IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 30, 2020, accepted September 26, 2020, date of publication October 1, 2020, date of current version October 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028185

On the Road With 16 Neurons: Towards Interpretable and Manipulable Latent Representations for Visual Predictions in Driving Scenarios

ALICE PLEBE[1] AND MAURO DA LIO[2], (Member, IEEE)
[1]Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy
[2]Department of Industrial Engineering, University of Trento, 38123 Trento, Italy

Corresponding author: Alice Plebe (alice.plebe@unitn.it)

This work was supported in part by the Horizon 2020 Dreams4Cars Research and Innovation Action, European Commission, under Grant 731593, and in part by the Deep Learning Laboratory from the ProM Facility funded by the Fondazione Cassa di Risparmio di Trento e Rovereto, Italy.

ABSTRACT This paper proposes a strategy for visual perception in the context of autonomous driving. Humans, when not distracted or drunk, are still the best drivers you can currently find. For this reason, we take inspiration from two theoretical ideas about the human mind and its neural organization. The first idea concerns how the brain uses structures of neuron ensembles that expand and compress information to extract abstract concepts from visual experience and code them into compact representations. The second idea suggests that these neural perceptual representations are not neutral but functional to predicting the future state of affairs in the environment. Similarly, the prediction mechanism is not neutral but oriented to the planning of future action. We identify within the deep learning framework two artificial counterparts of the aforementioned neurocognitive theories. We find a correspondence between the first theoretical idea and the architecture of convolutional autoencoders, while we translate the second theory into a training procedure that learns compact representations which are not neutral but oriented to driving tasks, from two distinct perspectives. From a static perspective, we force separate groups of neural units in the compact representations to represent specific concepts crucial to the driving task distinctly. From a dynamic perspective, we bias the compact representations to predict how the current road scenario will change in the future. We successfully learn compact representations that use as few as 16 neural units for each of the two basic driving concepts we consider: `cars` and `lanes`. We maintain the two concepts separated in the latent space to facilitate the interpretation and manipulation of the perceptual representations. The source code for this paper is available at https://github.com/3lis/rnn_vae.

INDEX TERMS Autonomous driving, convergence-divergence zones, deep learning, predictive brain, variational autoencoder.

I. INTRODUCTION

Road traffic injuries are the leading cause of death for the age group between 5 and 29 years [1]. The World Health Organization reported that in 2018 the number of road traffic deaths was 16 times larger than in war conflicts from that same year [1]. This suggests that mitigation of motor vehicle accidents will probably be one of the most beneficial outcomes expected from artificial intelligence and automation [2]. In fact, in the US only 2% of vehicle crashes are due to technical failures; the rest is attributable to the human drivers. Among the major causes of accidents are inattention, fast or reckless driving, illegal maneuvers, the influence of alcohol or drugs, and tiredness [3].

Self-driving cars will be immune to all the risky factors depending on human drivers. The development of fully autonomous vehicles has always been considered a coveted achievement for modern society. The research on this field has a long history that dates back to the late 70s [4], but it became a reality – at an unusually fast pace – no longer than a decade ago [5]. While most of the components of a self-driving system (such as sensors) have improved at

The associate editor coordinating the review of this manuscript and approving it for publication was Junhua Li.

