Concrete Problems for Autonomous Vehicle Safety: Advantages of Bayesian Deep Learning

Rowan McAllister, Yarin Gal†, Alex Kendall, Mark van der Wilk, Amar Shah, Roberto Cipolla, Adrian Weller†
Department of Engineering, University of Cambridge, UK
† also Alan Turing Institute, London, UK
{rtm26, yg279, agk34, mv310, as793, rc10001, aw665}@cam.ac.uk

Abstract

Autonomous vehicle (AV) software is typically composed of a pipeline of individual components, linking sensor inputs to motor outputs. Erroneous component outputs propagate downstream, hence safe AV software must consider the ultimate effect of each component's errors. Further, improving safety alone is not sufficient. Passengers must also *feel* safe to trust and use AV systems. To address such concerns, we investigate three under-explored themes for AV research: safety, interpretability, and compliance. *Safety* can be improved by quantifying the uncertainties of component outputs and propagating them forward through the pipeline. *Interpretability* is concerned with explaining what the AV observes and why it makes the decisions it does, building reassurance with the passenger. *Compliance* refers to maintaining some control for the passenger. We discuss open challenges for research within these themes. We highlight the need for concrete evaluation metrics, propose example problems, and highlight possible solutions.

1 Introduction

Autonomous vehicles (AVs) could bring great benefits to society, from reducing[1] the 1.25 million annual road fatalities [WHO, 2015] and injuries (≈100x fatality rate [NSC, 2016]), to providing independence to those unable to drive. Further, AVs offer the AI community many high-impact research problems in diverse fields including: computer vision; probabilistic modelling; gesture recognition; mapping; pedestrian and vehicle modelling; human-machine interaction; and multi-agent decision making.

Whilst great improvements have been made in vehicle hardware and sensor technology, we argue that one barrier to AV adoption is within *software*: particularly how to integrate different software components while considering how errors propagate through the system. AV systems are typically built of a pipeline of individual components, linking sensor inputs to motor outputs. Raw sensory input is first processed by object detection and localisation components, resulting in scene understanding. Scene understanding can then be inputted into a scene prediction component to anticipate other vehicles' motions. Finally, decision components transform scene predictions into motor commands. Such components are increasingly implemented with deep learning tools, as recent engineering advances have dramatically improved the performance of such tools [Kendall and Gal, 2017]. Deep learning methods supersede previous approaches in an array of tasks, and have become the de facto tools of choice in many applications. Yet building an AV pipeline out of deep learning building blocks poses new challenges.

Difficulties with a pipeline arise when erroneous outputs from one component propagate into subsequent components. For example, in a recent tragic incident, an AV perception component confused the white side of a trailer with bright sky. This misclassification propagated forwards to the motion planning process, resulting in the first fatality of an assisted driving system [NHTSA, 2017]. Such incidents damage public perception of AV safety, where there is already understandable concern [Giffi and Vitale, 2017]. Further, whilst a major goal for society is safer roads, it is not sufficient for AV software simply to be safe. Passengers need to *feel* safe in order to build trust and use the technology. Passenger apprehension can arise from both unfamiliarity and a lack of control over AV decisions. We therefore investigate three underappreciated research themes for the successful adoption of fully autonomous vehicles: *safety*, *interpretability*, and *compliance* (S.I.C. for short).

The *Safety* theme concentrates on reliably performing to a safe standard across all reasonable contexts. A crucial requirement for safety is to understand when an AV component faces unfamiliar contexts. Modelling an AV's uncertainty is a sensible way to measure unfamiliarity, and enables appropriate subsequent decisions to be made under such uncertainties. Bayesian probability theory is a formal language of uncertainty, providing tools to model different explanations of the world. Both *interpretability* and *compliance* help reassure passengers and make them feel safe, as well as to build trust. Interpretability is concerned with giving insights into the decisions an AV makes, and the outputs it produces. Such insights can take a visual form, be text based, or even auditory. Two aspects of compliance are of interest: 1) compliance with passenger preferences; 2) compliance with the law, regulations, and societal norms. By complying with passenger preferences (under a safety envelope), passengers can feel re-

[1] Most crashes involve human error [NHTSA, 2008, Tables 9a-c].

