Dream-like simulation abilities for automated cars

DREAMS4CARS

Grant Agreement No. 731593

Deliverable:	D3.3. – Report on the simulation system (public version)
Dissemination level:	PU – Public
Delivery date:	29/October/2019
Status:	Final

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731593

Deliverable Title	Report on the simulation system (public version)		
WP number and title	WP3 Dream-like states		
Lead Editor	Mauro Da Lio UNITN		
Contributors			
Creation Date	29/October/2019	**Version number**	1.0
Deliverable Due Date	31/October/2019	**Actual Delivery Date**	29/October/2019
Nature of deliverable	**X**	R - Report	
Dissemination Level/ Audience	**X**	PU – Public, fully open	

Version	Date	Modified by	Comments
1.0	29/October/2019	Mauro Da Lio	Public version created from D3.2.

Executive Summary

This is the open (public) version of deliverable D3.2 Simulation system (release 2). We provide a short description of the main findings and conclusions of WP3 and some parts of D3.2 itself.

There are other parts of D3.2 that are not published because of two main reasons: a) they are suited for possible technical-economical exploitations and/or b) they are still confidential in order to avoid anticipated disclosure of the contents of future publications.

We provide the full Table of Contents of D3.2 (with omitted sections) in order to promote the contents to raise interests for possible adopters in case of technical-economical exploitations.

The complete list of references is also included.

Table of Contents

List of Diagrams

List of Tables

1 Summary of findings

Deliverable D3.2 (Simulation system) represents the main research findings of Dreams4car, describing a process for bootstrapping sensorimotor systems based on *learning models of the wold* (both models that predict the effect of actions and models that predict events) *that are then manipulated* offline to *synthetize* action strategies (see Figure 1).

This process is described in section 3. It starts with the supervised learning of vehicle dynamics “forward” (or predictive models) from data. We here introduce an efficient architecture for the networks that makes the models explainable (thus solving the black box issue that would otherwise prevent adoption of neural network for motion control due to liability issues and the ISO 26262 mandatory requirement). We show how this architecture allows disentangling independent causes (weak superposition effect) and how it is very similar to cerebellar filters. Stochastic forward models are then introduced resorting to the bootstrapping technique which does not make any assumption about the distributions and correlations of data. Progressive refinement of models via lifelong learning (e.g., learning nonlinearities on top of linear models) is also explained. A particular form of overfitting (high-frequency spectral overfitting) is studied and methods of regularization are provided. Statistical methods to compare predictions of learned forward models are finally introduced. This section is partly included in the public version D3.3 of the deliverable.

In section 3.2 the learning of inverse models is introduced showing two possible approaches: from data (supervised learning by swapping input and output) and via the first form of episodic simulations (unsupervised learning). We show that the latter is generally more robust, and it may be used to train inverse model for robust predictive control (an aspect that is still a research topic for traditional MPC).

In section 3.3 we step to a higher level of motor control introducing the notion “short-cuts” in the simulation paths. This is a form of learning neural network abstractions that allows making prediction of action outcomes without needing to simulate the entire action in details. It can be regarded as the transition between embodied (detailed) and episodic (abstract) simulations. This in turns allows to progressively build more and more abstract, *fast and efficient* simulation blocks for simulations of further higher levels, ending with the tools for accelerating Reinforcement Learning.

We finally end up with the learning of action values, i.e., the salience stored in the “motor cortex” and we show how it is functionally equivalent to the notion of reward in Reinforcement Learning, except that it is obtained via a synthesis process that manipulates learned models of the world, rather than via trial and error exploration (the latter is RL).

In section 4 Reinforcement Learning applications are introduced. The choice of safe speed (as the most important behavioural choice for safe driving) in studied in details for the case of pedestrians possibly crossing the road, obtaining a network that interacts with the lower-levels of the agent via setting recommended safe speed. Open issues related to RL are also discussed (RL in itself is not the research focus of Dreams4Cars).

We introduce also possible research lines for the future in section 5; in particular how the sensorimotor system here developed can be integrated with self-organizing perception system to form a whole. We take inspiration from the CDZ hypothesis of Damasio and show some proof of concept with implementations based on Variational Autoencoders. This part is included in the deliverable as already published.

Across all the document we make as many efforts as possible to compare the methods here with more or less traditional alternatives and point many advantages: so, section 3.1.5 evaluates the performance of forward models networks, section 3.1.9 deals with statistical comparisons of forward models, section 3.3.4 compare stochastic motor models to the ones in the literature, section 3.3.6 compares to Optimal Control, section 3.3.7.2 makes an internal comparison between the core bootstrapping approach of section 3 and Reinforcement Learning. Section 4 also presents some qualitative comparisons of the RL implementation within similar ones in the literature. Worth to be noted is the fact that here RL creates networks that are built on top of an agent the can operate on the real world, whereas the literature examples are for simulated worlds and not directly transferable to the real world (transfer here is permitted by acting with high-level directives on an agent that can operate on the real world by itself).

1.1 Objectives of D3.2

The two primary goals of the “dreaming” machinery are creating *useful* simulation scenarios (section 2) and *learning* from the simulations (sections 3 and 4).

This deliverable describes the methods that have been developed by Dreams4Cars (D4C). Together with D2.2 (the Agent implementation with reconfiguration abilities) and D1.2 (the system architecture) constitutes the main findings of Dreams4Cars.

This deliverable presents an organized description of the methods that have been developed. A limited number of examples are given in this document in order not to disturb the main narrative line (when necessary we will point to examples given in version 1 of this deliverable, in published papers and/or we will point examples in deliverable D5.3; this is similar to the organization given to D2.2 for WP2).

More examples with the quantification of the improvements that D4C technology produce for automated driving will be given in D5.3, D1.4 and D5.4 instead.

1.2 Embodied and episodic simulations

One of the main findings of D3.1 was the introduction of two distinct forms of simulations termed as *embodied* and *episodic* simulations. The existence of two forms is supported by psychological, behavioural and biological findings [1]. In brief embodied simulations are associated with fast, unconscious, rigid, short processing; while episodic simulations are related to controlled, flexible and longer-term processing [1].

The development and learning of embodied and episodic simulations will involve diverse learning mechanisms and while interplay between the two is an open question it is possible to outline some suggestions on how this is achieved in the human brain. In the simplest case, simulation can be described as covert actions generating predictions of the sensory effects if those actions had they been executed. The predicted sensory effects (stimulus S) should then be able to generate new possible covert actions (response R) and so on. Thus, there is a need to learn both the association between S and R (so called procedural predictions) and the association between R and S (so called *declarative* predictions) [2]. While procedural predictions can be learnt by different learning systems in the brain depending on the level of granularity of the prediction [3], perhaps the most common example would be supervised learning in the cerebellum or reinforcement learning by the basal ganglia [2]. Declarative predictions on the other hand seems mainly related to different types of cortical learning pathways (generally thought of as being instances of unsupervised learning). Doya [4] provides an accessible overview of how the common types of learning in artificial neural networks relates to learning in different parts of the brain.

Since our agent (D2.2) includes loops that learns forward models (cerebellum, D2.2 section 2.4) and mimic back signalling in the frontal cortex (the subsumption architecture, D2.2 section 2.2.5) and in the dorsal stream (the convergent divergent structure, D2.2 section 2.1.5) we can exploit these artificial structures to implement the two forms (declarative and procedural) of simulation and learning.

There are two different approaches:

1) **Bottom up approach**. Learning proceeds bottom up (see section 3 and Figure 1), starting with the learning of forward models and gradually bootstrapping more and more complex motor abilities (motor primitive units, primitive chains and action sequences). Simple episodes may be used/created to set the training goals for each level of competence without needing the entire simulation environment (OpenDS is not necessary except, in some cases, for the very last level).
2) **Top down approach**. Learning proceeds via instantiation of putative goals in the Logical Reasoning Module and produces the biases of the high-level action selection via reinforcement learning. The whole simulation environment is required to insatiate a wide variety of episodes (section 4).

In any case there is some overlap between the two approaches, because 1) can be extended to (very effectively) cover situations that could be generated with 2) top down.

2 Generation of episodes

This section deals with the problem of how to create useful situations for dreaming.

2.1 Level specific episodes

Within a subsumption architecture, sensorimotor abilities are organized in levels of increasing complexity (see Figure 1, section 3).

Every level, except level 1 (Figure 1), may be engaged in offline simulations. For example, at level 2 a sequence of hypothetical motor commands may be used to predict the consequences of those commands (direct problem). Such predictions use the forward model learned at level 1. In turn, this prediction ability may be used to solve inverse problems too: for example, finding the motor commands that produce a desired trajectory (section 0). Similar considerations can be developed for each level. For example, at level 3 one might conceive the problem of finding convenient trajectories to reach hypothetical target states (a level 3 inverse problem) and using level 2 findings to solve it.

We will deal in details with the solution of these inverse problems in sections 3 and 4 (as they form the core of learning). Here we note that, in order to carry out simulations, two ingredients are necessary:

1) a level specific prediction model (which may be the result of training the immediately bottom layer) and
2) a level specific large number of hypothetical goals.