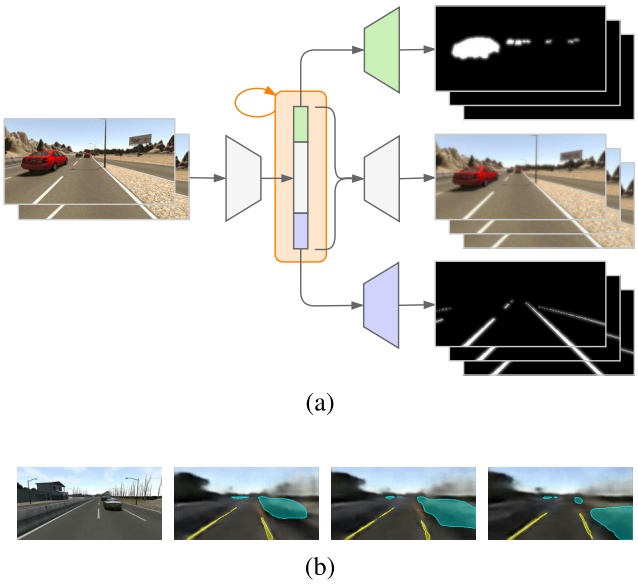

FIGURE 1. The idea behind our approach. (a) A first model learns to represent visual scenarios into compact vectors that are at once semantically organized and temporally coherent. By exploiting semantic segmentation as a supporting task, the model forces separate groups of neurons to distinctly represent the basic concepts of `cars` and `lanes`, while self-supervision is adopted to bias the internal representation towards the ability to predict the dynamics of objects in the scene. (b) A second neural network uses the compact representations to perform imagery and predict long-term future frames.

the typical rate of technological progress without any specific crucial innovations, the impressive advances have been mainly fueled by the emerging "deep" versions of artificial neural networks [6]–[8].

Since the early beginnings, the greatest challenge for autonomous driving systems has been the perception and understanding of the road environment. This is precisely one of the successful fields of application of deep neural models [9]–[11], which have quickly become the method of choice for driving scene perception [12]–[15]. However, despite the impressive progress, perception remains the major obstacle towards fully autonomous vehicles. The core of this issue can be identified in the narrow conception of "perception" usually assumed in autonomous driving, which lacks a fundamental aspect: to gather knowledge about objects and events in the environment oriented to plan future actions [16], [17]. Hence, perception is not a mere elucidation of objects in the world but the detection of action possibilities.

In this respect, it might be useful to reflect on how humans are able to drive. When not distracted, asleep, or deliberately engaged in dangerous maneuvering, humans are excellent at driving, as at many other complex and highly specialized sensorimotor behaviors. How the human brain realizes such sensorimotor behaviors is far from being fully understood, but a few general neurocognitive theories try to shed light on this. We believe it is useful to borrow in particular two theoretical ideas to design the perception strategy of autonomous vehicles.

The first neurocognitive theory we take inspiration from concerns how sensory information is coded into low-dimensional representations in the brain. These perceptual representations can capture aspects related to the actions that caused the perceptual stimulus. Because of the sensorial information saved in the representations, the brain can recreate the original stimulus in an approximated form, during a phenomenon called *mental imagery* [18], [19]. One of the first pieces of evidence of these internal representations was found in the work of Damasio [20], who identified neuron ensembles exhibiting a convergent structure, where neural signals are projected onto multiple cortical regions in a many-to-one fashion. Damasio later developed a broader theory [21] identifying more sophisticated neural structures he called *convergence-divergence zones* (CDZs). In this case, the very same neuron ensembles can perform both convergent and divergent projections, depending on the current action the brain is engaged with: the convergent flow is dominant during perceptual recognition, while the divergent flow occurs during mental imagery. For this reason, CDZs have been recognized as a crucial component in the formation of concepts in the brain [22]. Therefore, we believe it useful to design an artificial model with a similar hierarchical architecture for learning the abstract concepts relevant to the driving context.

The second theoretical idea concerns the nature of the neural representations in the brain. In most cases, neural representations are not abstract representations of the environment but neural states functional to predicting the state of affairs in the future environment. The ability to predict appears to be the primary goal of intelligence [23], [24]. There is evidence for the existence in the brain of various circuits that provide prediction from perceptual representations. In particular, two forms of prediction – procedural and declarative – are typically acknowledged in different brain structures [25]. However, one of the most popular theories in the field interprets the mental mechanism of prediction in mathematical terms [26], [27]. This theory, called *predictive brain*, explains the behavior of the brain as the minimization of free-energy, a quantity that can be expressed in mathematical form. We will show how this formulation can be adopted as a loss function to train our model.