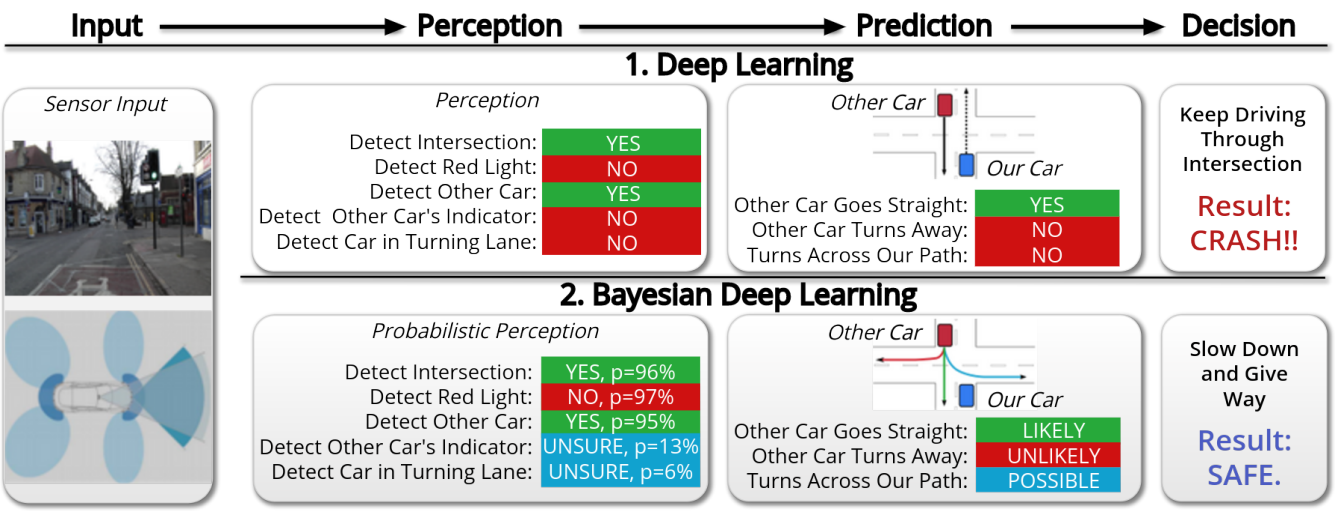


Figure 1: End-to-end Bayesian deep learning architecture. This figure illustrates our architecture and the key benefit of propagating uncertainty throughout the AV framework. We consider a hypothetical situation where an AV (our blue car) approaches an intersection, where another car (red) will turn left into our path. We compare a framework built on traditional (non-Bayesian) and Bayesian deep learning. Although both systems use the same initial sensory information, propagating uncertainty through the prediction and decision layers allows the Bayesian approach to avert disaster. Deep learning methods are required for state-of-the-art performance, hence more traditional Bayesian methods are not possible.

assured they retain high-level situational control. Compliance with the law, and the integration of regional regulations into an AV pipeline, is also a crucial requirement which leads to problems relating to integrating logical rules into deep learning systems. Research on such problems is extremely sparse, and we comment on this below.

In this document we survey technical challenges and suggest important paths for future research in S.I.C. from a machine learning research perspective. We highlight the need for concrete evaluation metrics, and suggest new metrics to evaluate safety by looking at both model performance and model uncertainty. We suggest example problems in each of the S.I.C. themes, and highlight possible solutions.

2 Safety

In AV software, no individual component exists in isolation. Improving safety by reducing individual component errors is necessary for safe AV software—but not sufficient. Even if each component satisfies acceptable fault tolerances in isolation, the accumulation of errors can have disastrous consequences. Instead, an understanding of how errors propagate forwards through a pipeline of components is critical. For example, in Figure 1, a perception component—which detects another vehicle's indicator lights—influences how a prediction component anticipates the vehicle's motion, ultimately influencing the driving decision. Since misclassification by components early in the pipeline affects components downstream, at the very least, each component should pass on the limitations and uncertainties of its outputs. Correspondingly, each component should be able to accept as input the uncertainty of the component preceding it in the pipeline. Further, these component uncertainties must be assembled in a principled way to yield a meaningful measure of overall system uncertainty, based on which safe decisions can be made.

A principled approach to modelling uncertainty is *Bayesian probability theory* (in fact, it can be shown that a rational agent *must* be Bayesian in its beliefs [Berger, 2013]). Bayesian methods use *probabilities* to represent uncertainty, and can be used for each individual component to represent its subjective confidence in its output. This confidence can then be propagated forward through the pipeline. To do so, each component must be able to input and output *probability distributions* rather than numbers. Together, component outputs downstream become functions of the uncertainty in predictions made upstream, enabling a decision layer to consider the consequences of plausible misclassifications made upstream, and act accordingly. Other approaches to conveying uncertainty exist in the field, including, for example, ensembling [Gal, 2016]. But the uncertainties of such techniques cannot necessarily be combined together in a meaningful way (a necessity with a complex software pipeline). Further, the *type* of uncertainty captured by these methods is not necessarily appropriate in safety applications (discussed further in §5). Traditional AV research has used Bayesian tools to capture uncertainty in the past [Paden *et al.*, 2016]. But the transition of many such systems to deep learning poses a difficulty. How would deep learning systems capture uncertainty?

While many Bayesian models exist, deep learning models obtain state-of-the-art perception of fine details and complex relationships [Kendall and Gal, 2017]. Hence we propose the use of Bayesian Deep Learning (BDL). BDL is an exciting field lying at the forefront of research. It forms the intersection between deep learning and Bayesian probability theory, offering principled uncertainty estimates within deep architectures. These deep architectures can model complex tasks by leveraging the hierarchical representation power of deep learning, while also being able to infer complex multi-modal posterior distributions. Bayesian deep learning models typically form uncertainty estimates by either placing distribu-

tions over model weights, or by learning a direct mapping to probabilistic outputs. One pragmatic approach is to use dropout to approximate a Bernoulli distribution over the neural network's weights. One can then sample from this network with different dropout masks to obtain the posterior distribution [Gal, 2016, Chapter 2].

Figure 1 illustrates an example where propagating uncertainties throughout the entire pipeline can prevent disaster. It contrasts standard deep learning with Bayesian deep learning. In this case, standard deep learning makes hard predictions, whereas Bayesian deep learning outputs probabilistic predictions accounting for each *model's ignorance about the world*. In this scenario, the AV is approaching an intersection which has an oncoming vehicle whose intention is to turn across the path of our AV. Consider the situation where our model detects the other car, but fails to detect that it is in the turning lane or that it is flashing its indicator. Using deep learning, and passing hard classification results through the architecture, the system would be unable to predict that the oncoming car might turn. In contrast, a Bayesian deep learning system might still mistake the oncoming car's position and indicator, but it would be able to propagate its uncertainty to the next layers. Given the probabilities of the other vehicle's existence, indicator lights, and location in the turning lane, the prediction component can now compute the probability that the oncoming would obstruct the AV's motion. At the third stage, the decision component reacts to the range of possible predicted movements of other vehicles, resulting in a completely different outcome and averting a crash.

