (Visited on 06/30/2021).

[474] Jiayuan Mao et al. "The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision". In: *arXiv:1904.12584 [cs]* (Apr. 2019). arXiv: 1904.12584. (Visited on 06/25/2021).

[475] Zhiping Mao, Ameya D Jagtap, and George Em Karniadakis. "Physics-informed neural networks for high-speed flows". In: *Computer Methods in Applied Mechanics and Engineering* 360 (2020). Publisher: Elsevier, p. 112789.

[476] Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. "Imposing hard constraints on deep networks: Promises and limitations". In: *arXiv preprint arXiv:1706.02025* (2017).

[477] E. Mazor et al. "Interacting multiple model methods in target tracking: a survey". In: *IEEE Transactions on Aerospace and Electronic Systems* 34.1 (Jan. 1998), pp. 103–123. (Visited on 06/28/2021).

[478] Paul McNamara and Frederik Van De Putte. *Deontic Logic*. Ed. by Edward N. Zalta. 2021. URL: https://plato.stanford.edu/archives/spr2021/entries/logic-deontic/ (visited on 06/29/2021).

[479] Shike Mei, Jun Zhu, and Jerry Zhu. "Robust regbayes: Selectively incorporating first-order logic domain knowledge into bayesian models". In: *International Conference on Machine Learning*. PMLR, 2014, pp. 253–261.

[480] Stefano Melacci et al. "Can Domain Knowledge Alleviate Adversarial Attacks in Multi-Label Classifiers?" In: *arXiv preprint arXiv:2006.03833* (2020).

[481] Ali Mesbah. "Stochastic model predictive control with active uncertainty learning: A survey on dual control". In: *Annual Reviews in Control* 45 (2018). Publisher: Elsevier, pp. 107–117.

[482] Moustafa Meshry et al. "Neural rerendering in the wild". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6878–6887.

[483] Rémi Mignot and Geoffroy Peeters. "An analysis of the effect of data augmentation methods: experiments for a musical genre classification task". In: *Transactions of the International Society for Music Information Retrieval* 2.1 (2019).

[484] Damien de Mijolla et al. "Human-interpretable model explainability on high-dimensional data". In: *arXiv preprint arXiv:2010.07384* (2020).

[485] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. December 5-8, 2013, Lake Tahoe, Nevada, United States.* 2013, pp. 3111–3119.

[486] Dimity Miller et al. "Benchmarking sampling-based probabilistic object detectors". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops*. June 2019.

[487] Dimity Miller et al. "Dropout sampling for robust object detection in open-set conditions". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3243–3249.

[488] M.L. Miller, H.S. Stone, and I.J. Cox. "Optimizing Murty's ranked assignment method". In: *IEEE Transactions on Aerospace and Electronic Systems* 33.3 (July 1997), pp. 851–862. (Visited on 06/28/2021).

[489] Akshay Minocha and Navjyoti Singh. "Legal Document Similarity using Triples Extracted from Unstructured Text". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), 2018.

[490] *MIREL - MIning and REasoning with Legal texts*. URL: https://www.mirelproject.eu/ (visited on 07/06/2021).

[491] Sudhanshu Mittal et al. "Parting with illusions about deep active learning". In: *arXiv preprint arXiv:1912.05361* (2019).

[492] Takeru Miyato et al. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". In: *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2018), pp. 1979–1993.

[493] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 2015). Publisher: Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved., pp. 529–533.

[494] Mobileye. *Implementing the RSS Model on NHTSA Pre-Crash Scenarios*. July 2021. URL: https://static. mobileye.com/website/corporate/rss/rss_on_nhtsa. pdf.

[495] Arvind T Mohan et al. "Embedding hard physical constraints in neural network coarse-graining of 3d turbulence". In: *arXiv preprint arXiv:2002.00021* (2020).

[496] Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. 2019.

[497] Grégoire Montavon et al. "Explaining nonlinear classification decisions with deep Taylor decomposition". In: *Pattern Recognition* 65 (May 2017). Publisher: Elsevier BV, pp. 211–222.

[498] Yang-Sae Moon, Kyu-Young Whang, and Wook-Shin Han. "General match: a subsequence matching method in time-series databases based on generalized windows". en. In: *Proceedings of the 2002 ACM SIGMOD international conference on Management of data - SIGMOD '02*. Madison, Wisconsin: ACM Press, 2002, p. 382. (Visited on 06/25/2021).

[499] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2574–2582.

[500] Raha Moraffah et al. "Causal Interpretability for Machine Learning – Problems, Methods and Evaluation". In: *arXiv:2003.03934 [cs, stat]* (Mar. 2020). arXiv: 2003.03934. (Visited on 06/14/2021).

[501] Satya M Muddamsetty, NS Jahromi Mohammad, and Thomas B Moeslund. "Sidu: Similarity difference and uniqueness method for explainable ai". In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3269–3273.

[502] Jishnu Mukhoti and Yarin Gal. "Evaluating bayesian deep learning methods for semantic segmentation". In: *arXiv preprint arXiv:1811.12709* (2018).

[503] Prateek Munjal et al. "Towards robust and reproducible active learning using neural networks". In: *ArXiv, abs/2002.09564* (2020).

[504] Nikhil Muralidhar et al. "Incorporating prior domain knowledge into deep neural networks". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 36–45.

[505] Rohan Nanda et al. "Unsupervised and supervised text similarity systems for automated identification of national implementing measures of European directives". en. In: *Artificial Intelligence and Law* 27.2 (June 2019), pp. 199–225. (Visited on 07/14/2021).

[506] Yatin Nandwani, Abhishek Pathak, and Parag Singla. "A primal dual formulation for deep learning with constraints". In: *Advances in Neural Information Processing Systems* 32 (2019).

[507] Arpit Narechania, Ahsan Qamar, and Alex Endert. "Safetylens: Visual Data Analysis of Functional Safety of Vehicles". In: *IEEE transactions on visualization and computer graphics* (2020). Publisher: IEEE.

[508] Radford M. Neal. "MCMC Using Hamiltonian Dynamics". In: *Handbook of Markov Chain Monte Carlo* 54 (2010), pp. 113–162.

[509] Michael Neunert et al. "Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking". In: *2016 IEEE International Conference*

on Robotics and Automation (ICRA). Stockholm: IEEE, May 2016, pp. 1398–1404. (Visited on 06/24/2021).

[510] Jerzy Neyman. *Sur les applications de la thar des probabilities aux experiences Agaricales: Essay des principle*. English translation of excerpts (1990) by D. Dabrowska and T. Speed, in Statistical Science, 5:463–472. 1923.

[511] Andrew Ng and Stuart Russell. "Algorithms for Inverse Reinforcement Learning". In: *ICML*. 2000. (Visited on 06/19/2021).

[512] Cuong V. Nguyen et al. "Variational Continual Learning". In: *Proceedings of the 6th International Conference on Learning Representations*. OpenReview.net, 2018.

[513] Dai Quoc Nguyen et al. "QuatRE: Relation-Aware Quaternions for Knowledge Graph Embeddings". In: *arXiv preprint arXiv:2009.12517* (2020).

[514] Hieu T Nguyen and Arnold Smeulders. "Active learning using pre-clustering". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 79.

[515] NHTSA. *Automated Driving Systems: A Vision for Safety*. en. 2017. URL: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf (visited on 06/20/2021).

[516] NHTSA. *Federal Automated Vehicles Policy*. 2016. URL: https://www.hsdl.org/?view&did=795644 (visited on 06/20/2021).

[517] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. "A Three-Way Model for Collective Learning on Multi-Relational Data". In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11. event-place: Bellevue, Washington, USA. Madison, WI, USA: Omnipress, 2011, pp. 809–816.

[518] Matthew Niedoba et al. "Improving movement prediction of traffic actors using off-road loss and bias mitigation". In: *Workshop on'Machine Learning for Autonomous Driving'at Conference on Neural Information Processing Systems*. 2019.

[519] Giannis Nikolentzos, Polykarpos Meladianos, and Michalis Vazirgiannis. "Matching Node Embeddings for Graph Similarity". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. event-place: San Francisco, California, USA. AAAI Press, 2017, pp. 2429–2435.

[520] David Nistér et al. *An Introduction to the Safety Force Field*. Mar. 2019. URL: https://www.nvidia.com/content/dam/en-zz/Solutions/self-driving-cars/safety-force-field/an-introduction-to-the-safety-force-field-v2.pdf.

[521] Guanglin Niu et al. "Rule-Guided Compositional Representation Learning on Knowledge Graphs". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 2950–2958.

[522] Jeremy Nixon et al. "Measuring Calibration in Deep Learning". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 38–41.

[523] Partha Niyogi, Federico Girosi, and Tomaso Poggio. "Incorporating prior information in machine learning by creating virtual examples". In: *Proceedings of the IEEE 86.11* (1998), pp. 2196–2209.

[524] *NLLP*. URL: https://nllpw.org/about/ (visited on 07/02/2021).

