Hierarchical Decision-Making for Autonomous Driving

Simon Chauvin, *ESR Labs AG*

Motivation - *ESR Labs AG* works on all software components of Autonomous Driving. It includes the relatively well understood modules of *Perception*, *Localization* and *Drive Control*. The *Machine Learning Team* also explores the potential of two components that form the brain of Autonomous Vehicles: *Scene Understanding* and *Decision-Making*. In this informal paper, I want to share my vision on these two cognitive elements: *what are their roles in the above-mentioned modular architecture? What are some of the difficulties regarding their implementation? How can they be overcome?* In addition, I would like to share recent references for further reading among some of my favourite sources of reflexion.

Abstract - This paper is structured as follows: Sections I emphasizes the role of *Scene Understanding* in Autonomous Driving. It is responsible for designing a representation of the traffic scene that is intelligible for *Decision-Making*. A representation based on a multi-level structure is proposed. Three description levels are introduced: *Metrical*, *Topological* and *Semantic*. Section II explains the benefit of combining semantic and numerical approaches in *Decision-Making*. Section III proposes to structure the *Decision-Making* process in a hierarchical architecture, based on the here-introduced *3M* decomposition: *Mission - Manoeuvre - Motion*. The Section IV focuses on *Manoeuvre Selection* and shows how *Reinforcement Learning* can be merged with traditional rule-based methods. This ensures safety and traffic rules compliance while offering generalization capabilities to address new complex scenarios. Techniques to handle uncertainty in *Decision-Making* are developed in section IV. Section V finally shows that the proposed approach for *Manoeuvre Selection* can fit with state-of-the-art methods for *Motion Planning*.

***Index Terms* - Decision-Making, Scene Understanding, Uncertainty Handling, Reinforcement Learning, Autonomous Driving**

I. Scene Understanding as Foundation for Decision-Making

Decision-Making for Autonomous Driving requires an accurate and adequate representation of the environment. Raw sensor information cannot be processed as such for reasoning. Instead, traffic situation must first be assessed via *Scene Modelling* and *Semantic Interpretation*. For this purpose, an ontology-based *Environment Model* can be developed.

Three sources of knowledge are fused:

- The information gained by the *Perception* module
- The result of the *Localization and Mapping* module
- The prior knowledge on the road network

This *Environment Model* is combined with the self-representation of the ego vehicle to yield a *Context Model*. In doing so, *Metrical* (also called *Geo-Spatial*), *Topological* and *Temantic* representations are all combined into a hierarchical structure, as shown in Fig. 1.

Examples of features in the *Context Model* include:

- Lateral gap to the centre line (*Geo-Spatial*)
- Number of merging lanes at next intersection (*Topological*)
- Presence of vehicles in the opposing lane (*Semantic*)

Different structures of *Context Model* are designed in [1], [2], [3], [4], [5], [6], [7], [8] and [9]. In the same vein, *Scene Graph* is presented in [10], *Urban Road Situation* in [11], *State Space Assessment* in [12] while [13] introduces a *Compact Semantic State* to generate driving behaviours. Yet, some of them are designed for specific scenarios only.

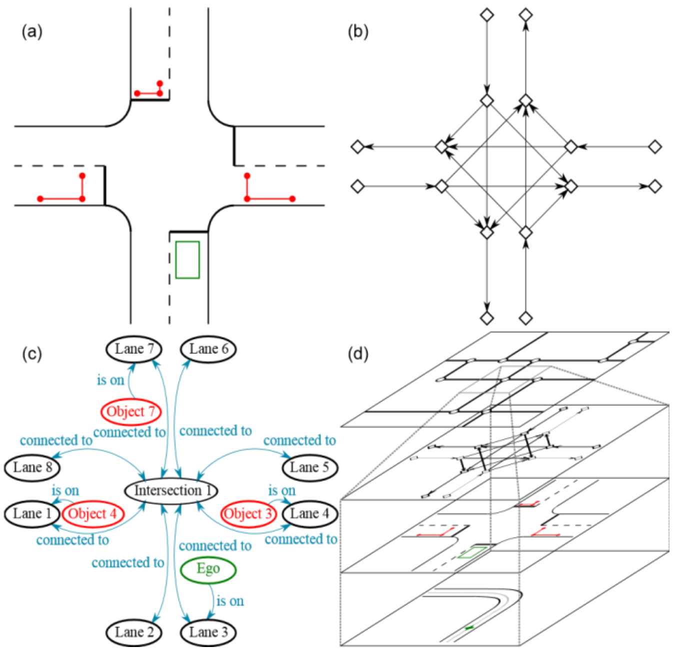

Fig. 1: *Metrical* (a), *Topological* (b) and *Semantic* (c) levels are all combined (d) in a context representation for *Environment Modelling*. The *Semantic* layer is hidden in (d) for readability. Taken from [2].