Methods for representing uncertainty in deep learning are critical for safe AVs, and raise many imperative research questions. For instance: to improve performance, should one focus on developing a better uncertainty estimate, or a better model? These competing hypotheses can be hard to tease apart. Other important research directions for BDL include the development of improved inference approximations (existing approaches can under-estimate model variance through excessive independence assumptions), and identifying when a model is misspecified (for example if a priori we give high probability to incorrect models, and low probability to correct models). The accumulation of errors arising from various approximations, such as inference approximations as well as potential model misspecification, is an important area of open research.

To assist training and analysing the pipeline as a whole, we propose that not one, but *all components* should use BDL tools. This allows for end-to-end training and the combination of uncertainties through the pipeline, and also simplifies the writing and maintenance of code (compared to a heterogeneous collection of algorithms). Often a modular approach (i.e. not end-to-end) is taken with AVs, developing each component independently, since it is easier to validate and unit-test individual components. However, end-to-end systems can perform better when the components are optimised jointly, discussed in §5.4.

3 Interpretability

The next theme we review is *interpretability.* Interpretable algorithms give insights into the decisions they make, and the outputs they produce. Such insights could be visual (e.g. a heat map highlighting model uncertainty), text based (e.g. asking a system 'why did you turn left?'), or auditory (an alarm sounding when the AV does not know what to do, and the passenger is required to take control). An AV should be able to explain to a passenger what it observes and why it makes the decisions it does, rather than expecting passengers to trust their lives to a black box. Trust increases when a product is shown to base decisions on environmental aspects that appear reasonable to a human. Interpretability is useful when a task is hard to define, or when a model is difficult to test against the huge variety of possible inputs it could receive [Doshi-Velez and Kim, 2017].

We highlight several important goals for interpretability:

- to help *passengers* trust AV technology by informing passengers what the AV is doing and why,
- to help *society broadly* understand and become comfortable with the technology, overcoming a reasonable fear of the unknown,
- to aid *engineers* understand the model to validate against safety standards,
- accountability for *insurance and legal* liability by reviewing and explaining decisions if something goes wrong.

We are particularly interested in interpretability by enabling interrogation of the system by the passenger. For example, asking, 'Why did you suddenly shift left?' should yield a simple intelligible response from the AV such as 'I shifted left to give the cyclist on our right more space'. Such communication could be done through natural-language or visual interfaces. Specifically, we discuss two technical aspects to this problem: model saliency and auxiliary outputs.

3.1 Model Saliency

Saliency detection methods show which parts of a given image are the most relevant to a given model and task. Such visual interpretations of how machine learning models make decisions is an exciting new area of AI research.

In deep learning, a few techniques have been proposed. One method uses the derivatives back-propagated through a network to quantify saliency [Simonyan *et al.*, 2013], but this often results in low quality pixel-level output. Another technique is visualising the model's parameters [Zeiler and Fergus, 2014] or by perturbing the input [Zhou *et al.*, 2014; Ribeiro *et al.*, 2016]. [Dabkowski and Gal, 2017] show how to learn a model to visualise saliency in real-time. When applied to autonomous vehicle models, research shows that driving decisions depend most on the boundaries of other cars and lane markings immediately ahead [Bojarski *et al.*, 2017]. Recent work [Zintgraf *et al.*, 2017] adds an interesting new element: an input item is relevant if it is important in explaining the prediction *and* it could not easily be predicted by other parts of the input. Such ideas are still in their infancy, and much work is needed before these could be used for real-world systems.

Bayesian deep learning suggests further ways to interpret saliency. We can understand how perturbations affect the estimated uncertainty in perception, as well as further down the pipeline. Further research is needed to apply these techniques in real-time, and to communicate this effectively to users.

3.2 Auxiliary Outputs

Interpretability is particularly important for *end-to-end* systems. One approach to introducing transparency in end-to-end pipelines is to examine intermediate, or auxiliary, outputs. In Section 5 we propose an end-to-end framework to map sensory inputs to control outputs. However, we also advocate for human-interpretable auxiliary outputs at each major stage. For example, in addition to the end-to-end driving loss, it would be useful to learn intermediate representations for depth, motion, geometry, semantic segmentation and prediction components [Kendall *et al.*, 2017]. These outputs may help in training the model by suggesting sensible intermediate representations, and they provide visual output to help humans understand and debug the model. Existing literature on this topic is extremely sparse. One challenge is to achieve effective intermediate outputs which explain the model's representation while maintaining the flexibility of end-to-end learning.

4 Compliance

The last major theme of research we discuss is *compliance*. In our setting, we suggest there are two important aspects to compliance: passenger reassurance, and law abiding. First, consider entering a car driven by someone unfamiliar, whose driving habits quickly make you concerned. You might wish the driver would slow down, or give the cyclist ahead more space. With AVs, such compliance to passengers' high-level directives could help to reassure them by giving a sense of control [Baronas and Louis, 1988]. Human drivers can adjust to non-verbal cues from passengers. If a passenger looks uneasy, or perhaps if a passenger is clearly holding something delicate then smoother driving is required. AVs cannot currently observe such cues, therefore passenger requests to an AV may be frequent.

Second, compliance also means meeting the requirements of the law and societal norms. An AV should only accept passenger requests that are safe and legal, hence protecting passengers from liability in the event of a crash. A manufacturer (or perhaps the software provider) would need to balance the flexibility of the operational framework with their own tolerance for liability risk.