The latter are used to train the level-specific inverse models (which are neural networks). We also note at this point that such inverse problems have close similarities with various forms of optimal control: for example, a network that solves the inverse problem of level 2 is functionally equivalent to Model Predictive Control (MPC) with the difference that the network operates on a learned (rather than an engineered) model and with some operative advantages that will be discussed later. Similarly, the inverse problems at levels 3-4 is equivalent to an Optimal Control Problem (but on learned dynamics) and the inverse problems at levels 4-5 are equivalent to Reinforcement Learning (but with the advantage of having previously constructed a vocabulary or actions).

Hence, to summarize, depending on the *level at which simulations occur*, episodes have correspondingly different nature and complexity. To give just two examples, episodes may range from the generation of complex hypothetical driving situations at the very highest level (e.g., pedestrians possibly crossing the road – see section 0) or they may be very simple target motor tasks at the lowest level (e.g., target trajectories for learning predictive control – see section 0 or 3.3).

With the above clarifications, the objectives of this section can now be stated as dealing with how to generate (at each level) episodes to be used for training of the level-specific inverse models.

2.2 Genetic Operators

The notion of genetic operators was described in D3.1 with the motivation to be able to facilitate the learning of safe driving by providing previously unseen and possibly useful scenarios. The motivation of using techniques inspired by genetic algorithms for the episodic generator is the basic idea of recombining and mutating the DNA of two parents to create a child with other properties than its parents. Thus, while genetic algorithms are usually seen as search method for better solutions, the main motivation for its use here is the ability to generate new distributions of target/environment states that is applicable to all levels of the subsumption architecture as shown with the labels “episodes” in Figure 1.

In brief, the crossover operator enables the system to simulate new traffic contexts. The mutation operator enables the system to perturb the traffic configuration in a particular context. The selection operators are used to control the complexity of the traffic environment.

The scenario generation mechanisms and toolchain for OpenDS is described in D3.1 and has been, for this deliverable, updated with pedestrian models (see Section 0) as well as a integrated it with the top-level LRM architecture (D.2.2).

The genetic operator concept has now been applied within both the bottom up part of the system (section 3) and the top down part of the system (sction 4) and the level specific applications is described there.

3 Bootstrapping hierarchical motor abilities via embodied and episodic simulations.

Figure 1 shows the bottom up approach for the learning of hierarchical motor abilities via offline simulations.

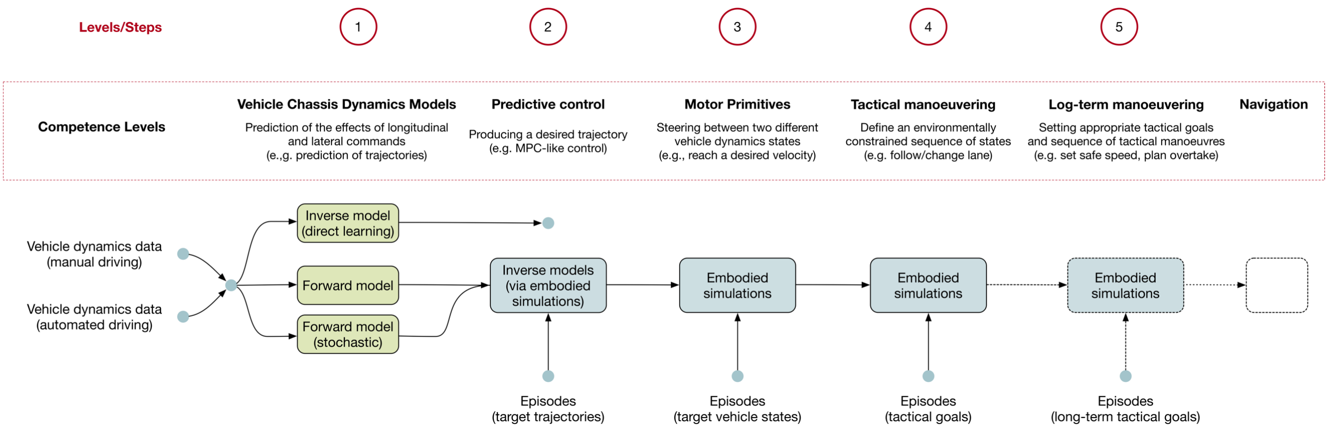

Figure 1: Bootstrapping hierarchical motor abilities proceeds bottom-up.

The process begins with the learning of forward models for the vehicle dynamics (level 1), which become the building blocks for episodic simulations (levels 2-5) that gradually bootstraps layers of competence of increasing complexity.

At every hierarchical level, the abilities learned at the previous levels may be exploited in a way that simplify the learning task. For example, learning tactical manoeuvring (level 4) may exploit the predictive models of level 3 (motor primitives), without necessarily carry the simulation at the lowest level of the chassis dynamics (level 1).

In other words, at every level, a level-specific forward model is trained using the previous level forward model in combination with level specific episodes.

At the lowest levels, episodes are quite simple and do not require the driving simulation environment (nonetheless the process requires episodes in combination with embodied simulations). The highest levels overlap with the competences that can be learned with the top down approach described in the next chapter.

At the end of the bootstrapping process a subsumption architecture is obtained. It can be further extended/improved/refined as new data (and new episodes) become available in a lifelong fashion.

At every level, episodes are created with a twofold goal:

a) to augment the training data set and
b) to generate validation sets.

The latter, in turn, may be used to detect situations calling for new experimental data. This way the agent not only learns behaviours, but it also becomes “aware” of the limitations in the training domain and (especially if a stochastic forward model is used) becomes “aware” of the uncertainties of desired behaviours and can, consequently, choose behaviours for which the uncertainties are sufficiently bounded (for example see section 0).

3.1 Learning forward models (level 1)

Learning forward models is the necessary first step to create the simulation blocks for embodied, and later episodic, simulations. This corresponds to the writing of the mathematical model equations in traditional (human-directed) engineering approach.

There may be, in principle, several machine learning frameworks suited for forward model implementation (for example, in the original Dreams4Cars proposal Locally Weighted Projection Regression (LWPR) were initially mentioned as a possible implementation).

The machine learning framework used by Dream4Cars is Neural Networks (and Deep Learning when necessary) because of several reasons:

a) mature neural network frameworks are readily available and relatively easy to use;
b) as will be shown, neural network frameworks provide a uniform modelling tool across all the layers of the hierarchical architecture (Figure 1), i.e., the same neural network framework can be used to create predictive and inverse models specific of each level;
c) albeit artificial neural networks are different than natural neural network in many aspects, they are similar enough that implementation of biological principles is simplified (and Dream4Cars uses significant biological inspiration).

3.1.1 Cerebellar filter principles

Figure 2 shows a schematic representation of a cerebellar filter, adapted from ([2], Fig.3), used to introduce the principles that are borrowed by the artificial forward model networks.

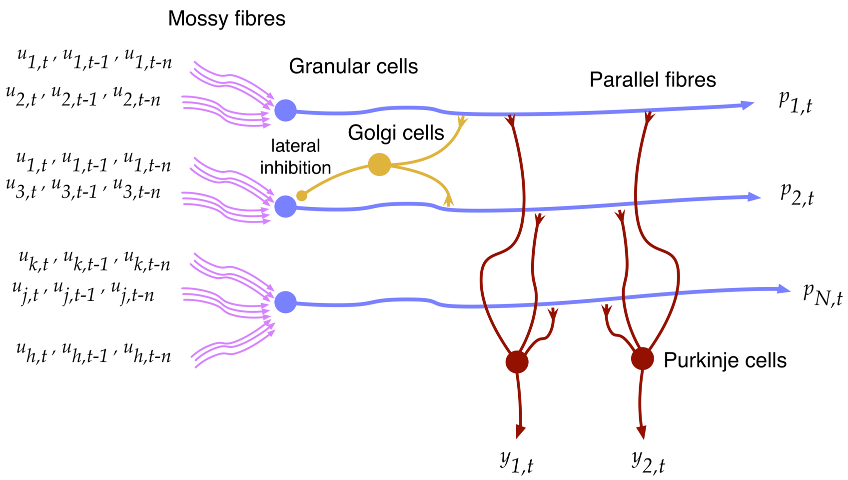

Figure 2: Architecture of the cerebellar filters.

The output of one filter at time t is produced by one Purkinje cell as a (linear) combination of the signals in the parallel fibres that synapse on the cell (e.g., $y_{1,t} = w_1\, p_{1,t} + w_2\, p_{2,t} + \cdots + w_N\, p_{N,t}$)[1], where N may be as large as 1 million fibres. Each parallel fibre is the axon of one Granular cell, which receives input signals from the Mossy fibres. Mossy fibres may carry the same signal sampled at different past times (delayed copies of that signal) such as, e.g., $u_{1,t}$, $u_{1,t-1}$, $u_{1,t-N}$ as well as signals of different types such as, e.g., $u_1, u_2, \dots, u_k, \dots$. These signals may have sensory and cortical origin, or they may be copies of the motor commands.

Each individual granular cell receives a limited number of inputs (it does not receive all the types of signals $u_1, u_2, \dots, u_k$). Hence the signals p_i represent *elementary effects* that are *superimposed by the Purkinje cells*. The Granular cells, individually, operate as detectors of "relatively simple contexts" [2] and their output may be written as $p_{i,t} = G_i(u_j)$, here j belongs to a subset of all possible u_{k,t_i}. Functions $G_i(.)$, are also called "basis" functions [5], stressing the functional fact that the output of one filter (one Purkinje cell) is a linear combination of a large number of "basis functions" operating as *context detectors*.

Finally, the Golgi cells, by inhibiting individual granular cells output, have the practical consequence of realizing piecewise output maps (some p_i being ignored in particular contexts p_j).