On the opposite, the presented representation is independent of the road geometry and the number of surrounding vehicles, making such a *Context Model* applicable to multiple urban scenarios.

In real-world driving problems, the agent cannot fully perceive and capture all details of surrounding objects. Therefore, for Autonomous Vehicles to take decisions based on incomplete observations, *Tracking*, *Consistency Check* and *Uncertainty Handling* must be considered.

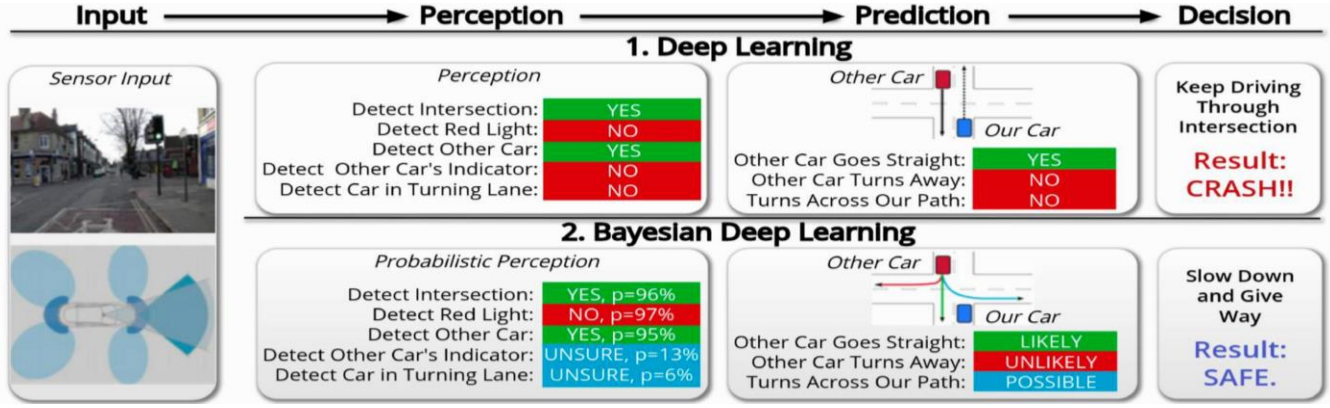

Fig. 2: Cross-module *Bayesian Deep Learning* architecture, illustrating the key benefit of propagating uncertainty throughout the modular pipeline. The other car's turning indicator can hardly be detected on the camera image. Nevertheless, the corresponding detection has a non-zero probability. For this reason, the option *Turns Across Our Path* cannot be ruled out. This causes a more careful manoeuvre to be taken, eventually avoiding collision. Taken from [75]

A. *Capture sequential and temporal information*

A car can disappear from the sensor view for one frame. But this should not radically change the driving decision. Hence *Tracking* of surrounding objects must be organised.

A first option is to use a *Recurrent Neural Network* (RNN), due to its ability to capture long-term dependencies in the sequential information. Applications include function approximation for *Reinforcement Learning* and prediction framework as in [14], [15], [16], [17], [18] and [19].

Another solution consists in equipping the agent with a memory of past observations and actions. Augmenting the state space by stacking consecutive *Contexts* can bring a RNN-like memory effect, as noted in [20], [21], [22], [23] and [24]. This trick can also compensate for the lack of *Markovity* in *Markov Decision Process* (MDP) models which are used to formulate the *Decision-Making* problem.

In addition, this stacking technique enriches *Context Model* with a temporal dimension.

B. *Verify consistency*

So far, no critical thinking has been applied. Detecting a pedestrian running at 80km/h does not raise any scepticism.

To address that and to tackle the lack of information, the context model goes through additional processing blocks for *Situation Assessment* and *Context Augmentation* as in [25]. The problem of uncertainty and plausibility is analysed by a *Consistency and Applicability Checker*.

The model is then enriched with semantic-based intention and motion prediction. Model-based filtering can be used, similar to [26], [27], [28], [29] and [30]. One strategy is to first perform manoeuvre recognition to then apply probabilistic trajectory prediction. Inference of motion intention can be completed using *Hidden Goal Methods* (*Where will the obstacle vehicle go?*) [31], [32] or *Reaction Models* (*How will the obstacle vehicle react?*) [33].

C. *Taking uncertainty into account*

When finally encoding the context model, (d) in Fig. 1 e.g., attention should be paid not to restricts features to point-estimates. Indeed, a knowledge representation such as *"Leading Vehicle is 5.4 m Ahead"* does not use any confidence level (e.g. *+/- 0.5 m at 95%*), yet necessary for robust driving *Decision-Making*. Instead, features should preferably be expressed as statistical distributions. Taking sensor uncertainties into account benefits the process of *Decision-Making* as shown in [34], [35] and illustrated on Fig. 2.