Our work aims to learn conceptual representations of the driving scenario from visual information. We intend to learn compact and informative representations that can be useful for a variety of downstream driving tasks, primarily the tasks requiring predictive capabilities. We propose a cognitive-inspired approach that forces the representations to be oriented to the driving tasks, under two distinct perspectives.

1) From a static perspective, we force separate groups of neural units to encode specific concepts crucial in the driving task distinctly. Specifically, we use as few as 16 neurons for each of the two basic concepts we adopt: `cars` and `lanes`. The latent space is explicitly partitioned in regions that encode different concepts so that they can be manipulated individually.

2) From a dynamic perspective, we bias the compact representations to predict how the current road scene would change in the future. Albeit this work does not fully develop visual mental imagery, it constitutes progress from mere perception to the creation of manipulable concepts that may increase the cognition abilities of intelligent vehicles, such as action-selection based on online imagery or developing improved sensorimotor abilities based on episodic simulation.

We achieve the conceptual representations by implementing an artificial neural model in line with the two aforementioned neurocognitive theories. We would like to note the term ''neural'' in artificial neural models by no means implies a faithful replication of the computations performed by biological neurons. On the contrary, the mathematics of deep learning shares little resemblance with the way the brain works [28], [29]. However, we identify two methods within the framework of artificial neural networks (ANNs) that appear, at least in part, rough algorithmic counterparts of the neurocognitive theories described above. Specifically, the CDZs may find a correspondence in the idea of convolutional autoencoders [30], while the predictive brain theory resonates with the adoption of Bayesian variational inference in combination with autoencoders [31], [32].

This work is part of the H2020 project Dreams4Cars,[1] aimed at developing an artificial driving agent inspired by the neurocognition of human driving [33]. In the following section, we further describe the objective of our work in more detail, and we discuss in §III the most significant related works. In §IV, we describe the implementation of 4 different neural models that successfully learn informative and compact representations. Lastly, Section §V presents the results of our models on the SYNTHIA dataset.

II. WHAT THIS PAPER IS (AND IS NOT) ABOUT

In the next section, we will review other works in the domain of autonomous driving that share objectives or methods with our proposal. Before that, we consider it useful to frame our proposal within the broader context of computer vision, trying to clarify similarities and differences between our approach and other relevant works in the domain of computer vision.

When looking at the results produced here, for example Fig. 7 and 9, it may seem that the outcome of our model is essentially image segmentation. Image segmentation is the process of partitioning of an image into meaningful subsets, and it has been one of the popular tasks in classical image processing [34]–[36] and continues to be a major topic in the era of deep learning for computer vision [37]–[40]. However, image segmentation has limited relevance to our work. Even if the outputs of the networks here presented indeed include the segmentation of cars and lanes, this is not the objective of the model.

[1] www.dreams4cars.eu

Our model aims at learning representations of the driving scenario that can be exploited for imagination in the driving context. We want these representations to be, first of all, meaningful. The representations must bear a semantic explanation, i.e., parts of the latent space are associated with concepts useful in the context of driving – `cars` and `lanes` in this case, but the work is open to further extensions such as `pedestrians` or `bikes`. The model learns these meaningful representations by exploiting semantic segmentation as a supporting task, as we will show in §IV-B, using a multi-decoder network which forces the partitioning of the internal representations into distinct concepts. In this context, segmentation can be therefore considered a practical way to achieve the separation of the semantic concepts in the latent space. Albeit this idea of partitioning the latent space may look as an expedient, we think however that it may be related to the notion of topographic organization largely present in the brain, where similar concepts are encoded in close groups of neurons [41]–[43].

Besides having a semantic organization, the representations learned by our model have a second important feature: they can be exploited for imagery, much like the brain's CDZs do, as described in §I. In this context, imagery is essentially constructing a static scene with attention to the conceptual entities considered – `cars` and `lanes` in our case. This process can result from a latent representation of a scenario seen before, or it can be triggered by a prediction of a future scenario based on past ones. It can also results from manipulating a latent space, generating scenarios the model has never seen before. In conclusion, now it is evident how semantic segmentation is just a byproduct of our entire model and not its primary focus.