This theme suggests important new research directions. Since the community has already identified that vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication protocols would be useful, perhaps an analogous 'vehicle-to-user' (V2U) protocol is also possible. Alternatively, a passenger might provide feedback to an AV in pursuit of a personalized service. But an individual cannot be expected to provide thousands of training examples. It is therefore important to research methods for data efficient learning, or more broadly "individualised learning". We believe that relevant ideas might come from transfer learning or data efficient reinforcement learning—learning to comply with a minimal number of interactions with the passenger. Solutions to problems arising from this under-explored theme have the potential to transform our current perception of AVs.

5 Propagating Uncertainty for Safe Systems

We next give a concrete example for realising the ideas above. We review a typical set of individual components used in an AV pipeline, and discuss how to propagate uncertainty between the individual components. We use this example to highlight concrete problems in S.I.C. AV research.

A variety of AV software architectures exist in the literature [Luettel *et al.*, 2012; Sivaraman and Trivedi, 2013]. However, we refer to the generic pipeline structure used in Figure 1 to guide our discussion on how software components commonly affect each other. We describe each component's ability to handle input and output uncertainty. Note that while Bayesian probabilistic methods are already ingrained in most robotics research to provide probabilistic outputs [Thrun *et al.*, 2005], probabilistic inputs have been explored much less.

Before we begin, it is important to highlight the two major types of uncertainty we wish to capture with BDL models. First, *aleatoric* (data dependent) uncertainty models noise which is inherent in the observations. Aleatoric uncertainty reflects the difficulty and ambiguity in the input, for example an over-exposed image or a featureless wall. Second, epistemic (model dependent) uncertainty accounts for ambiguity in the model itself. This uncertainty can arise, for example, from different possible ways of explaining the data. Epistemic uncertainty can be mitigated given enough data, and is a measure of "familiarity"—how close a new context is to previously observed contexts. An example of what these uncertainties looks like from a prototype scene understanding system is shown in Figure 2 and Figure 3. We next discuss architecture pipelines composed of three components: a perception layer, a prediction layer, and a decision layer.

5.1 Perception Layer

A perception system should be able to solve three important tasks: estimate the semantics of the surrounding scene, understand its geometry, and estimate the position of the vehicle itself.

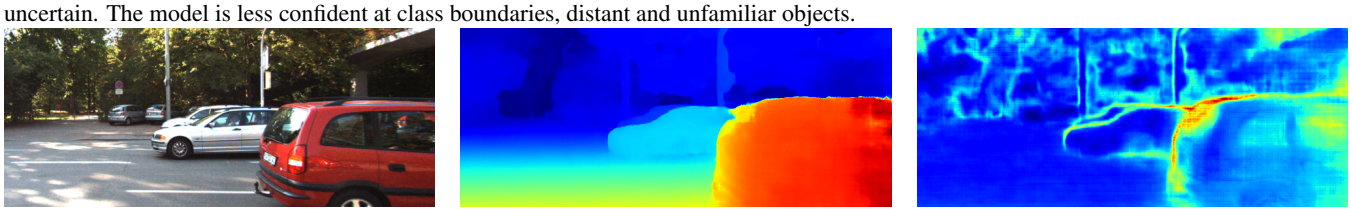
Semantics. The role of a semantic scene understanding system is to classify the class and state of the surroundings based on their appearance. Visual imagery is highly informative but non-trivial for algorithms to understand. Semantic segmentation is a common task where we wish to segment pixels and classify each segment according to a set of pre-defined class labels. For example, given a visual input (Figure 2a), the image is segmented into road, paths, trees, sky, and buildings (Figure 2b). Most algorithms make hard-classifications, however uncertainty can be captured with semantic segmentation algorithms which use Bayesian deep learning [Kendall *et al.*, 2015].

Geometry. In addition to semantics, a perception system must also infer the geometry of the scene. This is important to establish obstacles and the location of the road surface. This problem may be approached with range sensors such as ultrasound or LIDAR. However, visual solutions have been demonstrated using supervised [Eigen and Fergus, 2015] and unsupervised [Garg *et al.*, 2016] deep learning (see Figure 3).

Localisation. Finally, a perception system must estimate the vehicle's location and motion. In unfamiliar environments

(a) Sensor input (b) Semantic segmentation (c) Uncertainty

Figure 2: Bayesian deep learning for semantic segmentation. Typically, deep learning models make predictions (b) without considering the uncertainty (c). Our method proposes to estimate uncertainty (c) from each layer to pass down our Bayesian pipeline. (b) shows semantic segmentation, where classes are coloured individually. (c) shows uncertainty where colours other than dark blue indicate a pixel is more uncertain. The model is less confident at class boundaries, distant and unfamiliar objects.



(a) Sensor input (b) Depth prediction (c) Uncertainty

Figure 3: Bayesian deep learning stereo depth estimation algorithm. This example illustrates a common failure case for stereo vision, where we observe that the algorithm incorrectly handles the reflective and transparent window on the red car. Importantly, we observe that the BDL algorithm also predicts a high level of uncertainty (c) with this erroneous output (b).

this is commonly referred to as *simultaneous localisation and mapping* (SLAM) [Durrant-Whyte and Bailey, 2006]. It is beneficial to register the vehicle to a global map and also locally to scene features (such as lane position and distance to intersection). An AV must infer which lane it is in and where it is relative to an oncoming intersection. For this task as well there exist computer vision techniques which use Bayesian deep learning [Kendall and Cipolla, 2016].