[525] Natalya F Noy, Deborah L McGuinness, et al. *Ontology development 101: A guide to creating your first ontology*. 2001.

[526] Michael Oberst and David Sontag. "Counterfactual Off-Policy Evaluation with Gumbel-Max Structural Causal Models". en. In: *arXiv:1905.05824 [cs, stat]* (June 2019). arXiv: 1905.05824, p. 10.

[527] Ozan Oktay et al. "Anatomically constrained neural networks (ACNNs): application to cardiac image enhancement and segmentation". In: *IEEE transactions on medical imaging 37.2* (2017). Publisher: IEEE, pp. 384–395.

[528] Alessandro Oltramari et al. "Neuro-symbolic Architectures for Context Understanding". In: *arXiv:2003.04707 [cs]* (Mar. 2020). arXiv: 2003.04707. (Visited on 06/25/2021).

[529] Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. "Scalable Rule Learning via Learning Representation". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI'18. event-place: Stockholm, Sweden. AAAI Press, 2018, pp. 2149–2155.

[530] OpenAI et al. "Dota 2 with Large Scale Deep Reinforcement Learning". In: *arXiv:1912.06680 [cs, stat]* (Dec. 2019). arXiv: 1912.06680. (Visited on 07/09/2021).

[531] Manfred Opper and David Saad, eds. *Advanced mean field methods: theory and practice*. Neural information processing series. Cambridge, Mass: MIT Press, 2001.

[532] Maxime Oquab et al. "Learning and transferring mid-level image representations using convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1717–1724.

[533] Yaniv Ovadia et al. "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift". In: *Proceedings of the 33rd international conference on neural information processing systems*. Number of pages: 12 tex.articleno: 1254. Red Hook, NY, USA: Curran Associates Inc., 2019.

[534] *OWL - Semantic Web Standards*. URL: https://www.w3.org/OWL/ (visited on 06/30/2021).

[535] Brian Paden et al. "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles". en. In: *arXiv:1604.07446 [cs]* (Apr. 2016). arXiv: 1604.07446. (Visited on 11/29/2020).

[536] Jiyo Palatti et al. "Planning for Safe Abortable Overtaking Maneuvers in Autonomous Driving". In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 508–514.

[537] Monica Palmirani et al. "LegalRuleML: XML-Based Rules and Norms". en. In: *Rule-Based Modeling and Computing on the Semantic Web*. Ed. by Frank Olken,

Monica Palmirani, and Davide Sottara. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 298–312.

[538] Monica Palmirani et al. "PrOnto: Privacy Ontology for Legal Reasoning". In: *Electronic Government and the Information Systems Perspective*. Ed. by Andrea Kő and Enrico Francesconi. Vol. 11032. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 139–152. (Visited on 06/25/2021).

[539] Pingbo Pan et al. "Continual Deep Learning by Functional Regularisation of Memorable Past". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*. Ed. by Hugo Larochelle et al. 2020.

[540] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009). Publisher: IEEE, pp. 1345–1359.

[541] Yanwei Pang et al. "Mask-Guided attention network for occluded pedestrian detection". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019, pp. 4967–4975.

[542] Dim P Papadopoulos et al. "Training object class detectors with click supervision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6374–6383.

[543] Emilio Parisotto et al. "Neuro-Symbolic Program Synthesis". In: *arXiv:1611.01855 [cs]* (Nov. 2016). arXiv: 1611.01855. (Visited on 06/20/2021).

[544] Mee Young Park and Trevor Hastie. "L_1-regularization path algorithm for generalized linear models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.4 (2007). Publisher: Wiley Online Library, pp. 659–677.

[545] Razvan Pascanu and Yoshua Bengio. "Revisiting Natural Gradient for Deep Networks". In: *Proceedings of the 2nd International Conference on Learning Representations*. Ed. by Yoshua Bengio and Yann LeCun. 2014.

[546] Gaurav Patel and Jose Dolz. "Weakly supervised segmentation with cross-modality equivariant constraints". In: *Medical Image Analysis* 77 (2022), p. 102374.

[547] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. "Constrained convolutional neural networks for weakly supervised segmentation". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1796–1804.

[548] Sujoy Paul, Jawadul H Bappy, and Amit K Roy-Chowdhury. "Non-uniform subset selection for active learning in structured data". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6846–6855.

[549] Patricia Pauli et al. "Neural network training under semidefinite constraints". In: *arXiv preprint arXiv:2201.00632* (2022).

[550] Patricia Pauli et al. "Training Robust Neural Networks Using Lipschitz Bounds". In: *IEEE Control Systems Letters* 6 (2022), pp. 121–126.

[551] Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. "Deep Structural Causal Models for Tractable Counterfactual Inference". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 857–869.

[552] Ali Payani and Faramarz Fekri. "Inductive Logic Programming via Differentiable Deep Neural Logic Networks". In: *arXiv:1906.03523 [cs]* (June 2019). arXiv: 1906.03523. (Visited on 06/25/2021).

[553] Judea Pearl. *Causality: models, reasoning, and inference*. Cambridge, U.K. ; New York: Cambridge University Press, 2000.

[554] Judea Pearl. *Causality: models, reasoning, and inference*. en. 2. ed., repr. with corr. Cambridge: Cambridge University Press, 2013.

[555] Judea Pearl. "The seven tools of causal inference, with reflections on machine learning". en. In: *Communications of the ACM* 62.3 (Feb. 2019). Number: 3, pp. 54–60. (Visited on 09/05/2019).

[556] Judea Pearl. "Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution". In: *arXiv:1801.04016 [cs, stat]* (Jan. 2018). arXiv: 1801.04016. (Visited on 06/09/2021).

[557] Fengchao Peng et al. "Active Learning for Lane Detection: A Knowledge Distillation Approach". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15152–15161.

[558] Jizong Peng et al. "Discretely-constrained deep network for weakly supervised segmentation". In: *Neural Networks* 130 (2020). Publisher: Elsevier, pp. 297–308.

[559] David Perdomo Lopez et al. "Scenario Interpretation based on Primary Situations for Automatic Turning at Urban Intersections:" en. In: *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems*. Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2017, pp. 15–23. (Visited on 07/06/2021).

[560] Valentin Peretroukhin et al. "A smooth representation of belief over so (3) for deep rotation learning with uncertainty". In: *arXiv preprint arXiv:2006.01031* (2020).

[561] Juan-Manuel Perez-Rua et al. "Incremental Few-Shot Object Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[562] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "DeepWalk: Online Learning of Social Representations". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. event-place: New York, New York, USA. New York, NY, USA: Association for Computing Machinery, 2014, pp. 701–710.

[563] Matthew E. Peters et al. "Deep contextualized word representations". In: *CoRR* abs/1802.05365 (2018).

[564] Vitali Petsiuk, Abir Das, and Kate Saenko. "Rise: Randomized input sampling for explanation of black-box models". In: *arXiv preprint arXiv:1806.07421* (2018).

[565] Vitali Petsiuk et al. "Black-box explanation of object detectors via saliency maps". In: *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 11443–11452.

[566] Tung Phan-Minh et al. "CoverNet: Multimodal Behavior Prediction Using Trajectory Sets". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[567] Dean A. Pomerleau. "Efficient Training of Artificial Neural Networks for Autonomous Navigation". en. In: *Neural Computation* 3.1 (Feb. 1991), pp. 88–97. (Visited on 06/20/2021).

[568] Janis Postels et al. "Sampling-free epistemic uncertainty estimation using approximated variance propagation". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 2931–2940.

[569] Prakash Poudyal et al. "Echr: legal corpus for argument mining". In: *Proceedings of the 7th Workshop on Argument Mining*. 2020, pp. 67–75.

[570] Ameya Prabhu, Charles Dognin, and Maneesh Singh. "Sampling bias in deep active classification: An empirical study". In: *arXiv preprint arXiv:1909.09389* (2019).

[571] Henry Prakken. "On the problem of making autonomous vehicles conform to traffic law". en. In: *Artificial Intelligence and Law* 25.3 (Sept. 2017), pp. 341–363. (Visited on 07/09/2021).

[572] Sergey Prokudin, Peter Gehler, and Sebastian Nowozin. "Deep directional statistics: Pose estimation with uncertainty quantification". In: *European conference on computer vision (ECCV)*. Vol. 9. tex.month_numeric: 9. Sept. 2018, pp. 542–559.

[573] Mattia Prosperi et al. "Causal inference and counterfactual prediction in machine learning for actionable healthcare". en. In: *Nature Machine Intelligence* 2.7 (July 2020), pp. 369–375. (Visited on 06/09/2021).

[574] Muni Sreenivas Pydi and Vishnu Suresh Lokhande. "Active Learning with Importance Sampling". In: *arXiv preprint arXiv:1910.04371* (2019).

[575] Jiaxin Qi et al. "Two Causal Principles for Improving Visual Dialog". In: *arXiv:1911.10496 [cs]* (Mar. 2020). arXiv: 1911.10496. (Visited on 06/18/2021).

[576] Xipeng Qiu et al. "Pre-trained Models for Natural Language Processing: A Survey". In: *CoRR* abs/2003.08271 (2020). _eprint: 2003.08271.