Having clarified the role of segmentation in our work, we want to discuss the connection with another important machine learning domain called *self-supervision*. Unlike unsupervised learning, self-supervision is not motivated by biological plausibility; it is instead a way around the ever-present issue of manual data labeling in large datasets of images [44], [45]. Usually, self-supervision is realized by designing pretext tasks without any particular relevance for the agent but useful for the automatic generation of pseudo-labels. While learning to solve the pretext tasks, the model is forced to capture certain visual features of images that are ideally useful for the core task of the agent.

The computer vision community has proposed several kinds of creative pretext tasks for self-supervision. A prevalent task is *colorization* [46], where a color image is first converted to graylevel, and the model learns to reconstruct the color version. Another kind of task is solving jigsaw puzzles made from patches of the input image [47]. There are also self-supervision tasks that are indeed useful to the overall objective of the model, but the labeling is assumed by analytical methods [48]: a common example is the exploitation of the epipolar constrains in the stereo image pair as supervision for training a monocular image depth estimation model [49].

On the other hand, a small number of approaches exploit prediction as a self-supervision task. Our model adopts this idea, using prediction of future frames to bias the internal representation towards the ability to learn the dynamics of objects in the scene. In this sense, prediction for self-supervision shows a connection with the cognitive idea of *predictive brain* we mentioned before in §I.

Still, not all approaches maintain a sound cognitive account of prediction in the context of vision. For example, [50] arranges images in overlapping blocks by rows and columns, scanned in sequence with recursive networks attempting to ''predict'' the next block. This account of prediction is clearly an artifact with no correspondence in a cognitive agent. Instead, our work aims to include effective forms of prediction: prediction as imagination, and prediction as the construction of a probable future scenario. The DeepMind research group also widely adopts prediction for self-supervision [51]–[53] in a way more similar to ours.

One of the few works based on a cognitive account of prediction is the model proposed by Ha and Schmidhuber [54]. This model shares some fundamental components with our architectures: the use of variational autoencoders and recursive neural networks. There is, however, a significant difference in the objectives of the models. The work of Ha and Schmidhuber is a complete agent and includes other components not considered in our model, like a controller responsible for determining the course of actions of the agent. Their wider architecture comes at the expense of a very shallow perceptual capability. Much like complex neural networks of the past generation, this model is an interesting proof of concept working in synthetic simplified examples. The simple game-like scenario on which the model has been tested has an overly simplified visual appearance, not using perspective and very low resolution. Conversely, our aim is not training an agent, but learning the perceptual capability needed for visual imagery, including the projection of hypothetical driving scenarios in visual space.

III. RELATED WORKS

It is not uncommon for works adopting neural networks for perception in autonomous vehicles to declare virtues of a neurocognitive inspiration [55]–[57]. However, often these ideas do not transfer the specific brain mechanisms into algorithms. To the best of our knowledge, the two neurocognitive principles embraced by this work – Damasio's CDZs and Friston's predictive brain – have not been proposed in any work on perception for autonomous driving. Besides, the striking similarity between the formulation of brain predictivity given by Friston and the variational autoencoder algorithm seems to remain unnoticed, with few exceptions [58].

The idea of autoencoder has been at the heart of the ''deep'' turn of ANNs [59]–[61], and the variational version has rapidly gained attention [62]. Still, in the domain of autonomous vehicle perception, this architecture is not as popular as other approaches like *end-to-end*. In the end-to-end strategy, images from a front-facing camera are fed into a stack of convolutions, followed by feedforward layers which generate the low-level commands. The first attempt in this direction dates before the rise of deep learning [63], and it has been the groundwork for the later popular NVIDIA's PilotNet [13], [64]. One of the most severe drawbacks of end-to-end systems based on static frame processing is the erratic variation of steering wheel angle within short time periods. A potential solution is to provide a temporal context in the models, combining convolutions with recurrent networks [65].