With the *Situational Awareness* offered by this environment representation, efficient *Decision-Making*, especially for *Manoeuvre Selection*, can be now performed.

II. Human Reasoning is neither Geometric nor Numerical but Semantic and Linguistic

While driving, no one says: *"I go 9.14 m at the current speed and then accelerate at a rate of 0.32 m/s²"*? Instead of geometric units and acceleration vectors, driving policies in human language are of a semantic nature, such as *"I follow the vehicle in front of me"* or *"I overtake the car on my right"*. On the other hand, pure numerical methods that do not try to mimic human cognition show computing performance in many fields, e.g. motion optimization.

Therefore, it can be decided to structure the decision part with multiple levels of abstraction. And to combine human concepts like *Driving Tasks* and *Processing Levels* with numerical reasoning, as in [36], [37], [38], [39] and [40].

Such hierarchical structure offers several benefits:

- It allows for reasoning in high-complexity situations and simplifies the handling of semantic information, especially traffic rules, as in [5].
- Based on *Functional Separation*, this hierarchy facilitates verifiability, testing and communication to the driver, while preventing unnecessary complexity.
- Furthermore, it avoids the information-loss of top-down approaches and signal-based architectures by personalizing the representation of the environment for each level. *Manoeuvre Selection* may prefer e.g. the representations (b) and (c) to perform reasoning on the 1D-dimensional abstraction of the lanes and their connectivity. On the opposite, path optimization can use a 2/3D risk map derived from (a) of Fig. 1.

Table I gives an overview of the proposed architecture for *Decision-Making*. To ensure robustness and safety, bidirectional communication occurs between modules as in [25], [41], [42] and similar to the *Closed-Loop Behavioural Policies Selection* of [29] and [43]. For conciseness, these feedback mechanisms are not represented in Table I.

III. 3M Concept: Mission, Manoeuvre and Motion

The block responsible for *Decision-Making*, sometimes called *Planning and Control*, is composed of three modules:

- A *Mission* is instantiated by the *Navigation* module. Based on a user-initiated destination, Dijkstra's algorithm is used to find the shortest way in the road network. The result is a list of *Lane Segments*, named *Route*.
- On the *Tactical* level, the previously detailed *Scene* is processed and augmented with additional information as in [25]. The derived representation then goes for *Situation Assessment.* The next step is *Behavioural Planning*. It entails not only *Manoeuvre Selection* but also planning about how a manoeuvre should be executed to complete the requested *Route*.
- The last block is the *Operational* layer. It is responsible for selecting a *Motion* that complies with the set of target poses and manoeuvres specified by the preceding *Tactical* level, also called *Guidance* level in [25].

The structure for *Behavioural Planning* in the *Tactical* level is further detailed in the following.

To address *Manoeuvre Selection*, a high-level semantic is employed. Different driving *Policy Options* such as *Take-Over*, *Follow Lane* and *Stop at Intersection* are considered. The term *Policy Option* is often used for *Hierarchical Reinforcement Learning* (HRL) as in [44], [45], [46] as well as in [47] to augment concepts of *Primitives* and *Actions* with a multi-level abstraction. It makes the module *Behaviour-Aware*, as described in [48].

The selection process consists of two complementary methods to balance between efficiency and safety:

- *Human-like Decision-Making*, as termed in [49], using *Reinforcement Learning*.
- *Knowledge-based Inference*, i.e. rule-based reasoning.

The main idea is to outsource high-level planning to a learning module and merge it with an algorithm based on strict rules such as *if-then-else* statements. In doing so, it aims at combining the best of both worlds, as detailed in the comparison of [50] on AI-based methods for high-level *Decision-Making*. These two parts are now presented.

A. Reinforcement Learning for POMDP

The problem is formulated as a *Partially Observable Markov Decision Process* (POMDP) to handle the inevitable uncertainty in perception and prediction.

As in [51], two models are learnt depending on whether the car is driving or is stopped:

- The *Sequential Actions* mode determines the most appropriate driving manoeuvre when moving.
- In *Time-to-Go* mode, the agent learns when to depart, as in [52] and [53].

To find the optimal policy, a *Model-Free* approach can be preferred. The reason is that because driving behaviours may be cooperative or adversarial, it is not trivial to model the environment explicitly with all possible future situations, as pointed out in [54].

To reduce the exhaustive exploration in the early stages, parameters can be warmed up with domain knowledge. Either via *Imitation Learning* using human demonstrations as in [55], or similarly to [56] where prior background knowledge, in the form of a value function, helps to reduce the amount of exploration.

The reward function of the POMDP is hand-engineered by a human expert. *Inverse Reinforcement Learning* (IRL), i.e. optimising the reward model to maximise the probability of the expert’s trajectories, can be beneficial as shown in [57], [58]. In addition, IRL offers the possibility to directly integrate learned models into deployable systems, which can straightforwardly be tested and benchmarked against existing hand-crafted reward functions.