But other problems exist in perception as well, and uncertainty plays a central role in these. Roboticists have long considered uncertainty in robotic perception. Uncertainty in geometric features has been considered important for motion planning [Durrant-Whyte, 1988] and filtering and tracking [Kalman, 1960; Julier *et al.*, 1995; Isard and Blake, 1998]. Another popular use is sensor fusion, given probabilistic models of each sensor (a likelihood function). Sensor fusion combines multiple sensor outputs using Bayes rule to construct a world-view more confident than any one sensor can alone. The benefit is a principled way to combine all available information, reducing mapping and localisation errors. This way of combining sensors of different types, in particular LIDAR, radar, visual, and infra-red, helps to overcome the physical limitations of any one individual sensor [Brunner *et al.*, 2013].

The perceptual component is usually placed at the start of the pipeline, and has to process raw sensory input. The perception component's output should consists of a prediction together with uncertainties, which are useful for outputting more precise object detection (via filtering and tracking) and more accurate state estimation (via improved mapping and localisation). Evaluating a hard decision should not be made at the output from the perception component. Instead, the uncertainties should be explicitly propagated forward to the prediction layer. Most research does not consider uncertainty beyond this point, but it is a critical input for the follow-up prediction and decision layers (see Figure 1). Uncertainties can be propagated as a set of samples from the perception system's output distribution, for example. However, this approach is limited in its use in real-time systems. It is an open problem to decide on a good representation which captures the epistemic uncertainty of the perception system efficiently.

5.2 Prediction Layer

Even though scene perception components can identify obstacle locations, many obstacles are dynamic agents such as pedestrians and vehicles. Scene prediction helps an AV anticipate agent motions to avoid dangerous situations, such as a vehicle ahead possibly braking suddenly, or a pedestrian possibly stepping onto the road. Given a typical LIDAR range ($\sim$100m) and legal speed limit ($\sim$30m/s), predictions are generally not made more than $\sim$3.3s ahead.

Given a model of external agents' behaviour, scene prediction can be framed as a partially observable Markov decision process (POMDP) [Bandyopadhyay *et al.*, 2013]. In the absence of a model, inverse reinforcement learning (IRL) can be used [Ziebart *et al.*, 2009]. IRL is the problem of inferring another agent's latent reward function from observations [Ng and Russell, 2000]. An inferred reward function can then be used in the decision layer to emulate examples of human driving for example, called *apprenticeship learning* [Abbeel and Ng, 2004], or simply to predict external agents' motions. In this method we learn a reward function given features from a particular scene, aiming to generalise well to new scenes [Levine *et al.*, 2010]. Variants of IRL include Bayesian IRL which use Gaussian processes [Levine *et al.*, 2011], as do deep IRL which use neural networks [Wulfmeier *et al.*, 2015a]. Probabilistic deep methods have been introduced recently [Wulfmeier *et al.*, 2015b]. Probabilistic output predictions of pedestrians or vehicles can be easily visualised, which is advantageous for debugging and interpretability purposes. For example, heat maps of state occupancy probabili-

ties might be used [Ziebart *et al.*, 2009]. Other approaches for prediction try to build an autoregressive model that predicts future segmentations given visual segments from a perception layer [Neverova *et al.*, 2017]. Such a method is visually interpretable as well, but inaccurate beyond 0.5s due to lack of geometrical knowledge.

Input uncertainty, however, has not been investigated with respect to scene prediction, as far as we are aware. Input uncertainty outputted by the perceptual layer in object existence and location would be beneficial for prediction in a complex pipeline. This can be extremely difficult computationally to achieve in real-time. If the output of the prediction layer is a map of occupancy probabilities, the contribution of each object being tracked could be weighted by its probability of existence, and convolved with the probability distribution function of its location for example. This is an example open problem which has not been addressed by the literature yet.

5.3 Decision Layer

A decision layer commonly includes route planning, motion planning, and feedback control components to control the AV [Paden *et al.*, 2016]. For each component, we strongly advocate a Bayesian decision theoretic approach, with Bayesian decision theory being a principled methodology to decide on actions which maximize a given utility function [Bernardo and Smith, 2001, Chapter 2]. Further, Bayesian decision theory provides a feedback mechanism in which actions can be taken to reduce the agent's uncertainty. This feedback is used to make better informed decisions at a later time (e.g. moving to get a better view). This relies on tools developed in the partially observable Markov Decision processes literature [Astrom, 1965; Sondik, 1971]. Two major tasks for the decision component is route planning, motion planing.

Route Planning

This task concerns navigation according to an a priori known graphical structure of the road network, solving the *minimum-cost path problem* using algorithms such as A*. Several algorithms exist that are more suited to large networks than A* [Bast *et al.*, 2016]. For many purposes route planning is mostly a solved problem.

Motion Planning

Given which streets to follow, in motion planning (MP) we navigate according to the current road conditions observed by the AV's sensors, taking into account e.g. traffic light states, pedestrians, and vehicles. Most MP methods are either sample based path planners or graph search algorithms [LaValle, 2006]. Classical path planning considers an AV's state space divided into binary 'free space' and 'obstacle' classes. Uncertainty in localisation and mapping will blur such a free-obstacle distinction, which several sample based planners consider [Bry and Roy, 2011; Melchior and Simmons, 2007].

To plan motions well under dynamics uncertainty, an accurate probabilistic dynamics model is critical. While kinematic models work well for low-inertia non-slip cases, more complex models are required to model inertia, and probabilistic models are needed to model slip well [Paden *et al.*, 2016]. Modelling dynamics uncertainty is especially relevant for safe driving on dirt or gravel where slipping is common and should be anticipated [McAllister *et al.*, 2012]. More recently, BDL dynamics models have been used in the design of controllers [Gal *et al.*, 2016; Depeweg *et al.*, 2016], works which should be extended into MP.