[1] Climbing fibres from the inferior Olive (not shown for clarity) carry a teaching signal that alter (all) the synaptic weights, effectively producing a form of supervised learning. However, the form of supervised learning used in the artificial network is based on classical backpropagation.

3.1.2 Artificial implementations

Figure 3 shows an example artificial neural network that was used in [6] to learn the longitudinal dynamics of a vehicle. The network structure resembles the cerebellar filters in the sense that blocks labelled 1,..,4, corresponding to the mossy fibres, individually receive only a subset of the input signals and learns the individual effect of air drag (1), brake pressure (2), road slope (3) and engine force (4), which are lastly superimposed at the final summation block.

Furthermore, branch 4 implements the equivalent of Golgi cell lateral inhibition: 8 parallel channels individually predict the acceleration for each gear. The gear channel G_k (the gear context) inhibits all outputs fibres except that corresponding to the engaged gear. In this way a nonlinear model that predicts the vehicle acceleration given the engine torque and gear is realized.

Channels 2, 3, and 4, except for the gear selection, are designed with linear layers (linear basis functions $G(.)$) which somehow restrict the generalization ability of the network on one side. On the other side, linear sub-models for brake (2), slope (3) and engine (4) effects, means that the weights of the neural network layers are *interpretable* as the (underlying) linear impulse response of the system.

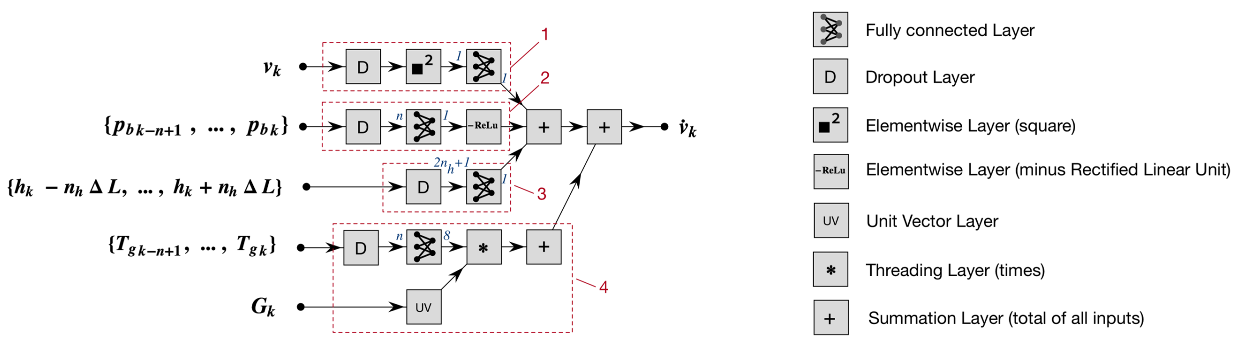

Figure 3: Example of artificial neural network architecture for learning forward models.

3.1.3 Interpretability of the networks

This section is omitted (confidential part)

3.1.4 Modelling nonlinearities

This section is omitted (confidential part)

3.1.5 Performance evaluation

Two questions concerning the accuracy of neural network implementations of forward models have been considered:

- How do neural network forward models compare to more traditional ones, such as, for example, parametric models based on ordinary differential equations?
- How does the biological architecture compare to other possible neural network architectures?

3.1.5.1 How do neural network forward models compare to more traditional ones, such as, for example, parametric models based on ordinary differential equations?

The first question has been studied in paper [6] where the longitudinal dynamics model given in Figure 3 has been compared to a) a commonly used analytical model for the longitudinal dynamics and to b) a state space data-driven black-box model. The three models were compared on two distinct datasets: one from the FP7 Interactive project (a Lancia Delta on a 50 km route with mixed types of roads) and another collected with the

Renegade vehicle in mixed roads with separate training, validation and test routes (for the Lancia Delta two parts of the same route were used for training and validation). This study found that the neural network performed consistently better than the state space model, which in turn was better than the parametric analytical model.

The following table summarizes the findings, reporting the coefficient of determination (R^2) for the three models.

On the Lancia delta, the three models perform fair (the Analytical model) to excellent. The R^2 on the validation set exceeds the training set (contrary to expectations) because the validation set had less stop and go events, which are the most difficult to model.

On the Jeep Renegade the performance of the Analytical model drops. Also, the State Space model witnesses a degradation of modelling capability. Instead the Neural Network remains very good. The degradation in the Renegade was found to depend on a noisier engine torque signal (this signal is the engine net delivered torque estimated by the engine control unit, which was different in the two cars).

The reason for the analytical model to underperform when compared to the neural network is its excessive rigidity. One wold expect the equations of the human engineered model to be very accurate, but in practice the engineer made a number of modelling assumptions and simplifications when developing the analytical model (see discussion in the paper), resulting in a strongly biased model which would perform well if the assumption were satisfied. This often is not the case and, to fit the training data, the few available model parameters are adjusted for best fitting, and their original meaning may be actually lost (see discussion in the paper concerning the implausible estimation of rolling and air drag resistance for the Renegade).

Table 1: Coefficient of determination (R^2) for the three models

	Analytical parametric model	State Space model	Neural Network model
Lancia Delta	94.87 % (training) 93.39 % (validation)	96.95 % (training) 97.89 % (validation)	97.82 % (training) 98.64 % (validation)
Jeep Renegade	86.47 % (training) 80.14 % (validation)	89.19 % (training) 90.26 % (validation)	96.9 % (training) 97.3 % (validation)

3.1.5.2 How does the biological architecture compare to other possible neural network architectures?

The neural networks introduced so far are feedforward networks that use a particular architecture inspired by cerebellar filters. This is, of course, not the only way to realize a neural network that produce a desired input-output map. Hence, the question of how this bioinspired architecture compares to other possible neural layer arrangements has been investigated in [7], where 4 combinations of feedforward/recurrent and structured/unstructured networks have been examined (see Figure 4 caption).

It was preliminarily argued that the dynamics of physical systems may often be described as the *superposition of individual independent causes* (forces with different origins). Hence a network architecture that combines independent effects is suggested by physical insight, and this is remarkably similar to the cerebellar architecture (Figure 2 and Figure 3). Nonetheless, one might wonder whether a shallow, but fully connected, feedforward network (the classical multilayer perceptron) could be a better choice. In a fully connected feedforward network, the hidden layer seeks combinations from all possible input (Figure 4, top right). Thus, is a more generic network architecture that can model input-output maps of any kind, and with any accuracy provided sufficient training data is available. On the other hand, networks with structure like (Figure 4, top left) are biased towards more efficient learning of particular types of functions. These networks have less (better focused) training parameters and are more "sample efficient", meaning that they can learn the types of function they are specialized in with less training data. For example, the network in Figure 4, top left, will not seek to explain the vehicle acceleration as (possibly) a nonlinear function of brake pressure and engine torque together, whereas the network in Figure 4, top right, will try to combine brake and gas pedal. To discover that they are independent a significant amount of training examples is required. Since brake and engine effects are indeed independent, the network on the left is more efficient. In one case, left, prior knowledge (independence of brake and engine sub-plants) is embedded in the network topology. In the other case, right, independence of the two inputs has to be learned and this requires more data and/or exposes to more overfitting risks.

In [10] (see also talk https://neuroscience.stanford.edu/videos/reinforcement-learning-fast-and-slow) M. Botvinik, while discussing the slowness of Deep Learning methods make similar considerations. He argues that fast learning may only occur with biased neural networks, which in nature are the result of evolution (i.e., of a "slow" learning process).

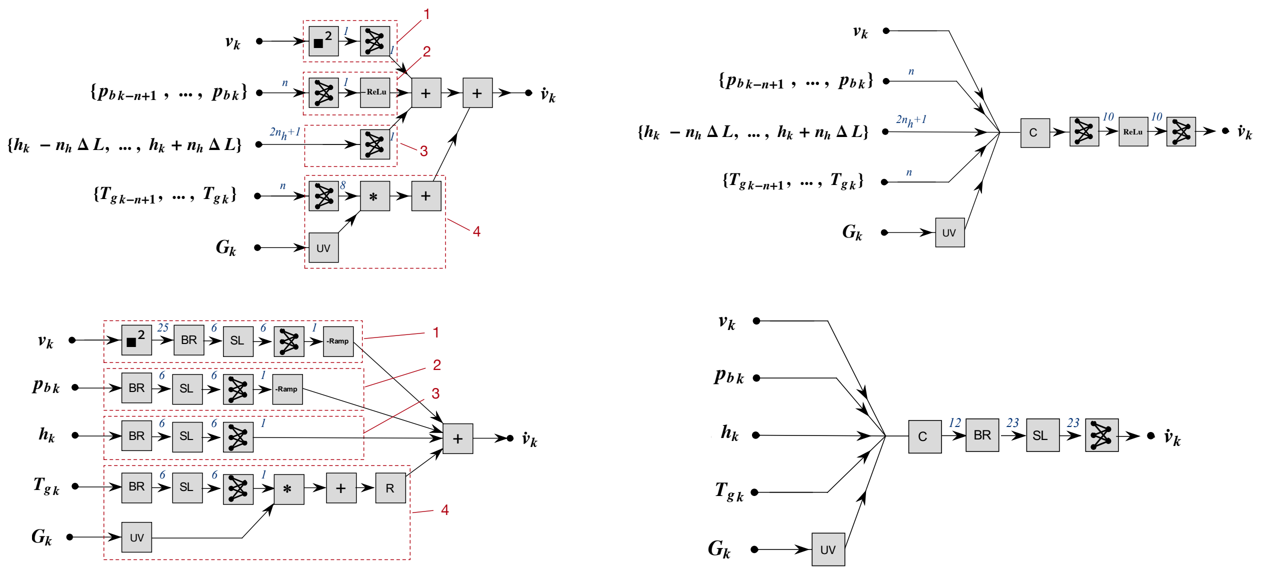

Figure 4: Candidate network topologies that were studied in [7] for implementation of forward models. Top row: feedforward networks. Bottom row: recurrent network. Left column: Networks with cerebellar structure (superposition of independent effects). Right column: Fully connected unstructured networks.