[577] Meng Qu et al. "RNNLogic: Learning Logic Rules for Reasoning on Knowledge Graphs". In: *arXiv:2010.04029 [cs]* (Oct. 2020). arXiv: 2010.04029. (Visited on 06/04/2021).

[578] Zhenshen Qu et al. "Deep Active Learning for Remote Sensing Object Detection". In: *arXiv preprint arXiv:2003.08793* (2020).

[579] *QuantLaw*. en. URL: https://github.com/QuantLaw (visited on 07/13/2021).

[580] Rahul Rai and Chandan K Sahu. "Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus". In: *IEEE Access* 8 (2020). Publisher: IEEE, pp. 71050–71073.

[581] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". en. In: *Journal of Computational Physics* 378 (Feb. 2019), pp. 686–707. (Visited on 01/11/2021).

[582] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. "Multistep neural networks for data-driven discovery of nonlinear dynamical systems". In: *arXiv preprint arXiv:1801.01236* (2018).

[583] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations". In: *arXiv preprint arXiv:1711.10561* (2017).

[584] Aravind Rajeswaran et al. "Meta-Learning with Implicit Gradients". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[585] Prajit Ramachandran et al. "Stand-alone self-attention in vision models". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 32. Curran Associates, Inc., 2019.

[586] Chengping Rao, Hao Sun, and Yang Liu. "Physics-informed deep learning for computational elastodynamics without labeled data". In: *Journal of Engineering Mechanics* 147.8 (2021). Publisher: American Society of Civil Engineers, p. 04021043.

[587] Chengping Rao, Hao Sun, and Yang Liu. "Physics-informed deep learning for incompressible laminar flows". In: *Theoretical and Applied Mechanics Letters* 10.3 (2020). Publisher: Elsevier, pp. 207–212.

[588] Sathya N Ravi et al. "Explicitly imposing constraints in deep networks via conditional gradients gives improved generalization and faster convergence". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. Issue: 01. 2019, pp. 4772–4779.

[589] J. Redmon and A. Farhadi. "YOLO9000: Better, Faster, Stronger". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6517–6525.

[590] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[591] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[592] Russell Reed, Seho Oh, RJ Marks, et al. "Regularization using jittered training data". In: *International joint conference on neural networks*. Vol. 3. 1992, pp. 147–152.

[593] D. Reid. "An algorithm for tracking multiple targets". en. In: *IEEE Transactions on Automatic Control* 24.6 (Dec. 1979), pp. 843–854. (Visited on 06/28/2021).

[594] Hongyu Ren, Weihua Hu, and Jure Leskovec. "Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings". In: *International Conference on Learning Representations*. 2020.

[595] Pengzhen Ren et al. "A Survey of Deep Active Learning". In: *ACM Comput. Surv.* 54.9 (Oct. 2021). Place: New York, NY, USA Publisher: Association for Computing Machinery.

[596] Shaoqing Ren et al. "Faster R-CNN: towards real-time object detection with region proposal networks". In: *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2016). Publisher: IEEE, pp. 1137–1149.

[597] Danilo Rezende and Shakir Mohamed. "Variational inference with normalizing flows". en. In: *Proceedings of the 32nd international conference on machine learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of machine learning research. Lille, France: PMLR, July 2015, pp. 1530–1538.

[598] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research. Issue: 2. Bejing, China: PMLR, June 2014, pp. 1278–1286.

[599] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016).

[600] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Anchors: High-precision model-agnostic explanations". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. Issue: 1. 2018.

[601] Jonathan G. Richens, Ciarán M. Lee, and Saurabh Johri. "Improving the accuracy of medical diagnosis with causal machine learning". en. In: *Nature Communications* 11.1 (Aug. 2020). Number: 1 Publisher: Nature Publishing Group, p. 3923. (Visited on 06/09/2021).

[602] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. "Playing for benchmarks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2213–2222.

[603] Ryan Riegel et al. "Logical Neural Networks". In: *arXiv:2006.13155 [cs]* (June 2020). arXiv: 2006.13155. (Visited on 06/30/2021).

[604] Matthew Riemer et al. "Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference". In: *arXiv:1810.11910 [cs, stat]* (May 2019). arXiv: 1810.11910. (Visited on 07/08/2021).

[605] Ralf Risser. "Behavior in traffic conflict situations". en. In: *Accident Analysis & Prevention* 17.2 (Apr. 1985), pp. 179–197. (Visited on 06/28/2021).

[606] Hippolyt Ritter, Aleksandar Botev, and David Barber. "A Scalable Laplace Approximation for Neural Networks". In: *International Conference on Learning Representations*. 2018.

[607] Hippolyt Ritter, Aleksandar Botev, and David Barber. "Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems*. Ed. by Samy Bengio et al. 2018, pp. 3742–3752.

[608] Albert Rizaldi and Matthias Althoff. "Formalising Traffic Rules for Accountability of Autonomous Vehicles". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. ISSN: 2153-0017. Sept. 2015, pp. 1658–1665.

[609] Livio Robaldo and Xin Sun. "Reified Input/Output logic: Combining Input/Output logic and Reification to represent norms coming from existing legislation". In: *Journal of Logic and Computation* 27.8 (Dec. 2017), pp. 2471–2503. (Visited on 03/19/2021).

[610] Livio Robaldo et al. "Formalizing GDPR Provisions in Reified I/O Logic: The DAPRECO Knowledge Base". en. In: *Journal of Logic, Language and Information* 29.4 (Dec. 2020), pp. 401–449. (Visited on 03/19/2021).

[611] Livio Robaldo et al. "Introduction for artificial intelligence and law: special issue "natural language processing for legal texts"". en. In: *Artificial Intelligence and Law* 27.2 (June 2019), pp. 113–115. (Visited on 07/10/2021).

[612] Hendrik Roehm et al. "STL Model Checking of Continuous and Hybrid Systems". en. In: *Automated Technology for Verification and Analysis*. Ed. by Cyrille Artho, Axel Legay, and Doron Peled. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 412–427.

[613] German Ros et al. "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3234–3243.

[614] Ariel Rosenfeld et al. "Leveraging human knowledge in tabular reinforcement learning: A study of human subjects". In: *arXiv:1805.05769 [cs]* (May 2018). arXiv: 1805.05769. (Visited on 07/07/2021).

[615] Francisca Rosique et al. "A systematic review of perception system and simulators for autonomous vehicles research". In: *Sensors* 19.3 (2019), p. 648.

[616] Grant M Rotskoff and Eric Vanden-Eijnden. "Learning with rare data: Using active importance sampling to optimize objectives dominated by rare events". In: *arXiv preprint arXiv:2008.06334* (2020).

[617] Matthias Rottmann, Karsten Kahl, and Hanno Gottschalk. "Deep bayesian active semi-supervised learning". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 158–164.

[618] Nicholas Roy and Andrew McCallum. "Toward optimal active learning through monte carlo estimation of error reduction". In: *ICML, Williamstown* (2001), pp. 441–448.

[619] Donald B Rubin. "Causal Inference Using Potential Outcomes: Design, Modeling, Decisions". en. In: *Journal of the American Statistical Association* 100.469 (Mar. 2005), pp. 322–331. (Visited on 06/10/2021).

[620] Donald B. Rubin. "Estimating causal effects of treatments in randomized and nonrandomized studies". In: *Journal of Educational Psychology* 66.5 (1974). Place: US Publisher: American Psychological Association, pp. 688–701.

[621] Laura von Rueden et al. "Combining machine learning and simulation to a hybrid modelling approach: Current and future directions". In: *International symposium on intelligent data analysis*. 2020, pp. 548–560.

[622] Laura von Rueden et al. "Informed Machine Learning – A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems". In: *IEEE Transactions on Knowledge and Data Engineering* (2021). arXiv: 1903.12394, pp. 1–1. (Visited on 06/29/2021).

[623] Laura von Rueden et al. "Informed Machine Learning–A Taxonomy and Survey of Integrating Knowledge into Learning Systems". In: *arXiv preprint arXiv:1903.12394* (2019).

[624] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". en. In: *International Journal of Computer Vision* 115.3 (Dec. 2015), pp. 211–252. (Visited on 07/21/2021).

[625] D. Sacha et al. "VIS4ML: An Ontology for Visual Analytics Assisted Machine Learning". In: *IEEE Transactions on Visualization and Computer Graphics* 25 (2019), pp. 385–395.

[626] Abbas Sadat et al. "Jointly learnable behavior and trajectory planning for self-driving vehicles". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3949–3956.

[627] Abbas Sadat et al. "Perceive, Predict, and Plan: Safe Motion Planning Through Interpretable Semantic Representations". en. In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Vol. 12368. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 414–430. (Visited on 01/29/2021).

[628] SAE. *J3016C: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International*. 2021. URL: https://www.sae.org/standards/content/j3016_202104/ (visited on 05/10/2021).