5.4 An End-to-End Learning Paradigm

In an *end-to-end learning* paradigm we jointly train all components of the system from perception, through prediction, and ending in decision. This is an attractive paradigm because it allows each component to form a representation which is optimal w.r.t. the desired *end task*. Alternatively, training a single component in isolation provides no signal or knowledge of the end task. For example, training a perception system in isolation might result in a representation which equally favours accurate segmentation of the clouds above the AV compared to identifying the vehicle immediately ahead. End-to-end training would allow the perception system to focus its learning on aspects of the scene most useful for the end task.

However, solely using an end-to-end training signal requires large amounts of data. This is because mapping the sensory inputs to driving commands is complex. To constrain the model in a way that learns faster, it is useful to use auxiliary losses with each component's task. This is also important for interpretability reasons discussed in Section 3. In practice, we might pre-train individual components (such as semantic segmentation, prediction, etc.) and then fine-tune them using end-to-end training with auxiliary losses.

A criticism of the sequential prediction-decision pipeline architecture is its assumption that the external world evolves independent to the AV's decisions. Yet surrounding pedestrians and vehicles certainly decide action based on an AV's decisions, and vice versa. For example, driving situations that require multi-agent planning include merging lanes, four ways stop signs, rear-end collision avoidance, giving way, and avoiding another's blind spots. Such multi-agent planning can be solved by coupling prediction and decision components to jointly predict the external world state *and* the conditional AV decisions. Such an architecture avoids basing future decisions on the *marginal* distribution of future external-scene states (which is problematic, since in 3-10s, the marginal probability distribution could cover the entire road with non-negligible probabilities of occupancy). Concretely, an approach to jointly reasoning about multi-agent scenarios is probabilistic model based reinforcement learning [Deisenroth and Rasmussen, 2011], which has been extended to Bayesian deep learning [Gal *et al.*, 2016].

6 Metrics

Being able to quantify performance of algorithms against a set benchmark is critical to developing safe systems. It allows acknowledgement of incremental development. Additionally, it provides an ability to validate a system against a set of safety regulations. Therefore, a set of specific metrics is needed to score and validate a system. Metrics and loss functions can be difficult to specify without unintended consequences [Amodei *et al.*, 2016]. Therefore, we wish to highlight key considerations when designing metrics for AV classification and regression systems, and specifically under a Bayesian framework.

6.1 Uncertainty Metrics

The core benefit of using a Bayesian framework is being able to propagate uncertainty to improve prediction and model reasoning. Therefore it is imperative that the probability distribution computed by each component accurately reflects the true uncertainty in the system. Metrics must capture the quantitative performance of the uncertainty the model is trying to measure. Typical metrics should include uncertainty calibration plots and precision recall statistics of regression and classification algorithms. Other task-specific metrics include inspecting test-log-predictive-probabilities—a measure of how "surprised" the system output is on a test-set of known outcomes.

It is most important to accurately quantify the uncertainty of intermediate representations, such as those from perception and scene prediction components. This is because the most important aspect is to correctly propagate uncertainty to the following layers so that we can make informed decisions. The final control output will be a hard decision. Therefore uncertainty metrics should focus on intermediate components and tasks such as perception, sensor fusion and scene prediction.

6.2 End-to-End Metrics

Under an end-to-end paradigm, ultimately we should test and validate our model using *downstream metrics*. Metrics such as ride comfort, the number of crashes and journey efficiency are what will be used to ultimately judge AVs. Each component should be judged on its ultimate effects towards these metrics, rather than in isolation. Does an improvement in component C by measure M correspond well to an improvement in final vehicle performance P? In isolation, a trained layer might focus on modelling things which do not matter to the end task.

Another challenge in quantifying the performance of end-to-end systems is the attribution of individual components to end performance. For example, if two components perform much better when combined in an end-to-end system, rather than when used individually, how do we quantify their contribution? In game theory, this is answered by the Shapley value [Štrumbelj and Kononenko, 2014]. However, refining such approaches and applying them to large scale systems is challenging.

6.3 Individual Component Metrics

We recognise that while an end-to-end set of metrics is necessary to fully evaluate the safety of a system, individual metrics for each module are necessary in practice. This requires an interpretable system, as the system must expose intermediary outputs for analysis (and cannot simply learn an end-to-end mapping). However, in the robotics and AI communities, these metrics often do not reflect the end use of the system. For a safe system, these metrics should be both end-goal related and also quantify the uncertainty in each module. To this point, we make some specific suggestions of metrics for each component to use during development.

Perception Layer Metrics

Currently, low level vision tasks such as semantic segmentation and depth perception focus on improving pixel-wise metrics such as pixel-accuracy or pixel intersection over union. However, in an integrated end-to-end system, it is more important to capture object and obstacle metrics. Object false-positive and false-negative scores are more important than precise segmentation boundary delineation and pixel accuracy. For geometry estimation, it is more important to minimise large failures rather than to improve millimetre precision—which can be achieved by considering threshold accuracy metrics rather than metric errors. Consideration should also be made to develop metrics that reflect the calibration of perception uncertainty outputs. Sensor fusion is an important process here therefore it is essential for metrics to quantify relative uncertainty between sensory inputs.