Another question addressed in [7] concerns whether recurrent neural network might be better that feedforward networks (Figure 4 bottom row versus top row). The dynamics of a vehicle (as well as any physical system) have an infinite response but the feedforward networks on top operate with a finite memory (hence they neglect input past the memory length). The response of system modelled like in (Figure 4 top row) is truncated as if the system were a Finite Impulse Response filter. Nonetheless, since the response of all stable physical system vanishes in some time, the truncation error may be negligible if memory of sufficient length (n large enough) is used. Recursive neural network (Figure 4 bottom row) have, on the other hand, infinite memory. This, far from being a real advantage, brings about more trainable parameters that makes the recurrent network more prone to overfit. Recurrent network would be more general, capable of learning the response whichever its length, whereas feedforward networks are biased towards learning system with a given maximum memory length. Again, the more general network is less sample-efficient and may use its (otherwise null) parameters concerned with old events to overfit the training data.

Table 2: Coefficient of determination (R2) feedforward/recurrent, structured/unstructured neural networks

	Feedforward network		Recurrent network	
	Unstructured	Structured	Unstructured	Structured
Training	93.8 %	97.8 %	98.7 %	98.0 %
Validation	93.5 %	97.0 %	95.6 %	96.5 %

Table 2, from [7], confirms the above arguments. The best fit on the validation set is from the structured feed-forward network. The generic feedforward network has less modelling ability. The recurrent networks overfit the training data.

3.1.6 Stochastic forward models

This section is omitted (confidential part)

3.1.7 Example. Stochastic forward model for the Renegade steering actuator

This section is omitted (confidential part)

3.1.8 Spectral considerations and impulse response

This section is omitted (confidential part)

3.1.9 Comparing learned forward models

This section is omitted (confidential part)

3.2 Learning inverse models

This section is omitted (confidential part)

3.2.1 Supervised learning of inverse models from recorded input-output example data (level 1)

This section is omitted (confidential part)

3.2.2 Unsupervised learning of inverse models via episodic-embodied simulations (level 2)

This section is omitted (confidential part)

3.2.3 Robust unsupervised learning via simulations with bootstrapped forward models (level 2)

This section is omitted (confidential part)

3.3 Learning motor primitives (level 3)

This section is omitted (confidential part)

3.3.1 Control with inverse model

This section is omitted (confidential part)

3.3.2 Robust optimal control (choice of desired trajectory that is likely to be executed best)

This section is omitted (confidential part)

3.3.3 Short-cutting the simulation paths

This section is omitted (confidential part)

3.3.4 Comparison with literature probabilistic motion models

This section is omitted (confidential part)

3.3.5 Example. Predicting uncertainty in the trajectory curvature

This section is omitted (confidential part)

3.3.6 Relation to Optimal Control

This section is omitted (confidential part)

3.3.7 Learning the motor cortex salience

This section is omitted (confidential part)

3.3.7.1 Example learning the lateral salience

This section is omitted (confidential part)

3.3.7.2 Relation to Reinforcement Learning

This section is omitted (confidential part)

3.4 Learning of the salience and action sequences (level 4)

This section is omitted (confidential part)

4 Top down approach

This section is omitted (confidential part)

4.1 Pedestrian Models: learning the safe speed (level 5)

4.1.1 Motivation

As suggested above (also Figure 1), level 5 of the hierarchy concerns high-level behaviours such as setting the safe speed and navigation goals. To address the mechanisms needed for those levels we have focused on speed adjustment to pedestrians. Interpreting pedestrian intentions are sometimes difficult for human drivers. A common type of car-to-pedestrian interaction situation involves a pedestrian near or in the road and the need to determine whether he or she will attempt to cross the road. Naturalistic studies of car-to-pedestrian incidents have found that failure of adjusting the speed the main cause of incidents and sudden hard breaking [29]. By applying a novel method of analysis Habibovic et al. [29, pag. 562] drew the following conclusion regarding the design of ADAS systems in cars:

> "When driving straight through intersections, incidents were mainly associated with the late timing of action(s) and high speed. These events were, in turn, preceded by misjudgement of the traffic situation. In several cases, the drivers who had enough information in the traffic environment to anticipate that pedestrians might enter the road did not adjust their speed to accommodate a gentle stop. This finding indicates that ADAS need to support timely notification of potential conflict pedestrians even when there is information in the environment that suggests that a pedestrian might cross the car's path."

The fact that information gained from pedestrian's trajectories, speed, and other possible cues in a given context is insufficient for a human driver may indicate that it is not a trivial problem, even from a machine learning perspective. For example, as noted above, only a small fraction of the visible pedestrians are likely to intersect with the vehicle path; and the cause of the pedestrian's behaviour might not be externally observable, such as e.g., suddenly remembering having forgotten something and suddenly turning back. Another commonly observed problem, is occlusion caused by temporary or permanent obstructing objects, such as vehicles or other types of constructions [29]. While there are rules and laws for pedestrian behaviors these are not always followed for many reasons. For example, distracted walking caused by e.g. head-phone use or engagement with an electronic device contributes to accidents [30], [31]. To help drivers and reduce the number of pedestrian related accidents many car manufacturers have included different types of pedestrian detection systems with automatic braking. A recent report from the American Automotive Association (2019, October) showed that the performance of four existing such systems were limited in complex scenarios and at speeds above 20mph [32]. The study suggested that the systems could prevented collision with a crossing adult pedestrian crossing perpendicular to the road when traveling at 20 mph, but that in speeds of 30 mph, or more complex scenarios of a child running into the road, the systems were significantly less ineffective. Thus, in these cases the human driver still needs to be attentive. A possible addition to automatic braking systems are so called safe speed systems which recommend or set a safe speed based on contextual cues, such as the existence of a pedestrian crossing or upcoming curves [33], [34]. Nevertheless, avoiding car-to-pedestrian incidents is a challenging task and we study this by means of episodic simulations and reinforcement learning with deep q-networks.

As will be noted in the following sections, there are some limitations to the realism of simulations and one additional limitation is that we have not included signalized crossings, footbridges and under passes in our simulations; but it should be noted that the addition of such constraints would influence the pedestrian behaviours as well, such as perhaps reducing, but not eliminating, the likelihood to cross outside zebra crossing e.g. [35].

4.1.2 Pedestrian interface for the OpenDS simulation enviornment

This section is omitted (confidential part)

4.1.3 Pedestrian models

This section is omitted (confidential part)

4.2 Reinforcement learning

This section is omitted (confidential part)

4.2.1 Algorithm

This section is omitted (confidential part)

4.3 Reinforcement Learning within simulations in OpenDS

This section is omitted (confidential part)

4.4 Applications

This section is omitted (confidential part)

4.4.1 Application 1: lane keeping

This section is omitted (confidential part)

4.4.2 Application 2: progressive learning with episodic simulations

This section is omitted (confidential part)

4.4.3 Application 3: safe speed with visible pedestrian

This section is omitted (confidential part)

4.4.4 Application 4: safe speed with occlusion

This section is omitted (confidential part)

4.4.5 Application 5: safe speed with diverse pedestrian behaviour

This section is omitted (confidential part)

4.4.6 Application 6: learning to overtake safely (level 5)

This section is omitted (confidential part)

5 Conclusion

The main contribution of this documents, which in fact are the main findings of Dreams4Cars, is perhaps well summarized by the learning process of Figure 1. This process is about learning models of the wold (both models that predict the effect of actions and models that predict events) that are then manipulated offline to *synthetize* action strategies.

This process appears to be very efficient both with respect to synthetizing good action models (inverse models) and for sample efficiency (optimal use of available data). The process bootstraps a hierarchical sensorimotor architecture where each level learning is accelerated by what was learned at the immediately precedent level. The obtained sensorimotor system replaces corresponding aspects of traditional motor control (MPC, Optimal Control, Trajectory planning, etc.).

Overall the process can be seen as an unsupervised training procedure: forward models learned at a given level are used to create the inverse models for the next level of control. The only necessary data are examples of events (used for creating the training episodes) and the learning of the lowest-level forward model (which is a form of supervised learning using input-out pairs recorded at the wake state).

The lower levels of this procedure (such as. e.g., the training of inverse models of the vehicle dynamics) appear to have some resemblance with the restructuring of neural networks in sleep stage N2. Higher-levels of episodic simulations looks somehow similar to REM dreams.