[629] Gobinda Saha, Isha Garg, and Kaushik Roy. "Gradient Projection Memory for Continual Learning". In: *arXiv:2103.09762 [cs]* (Mar. 2021). arXiv: 2103.09762. (Visited on 05/19/2021).

[630] Olimpiya Saha, Prithviraj Dasgupta, and Bradley Woosley. "Real-time robot path planning from simple to complex obstacle patterns via transfer learning of options". In: *Autonomous Robots* 43 (2019). Publisher: Springer US, pp. 2071–2093.

[631] Francisco Sahli Costabal et al. "Physics-informed neural networks for cardiac activation mapping". In: *Frontiers in Physics* 8 (2020). Publisher: Frontiers, p. 42.

[632] Tim Salzmann et al. "Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVIII*. Ed. by Andrea Vedaldi et al. Vol. 12363. Lecture Notes in Computer Science. Springer, 2020, pp. 683–700.

[633] Adam Santoro et al. "A simple neural network module for relational reasoning". In: *arXiv:1706.01427 [cs]* (June 2017). arXiv: 1706.01427. (Visited on 06/30/2021).

[634] M. Sato and H. Tsukimoto. "Rule extraction from neural networks via decision tree induction". en. In: *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*. Vol. 3. Washington, DC, USA: IEEE, 2001, pp. 1870–1875. (Visited on 06/22/2021).

[635] Stefan Schaal. "Learning from Demonstration". In: *Advances in Neural Information Processing Systems 9*. MIT Press, 1997, pp. 12–20.

[636] Tom Schaul et al. "Prioritized Experience Replay". In: *International Conference on Learning Representations*. Puerto Rico, 2016.

[637] Michael Schlichtkrull et al. "Modeling Relational Data with Graph Convolutional Networks". In: *The Semantic Web*. Ed. by Aldo Gangemi et al. Cham: Springer International Publishing, 2018, pp. 593–607.

[638] G.P.J. Schmitz, C. Aldrich, and F.S. Gouws. "ANN-DT: an algorithm for extraction of decision trees from artificial neural networks". In: *IEEE Transactions on Neural Networks* 10.6 (Nov. 1999), pp. 1392–1401. (Visited on 06/22/2021).

[639] Bernhard Schölkopf. "Causality for Machine Learning". In: *arXiv:1911.10500 [cs, stat]* (Dec. 2019). arXiv: 1911.10500. (Visited on 08/12/2020).

[640] Bernhard Schölkopf et al. "Toward causal representation learning". In: *Proceedings of the IEEE* 109.5 (2021), pp. 612–634.

[641] Matthias Schraps and Bosler Alexander. "Knowledge Extraction from German Automotive Software Requirements using NLP-Techniques and a Grammar-based Pattern Detection". In: *Conference: The Eighth International Conference on Pervasive Patterns and Applications*. 2016.

[642] Matthias Schreier, Volker Willert, and Jurgen Adamy. "An Integrated Approach to Maneuver-Based Trajectory Prediction and Criticality Assessment in Arbitrary Road Environments". In: *IEEE Transactions on Intelligent Transportation Systems* 17.10 (Oct. 2016), pp. 2751–2766. (Visited on 06/28/2021).

[643] Matthias Schreier, Volker Willert, and Jurgen Adamy. "Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Qingdao, China: IEEE, Oct. 2014, pp. 334–341. (Visited on 06/28/2021).

[644] Julian Schrittwieser et al. "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model". In: *Nature* 588.7839 (Dec. 2020). arXiv: 1911.08265, pp. 604–609. (Visited on 06/29/2021).

[645] Brigit Schroeder and Alexandre Alahi. "Using a Priori Knowledge to Improve Scene Understanding". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[646] Robin Schubert, Eric Richter, and Gerd Wanielik. "Comparison and evaluation of advanced motion models for vehicle tracking". In: *2008 11th international conference on information fusion*. IEEE, 2008, pp. 1–6.

[647] John Schulman et al. "Proximal Policy Optimization Algorithms". In: *arXiv:1707.06347 [cs]* (Aug. 2017). arXiv: 1707.06347. (Visited on 06/29/2021).

[648] John Schulman et al. "Trust Region Policy Optimization". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1889–1897.

[649] Matthias Schultheis et al. "Receding horizon curiosity". In: *Conference on Robot Learning*. PMLR, 2020, pp. 1278–1288.

[650] Gesina Schwalbe and Bettina Finzel. "XAI Method Properties: A (Meta-) study". In: *arXiv preprint arXiv:2105.07190* (2021).

[651] Jonathan Schwarz et al. "Progress & Compress: A scalable framework for continual learning". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 4535–4544.

[652] Dexter R. R. Scobee and S. Shankar Sastry. "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning". In: *arXiv:1909.05477 [cs, eess, stat]* (Apr. 2020). arXiv: 1909.05477. (Visited on 01/08/2021).

[653] Sean Segal et al. "Just Label What You Need: Fine-Grained Active Selection for P&P through Partially Labeled Scenes". In: *Conference on Robot Learning*. PMLR, 2022, pp. 816–826.

[654] Ramprasaath R. Selvaraju et al. "Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization". In: *CoRR* abs/1610.02391 (2016). _eprint: 1610.02391.

[655] Ozan Sener and Silvio Savarese. "Active learning for convolutional neural networks: A core-set approach". In: *arXiv preprint arXiv:1708.00489* (2017).

[656] Murat Sensoy, Lance Kaplan, and Melih Kandemir. "Evidential deep learning to quantify classification uncertainty". In: 2018, pp. 3179–3189.

[657] Luciano Serafini and Artur d'Avila Garcez. "Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge". In: *arXiv:1606.04422 [cs]* (July 2016). arXiv: 1606.04422. (Visited on 06/30/2021).

[658] Kamal Kumar Sethi, Durgesh Kumar Mishra, and Bharat Mishra. "Extended taxonomy of rule extraction techniques and assessment of kdruleex". In: *International Journal of Computer Applications* 50.21 (2012). Publisher: Citeseer.

[659] Kamal Kumar Sethi, Durgesh Kumar Mishra, and Bharat Mishra. "KDRuleEx: A Novel Approach for Enhancing User Comprehensibility Using Rule Extraction". en. In: *2012 Third International Conference on Intelligent Systems Modelling and Simulation*. Kota Kinabalu, Malaysia: IEEE, Feb. 2012, pp. 55–60. (Visited on 06/22/2021).

[660] Rudy Setiono and Wee Kheng Leow. "FERNN: An algorithm for fast extraction of rules from neural networks". In: *Applied Intelligence* 12.1 (2000). Publisher: Springer, pp. 15–25.

[661] Burr Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison, 2009.

[662] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. "Query by committee". In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 287–294.

[663] Cosma Rohilla Shalizi and James P. Crutchfield. "Computational Mechanics: Pattern and Prediction, Structure and Simplicity". In: *Journal of Statistical Physics* 104.3/4 (2001), pp. 817–879. (Visited on 06/25/2021).

[664] Chao Shang et al. "End-to-end Structure-Aware Convolutional Networks for Knowledge Base Completion". In: *CoRR* abs/1811.04441 (2018). _eprint: 1811.04441.

[665] Ramalingam Shanmugam. "Elements of causal inference: foundations and learning algorithms". en. In: *Journal of Statistical Computation and Simulation* 88.16 (Nov. 2018). Number: 16, pp. 3248–3248. (Visited on 07/15/2020).

[666] Shuai Shao et al. "Crowdhuman: A benchmark for detecting human in a crowd". In: *arXiv preprint arXiv:1805.00123* (2018).

[667] Siyu Shao et al. "Highly accurate machine fault diagnosis using deep transfer learning". In: *IEEE Transactions on Industrial Informatics* 15.4 (2018). Publisher: IEEE, pp. 2446–2455.

[668] Lloyd S Shapley. "A value for n-person games". In: *Contributions to the Theory of Games* 2.28 (1953), pp. 307–317.

[669] Pengpeng Shi, Zhi Zeng, and Tiangshou Liang. "Physics-informed ConvNet: Learning Physical Field from a Shallow Neural Network". In: *arXiv preprint arXiv:2201.10967* (2022).

[670] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. "Pointrcnn: 3d object proposal generation and detection from point cloud". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 770–779.

[671] Hikaru Shindo, Masaaki Nishino, and Akihiro Yamamoto. "Differentiable Inductive Logic Programming for Structured Examples". In: *arXiv:2103.01719 [cs]* (Mar. 2021). arXiv: 2103.01719. (Visited on 06/25/2021).

[672] B. Shneiderman. "The eyes have it: a task by data type taxonomy for information visualizations". In: *Proceedings 1996 IEEE Symposium on Visual Languages*. 1996, pp. 336–343.

[673] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of Big Data* 6.1 (2019), pp. 1–48.

[674] Ilya Shpitser and Judea Pearl. "Complete Identification Methods for the Causal Hierarchy". en. In: *Journal of Machine Learning Research* 9.64 (2008), pp. 1941–1979.