Online POMDP solvers such as DESPOT [59], SARSOP, MODIA as well as Adaptive Belief Tree with TAPIR can be used. They have been successfully applied by [11], [60], [53], [61] and [31] for *Decision-Making* under uncertainty for Autonomous Driving. Alternatively, a MDP with a *k-Markov Property*, where the state consists of the last k observations, can approximate a POMDP as explained in [62] and implemented in [56].

TABLE I - Overview of the proposed Hierarchical Structure for Decision-Making in Autonomous Driving

		Level	Role	Task		Generated Object
Scene Understanding	→	Strategic	Navigation	Route Planning	→	Mission
Scene Understanding	→	Tactical	Behavioural Planning	Policy Selection	→	Manoeuvre
Scene Understanding	→	Operational	Stabilization	Trajectory Selection	→	Motion
			↓	↓		
			Manoeuvre Execution	Drive Control		

Functional separation enables to encapsulate the *Driving Policy Selection* such that it is not directly exposed to raw perceptual inputs or low-level vehicle dynamics. In the left-hand column, *Scene Understanding* denotes models coming from the *Environment Modelling* and *Self-Perception* modules. As in [25], these models are adapted to offer different levels of abstraction. Concretely, the *Strategic* level expects a *Road Network and Pose on Map* as input. A *Scene* is processed by the *Tactical* level. Finally, the *Operational* level uses some specific *Features in the States of the Environment and the Ego Vehicle*.

The RL data-driven method offers several advantages:

- First, it allows for complex reasoning. Common sense and intuition are too complex for manual rules. Hence learning from data can be beneficial to address the interactive and intuitive parts of driving that cannot be governed by strict and easy-to-define rules. For instance, human drivers' unpredictable behaviour would be complicated to address only with a rule-based algorithm where a threshold, like the *Time-to-Collision* (TTC), is used at intersections to decide when to go.
- Contrary to pure greedy and reactive approaches that ignore almost all information about driver intent and are overly cautious, RL methods can take full benefits of the time-sequential process evolution. This enables to handle uncertainty, to anticipate and to reason about the long-time horizon effects of immediate actions.
- Furthermore, RL-based methods offer, scalability as well as great generalization capabilities on new and unforeseen scenarios. They do not require any repeated hyperparameter or heuristic tuning for different environments and situations.
- Last but not least, RL techniques applied to Autonomous Driving can improve with experience, as illustrated in [63], using feedbacks of many different safety drivers.

B. *Knowledge-based Inference Model*

Integrating traffic rules and advanced safety considerations in the RL approach is complicated. One solution consists in explicitly separating traffic-rules and safety constraints from the reward function of the POMDP, as developed in [64]. In [65], a technique called *Action-Q-Masking* is implemented. Simply put, hand-coded traffic rules prevent certain *Policy Options* to be taken by imposing hard constraints on the policy. For example, in the lane changing problem, if the ego car is in the left most lane, then taking a left action will result in getting off the road. Therefore, a mask can be put on the Q-value associated with the left action to ensure it is never selected in such a state. Similar to *Masking*, [66] introduces the concept of *Shield* to guarantee safety by enforcing properties expressed in *Temporal Logic* (TL) while learning optimal policies. The *Shield* monitors the actions from the learner and corrects them only if the chosen action causes a violation of the safety and traffic rules specification.

To start with, a rule-based method using TTC heuristics can be implemented, similarly to the concept of *Risk Factor* introduced in [67]. Risk measures such as the *Time-To-Closest-Encounter* (TTCE) or the more general *Survival Analysis* presented in [68] could also be applied.

C. *Benefits of the Fusion*

As pointed by [10], integrating *Domain Knowledge* in form of rules, together with *Reinforcement Learning* reasoning techniques offers several benefits:

- Coupling *Human-Like Cognition* with *Knowledge-based Inference* approaches enables the interpretation of semantically complicated and real-world scenes, while keeping the computational complexity reasonable.
- It relieves the learning process by incorporating prior information about the problem that the agent does not need to learn from scratch through exploration.
- At the same time, hard constraints from the rule-based part guarantee robust functional safety and ensure traffic-rule compliance.
- Finally, it offers ease of testing, interpretability and separation of responsibilities. This is important in the context of safety-critical applications such as Autonomous Driving.

Combining a high-level RL-based policy with a non-learnable low-level rule is recommended in additional recent research papers. For instance in [47] and [69].

A similar concept can be found in [70]. To address *Decision-Making* in a dynamic environment with uncertainties, the authors introduced the PEORL framework (*Planning - Execution - Observation - Reinforcement - Learning*). The idea is to use *Symbolic Planning* for *Option Discovery* with *Hierarchical Reinforcement Learning* (HRL): the agent carries prior knowledge of the dynamic system and utilizes an abstract symbolic representation to generate plans that guide its RL process. By combining RL with pre-defined, manually crafted rules, the PEORL framework offers a rapid policy search and robust symbolic planning. It also provides a learning capability, i.e. it leverages learned experience to enrich symbolic knowledge and improve planning.