While the procedure creates the blocks for higher-levels of simulation abstractions, these blocks become also available for the very highest-level which may be implemented via Reinforcement Learning. This way RL may proceed on a low-dimensional state-action space and be, in principle, more efficient because on top of efficient neural networks. It is also worth observing that, beginning at the level of motor primitives, the main goal for learning inverse models is not just learning the action for *one* specific goal, but a function that returns *the value of actions for posterior action selection*. This value function is the salience stored in the “motor cortex”. It is functionally equivalent to the reward function in RL, but its computation is not achieved via a trial and error exploration (the RL way) but via a *synthesis* process based on the *offline manipulation of learned models of the world* (the inspiring idea of Dreams4Cars).

Concerning perception aspects, Dreams4Cars is based on symbolic representations produced by off the shelf sensors because these sensors will be used for some (quite long) time in the first generations of automated vehicles. Dealing with symbolic representations means that it was a human designer that defined the symbol systems (with unavoidable omissions on one side and unnecessary information on the other side). On the other hand, crossover and mutation operators in such a symbolic context is quite easy to do, and so it is easy to create “dreams” or episodes. Prospectively, however, one might wish replacing the human choice of symbols with a self-organized perception system that compresses high-dimensional sensor data into the low dimensions that are *most useful for action*. Such an architecture is described next as a possible evolution, and leverages on the notion of Convergence Divergence Zones as proposed by Damasio. Hence the finding of Dreams4Cars may be, in principle transferred to systems that, besides learning behaviours, may also contextually learn to interpret the world (what is relevant for action).

5.1.1 Note on Reinforcement Leaning

Reinforcement learning was chosen to a large extent on the basis of the bio-inspired approach taken here. Not only insofar as the inspiration of how the animal brain uses reinforcement learning for certain tasks, but also to align with a view where cognition is not separated from perception and action (see e.g. [49]) which separates our approach from the mainstream approach of providing explicit borders between perception and decision making/motor control (as exemplified by the review in [50]). However, as already noted above DRL is not an automatic solution to the problems posed and we experienced some difficulties, such as integrating the DRL with the developed Action Selection system (D2.1) mainly because of the dimensionality of the problem (Section 0). In comparison to the bottom-up approach, the top-down approach with automatic scenario generation and reinforcement learning methodologically required more computational time, it requires significant re-searcher/engineering top down design of reward function and choice of hyperparameters, which limits the use-fulness of the approach. Although not pursued here, future research to achieve a fully automatic training system for AD, more research needs to be devoted to research independent methods for reward function development and hyperparameter setting. However, it should be noted that the DRL added possible benefits to the existing co-driver architecture by adding suggestions of the appropriate speed given contextual clues in the environment. Another notable outcome of the work on reinforcement learning, is the results on and new ways of implementing a replay buffer [51]. Although previous studies have found increased performance with the addition of a replay buffer [51], [52] our results indicate that the performance of the replay buffer depends on the input-data and task requirements (see Section 0). Furthermore, we developed a method for handling rare events, which is an important aspect of autonomous driving and can be used in future (see Section **Errore. L'origine riferimento non è stata trovata.**).

5.2 Future developments

So far, Dreams4Cars is focused on the synthesis of behaviours assuming the use of existing sensor technology. This choice was motivated in the original proposal by the consideration that first generations of automated vehicles will use perception systems based on the (evolution of) the current sensor technology.

The input to Dreams4Cars is thus the *symbolic* output of the used sensor subsystems, in which the symbols have been chosen by the engineers that developed the "smart" sensors.

It makes however sense to reason about possible future developments, in which the perception system itself is integral part of the Agent: the Agent not only learns better behaviours, but it can also learn to extract action relevant information from raw sensory data; i.e. creating its own concepts and representations.

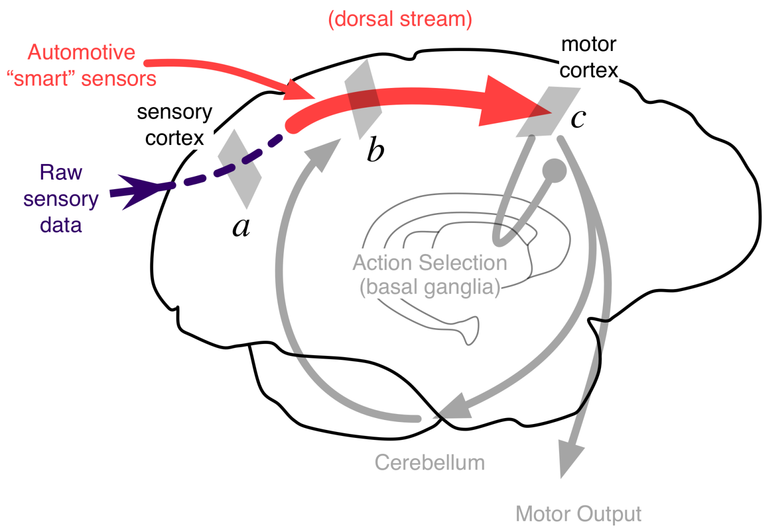

Figure 5: Evolution of Agent architecture to allow learning in the entire perception action system, with in particular the ability learn at perception and forming novel concepts.

This idea is expressed in Figure 5. The current Dreams4Cars sensorimotor system is shown in red colour. Because the automotive sensors provide pre-processed high-level symbols, part of the dorsal stream is in truth bypassed and the sensor output signals are used to detect affordances and compute motor cortex as they are.

In a certain sense it is like the Agent does not see the real world, but the encoding of the environment with the *concepts/symbols that have been chosen by the designer*. To clarify this point, let us refer to Figure 6, which is an example of symbolic representation based on human selected concepts. In the example, a neural network (Ademxapp Model A1 Trained on Cityscapes Data) was used to produce a semantic segmentation of a camera view. It is important to observe that the classes in which the image is segmented (shown in the legend) were decided by the engineers that developed the neural network; and in this case, they did not conceive different labels for the minivan with the open door and the other cars. Consequently, the dangerousness of the minivan is not perceived, because all vehicle types are projected onto the same, unique, vehicle class. Re-training the semantic segmentation network with more classes would be a humongous endeavour: first it would be difficult to know exactly how many vehicle sub-classes are necessary; then re-labelling the training images would be a great cost and, finally, even if it is known that there is a minivan with an open rear door in itself this does not give any precise information about what kind of behaviour is necessary to manage the risk (it would make the downstream decision system to explode in complexity).

It makes thus sense to evaluate the possibility that the perception system becomes integral part of the Agent, which receives raw (or lower-level) sensory data and learns to form/update concepts by itself. This is shown with the dark violet dashed line in Figure 5.

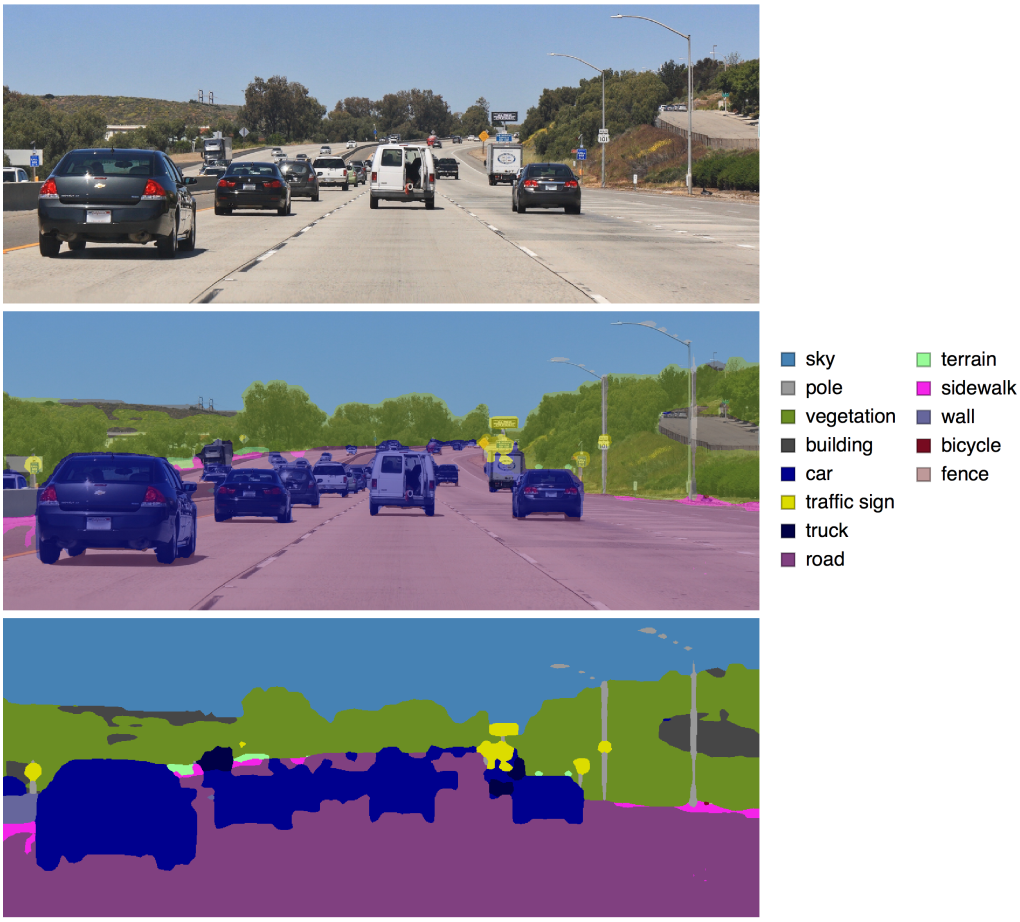

Figure 6: Example of semantic classification based on classes (hence concepts) defined by a human designer. The dangerousness of the minivan with the open door is lost is lost because the real situation (on top) is projected onto a predefined number of classes that do not make distinction between the minivan and the other cars.