[675] Ilya Shpitser and Judea Pearl. "Identification of Conditional Interventional Distributions". en. In: *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*. UAI'06. Arlington, Virginia, USA: AUAI Press, July 2006, pp. 437–444.

[676] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning important features through propagating activation differences". In: *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.

[677] Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. "Model-based active exploration". In: *International Conference on Machine Learning*. PMLR, 2019, pp. 5779–5788.

[678] Joachim Sicking et al. "A Novel Regression Loss for Non-Parametric Uncertainty Optimization". In: *Third Symposium on Advances in Approximate Bayesian Inference* (2021).

[679] Joachim Sicking et al. "Characteristics of Monte Carlo Dropout in Wide Neural Networks". In: *ICML UDL Workshop* (2020). _eprint: arXiv:2007.05434.

[680] Yawar Siddiqui, Julien Valentin, and Matthias Nießner. "Viewal: Active learning with viewpoint entropy for semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9433–9443.

[681] David Silver et al. "Deterministic Policy Gradient Algorithms". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research. Issue: 1. Bejing, China: PMLR, June 2014, pp. 387–395.

[682] David Silver et al. "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". In: *arXiv:1712.01815 [cs]* (Dec. 2017). arXiv: 1712.01815. (Visited on 06/29/2021).

[683] David Silver et al. "Mastering the Game of Go with Deep Neural Networks and Tree Search". In: *Nature* 529.7587 (Jan. 2016). Publisher: Nature Publishing Group, pp. 484–489.

[684] Oriane Siméoni et al. "Rethinking deep active learning: Using unlabeled data at model training". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 1220–1227.

[685] Noah Simon et al. "A sparse-group lasso". In: *Journal of Computational and Graphical Statistics* 22.2 (2013). Publisher: Taylor & Francis, pp. 231–245.

[686] K. Simonyan, A. Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *CoRR* abs/1312.6034 (2014).

[687] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". en. In: *arXiv:1409.1556 [cs]* (Apr. 2015). arXiv: 1409.1556. (Visited on 07/12/2021).

[688] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. "Variational adversarial active learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5972–5981.

[689] Amin Sleimi et al. "A Query System for Extracting Requirements-Related Information from Legal Texts". In: *2019 IEEE 27th International Requirements Engineering Conference (RE)*. Jeju Island, Korea (South): IEEE, Sept. 2019, pp. 319–329. (Visited on 06/28/2021).

[690] Daniel Smilkov et al. "Smoothgrad: removing noise by adding noise". In: *arXiv preprint arXiv:1706.03825* (2017).

[691] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical Networks for Few-shot Learning". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[692] Richard Socher et al. "Reasoning with Neural Tensor Networks for Knowledge Base Completion". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'13. event-place: Lake Tahoe, Nevada. Red Hook, NY, USA: Curran Associates Inc., 2013, pp. 926–934.

[693] Petru Soviany and Radu Tudor Ionescu. "Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction". In: *20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2018, Timisoara, Romania, September 20-23, 2018*. IEEE, 2018, pp. 209–214.

[694] Jost Tobias Springenberg et al. "Striving for simplicity: The all convolutional net". In: *arXiv preprint arXiv:1412.6806* (2014).

[695] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[696] Russell Stewart and Stefano Ermon. "Label-free supervision of neural networks with physics and domain knowledge". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[697] Shane Storks, Qiaozi Gao, and J. Chai. "Commonsense Reasoning for Natural Language Understanding: A Survey of Benchmarks, Resources, and Approaches". In: *ArXiv abs/1904.01172* (2019).

[698] Linda Studer et al. "A comprehensive study of imagenet pre-training for historical document image analysis". In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR* (2019), pp. 720–725.

[699] Rudi Studer, V.Richard Benjamins, and Dieter Fensel. "Knowledge engineering: Principles and methods". en. In: *Data & Knowledge Engineering* 25.1-2 (Mar. 1998), pp. 161–197. (Visited on 06/29/2021).

[700] Jiankai Sun et al. *Neuro-Symbolic Program Search for Autonomous Driving Decision Module Design*. en. 2020.

[701] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[702] Simon Suo et al. "TrafficSim: Learning to simulate realistic multi-agent behaviors". In: *arXiv preprint arXiv:2101.06557* (2021).

[703] Richard S Sutton et al. "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press, 2000.

[704] Richard S. Sutton. "Dyna, an Integrated Architecture for Learning, Planning, and Reacting". In: *SIGART Bull.* 2.4 (July 1991). Place: New York, NY, USA Publisher: Association for Computing Machinery, pp. 160–163.

[705] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An Introduction (second edition)*. en. Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 2018.

[706] Siddharth Swaroop et al. "Improving and Understanding Variational Continual Learning". In: *CoRR* abs/1905.02099 (2019). _eprint: 1905.02099.

[707] Christian Szegedy et al. "Going Deeper with Convolutions". In: *Computer Vision and Pattern Recognition (CVPR)*. 2015.

[708] I.A. Taha and J. Ghosh. "Symbolic interpretation of artificial neural networks". en. In: *IEEE Transactions*

on Knowledge and Data Engineering 11.3 (June 1999), pp. 448–463. (Visited on 06/22/2021).

[709] Ardi Tampuu et al. "A Survey of End-to-End Driving: Architectures and Training Methods". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020). arXiv: 2003.06404, pp. 1–21. (Visited on 06/20/2021).

[710] Chuanqi Tan et al. "A survey on deep transfer learning". In: *International conference on artificial neural networks*. Springer, 2018, pp. 270–279.

[711] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[712] Mingxing Tan, Ruoming Pang, and Quoc V Le. "Efficientdet: Scalable and efficient object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10781–10790.

[713] Jian Tang et al. "LINE: Large-scale Information Network Embedding". In: *Proceedings of the 24th International Conference on World Wide Web*. WWW '15. event-place: Florence, Italy. New York, NY, USA: ACM, 2015, pp. 1067–1077.

[714] Vladimir Tchuiev and Vadim Indelman. "Inference over distribution of posterior class probabilities for reliable bayesian classification and object-level perception". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4329–4336.

[715] Gerald Tesauro. "Temporal Difference Learning and TD-Gammon". In: *Commun. ACM* 38.3 (Mar. 1995). Place: New York, NY, USA Publisher: Association for Computing Machinery, pp. 58–68.

[716] Sebastian Thrun. "Extracting rules from artificial neural networks with distributed representations". In: *Advances in neural information processing systems* 7 (1994).

[717] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. Intelligent robotics and autonomous agents. OCLC: ocm58451645. Cambridge, Mass: MIT Press, 2005.

[718] Sebastian Thrun and Lorien Pratt. "Learning to learn: Introduction and overview". In: *Learning to learn*. Springer, 1998, pp. 3–17.

[719] Sebastian B. Thrun. *Extracting Provably Correct Rules from Artificial Neural Networks*. Tech. rep. University of Bonn, 1993.

[720] Zhi Tian et al. "FCOS: Fully Convolutional One-Stage Object Detection". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9626–9635.

[721] Robert Tibshirani. "Regression Shrinkage and Selection Via the Lasso". In: *Journal of the Royal Statistical Society, Series B* 58 (1994), pp. 267–288.

[722] Michalis K. Titsias et al. "Functional Regularisation for Continual Learning with Gaussian Processes". In: *Proceedings of the 8th International Conference on Learning Representations*. OpenReview.net, 2020.

[723] Josh Tobin et al. "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. 2017, pp. 23–30.

[724] Simon Tong and Daphne Koller. "Support vector machine active learning with applications to text classification". In: *Journal of machine learning research* 2.Nov (2001), pp. 45–66.

[725] Geoffrey G. Towell and Jude W. Shavlik. "Extracting refined rules from knowledge-based neural networks". en. In: *Machine Learning* 13.1 (1993), pp. 71–101. (Visited on 06/22/2021).

[726] Dustin Tran, Rajesh Ranganath, and David M Blei. "The variational Gaussian process". In: *arXiv preprint arXiv:1511.06499* (2015).

[727] Toan Tran et al. "Bayesian generative active deep learning". In: *International Conference on Machine Learning*. PMLR, 2019, pp. 6295–6304.

[728] Théo Trouillon et al. "Complex Embeddings for Simple Link Prediction". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. event-place: New York, NY, USA. JMLR.org, 2016, pp. 2071–2080.

[729] J.N. Tsitsiklis and B. Van Roy. "An analysis of temporal-difference learning with function approximation". In: *IEEE Transactions on Automatic Control* 42.5 (1997), pp. 674–690.

[730] H. Tsukimoto. "Extracting rules from trained neural networks". en. In: *IEEE Transactions on Neural Networks* 11.2 (2000), pp. 377–389. (Visited on 06/22/2021).

[731] Hideomi Tsunakawa et al. "Contrastive Relevance Propagation for Interpreting Predictions by a Single-Shot Object Detector". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019, pp. 1–9.

[732] Simon Ulbrich et al. "Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. Gran Canaria, Spain: IEEE, Sept. 2015, pp. 982–988. (Visited on 07/01/2021).