Details and results of the here-proposed approach will be more formally documented in future publications.

As in [13], [20], [65], [71], [72] and [73], the open-source SUMO simulator [74] is used to run experiments, as shown on Fig. 3. An in-house developed interface structures the interaction with SUMO in a similar way *Environments* of the *gym* library of OpenAI does. SUMO helps modelling a variety of traffic conditions in different scenarios via the configuration of road networks, signs, pedestrians, vehicles, etc. Uncertainty can also be included in simulation by playing with the distribution of parameters, e.g. start positions, driving styles and sensor noise.

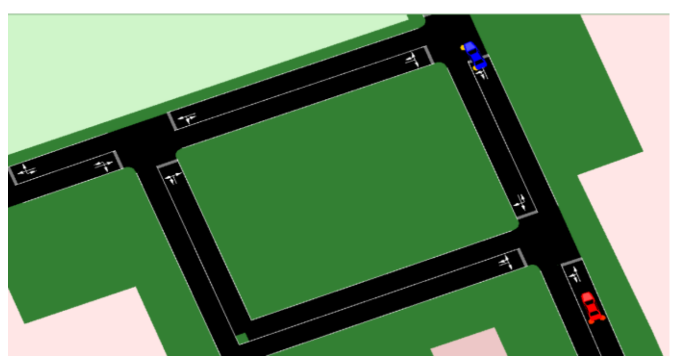

Fig. 3. The open source package SUMO enables to train and test the RL agent on a model of the Munich test area used by ESR Labs AG.

IV. Uncertainty-based Cognition

Regardless of their implementations, *Scene Understanding* and *Decision-Making* must deal with uncertainty. Some elements that could improve the robustness of the proposed approach are now presented.

Not all mistakes are equal: making a mistake confidently can be much more harmful than demonstrating uncertainty about the situation. With the support of Fig. 2, one could ask: "*Who can be satisfied or can trust one algorithm that returns only one type of Manoeuvre to follow?*"

To handle uncertainties, it is preferable for behaviour suggestions to come with probabilities and confidence levels. This is detailed by [75] which explains the key benefit of propagating uncertainty throughout the modular framework using *Bayesian Deep Learning* architectures.

In addition, [72] makes use of *Belief States* to handle uncertainty at urban intersections, while [76] argues that performance metrics traditionally used for *Machine Learning,* such as *Precision* and *Recall*, can be insufficient in Autonomous Driving applications. The authors propose the concept of *Introspection* to assess uncertainty for prediction and intention recognition.

Finally, [77] and [34] distinguish two different types of uncertainty and investigate their implications in the context of Autonomous Driving:

A. Aleatoric Uncertainty

On the one hand, *Aleatoric Uncertainty* is of statistical origin and applies especially for sensors. The noise inherent to measurement is often modelled by introducing probability densities, such as Gaussian distributions in the case of Kalman filters for instance.

B. Epistemic Uncertainty

On the other hand, *Epistemic Uncertainty* describes missing or incomplete information such as occluded areas. [78] develops a generic approach for tactical *Decision-Making* to deal with stochastic behaviour of other road users and limited perception capabilities. Nevertheless, methods that explicitly represent *Epistemic Uncertainty* are not very common in the field of Autonomous Driving.

Other approaches aim at handling these two kinds of uncertainty as well as formulating safe and reliable predictions.

- *Hidden Markov Models* ([79], [80], [81], [82]), *Dynamic Bayesian Networks* ([83], [84], [85], [86], [87]) and *Hybrid Gaussian Mixture Models* ([88]) are among the most commonly used solutions to deal with incomplete sensor information.
- [84] proposes new heuristics to make decisions and predict trajectories. For instance, the *Time-To-Critical-Collision-Probability* (TTCCP) should substitute traditional *Time-To-X* metrics (with *X* e.g. *Intersection or Collision*) which cause over-conservative decisions.
- In the same vein, it seems useful to compute, for each contemplated manoeuvre, a *Probability of Success,* as introduced by [40], and use methods discussed in [89] to rigorously quantify uncertainty.

Such additional information about uncertainty could enrich the *Behavioural Planning* presented before.

The proposed *Tactical* module yields two target elements for *Poses* (i.e. position and orientation) and *Manoeuvres*:

- A *Manoeuvre Decision*. Its intelligible format makes it convenient for interpretation, e.g. *Follow Leader with Care for Lateral Distance*. It is then converted to a numerical representation. The metric encoding is used to adjust parameters. In this example, the *Lateral Size of Ego Car* can be increased in the cost function used for trajectory optimization.
- A *Predefined Path*. It is engineered from static road knowledge and represents the desired driving paths under normal circumstances, as in [90]. It helps to guide the motion search and specifies the pose to reach.