The theoretical framework that looks better suited for this appears to be the notion of Damasio's convergence-divergence zones (CDZ) that has been mentioned in previous deliverables (e.g., D2.3 section 2.1.5). According to this guiding theory, sensory information is forcefully compressed in convergence zones (*a*-*b* in Figure 5), encoding in compact form the environment and agent states. Since these concepts are then used for generation of affordances and computation of the value of each one (*b*-*c* in Figure 5), any optimization/learning process will lead to retain only the information that is relevant for action. At the same time, should there exist two similar environmental states that require different actions (car vs. minivan) a learning process (we do not argue here of what type) will necessary lead to differentiate the representations of those two states at *b*.

As already explained in previous deliverables the CDZ architecture can be (approximately) constructed into an artificial neural network with branches for the forward and inverse data flows (D2.3, Figure 5).

Avery simple way to train the convergence part of the CDZ architecture is by means of an autoencoder that implements the forward a-b and backward b-a paths (D2.3 section 2.1.5). It must be observed that this form of training, albeit simple, is not ideal because the latent space formed in this way is optimized for reproducing the sensory input as best as possible. Instead the ultimate goal should be to optimize the latent space in detecting patterns that are mapped onto different actions.

In the following we present a first investigation of the implementation of the CDZ architecture for visual sensory input. The latent space is trained with the autoencoder approach on the *a-b-a* path but this process is augmented by branches trained to detect vehicles and lanes with (for the moment) semi-supervised approach. We show how the concepts formed in this latent space can be, in principle, manipulated to predict near future states and/or to form imaginary environmental states using the crossover operator. Finally, we discuss how to train the CDZ architecture on the (ultimate) *a-b-c* path via simulations.

5.2.1 Mental imagery for Automated Vehicles using Damasio's CDZ architecture

Over the years the CDZ hypothesis has found support of a large body of neurocognitive and neurophysiological evidence. However, it is a purely descriptive model and does not address the crucial issue of how the same neural assembly, which builds connections by experiences in the convergent direction, can computationally work in the divergent direction as well. At the moment, there are no computational models that faithfully replicate the behaviour of CDZs, however, we found a number of independent notions, introduced in the field of artificial intelligence for different purposes, which bear significant similarities with the CDZ scheme.

In the realm of artificial neural networks, the computational idea that most closely resonate with CDZ is the *autoencoder*. There is a clear correspondence between the encoder and the convergence zone in the CDZ neurocognitive concept, and similarity between the decoder and the divergence zone.

5.2.1.1 Convergence-Divergence as Convolutional-Deconvolutional Autoencoder

One of the major challenges in cognitive science is explaining the mental mechanisms by which we build conceptual abstractions. The conceptual space is the mental scaffolding the brain gradually learns through experience, as internal representation of the world. CDZs are a valid systemic candidate for how the formation of concepts takes place at brain level. However, the idea of CDZ is just sketched and cannot provide a detailed mechanism for conceptual abstractions.

According to the historical empiricist tradition, conceptual abstractions is derived from experience, mostly perceptual experience. This direction fits perfectly with the approach implemented by artificial neural networks.

Still, a difficulty with acquiring even moderately abstract categories lies in the mutually inconsistent manifestations of the characteristic features of a category, in each of its real exemplars. In visual data, for example, object translation, rotation, motion in depth, deformation and lighting changes can drastically entangle features of objects belonging to the same category. Conversely, the perceptual appearance of two unrelated objects, like a close flying insect and a far distant vulture, can be very similar. A suggested solution to this difficult issue is in the transformational abstraction [53] performed by a hierarchy of cortical operations, as in the ventral visual cortex. The essence of transformational abstraction, from a mathematical point of view, should lie in the combination of two operations: linear convolutional filtering and nonlinear down-sampling. Operations of this sort have been identified in the primary visual cortex and the staking of this process in hierarchy is well recognized in the primate ventral visual path [54].

Let us now dive into detail of how convergence can be achieved inside autoencoders. The most common way is stacking feed-forward layers with decreasing number of units. There is, however, an interesting alternative closely related to the transformational abstraction hypothesis: the *deep convolutional neural networks* (DCNNs). The DCNN implements the hierarchy of convolutional filtering alternated with nonlinear down-sampling, and it is considered the essence of transformational abstraction.

DCNNs do not only resonate with the theoretical proposal of transformational abstraction, there is a growing evidence of striking analogies between patterns in DCNN models and patterns of voxels in the brain visual system [55].

DCNNs are therefore a highly biologically plausible implementation for the convergence zone in CDZs, at least in the case of visual information. Convolutional neural models do not include a divergence counterpart, typically the outputs of the last convolutions are fed into ordinary feed forward layers to produce a classification. This gap was filled with the *deconvolutional* neural networks, performing alternation of un-pooling and linear filtering. Each step of these two operations reconstruct a higher level of spatial dimension of the data, up to the full high dimension of the original image. This stacked combination of deconvolution and un-pooling is the current neural implementation closer to the idea of divergence zone of CDZs.

5.2.1.2 Predictive Brain as Variational Autoencoder

The reason why cognition is mainly explicated as simulation, as in the simulation theory of cognition of Hesslow, is because the brain through simulation can achieve the most precious information of an organism: a prediction of the state of affairs in the environment in the future. The need of predicting, and how it moulds the entire cognition, has become the core of a theory which has gained large attention in the last decade, made popular under the term “predictive brain”, or “free-energy principle for the brain”, by Karl Friston [56]. According to Friston, the behaviour of the brain – and of an organism as a whole – can be conceived as minimization of free-energy, a quantity that can be expressed in several ways depending on the kind of behaviour and the brain systems involved.

Free-energy is a concept originated in thermodynamics, as a measure of the amount of work that can be extracted from a system. What is borrowed by Friston is not the thermodynamic meaning of the free-energy, but its mathematical form only. The basic form of the free-energy under the variational Bayesian framework is borrowed by Friston for abstract entities of cognitive value. For example, this is his free-energy formulation in the case of perception:

$$F_P = \mathit{\Delta}_{\mathrm{KL}}\Big(\check{p}(\boldsymbol{c}|\boldsymbol{z})\|p(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{a})\Big) - \log p(\boldsymbol{x}|\boldsymbol{a}) \tag{1}$$

where ***x*** is the sensorial input of the organism, ***c*** is the collection of the environmental causes producing ***x***, ***a*** are actions that act on the environment to change sensory samples, and ***z*** are inner representations of the brain. The quantity $\check{p}(\boldsymbol{c}|\boldsymbol{z})$ is the encoding in the brain of the estimate of causes of sensorial stimuli. The quantity $p(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{a})$ is the conditional probability of sensorial input conditioned by the actual environmental causes ***c***. The discrepancy between the estimated probability and the actual probability is given by the Kullback-Leibler divergence Δ_{KL}. The minimization of F_P in equation optimizes ***z***.

In the last few years there has been renewed interest in the area of Bayesian probabilistic inference in learning models of high dimensional data. The Bayesian framework, variational inference in particular, has found a fertile ground in combination with neural models. A new approach connecting autoencoders and variational inference became quickly popular under the term *variational autoencoder*, and a variety of neural models including such idea have been proposed over the years.

The interesting aspect is that the adoption of variational inference lead to a mathematical formulation impressively similar to the concept of free energy in Friston. The following equation shows the loss function of a variational autoencoder:

$$\mathcal{L}(\Theta,\Phi|\mathbf{x}) = \Delta_{\mathrm{KL}}\left(q_\Phi(\mathbf{z}|\mathbf{x})\|p_\Theta(\mathbf{z})\right) - \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log p_\Theta(\mathbf{x}|\mathbf{z})\right] \tag{2}$$

where ***x*** is a high dimensional random variable, ***z*** the representation of the variable in the low-dimensional latent space. Θ and Φ are parameters describing, respectively, the decoder and encoder of the network. p_Θ is computed by the decoder and represents the desired approximation of the unknown input distribution p, and q_Φ is the auxiliary distribution computed by the encoder from which to sample ***z.*** E is the expectation operator, and $\Delta_{KL}$is the Kullback-Leibler divergence.

It is evident how equations (1) and (2) are impressively similar. This close analogy went unnoticed by all the main developers of variational autoencoder. It is not so surprising because mainstream deep learning is driven by engineering goals without any interest in connections with cognition. Within the philosophy of Dreams4Cars, the strong connection between a well-established cognitive theory and a computational solution, greatly argues in favour of adopting such solution.

5.2.1.3 A CDZ-like Model for Driving

As described above, we reviewed several components that match quite closely the relevant neurocognitive theories mentioned before. Our proposed model attempts to weave together these components, finalized at visual perception in an autonomous driving agent.

Similarly to the hierarchical arrangement of CDZs in the brain, our model is provided with different levels of processing paths. A first processing path starts from the raw image data and converges up to a low-dimension representation of visual features. Consequently, the divergent path outputs in the same format as the input image. The other processing path leads to representations that are no more in terms of visual features, rather in terms of concepts. As discussed before, our brain naturally projects sensorial information – especially visual – into conceptual space, where the local perceptual features are pruned and neural activation code the nature of entities present in the environment that produced the stimuli.