Prediction Layer Metrics

The goal of IRL is to learn a reward function. Existing IRL metrics in the literature use distances between inferred (and real) functions, e.g. reward functions [Ramachandran and Amir, 2007; Aghasadeghi and Bretl, 2011], value functions [Levine *et al.*, 2011; Wulfmeier *et al.*, 2015a; 2015b], or policies [Levine *et al.*, 2010; Abbeel and Ng, 2004]. However existing metrics in the field are not appropriate for AVs. Instead, we wish to evaluate probabilistic predictions of agents in continuous 2D maps, therefore we assert that log probabilities [Ziebart *et al.*, 2008; 2009] are a better suited metric for AVs. This is especially the case since motion planning algorithms should be optimised along a similar metric—e.g. minimising the probability of collision, which is directly related to the probability of a map cell being occupied or not. Since IRL relies on reward *features* of scenes to generalise well to new scenes, evaluation should be done on a new scene to the one the IRL algorithm was trained on. A test-set of new scenes should be used to test 1) how *transferable* the reward features are, and 2) how well the model uses those features [Levine *et al.*, 2011].

Decision Layer Metrics

The ultimate aim of AV systems is to drive significantly more safely than human drivers. It is therefore not sensible to evaluate AVs on how well they emulate human drivers. For example, several datasets evaluate steering performance by comparing to human drivers using steering angle RMSE metrics. However steering angle RMSE makes little sense in the absence of destination inputs. Much worse, this metric makes little sense in the absence of the recorded human driver's *intentions* in choosing intersection and lane changes.

Given the probabilistic predictions of external agents such as the human driver, a more sensible metric for MP is not immediately clear. However, two suggested options include minimising either the 'probability of a collision' or the 'expected number of agents collided with', as well as taking into account the ride comfort and distance of the agent from the desired final destination at the end of the task. Additionally, it is desirable to be able to choose paths which bound the probability of collision [Bry and Roy, 2011]. However, the probability of collisions cannot be the only factor accounted for (since optimising collision safety alone would result in the vehicle never moving). A trade-off must be made between safety and the requirements of the vehicle. This trade-off may be adjusted by the user for systems which exhibit compliance.

6.4 AV Classifications

Currently, the accepted standard of AV classification is SAE J3016 [SAE, 2014]. Whilst the standard focuses on the degree of human input required, it is also related to the vehicle's capabilities. The most advanced class is SAE Level 5: '*the full-time performance by an Automated Driving System of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver*'. No higher class currently exists. However, one of the great potentials of AV technology is driving capabilities that greatly *exceed* what can be managed by human drivers. Human drivers are limited by their observational abilities, processing power, and slow reaction times. By contrast, AVs can access more of the electromagnetic spectrum to better perceive through rain, fog, dust, smoke, darkness [Brunner *et al.*, 2013], can have more processing power, and much faster reaction times. The development of higher SAE Levels could help encourage AV research to help reach this potential.

7 Conclusions

We believe that autonomous vehicles will bring many benefits to society, while presenting important and fascinating research challenges to the artificial intelligence community. In this paper we highlighted three themes which will be critical for a smooth adoption of AV systems by society. The first is safety through the use and propagation of uncertainty from every component throughout the entire system pipeline, following a principled Bayesian framework. We discussed the dangers with making hard decisions with perceptual components for example, asserting that soft (uncertain) classifications should be propagated through to the decision layer. This enables the AV to act more cautiously in the event of greater uncertainty. To implement each component in a safe system, we suggested Bayesian deep learning which combines the advantages of the highly flexible deep learning architectures with Bayesian methods. We also discussed the themes of interpretability and compliance as ways to build trust and mitigate fears which passengers might otherwise reasonably have about unfamiliar black-box AV systems. Lastly, we discussed the importance of clear metrics to evaluate each component's probabilistic output based on their ultimate effect on the vehicle's performance.

Acknowledgements

Adrian Weller acknowledges support by the Alan Turing Institute under the EPSRC grant EP/N510129/1, and by the Leverhulme Trust via the CFI.

References

[Abbeel and Ng, 2004] Pieter Abbeel and Andrew Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, page 1, 2004.

[Aghasadeghi and Bretl, 2011] Navid Aghasadeghi and Timothy Bretl. Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals. In *IROS*, pages 1561–1566, 2011.

[Amodei *et al.*, 2016] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[Astrom, 1965] Karl Astrom. Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

[Bandyopadhyay *et al.*, 2013] Tirthankar Bandyopadhyay, Kok Won, Emilio Frazzoli, David Hsu, Wee Lee, and Daniela Rus. *Intention-Aware Motion Planning*, pages 475–491. Berlin, Heidelberg, 2013.

[Baronas and Louis, 1988] Ann-Marie Baronas and Meryl Louis. Restoring a sense of control during implementation: How user involvement leads to system acceptance. *MIS Quarterly*, 12(1):111–124, 1988.

[Bast *et al.*, 2016] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato Werneck. Route planning in transportation networks. In *Algorithm Engineering*, pages 19–80. 2016.

[Berger, 2013] James Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[Bernardo and Smith, 2001] José M Bernardo and Adrian Smith. Bayesian theory. *Measurement Science and Technology*, 12(2):221, 2001.

[Bojarski *et al.*, 2017] Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv preprint arXiv:1704.07911*, 2017.

[Brunner *et al.*, 2013] Christopher Brunner, Thierry Peynot, Teresa Vidal-Calleja, and James Underwood. Selective combination of visual and thermal imaging for resilient localization in adverse conditions: Day and night, smoke and fire. *Journal of Field Robotics*, 30(4):641–666, 2013.