These two outputs are processed by a *Geometric Path Planner* module, responsible for *Motion Selection*.

V. Motion Selection - Both Exploration and Exploitation

The *Geometric Path Planner* generates drivable and collision-free trajectories for a short-horizon based on the requested sub-goals. It proceeds in three steps: first *Trajectory Generation*. Then *Trajectory Assessment and Selection*. Finally, *Trajectory Transformation* to forward readable objects to the low-level controller. It takes advantage of *Predefined Trajectories* but can also cope with unknown routes. Different methods are coupled such as:

- *Sampling-based* approaches. They enable to find admissible trajectories with or without prior proposal.
- *Optimization-based* algorithms. They are applied to ensure safety and comfort.

The *Any-Time* property [91] can ensure that a valid and safe solution is returned even if the computation is interrupted. In other words, given more time or computational power, the *Motion* output will improve.

The resulting list of waypoints that defines the *Motion* is stored as a *Path-to-Drive* for the next iteration and sent to the *Drive Control* module responsible for *Manoeuvre Execution*.

VI. Conclusion - The Brain

With the benefits of combining learning methods with traditional inference approaches, the *Scene Understanding & Decision-Making* hybrid module completes several tasks.

It understands complex and dynamic scenes. Then it thinks and elaborates informed driving decisions in an efficient, safe, rule-abiding, human-like and real-time way.

As a *Brain*, it connects the *Eyes* to the *Muscles* of the Autonomous Vehicle.

FUTURE WORK

The author intends to pursue the approach, and direct further research to these topics.

REFERENCES

[1] X. Geng, H. Liang, B. Yu, P. Zhao, L. He, and R. Huang, "A Scenario-Adaptive Driving Behavior Prediction Approach to Urban Autonomous Driving," 2017.

[2] S. Ulbrich, T. Nothdurft, M. Maurer, and P. Hecker, "Graph-Based Context Representation , Environment Modeling and Information Aggregation for Automated Driving," 2014.

[3] R. Kohlhaas, T. Bittner, T. Schamm, and J. Z. Marius, "Semantic State Space for High-level Maneuver Planning in Structured Traffic Scenes," 2014.

[4] R. Kohlhaas, D. Hammann, T. Schamm, and J. M. Zollner, "Planning of High-Level Maneuver Sequences on Semantic State Spaces," 2015.

[5] C. Chen, A. Gaschler, M. Rickert, and A. Knoll, "Task Planning for Highly Automated Driving," 2015.

[6] J. Rieken, R. Matthaei, and M. Maurer, "Toward Perception-Driven Urban Environment Modeling for Automated Road Vehicles," 2015.

[7] M. Hülsen, J. M. Zöllner, and C. Weiss, "Traffic intersection situation description ontology for advanced driver assistance," 2011.

[8] L. Yu, X. Shao, Y. Wei, and K. Zhou, "Intelligent Land Vehicle Model Transfer Trajectory Planning Method Based on Deep Reinforcement Learning," 2018.

[9] R. Regele, "Using ontology-based traffic models for more efficient decision making of autonomous vehicles," 2008.

[10] L. Kunze, T. Bruls, T. Suleymanov, and P. Newman, "Reading between the Lanes : Road Layout Reconstruction from Partially Segmented Scenes," 2018.

[11] W. Liu, S. W. Kim, S. Pendleton, and M. H. Ang, "Situation-aware decision making for autonomous driving on urban road using online POMDP," 2015.

[12] M. Ardelt, P. Waldmann, F. Homm, and N. Kaempchen, "Strategic decision-making process in advanced driver assistance systems," 2010.

[13] P. Wolf, K. Kurzer, T. Wingert, F. Kuhnt, and J. Z. Marius, "Adaptive Behavior Generation for Autonomous Driving using Deep Reinforcement Learning with Compact Semantic States," 2018.

[14] D. Steckelmacher and P. Vrancx, "An Empirical Comparison of Neural Architectures for Reinforcement Learning in Partially Observable Environments," 2015.

[15] J. Morton, T. A. Wheeler, and M. J. Kochenderfer, "Analysis of Recurrent Neural Networks for Probabilistic Modeling of Driver Behavior," 2016.

[16] O. Scheel, L. Schwarz, N. Navab, and F. Tombari, "Situation Assessment for Planning Lane Changes: Combining Recurrent Models and Prediction," 2018.

[17] A. El Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep Reinforcement Learning framework for Autonomous Driving," 2017.

[18] X. Huang, J. Sun, and J. Sun, "A car-following model considering asymmetric driving behavior based on long short-term memory neural networks," 2018.

[19] N. Deo and M. M. Trivedi, "Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs," 2018.

[20] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning," 2017.