In the driving context it is not necessary to infer categories for every entity present in the scene, it is useful to project in conceptual space only the objects relevant to the driving task. In the model presented here we choose to consider the two main concepts of *cars* and *lane markings*.

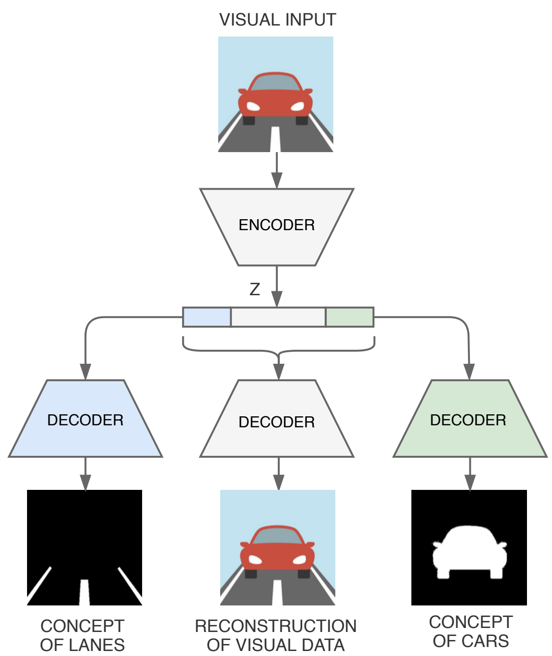

Figure 7: The architecture of our variational autoencoder.

As depicted in Figure 7: The architecture of our variational autoencoder., the presented variational autoencoder is composed of one shared encoder and three independent decoders, and all the components of the architecture are trained jointly. The encoder compresses an RGB image to a compact high-feature representation. Then the decoders map different part of the latent space back to separated output spaces: one into the same visual space of the input; the other two into conceptual space, producing binary images containing, respectively, *cars* entities and *lane markings* entities.

Therefore, in our implementation the entire latent vector $\boldsymbol{z}$ represents into the visual space, and at the same time two inner segments project specifically into the *car* and *lane marking* concepts. The rationale for this choice is that in mental imagery there is no clear-cut distinction between low-level features and semantic features: the entire scene is mentally reproduced but includes the "awareness" of the salient concepts present in the scene.

Note that the idea of partitioning the entire latent vector into meaningful components is not new. In related works the latent vector is sometimes partitioned in one segment for the semantic content and a second segment for the position of the object. Our approach is different. While we keep disjointed the two segments for the *car* and *lane* concepts, we fully overlap these two representations within the entire visual space. This way, we adhere entirely to the CDZ principle, and try to achieve the full scene by divergence, but at the same time including awareness for the *car* and *lane* concepts.

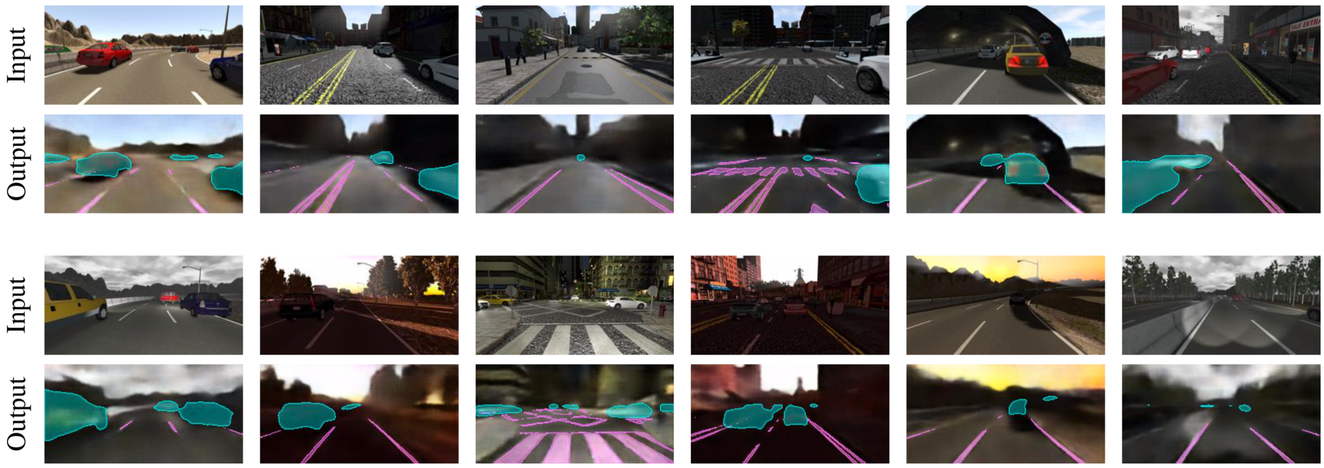

Figure 8: Results of our model for a selection of frames from the SYNTHIA dataset, with different environmental and lighting conditions.

Figure 8: Results of our model for a selection of frames from the SYNTHIA dataset, with different environmental and lighting conditions. shows a selection of results achieved with the model just described. The final architecture is trained for 200 epochs, and used 4 convolutional layers in the encoder, 4 deconvolutional layers for each decoder, and a latent space representation of 128 neurons, of which only 16 encoding the *car* concept and another 16 for the *lane marking* concept. We trained and tested the presented model on the SYNTHIA dataset, a large collection of synthetic images representing various urban scenarios.

The images are processed to better show at the same time the results on conceptual space and visual space. The coloured overlays highlight the concepts computed by the network: the cyan regions are the output of the *car* divergent path, and the pink overlays are the output of the *lane markings* divergent path. The figure includes a variety of driving situations, going from sunny environments (top rows) to very adverse driving conditions (bottom rows) in which the detection of other vehicles can be challenging even for a human.

These results nicely show how the projection of the sensorial input (original frames) into conceptual representation is very effective in identifying and preserving the sensible features of *cars* and *lane markings*, despite the large variations in lighting and environmental conditions.

For more details and results on this approach, refer to [57].

We presented a neural model for visual perception in the context of autonomous driving, grounded in a number of concepts from neuroscience and cognitive science. The main guiding principle is the CDZs proposed by Meyer and Damasio that in our context represent the neural correlate of mental imagery as simulation. CDZs find their best artificial cousin in the neural autoencoder architecture. For the choice of how to realize the convergence zone in the encoder, the guiding cognitive theory is that of transformational abstraction, suggesting the adoption of convolutional networks. One more theoretical contribution, the free-energy principle of Friston, further suggests refining the autoencoder architecture as variational autoencoder. Based on these premises, our model aims at gaining an internal low-level representation of two spaces: the visual one and the conceptual one. The latter is limited to the two most crucial concepts during driving: *cars* and *lane markings*. We succeeded in achieving an internal representation as compact as with 128 units only, of which 16 units are enough to recognize the *car* concepts in any location of the visual space, and similarly for the *lane* concepts. Our future plans involve the finalization of the higher-level model of the architecture which computes motor commands from the conceptual representation of the environment presented in this work.

5.2.2 Neuralized logical reasoning

This section is omitted (confidential part)

6 Bibliographical References

[1] H. Svensson e S. Thill, «Beyond bodily anticipation: Internal simulations in social interaction», *Cogn. Syst. Res.*, vol. 40, pagg. 161–171, 2016.

[2] K. L. Downing, «Predictive models in the brain», *Connect. Sci.*, vol. 21, n. 1, pagg. 39–74, mar. 2009.

[3] H. Svensson, A. F. Morse, e T. Ziemke, «Neural Pathways of Embodied Simulation», in *Anticipatory Behavior in Adaptive Learning Systems: From Psychological Theories to Artificial Cognitive Systems*, vol. 5499, G. Pezzulo, M. V. Butz, O. Sigaud, e G. Baldassarre, A c. di Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pagg. 95–114.

[4] K. Doya, «What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?», *Neural Netw.*, vol. 12, n. 7–8, pagg. 961–974, ott. 1999.

[5] J. Porrill, P. Dean, e S. R. Anderson, «Adaptive filters and internal models: Multilevel description of cerebellar function», *Neural Netw.*, vol. 47, pagg. 134–149, 2012.

[6] S. James, A. Sean, e D. L. Mauro, «Longitudinal Vehicle Dynamics: A Comparison of Physical and Data-Driven Models Under Large-Scale Real-World Driving Conditions», *IEEE Trans. Intell. Transp. Syst.*, vol. (submitted).

[7] M. Da Lio, D. Bortoluzzi, e G. P. Rosati Papini, «Modeling Longitudinal Vehicle Dynamics with Neural Networks», *Veh. Syst. Dyn.*, vol. in press.

[8] M. Da Lio, A. Plebe, e D. Bortoluzzi, «On Reliable Neural Network Sensorimotor Control in Autonomous Vehicles», *IEEE Trans. Intell. Transp. Syst.*, pag. submitted.

[9] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, e D. Hrovat, «Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation», *Int. J. Robust Nonlinear Control*, vol. 18, n. 8, pagg. 862–875, mag. 2008.

[10] M. Botvinick, S. Ritter, J. X. Wang, Z. Kurth-Nelson, C. Blundell, e D. Hassabis, «Reinforcement Learning, Fast and Slow», *Trends Cogn. Sci.*, vol. 23, n. 5, pagg. 408–422, mag. 2019.

[11] B. Efron e R. Tibshirani, *An introduction to the bootstrap*. New York: Chapman & Hall, 1994.