[733] Stefanie Urchs, Jelena Mitrović, and Michael Granitzer. "Design and Implementation of German Legal Decision Corpora". In: *13th International Conference on Agents and Artificial Intelligence*. July 2021, pp. 515–521. (Visited on 07/06/2021).

[734] Mevlut Uzun, Mustafa Umut Demirezen, and Gokhan Inalhan. "Physics Guided Deep Learning for Data-Driven Aircraft Fuel Consumption Modeling". In: *Aerospace* 8.2 (2021). Publisher: Multidisciplinary Digital Publishing Institute, p. 44.

[735] Sara A Van de Geer. "High-dimensional generalized linear models and the lasso". In: *The Annals of Statistics* (2008). Publisher: JSTOR, pp. 614–645.

[736] V. Vapnik. *Statistical learning theory*. Vol. 1. Wiley New York, 1998.

[737] Balakrishnan Varadarajan et al. "MultiPath++: Efficient Information Fusion and Trajectory Aggregation for Behavior Prediction". In: *CoRR* abs/2111.14973 (2021). arXiv: 2111.14973.

[738] Shikhar Vashishth et al. "Composition-based Multi-Relational Graph Convolutional Networks". In: *International Conference on Learning Representations*. 2020.

[739] Shikhar Vashishth et al. "InteractE: Improving Convolution-based Knowledge Graph Embeddings

by Increasing Feature Interactions". In: *CoRR* abs/1911.00219 (2019). _eprint: 1911.00219.

[740] Ashish Vaswani et al. "Attention is All You Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. event-place: Long Beach, California, USA. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[741] Marcell Vazquez-Chanlatte et al. "Learning Task Specifications from Demonstrations". In: *arXiv:1710.03875 [cs]* (Oct. 2018). arXiv: 1710.03875. (Visited on 03/03/2021).

[742] Petar Veličković et al. "Graph Attention Networks". In: *6th International Conference on Learning Representations* (2017).

[743] Vikas Verma et al. "Manifold mixup: Better representations by interpolating hidden states". In: *International conference on machine learning*. 2019, pp. 6438–6447.

[744] Alexander Vezhnevets, Joachim M Buhmann, and Vittorio Ferrari. "Active learning for semantic segmentation with expected change". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3162–3169.

[745] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". en. In: *Nature* 575.7782 (Nov. 2019). Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7782 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computer science;Statistics Subject_term_id: computer-science;statistics, pp. 350–354. (Visited on 06/29/2021).

[746] Oriol Vinyals et al. "Matching Networks for One Shot Learning". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016.

[747] Khuong Vo et al. "Combination of domain knowledge and deep learning for sentiment analysis". In: *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*. Springer, 2017, pp. 162–173.

[748] Ba-Ngu Vo, Ba-Tuong Vo, and Dinh Phung. "Labeled Random Finite Sets and the Bayes Multi-Target Tracking Filter". In: *IEEE Transactions on Signal Processing* 62.24 (Dec. 2014), pp. 6554–6567. (Visited on 06/28/2021).

[749] Carl Vondrick and Deva Ramanan. "Video annotation and tracking with active learning". In: *Advances in Neural Information Processing Systems* 24 (2011). Publisher: Citeseer, pp. 28–36.

[750] Niklas Wahlström, Thomas B. Schön, and Marc Peter Deisenroth. "From Pixels to Torques: Policy Learning with Deep Dynamical Models". In: *CoRR* abs/1502.02251 (2015). _eprint: 1502.02251.

[751] E.A. Wan and R. Van Der Merwe. "The unscented Kalman filter for nonlinear estimation". In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. Lake Louise, Alta., Canada: IEEE, 2000, pp. 153–158. (Visited on 06/25/2021).

[752] Chieh-Chih Wang et al. "Simultaneous Localization, Mapping and Moving Object Tracking". en. In: *The International Journal of Robotics Research* 26.9 (Sept. 2007), pp. 889–916. (Visited on 06/24/2021).

[753] Dequan Wang et al. "Monocular plan view networks for autonomous driving". In: *arXiv preprint arXiv:1905.06937* (2019).

[754] Hanxiao Wang et al. "Human-in-the-loop person re-identification". In: *European conference on computer vision*. Springer, 2016, pp. 405–422.

[755] Haofan Wang et al. "Score-CAM: Score-weighted visual explanations for convolutional neural networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 24–25.

[756] Huijuan Wang, Shuangyin Li, and Rong Pan. "An Adversarial Transfer Network for Knowledge Representation Learning". In: *Proceedings of the Web Conference 2021*. WWW '21. event-place: Ljubljana, Slovenia. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1749–1760.

[757] Juan Wang and Bin Xia. "Bounding Box Tightness Prior for Weakly Supervised Image Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 526–536.

[758] Keze Wang et al. "Cost-effective active learning for deep image classification". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2016). Publisher: IEEE, pp. 2591–2600.

[759] Mei Wang and Weihong Deng. "Deep visual domain adaptation: A survey". In: *Neurocomputing* 312 (2018), pp. 135–153.

[760] Min Wang et al. "Cost-sensitive active learning through statistical methods". In: *Information Sciences* 501 (2019). Publisher: Elsevier, pp. 460–482.

[761] Peifeng Wang, Shuangyin Li, and Rong Pan. "Incorporating GAN for Negative Sampling in Knowledge Representation Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018).

[762] Pengwei Wang et al. "Logic Rules Powered Knowledge Graph Embedding". In: *CoRR* abs/1903.03772 (2019). _eprint: 1903.03772.

[763] Ping Wang et al. "Context-aware virtual adversarial training for anatomically-plausible segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 304–314.

[764] Shipeng Wang et al. "Training Networks in Null Space of Feature Covariance for Continual Learning". In: *arXiv:2103.07113 [cs]* (Mar. 2021). arXiv: 2103.07113. (Visited on 05/19/2021).

[765] Sifan Wang, Yujun Teng, and Paris Perdikaris. "Understanding and mitigating gradient flow pathologies in physics-informed neural networks". In: *SIAM Journal on Scientific Computing* 43.5 (2021). Publisher: SIAM, A3055–A3081.

[766] Sifan Wang, Xinling Yu, and Paris Perdikaris. "When and why PINNs fail to train: A neural tangent kernel perspective". In: *Journal of Computational Physics* 449 (2022). Publisher: Elsevier, p. 110768.

[767] Tan Wang et al. "Visual Commonsense R-CNN". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Apr. 2020, pp. 10760–10770. (Visited on 06/02/2020).

[768] Xiaolong Wang et al. "Non-local neural networks". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 7794–7803.

[769] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. "Meta-Learning to Detect Rare Objects". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[770] Xueping Wang et al. "Learning Person Re-Identification Models From Videos With Weak Supervision". In: *IEEE Transactions on Image Processing* 30 (2021). Publisher: IEEE, pp. 3017–3028.

[771] Xuezhi Wang, Tzu-Kuo Huang, and Jeff Schneider. "Active transfer learning under model shift". In: *International Conference on Machine Learning*. PMLR, 2014, pp. 1305–1313.

[772] Zengmao Wang et al. "Multi-class active learning by integrating uncertainty and diversity". In: *IEEE Access* 6 (2018). Publisher: IEEE, pp. 22794–22803.

[773] Zhen Wang et al. "Knowledge Graph Embedding by Translating on Hyperplanes". In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI'14. event-place: Québec City, Québec, Canada. AAAI Press, 2014, pp. 1112–1119.

[774] Zihao Wang et al. "Tackling Long-Tailed Relations and Uncommon Entities in Knowledge Graph Completion". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 250–260.

[775] Zijie J Wang et al. "CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* (2020). Publisher: IEEE.

[776] Ziyu Wang et al. "Dueling Network Architectures for Deep Reinforcement Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1995–2003.

[777] Anne S Wannenwetsch, Margret Keuper, and Stefan Roth. "Probflow: Joint optical flow and uncertainty estimation". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1173–1182.

[778] Christopher J. C. H. Watkins and Peter Dayan. "Q-learning". In: *Machine Learning*. 1992, pp. 279–292.

[779] Manuel Watter et al. "Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images". In: *NIPS*. 2015, pp. 2746–2754.

[780] Waymore. *Off road, but not offline: How simulation helps advance our Waymo Driver*. Apr. 2020. URL: https://blog.waymo.com/2020/04/off-road-but-not-offline%E2%80%93simulation27.html (visited on 06/15/2021).

[781] Casey Weaver. *Self-Driving Cars Learn to Read the Body Language of People on the Street - IEEE Spectrum*. en. 2020. URL: https://spectrum.ieee.org/transportation/self-driving/selfdriving-cars-learn-to-read-the-body-language-of-people-on-the-street (visited on 07/06/2021).

[782] Bob Wei et al. "Perceive, Attend, and Drive: Learning Spatial Attention for Safe Self-Driving". In: *arXiv:2011.01153 [cs]* (Nov. 2020). arXiv: 2011.01153. (Visited on 02/07/2021).