[21] J. Liu, P. Hou, L. Mu, Y. Yu, and C. Huang, "Elements of Effective Deep Reinforcement Learning towards Tactical Driving Decision Making," 2017.

[22] M. A. F. Khan and O. H. Elibol, "Car Racing using Reinforcement Learning," 2016.

[23] M. Legrand, "Deep Reinforcement Learning for Simulated Autonomous Vehicle Control," 2016.

[24] S. W. Jang, J. Min, and C. Lee, "Reinforcement Car Racing with A3C," 2016.

[25] S. Ulbrich *et al.*, "Towards a Functional System Architecture for Automated Vehicles," 2017.

[26] Y. Hu, W. Zhan, and M. Tomizuka, "Probabilistic Prediction of Vehicle Semantic Intention and Motion," 2018.

[27] J. Schlechtriemen, F. Wirthmueller, A. Wedel, G. Breuel, and K. D. Kuhnert, "When will it change the lane? A probabilistic regression approach for rarely occurring events," 2015.

[28] B. Zhou, W. Schwarting, D. Rus, and J. Alonso-mora, "Joint Multi-Policy Behavior Estimation and Receding-Horizon Trajectory Planning for Automated Urban Driving," 2018.

[29] E. Galceran, A. Cunningham, R. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment," 2017.

[30] C. Dong, J. M. Dolan, and B. Litkouhi, "Interactive Ramp Merging Planning in Autonomous Driving : Multi-Merging Leading PGM (MML-PGM)," 2017.

[31] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated Driving in Uncertain Environments: Planning with Interaction and Uncertain Maneuver Prediction," 2018.

[32] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss, "Interactive Scene Prediction for Automotive Applications," 2013.

[33] L. Wei, "Planning under Uncertainties for Autonomous Driving on Urban Road," 2016.

[34] A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?," 2017.

[35] Y. Gal, "Uncertainty in Deep Learning," 2016.

[36] A. Furda and L. Vlacic, "Enabling safe autonomous driving in real-world city traffic using Multiple Criteria decision making," 2011.

[37] R. Matthaei and M. Maurer, "Autonomous driving - A top-down-approach," 2015.

[38] C. Paxton, V. Raman, G. D. Hager, and M. Kobilarov, "Combining Neural Networks and Tree Search for Task and Motion Planning in Challenging Environments," 2017.

[39] L. Li, K. Ota, and M. Dong, "Human-Like Driving : Empirical Decision-Making System for Autonomous Vehicles," 2018.

[40] C. Menendez-Romero, F. Winkler, C. Dornhege, and W. Burgard, "Maneuver planning for highly automated vehicles," 2017.

[41] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," 2016.

[42] Ö. Sahin Tas, F. Kuhnt, J. M. Zöllner, and C. Stiller, "Functional System Architectures towards Fully Automated Driving," 2016.

[43] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, "MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving," 2015.

[44] Y. Wang, H. He, and C. Sun, "Learning to Navigate through Complex Dynamic Environment with Modular Deep Reinforcement Learning," 2018.

[45] R. Liaw, S. Krishnan, A. Garg, D. Crankshaw, J. E. Gonzalez, and K. Goldberg, "Composing Meta-Policies for Autonomous Driving Using Hierarchical Deep Reinforcement Learning," 2017.

[46] L. P. Tuyen, N. A. Vien, T. Chung, C. Science, and S. Korea, "Deep Hierarchical Reinforcement Learning Algorithm in Partially Observable Markov Decision Processes," 2018.

[47] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving," 2016.

[48] W. Schwarting, J. Alonso-mora, and D. Rus, "Survey on Planning and Decision-Making for Autonomous Vehicles," 2018.

[49] J. Rasmussen, "Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models," 1983.

[50] L. Claussmann, M. Revilloud, S. Glaser, and D. Gruyer, "A study on ai-based approaches for high-level decision making in highway autonomous driving," 2017.

[51] D. Isele, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating Intersections with Autonomous Vehicles using Deep Reinforcement Learning," 2017.

[52] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A POMDP approach," 2014.

[53] V. Sezer, T. Bandyopadhyay, D. Rus, E. Frazzoli, and D. Hsu, "Towards autonomous navigation of unsignalized intersections under uncertainty of human driver intent," 2015.

[54] P. Wang, C. Chan, and A. La De Fortelle, "A Reinforcement Learning Based Approach for Automated Lane Change Maneuvers," 2018.

[55] X. Liang, T. Wang, L. Yang, and E. P. Xing, "CIRL: Controllable Imitative Reinforcement Learning for Vision-based Self-driving," 2018.

[56] M. Bouton, K. Julian, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Utility Decomposition with Deep Corrections for Scalable Planning under Uncertainty," 2018.

[57] M. Wulfmeier, D. Z. Wang, and I. Posner, "Watch This: Scalable Cost-Function Learning for Path Planning in Urban Environments," 2016.