[Bry and Roy, 2011] Adam Bry and Nicholas Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *ICRA*, pages 723–730, May 2011.

[Dabkowski and Gal, 2017] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. 2017.

[Deisenroth and Rasmussen, 2011] Marc Deisenroth and Carl Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *ICML*, pages 465–472, 2011.

[Depeweg *et al.*, 2016] Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Learning and policy search in stochastic dynamical systems with Bayesian neural networks. *arXiv preprint arXiv:1605.07127*, 2016.

[Doshi-Velez and Kim, 2017] Finale Doshi-Velez and Been Kim. A roadmap for a rigorous science of interpretability. *arXiv preprint arXiv:1702.08608*, 2017.

[Durrant-Whyte and Bailey, 2006] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[Durrant-Whyte, 1988] Hugh Durrant-Whyte. Uncertain geometry in robotics. *Journal on Robotics and Automation*, 4(1):23–31, 1988.

[Eigen and Fergus, 2015] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, pages 2650–2658, 2015.

[Gal *et al.*, 2016] Yarin Gal, Rowan McAllister, and Carl Rasmussen. Improving PILCO with Bayesian neural network dynamics models. In *Data-Efficient Machine Learning workshop, ICML*, 2016.

[Gal, 2016] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[Garg *et al.*, 2016] Ravi Garg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[Giffi and Vitale, 2017] Craig Giffi and Joe Vitale. Fact, fiction and fear cloud future of autonomous vehicles in the minds of consumers. Technical report, Deloitte, Jan 2017.

[Isard and Blake, 1998] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[Julier *et al.*, 1995] Simon Julier, Jeffrey Uhlmann, and Hugh Durrant-Whyte. A new approach for filtering nonlinear systems. In *American Control Conference*, volume 3, pages 1628–1632, Jun 1995.

[Kalman, 1960] Rudolph Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[Kendall and Cipolla, 2016] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. *ICRA*, 2016.

[Kendall and Gal, 2017] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

[Kendall *et al.*, 2015] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[Kendall *et al.*, 2017] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. 2017.

[LaValle, 2006] Steven LaValle. *Planning algorithms*. Cambridge University Press, 2006.

[Levine *et al.*, 2010] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Feature construction for inverse reinforcement learning. In *NIPS*, pages 1342–1350, 2010.

[Levine *et al.*, 2011] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with Gaussian processes. In *NIPS*, pages 19–27, 2011.

[Luettel *et al.*, 2012] Thorsten Luettel, Michael Himmelsbach, and Hans-Joachim Wuensche. Autonomous ground vehicles – concepts and a path to the future. *Proceedings of the IEEE*, 100:1831–1839, May 2012.

[McAllister *et al.*, 2012] Rowan McAllister, Thierry Peynot, Robert Fitch, and Salah Sukkarieh. Motion planning and stochastic control with experimental validation on a planetary rover. In *IROS*, pages 4716–4723, Oct 2012.

[Melchior and Simmons, 2007] Nik Melchior and Reid Simmons. Particle RRT for path planning with uncertainty. In *ICRA*, pages 1617–1624. IEEE, 2007.

[Neverova *et al.*, 2017] Natalia Neverova, Pauline Luc, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. *arXiv preprint arXiv:1703.07684*, 2017.

[Ng and Russell, 2000] Andrew Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *ICML*, pages 663–670, 2000.

[NHTSA, 2008] NHTSA. Dot hs 811 059: National motor vehicle crash causation survey. Technical report, U.S. Department of Transportation, National Highway Traffic Safety Administration, Jul 2008.

[NHTSA, 2017] NHTSA. PE 16-007. Technical report, U.S. Department of Transportation, National Highway Traffic Safety Administration, Jan 2017. Tesla Crash Preliminary Evaluation Report.

[NSC, 2016] NSC. Motor vehicle fatality estimates. Technical report, National Safety Council, Statistics Department, Feb 2016.

[Paden *et al.*, 2016] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *Transactions on Intelligent Vehicles*, 1(1):33–55, 2016.

[Ramachandran and Amir, 2007] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. *Urbana*, 51(61801):1–4, 2007.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *KDD*, pages 1135–1144, 2016.

[SAE, 2014] SAE. J3016 surfact vehicle information report. Technical report, Society of Automotive Engineers International, Jan 2014. Levels of driving automation.

[Simonyan *et al.*, 2013] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[Sivaraman and Trivedi, 2013] Sayanan Sivaraman and Mohan Manubhai Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *Transactions on Intelligent Transportation Systems*, 14(4):1773–1795, Dec 2013.

[Sondik, 1971] Edward Sondik. The optimal control of partially observable Markov processes. Technical report, 1971.

[Štrumbelj and Kononenko, 2014] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[Thrun *et al.*, 2005] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[WHO, 2015] WHO. *Global status report on road safety 2015*. World Health Organization, Violence and Injury Prevention, 2015.

[Wulfmeier *et al.*, 2015a] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Deep inverse reinforcement learning. *CoRR*, 2015.

[Wulfmeier *et al.*, 2015b] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

[Zeiler and Fergus, 2014] Matthew Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833, 2014.

[Zhou *et al.*, 2014] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene CNNs. *arXiv preprint arXiv:1412.6856*, 2014.

[Ziebart *et al.*, 2008] Brian Ziebart, Andrew Maas, Andrew Bagnell, and Anind Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438, 2008.

[Ziebart *et al.*, 2009] Brian Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, Andrew Bagnell, Martial Hebert, Anind Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *IROS*, pages 3931–3936, 2009.

[Zintgraf *et al.*, 2017] Luisa Zintgraf, Taco Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.