[783] Jason Wei and Kai Zou. "Eda: Easy data augmentation techniques for boosting performance on text classification tasks". In: *arXiv preprint arXiv:1901.11196* (2019).

[784] Yeming Wen, Dustin Tran, and Jimmy Ba. "BatchEnsemble: an Alternative Approach to Efficient Ensemble and Lifelong Learning". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[785] Yeming Wen et al. "Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[786] Florian Wenzel et al. "Hyperparameter Ensembles for Robustness and Uncertainty Quantification". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020.

[787] Ruwan Wickramarachchi, Cory A. Henson, and Amit P. Sheth. "An Evaluation of Knowledge Graph Embeddings for Autonomous Driving Data: Experience and Practice". In: *CoRR abs/2003.00344* (2020). _eprint: 2003.00344.

[788] Jared Willard et al. "Integrating physics-based modeling with machine learning: A survey". In: *arXiv preprint arXiv:2003.04919* (2020).

[789] Ronald J. Williams. "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning". In: *Mach. Learn.* 8.3–4 (May 1992). Place: USA Publisher: Kluwer Academic Publishers, pp. 229–256.

[790] Christian Wissing et al. "Interaction-Aware Long-term Driving Situation Prediction". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI: IEEE, Nov. 2018, pp. 137–143. (Visited on 06/28/2021).

[791] Daniela M Witten and Robert Tibshirani. "A framework for feature selection in clustering". In: *Journal of the American Statistical Association* 105.490 (2010). Publisher: Taylor & Francis, pp. 713–726.

[792] Sebastien C Wong et al. "Understanding data augmentation for classification: when to warp?" In: *2016 international conference on digital image computing: techniques and applications (DICTA)*. 2016, pp. 1–6.

[793] Mark Woodward and Chelsea Finn. "Active one-shot learning". In: *arXiv preprint arXiv:1702.06559* (2017).

[794] Mitchell Wortsman et al. "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6750–6759.

[795] Anqi Wu et al. "Deterministic Variational Inference for Robust Bayesian Neural Networks". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[796] Bichen Wu et al. "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud". In: *2018 IEEE international conference on robotics and automation (ICRA)*. 2018, pp. 1887–1893.

[797] Pengxiang Wu, Siheng Chen, and Dimitris Metaxas. "MotionNet: Joint Perception and Motion Prediction for Autonomous Driving Based on Bird's Eye View Maps". In: *arXiv:2003.06754 [cs]* (Mar. 2020). arXiv: 2003.06754 version: 1. (Visited on 07/07/2021).

[798] Qi Wu et al. "Visual question answering: A survey of methods and datasets". In: *Computer Vision and Image Understanding* 163 (2017). Publisher: Elsevier, pp. 21–40.

[799] Weibin Wu et al. "Towards global explanations of convolutional neural networks with concept attribution". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8652–8661.

[800] Yan-Xue Wu et al. "Cost-sensitive active learning with a label uniform distribution model". In: *International Journal of Approximate Reasoning* 105 (2019). Publisher: Elsevier, pp. 49–65.

[801] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. "Maximum Entropy Deep Inverse Reinforcement Learning". In: *arXiv:1507.04888 [cs]* (Mar. 2016). arXiv: 1507.04888. (Visited on 04/29/2021).

[802] Yuxuan Xia et al. "Extended Object Tracking with Automotive Radar Using Learned Structural Measurement Model". In: *2020 IEEE Radar Conference (RadarConf20)*. Florence, Italy: IEEE, Sept. 2020, pp. 1–6. (Visited on 06/24/2021).

[803] Xiao Xie, Fan Du, and Yingcai Wu. "A Visual Analytics Approach for Exploratory Causal Analysis: Exploration, Validation, and Applications". In: *IEEE Transactions on Visualization and Computer Graphics* (2020). Publisher: IEEE.

[804] Jingyi Xu et al. "A semantic loss function for deep learning with symbolic knowledge". In: *International Conference on Machine Learning*. PMLR, 2018, pp. 5502–5511.

[805] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. PMLR, 2015, pp. 2048–2057.

[806] Minjie Xu and Gary Kazantsev. "Understanding Goal-Oriented Active Learning via Influence Functions". In: *arXiv preprint arXiv:1905.13183* (2019).

[807] Peng-Fei Xu et al. "A Physics-Informed Neural Network for the Prediction of Unmanned Surface Vehicle Dynamics". In: *Journal of Marine Science and Engineering* 10.2 (2022).

[808] Xiaoran Xu et al. "Dynamically Pruned Message Passing Networks for Large-scale Knowledge Graph Reasoning". In: *International Conference on Learning Representations*. 2020.

[809] Xin Xu et al. "Person Re-Identification via Active Hard Sample Mining". In: *arXiv preprint arXiv:2004.04912* (2020).

[810] Xiaopeng Yan et al. "Meta R-CNN: Towards General Solver for Instance-Level Low-Shot Learning". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9576–9585.

[811] Yan Yan, Yuxing Mao, and Bo Li. "Second: Sparsely embedded convolutional detection". In: *Sensors* 18.10 (2018), p. 3337.

[812] Bin Yang, Ming Liang, and Raquel Urtasun. "HDNET: Exploiting HD Maps for 3D Object Detection". In: *CoRR* abs/2012.11704 (2020). arXiv: 2012.11704.

[813] Bishan Yang et al. "Embedding Entities and Relations for Learning and Inference in Knowledge Bases". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[814] Fan Yang et al. "Uncertainty-guided transformer reasoning for camouflaged object detection". In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*. Oct. 2021, pp. 4146–4155.

[815] Jianyi Yang and Shaolei Ren. "A Quantitative Perspective on Values of Domain Knowledge for Machine Learning". In: *arXiv preprint arXiv:2011.08450* (2020).

[816] Lin Yang et al. "Suggestive annotation: A deep active learning framework for biomedical image segmentation". In: *International conference on medical image computing and computer-assisted intervention*. Springer, 2017, pp. 399–407.

[817] Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. "A Survey of Deep Learning Techniques for Neural Machine Translation". In: *CoRR* abs/2002.07526 (2020). _eprint: 2002.07526.

[818] Xu Yang, Hanwang Zhang, and Jianfei Cai. "Deconfounded Image Captioning: A Causal Retrospect". In: *arXiv:2003.03923 [cs]* (Mar. 2020). arXiv: 2003.03923. (Visited on 06/18/2021).

[819] Xufeng Yang et al. "An active learning kriging model for hybrid reliability analysis with both random and interval variables". In: *Structural and Multidisciplinary Optimization* 51.5 (2015). Publisher: Springer, pp. 1003–1016.

[820] Yi Yang et al. "Multi-class active learning by uncertainty sampling with diversity maximization". In: *International Journal of Computer Vision* 113.2 (2015). Publisher: Springer, pp. 113–127.

[821] Yibo Yang and Paris Perdikaris. "Adversarial uncertainty quantification in physics-informed neural networks". In: *Journal of Computational Physics* 394 (2019). Publisher: Elsevier, pp. 136–152.

[822] Yang-Sae Moon, Kyu-Young Whang, and Woong-Kee Loh. "Duality-based subsequence matching in time-series databases". In: *Proceedings 17th International Conference on Data Engineering*. Heidelberg, Germany: IEEE Comput. Soc, 2001, pp. 263–272. (Visited on 06/25/2021).

[823] Deheng Ye et al. "Towards Playing Full MOBA Games with Deep Reinforcement Learning". In:

arXiv:2011.12692 [cs] (Dec. 2020). arXiv: 2011.12692. (Visited on 07/09/2021).

[824] Fei Ye et al. "A Survey of Deep Reinforcement Learning Algorithms for Motion Planning and Control of Autonomous Vehicles". In: *arXiv:2105.14218 [cs]* (May 2021). arXiv: 2105.14218. (Visited on 07/09/2021).

[825] Fei Ye et al. "Meta Reinforcement Learning-Based Lane Change Strategy for Autonomous Vehicles". In: *arXiv preprint arXiv:2008.12451* (2020).

[826] Rui Ye et al. "A Vectorized Relational Graph Convolutional Network for Multi-Relational Network Alignment". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 4135–4141.

[827] Kai Yi et al. "Knowledge-based recurrent attentive neural network for traffic sign detection". In: *arXiv preprint arXiv:1803.05263* (2018).

[828] Minglang Yin et al. "Non-invasive inference of thrombus material properties with physics-informed neural networks". In: *Computer Methods in Applied Mechanics and Engineering* 375 (2021). Publisher: Elsevier, p. 113603.

[829] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. "Center-based 3D Object Detection and Tracking". In: *CoRR* abs/2006.11275 (2020). arXiv: 2006.11275.

[830] Donggeun Yoo and In So Kweon. "Learning loss for active learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 93–102.

[831] Jaesik Yoon et al. "Bayesian Model-Agnostic Meta-Learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[832] Zhonghao Yu et al. "Tracking the Trackers". en. In: *Proceedings of the 25th International Conference on World Wide Web*. Montréal Québec Canada: International World Wide Web Conferences Steering Committee, Apr. 2016, pp. 121–132. (Visited on 06/24/2021).