[12] C. M. Harris, «Signal-dependent noise determines motor planning», *Nature*, vol. 394, n. August, pagg. 780–784, 1998.

[13] C. M. Harris, «Biomimetics of human movement: functional or aesthetic?», *Bioinspir. Biomim.*, vol. 4, n. 3, pag. 033001, set. 2009.

[14] F. Biral, D. Bortoluzzi, V. Cossalter, e M. Da Lio, «Experimental study of motorcycle transfer functions for evaluating handling», *Veh. Syst. Dyn.*, vol. 39, n. 1, pagg. 1–25, 2003.

[15] A. Bemporad e M. Morari, «Robust model predictive control: A survey», in *Robustness in identification and control*, vol. 245, A. Garulli e A. Tesi, A c. di London: Springer London, 1999, pagg. 207–226.

[16] C. T. Smith, J. B. Aubrey, e K. R. Peters, «Different Roles for REM and Stage 2 Sleep In Motor Learning: A Proposed Model», pag. 24.

[17] C. M. Harris, «Signal-dependent noise determines motor planning», *Nature*, vol. 394, n. August, pagg. 780–784, 1998.

[18] P. M. Bays e D. M. Wolpert, «Computational principles of sensorimotor control that minimize uncertainty and variability.», *J. Physiol.*, vol. 578, n. Pt 2, pagg. 387–96, gen. 2007.

[19] E. Todorov e M. I. Jordan, «A Minimal Intervention Principle for Coordinated Movement», *Adv. Neural Inf. Process. Syst.*, vol. 15, pagg. 27–34, 2003.

[20] P. Viviani e T. Flash, «Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning.», *J. Exp. Psychol. Hum. Percept. Perform.*, vol. 21, n. 1, pagg. 32–53, mar. 1995.

[21] S. Thrun, W. Burgard, e D. Fox, *Probabilistic Robotics*, vol. 1. 2005.

[22] S. James, A. Sean, e D. L. Mauro, «Longitudinal Vehicle Dynamics: A Comparison of Physical and Data-Driven Models Under Large-Scale Real-World Driving Conditions», *IEEE Trans. Intell. Transp. Syst.*, vol. (submitted).

[23] M. Da Lio, G. P. Rosati Papini, e R. Donà, «Hierarchical Robust Path Control for Automat-ed Driving in Roads Structured with Lanes», vol. Manuscript in Preparation, 2018.

[24] R. Donà, G. P. Rosati Papini, M. Da Lio, e L. Zaccarian, «On the Stability and Robustness of HierarchicalVehicle Lateral Control with Inverse/ForwardDynamics Quasi-Cancellation», *IEEE Trans. Veh. Technol.*, vol. Accepted.

[25] T. C. Stewart, T. Bekolay, e C. Eliasmith, «Learning to Select Actions with Spiking Neurons in the Basal Ganglia», *Front. Neurosci.*, vol. 6, 2012.

[26] J. Kober, J. A. Bagnell, e J. Peters, «Reinforcement learning in robotics: A survey», *Int. J. Robot. Res.*, vol. 32, n. 11, pagg. 1238–1274, 2013.

[27] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, e V. Koltun, «CARLA: An open urban driving simulator», *ArXiv Prepr. ArXiv171103938*, 2017.

[28] A. E. Sallab, M. Abdou, E. Perot, e S. Yogamani, «End-to-end deep reinforcement learning for lane keeping assist», *ArXiv Prepr. ArXiv161204340*, 2016.

[29] A. Habibovic, E. Tivesten, N. Uchida, J. Bärgman, e M. Ljung Aust, «Driver behavior in car-to-pedestrian incidents: An application of the Driving Reliability and Error Analysis Method (DREAM)», *Accid. Anal. Prev.*, vol. 50, pagg. 554–565, gen. 2013.

[30] R. A. Scopatz e Y. Zhou, «Effect of electronic device use on pedestrian safety: A literature review», National Highway Traffic Safety Administration, Washington DC, DOT HS 812 256, apr. 2016.

[31] C. H. Basch, D. Ethan, P. Zybert, e C. E. Basch, «Pedestrian Behavior at Five Dangerous and Busy Manhattan Intersections», *J. Community Health*, vol. 40, n. 4, pagg. 789–792, ago. 2015.

[32] «AUTOMATIC EMERGENCY BRAKING WITH PEDESTRIAN DETECTION», American Automobile Association, ott. 2019.

[33] O. M. Carsten e F. N. Tate, «Intelligent speed adaptation: accident savings and cost–benefit analysis», *Accid. Anal. Prev.*, vol. 37, n. 3, pagg. 407–416, 2005.

[34] E. Bertolazzi, F. Biral, M. Da Lio, A. Saroldi, e F. Tango, «Supporting drivers in keeping safe speed and safe distance: The SASPENCE subproject within the European framework programme 6 integrating project PReVENT», *IEEE Trans. Intell. Transp. Syst.*, vol. 11, n. 3, pagg. 525–538, 2010.

[35] V. Cantillo, J. Arellana, e M. Rolong, «Modelling pedestrian crossing behaviour in urban roads: A latent variable approach», *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 32, pagg. 56–67, lug. 2015.

[36] «The 2nd strategic highway research program naturalistic driving study dataset. Available from the SHRP 2 NDS InSight Data Dissemination web site.», Transportation Research Board of the National Academy of Sciences., Database, 2013.

[37] W. Daamen e S. P. Hoogendoorn, «Experimental research of pedestrian walking behavior», *Transp. Res. Rec.*, vol. 1828, n. 1, pagg. 20–30, 2003.

[38] T. S. Keller, A. M. Weisberger, J. L. Ray, S. S. Hasan, R. G. Shiavi, e D. M. Spengler, «Relationship between vertical ground reaction force and speed during walking, slow jogging, and running», *Clin. Biomech.*, vol. 11, n. 5, pagg. 253–259, 1996.

[39] R. E. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1957.

[40] N. D. Daw e P. Dayan, «The algorithmic anatomy of model-based evaluation», *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 369, n. 1655, pag. 20130478, nov. 2014.

[41] V. Mnih *et al.*, «Playing Atari with Deep Reinforcement Learning», *ArXiv13125602 Cs*, dic. 2013.

[42] H. van Hasselt, A. Guez, e D. Silver, «Deep Reinforcement Learning with Double Q-learning», 2015.

[43] D. Ha e J. Schmidhuber, «World Models», *ArXiv180310122 Cs Stat*, mar. 2018.

[44] A. E. Sallab, M. Abdou, E. Perot, e S. Yogamani, «Deep reinforcement learning framework for autonomous driving», *Electron. Imaging*, vol. 2017, n. 19, pagg. 70–76, 2017.

[45] V. Mnih *et al.*, «Human-level control through deep reinforcement learning», *Nature*, vol. 518, n. 7540, pagg. 529–533, feb. 2015.

[46] D. Ha e J. Schmidhuber, «World Models», *ArXiv Prepr. ArXiv180310122*, 2018.

[47] E. Santana e G. Hotz, «Learning a driving simulator», *ArXiv Prepr. ArXiv160801230*, 2016.

[48] P. Wang, C.-Y. Chan, e A. de La Fortelle, «A reinforcement learning based approach for automated lane change maneuvers», in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pagg. 1379–1384.

[49] M. L. Anderson, «Embodied Cognition: A field guide», *Artif. Intell.*, vol. 149, n. 1, pagg. 91–130, set. 2003.

[50] C. Badue *et al.*, «Self-driving cars: A survey», *ArXiv Prepr. ArXiv190104407*, 2019.

[51] V. Mnih *et al.*, «Playing atari with deep reinforcement learning», *ArXiv Prepr. ArXiv13125602*, 2013.

[52] T. P. Lillicrap *et al.*, «Continuous control with deep reinforcement learning», *ArXiv Prepr. ArXiv150902971*, 2015.

[53] C. Buckner, «Empiricism without magic: transformational abstraction in deep convolutional neural networks», *SY*, vol. 195, pagg. 5339–5372, 2018.

[54] D. C. Van Essen, «Organization of visual areas in macaque and human cerebral cortex», .

[55] B. T. Tripp, «Similarities and differences between stimulus tuning in the inferotemporal visual cortex and convolutional networks», presentato al International Joint Conference on Neural Networks, 2017, pagg. 3551– 3560.

[56] K. Friston, «The free-energy principle: a unified brain theory?», *NNSr*, vol. 11, pagg. 127–138, 2010.

[57] A. Plebe e M. D. Lio, «Variational Autoencoder Inspired by Brain's Convergence-Divergence Zones for Autonomous Driving Application», in *Proceedings of the 20th International Conference on Image Analysis and Processing*, 2019, vol. 11751, pagg. 367–377.

[58] S. Hölldobler, Y. Kalinke, e H.-P. Störr, «Approximating the semantics of logic programs by recurrent neural networks», *Appl. Intell.*, vol. 11, n. 1, pagg. 45–58, 1999.

[59] G. G. Towell e J. W. Shavlik, «Knowledge-based artificial neural networks», *Artif. Intell.*, vol. 70, n. 1–2, pagg. 119–165, 1994.

[60] T. Zia, U. Zahid, e D. Windridge, «Modelling Relation Paths with Generative Adversarial for Representation Learning of knowledge Bases», presentato al NeurIPS 2019 KR2ML workshop, Vancouver, 2019.