[58] M. Wulfmeier, "On machine learning and structure for driverless cars mobile robots," 2017.

[59] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "DESPOT : Online POMDP Planning with Regularization," 2017.

[60] C. Shaojun, "Online POMDP Planning for Vehicle Navigation in Densely Populated Area," 2014.

[61] K. H. Wray, S. J. Witwicki, and S. Zilberstein, "Online decision-making for scalable autonomous systems," 2017.

[62] C.-J. Hoel, K. Wolff, and L. Laine, "Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning," 2018.

[63] A. Kendall *et al.*, "Learning to Drive in a Day," 2018.

[64] M. El Chamie, Y. Yu, B. Acikmese, and M. Ono, "Controlled Markov Processes with Safety State Constraints," 2018.

[65] M. Mukadam, A. Cosgun, A. Nakhaei, and K. Fujimura, "Tactical Decision Making for Lane Changing with Deep Reinforcement Learning," 2017.

[66] M. Alshiekh, R. Bloem, and K. Bettina, "Safe Reinforcement Learning via Shielding," 2017.

[67] M. Bahram, A. Wolf, M. Aeberhard, and D. Wollherr, "A prediction-based reactive driving strategy for highly automated driving function on freeways," 2014.

[68] J. Eggert and T. Puphal, "Continuous Risk Measures for Driving Support," 2018.

[69] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a Formal Model of Safe and Scalable Self-driving Cars," 2017.

[70] F. Yang, D. Lyu, B. Liu, and S. Gustafson, "PEORL: Integrating Symbolic Planning and Hierarchical Reinforcement Learning for Robust Decision-Making," 2018.

[71] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian,

and K. Fujimura, “Navigating Occluded Intersections with Autonomous Vehicles using Deep Reinforcement Learning,” 2018.
[72] M. Bouton, A. Cosgun, and M. J. Kochenderfer, “Belief state planning for autonomously navigating urban intersections,” 2017.
[73] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, “Flow: Architecture and Benchmarking for Reinforcement Learning in Traffic Control,” 2017.
[74] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent Development and Applications of SUMO - Simulation of Urban MObility,” 2012.
[75] R. Mcallister *et al.*, “Concrete Problems for Autonomous Vehicle Safety : Advantages of Bayesian Deep Learning,” 2017.
[76] H. Grimmett, R. Triebel, R. Paul, and I. Posner, “Introspective classification for robot perception,” 2016.
[77] M. Nolte, S. Ernst, J. Richelmann, and M. Maurer, “Representing the Unkown - Impact of Uncertainty on the Interaction between Decision Making and Trajectory Generation,” 2018.
[78] S. Brechtel, T. Gindele, and R. Dillmann, “Probabilistic Decision-Making under Uncertainty for Autonomous Driving using Continuous POMDPs,” 2015.
[79] H. Berndt and K. Dietmayer, “Driver intention inference with vehicle onboard sensors,” 2009.
[80] D. Meyer-delius, “Probabilistic Situation Recognition for Vehicular Traffic Scenarios,” 2009.
[81] W. Song, G. Xiong, and H. Chen, “Intention-Aware Autonomous Driving Decision-Making in an Uncontrolled Intersection,” 2016.
[82] T. Streubel and K. H. Hoffmann, “Prediction of driver intended path at intersections,” 2014.
[83] J. R. Chandiramani, “Decision Making under Uncertainty for Automated Vehicles in Urban Situations,” 2017.
[84] M. Schreier, “Bayesian environment representation, prediction, and criticality assessment for driver assistance systems,” 2017.
[85] T. Gindele, S. Brechtel, and R. Dillmann, “A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments,” 2010.
[86] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, “Exploiting map information for driver intention estimation at road intersections,” 2011.
[87] J. F. P. Kooij, F. Flohr, E. A. I. Pool, and D. M. Gavrila, “Context-based Path Prediction for Targets with Switching Dynamics,” 2018.
[88] E. Galceran, E. Olson, and R. M. Eustice, “Augmented vehicle tracking under occlusions for decision-making in autonomous driving,” 2015.
[89] W. Zhan, C. Liu, C. Y. Chan, and M. Tomizuka, “A non-conservatively defensive strategy for urban autonomous driving,” 2016.
[90] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, “A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles,” 2016.
[91] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, “Anytime motion planning using the RRT,” 2011.

Simon Chauvin studied Mechanical Engineering and Embedded Systems at Ecole Centrale Paris, France, and Imperial College London, UK. He received a Master of Engineering in 2017. He is currently working at ESR Labs AG, Munich, Germany, as a Machine Learning Engineer. His research interests include Scene Understanding, Cognition and Behavioural Planning under uncertainty for Autonomous Driving, as well as Markov Decision Processes and Reinforcement Learning.