[833] Di Yuan et al. "Active Learning for Deep Visual Tracking". In: *arXiv preprint arXiv:2110.13259* (2021).

[834] Jun Yuan et al. "A survey of visual analytics techniques for machine learning". In: *Computational Visual Media* 7.1 (2021). Publisher: Springer, pp. 3–36.

[835] Ming Yuan and Yi Lin. "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (2006). Publisher: Wiley Online Library, pp. 49–67.

[836] Sangdoo Yun et al. "Cutmix: Regularization strategy to train strong classifiers with localizable features". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 6023–6032.

[837] Manzil Zaheer et al. "Deep Sets". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. event-place: Long Beach, California, USA. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 3394–3404.

[838] Matthew D. Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision (ECCV)*. Vol. 8689 LNCS. Issue: PART 1. Springer, 2014, pp. 818–833.

[839] Wenyuan Zeng et al. "DSDNet: Deep Structured self-Driving Network". In: *arXiv:2008.06041 [cs]* (Aug. 2020). arXiv: 2008.06041. (Visited on 02/01/2021).

[840] Wenyuan Zeng et al. "End-to-end Interpretable Neural Motion Planner". In: *arXiv:2101.06679 [cs]* (Jan. 2021). arXiv: 2101.06679. (Visited on 02/24/2021).

[841] Friedemann Zenke, Ben Poole, and Surya Ganguli. "Continual Learning Through Synaptic Intelligence". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3987–3995.

[842] Chen Zeno et al. "Task agnostic continual learning using online variational bayes". In: *arXiv preprint arXiv:1803.10123* (2018).

[843] Nicholas Zerbel and Logan Yliniemi. "Multiagent monte carlo tree search". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 2019, pp. 2309–2311.

[844] Beichen Zhang et al. "State-relabeling adversarial active learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8756–8765.

[845] Chao Zhang et al. "Regions, Periods, Activities: Uncovering Urban Dynamics via Cross-Modal Representation Learning". In: *Proceedings of the 26th International Conference on World Wide Web*. WWW '17. event-place: Perth, Australia. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, pp. 361–370.

[846] Dingwen Zhang et al. "Weakly Supervised Object Localization and Detection: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), pp. 1–18.

[847] Dongkun Zhang et al. "Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems". In: *Journal of Computational Physics* 397 (2019). Publisher: Elsevier, p. 108850.

[848] Gongjie Zhang et al. "Meta-detr: Few-shot object detection via unified image-level meta-learning". In: *arXiv preprint arXiv:2103.11731* 2.6 (2021).

[849] Hongyi Zhang et al. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).

[850] Jiacheng Zhang et al. "Prior knowledge integration for neural machine translation using posterior regularization". In: *arXiv preprint arXiv:1811.01100* (2018).

[851] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. "Interpretable convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8827–8836.

[852] Quanshi Zhang et al. "Interpreting CNNs via Decision Trees". en. In: *arXiv:1802.00121 [cs]* (Feb. 2019). arXiv: 1802.00121. (Visited on 06/21/2021).

[853] Ruiyang Zhang, Yang Liu, and Hao Sun. "Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling". In: *Engineering Structures* 215 (2020). Publisher: Elsevier, p. 110704.

[854] S. Zhang, R. Benenson, and B. Schiele. "CityPersons: A Diverse Dataset for Pedestrian Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4457–4465.

[855] Shanshan Zhang, Jian Yang, and Bernt Schiele. "Occluded Pedestrian Detection Through Guided Attention in CNNs". en. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, June 2018, pp. 6995–7003. (Visited on 07/06/2021).

[856] Shifeng Zhang et al. "Widerperson: A diverse dataset for dense pedestrian detection in the wild". In: *IEEE Transactions on Multimedia* 22.2 (2019). Publisher: IEEE, pp. 380–393.

[857] Weixuan Zhang et al. "Active Model Learning using Informative Trajectories for Improved Closed-Loop Control on Real Robots". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4467–4473.

[858] Wen Zhang et al. "Interaction Embeddings for Prediction and Explanation in Knowledge Graphs". In: *CoRR* abs/1903.04750 (2019). _eprint: 1903.04750.

[859] Wen Zhang et al. "Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning". In: *The World Wide Web Conference*. WWW '19. event-place: San Francisco, CA, USA. New York, NY, USA: Association for Computing Machinery, 2019, pp. 2366–2377.

[860] Wenwei Zhang, Zhe Wang, and Chen Change Loy. "Multi-modality cut and paste for 3D object detection". In: *CoRR* abs/2012.12741 (2020).

[861] Xinruo Zhang, Gan Zheng, and Sangarapillai Lambotharan. "Trajectory Design for UAV-Assisted Emergency Communications: A Transfer Learning Approach". In: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. 2020, pp. 1–6.

[862] Zhanqiu Zhang et al. "Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction". In: *Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 2020, pp. 3065–3072.

[863] Zhao Zhang et al. "Knowledge Graph Embedding with Hierarchical Relation Structure". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 3198–3207.

[864] Zhishuai Zhang et al. "STINet: Spatio-Temporal-Interactive Network for Pedestrian Detection and Trajectory Prediction". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020, pp. 11343–11352.

[865] Shengjia Zhao, Tengyu Ma, and Stefano Ermon. "Individual calibration with randomized forecasting". In: *International Conference on Machine Learning*. PMLR, 2020, pp. 11387–11397.

[866] Lucia Zheng et al. "When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset". In: *arXiv:2104.08671 [cs]* (May 2021). arXiv: 2104.08671. (Visited on 07/06/2021).

[867] Zhi-Hua Zhou, Shi-Fu Chen, and Zhao-Qian Chen. "A statistics based approach for extracting priority rules from trained neural networks". en. In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. Como, Italy: IEEE, 2000, 401–406 vol.3. (Visited on 06/22/2021).

[868] Haoxi Zhong et al. "How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence". In: *arXiv:2004.12158 [cs]* (May 2020). arXiv: 2004.12158. (Visited on 07/06/2021).

[869] Haoxi Zhong et al. "Iteratively Questioning and Answering for Interpretable Legal Judgment Prediction". en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01 (Apr. 2020), pp. 1250–1257. (Visited on 07/06/2021).

[870] Bolei Zhou et al. "Learning deep features for discriminative localization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[871] Bolei Zhou et al. "Learning deep features for scene recognition using places database". In: *Advances in neural information processing systems* 27 (2014).

[872] Jie Zhou et al. "Graph Neural Networks: A Review of Methods and Applications". In: *CoRR* abs/1812.08434 (2018). _eprint: 1812.08434.

[873] Yin Zhou and Oncel Tuzel. "Voxelnet: End-to-end learning for point cloud based 3d object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4490–4499.

[874] Yuyin Zhou et al. "Prior-aware neural network for partially-supervised multi-organ segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 10672–10681.

[875] Fengbin Zhu et al. "Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering". In: *CoRR* abs/2101.00774 (2021). _eprint: 2101.00774.

[876] Jia-Jie Zhu and José Bento. "Generative adversarial active learning". In: *arXiv preprint arXiv:1702.07956* (2017).

[877] Jun Zhu, Ning Chen, and Eric P Xing. "Bayesian inference with posterior regularization and applications to infinite latent svms". In: *The Journal of Machine Learning Research* 15.1 (2014). Publisher: JMLR. org, pp. 1799–1847.

[878] Brian D Ziebart et al. "Maximum entropy inverse reinforcement learning." In: *Aaai*. Vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[879] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. "DeepRED – Rule Extraction from Deep Neural Networks". en. In: *Discovery Science*. Ed. by Toon Calders, Michelangelo Ceci, and Donato Malerba. Vol. 9956. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 457–473. (Visited on 06/21/2021).

[880] Christoph Zimmer, Mona Meister, and Duy Nguyen-Tuong. "Safe active learning for time-series modeling with gaussian processes". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2018, pp. 2735–2744.

[881]   Matthieu Zimmer et al. "Differentiable Logic Machines". In: *arXiv:2102.11529 [cs]* (Apr. 2021). arXiv: 2102.11529. (Visited on 06/25/2021).

[882]   H. Zou. "The Adaptive Lasso and Its Oracle Properties". In: *Journal of the American Statistical Association* 101 (2006), pp. 1418–1429.

[883]   Qin Zou et al. "Robust lane detection from continuous driving scenes using deep neural networks". In: *IEEE transactions on vehicular technology* 69.1 (2019). Publisher: IEEE, pp. 41–54.

[884]   Xiaohan Zou. "A Survey on Application of Knowledge Graph". en. In: *Journal of Physics: Conference Series* 1487 (Mar. 2020). Publisher: IOP Publishing, p. 012016. (Visited on 07/01/2021).

[885]   Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. "DoSeR - A Knowledge-Base-Agnostic Framework for Entity Disambiguation Using Semantic Embeddings". In: *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings*. Ed. by Harald Sack et al. Vol. 9678. Lecture Notes in Computer Science. Springer, 2016, pp. 182–198.