Still, the most appealing feature of the end-to-end strategy – to dispense with internal representations – is also the primary source of its troubles. Learning the entire range of road scenarios from steering supervision alone, considering all possible appearances of objects relevant to the drive, is not achievable in practical settings. For this reason, several more recent proposals suggest the inclusion of intermediate representations, such as the so-called *mid-to-mid* strategy used in ChauffeurNet [66], Waymo's autonomous driving system. ChauffeurNet is essentially made of a convolutional network that consumes the input data to generate an intermediate representation with the format of a top-down view of the surrounding area and salient objects. Besides, ChauffeurNet has several higher-level networks that iteratively predict information useful for driving. Another work [67] proposes to overcome the object agnosticism of the end-to-end approach with an *object-centric* deep learning system for autonomous vehicles. In this proposal, a first convolutional neural module takes an image and produces an intermediate representation. Then, other downstream networks are diversified depending on a taxonomy of objects-related structures in the intermediate representation, and the structures are lastly converted into discrete driving actions. The system proposed by Valeo Vision also uses an internal representation [68] constructed using a standard ResNet50 model [69] with the top fully-connected layers removed. The feature representation is shared across many tasks relevant to visual perception in automated driving such as object detection, semantic segmentation, and depth estimation. All the downstream tasks are realized using the top parts of standard models like YOLO [70] for object detection or FCN-8 [37] for semantic segmentation.

None of the works reviewed so far builds the internal representations through the idea of the autoencoder. We found just two notable exceptions in the field of perception for autonomous driving. The first one is by the company *comma.ai* [71], where the latent representations of 2048 neurons are obtained with a variational autoencoder using input images of 160×80 pixels. Once trained, the latent representations are used for predicting successor frames in time with a recurrent neural network. The second exception is a work by Toyota in collaboration with MIT [72] and proposes a variational autoencoder learning representations of 25 neurons. The entire internal representations are decoded to restore the input image of 200×66 pixels as in a standard autoencoder. Besides, one neuron of the representation is

interpreted as steering angle, so end-to-end supervision for this neuron is mixed in the total training loss.

There are similarities between these last two approaches and the one we present, but also fundamental differences. The latent representation of Amini *et al.* [72] does not take into account the crucial time dimension of the perceptual driving scenario. On the other hand, Santana and Hotz [71] include their internal representation in a recursive network for prediction, but time dependency is not exploited when learning the compact representation. Moreover, the *comma.ai*'s model is agnostic about the meaning of the neurons composing the latent representation, while Amini *et al.* assign meaning to just the single neuron coding steering angles. We already discussed in §II how a key strategy of our model is to assign conceptual meaning to separate groups of neurons in the latent representation. In contexts different from autonomous vehicles, the idea is not new. For example, in computer vision, [73] proposed a work for the generation of head poses using a latent space with separate representations for viewpoints, lighting conditions, and shape variations. Also, in [74] the latent vector is partitioned in semantic content and geometric coding. We will show in IV-B how our partitioning of the latent spaces differs from these approaches.

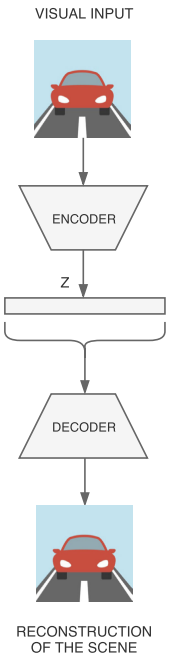

FIGURE 2. Architecture of our variational autoencoder (*Net1*).

IV. THE NEURAL MODELS

In this section, we present the details of our approach. We propose a model composed of two different networks: a first network generates compact representations of visual scenarios; a second network manipulates the latent vectors to predict future scenarios and to perform a rudimentary form of mental imagery.

Concerning the first part of the model, we have experimented some different architectures, all sharing the common feature of a hierarchical arrangement similar to the CDZs in the brain and following the strategy described in §I and §II. We compare three of these architectures, and each can be interpreted as a step forward in developing a more sophisticated way to learn the internal representations. Note that this series of steps can be interpreted as the opposite of what is commonly referred to as "ablation study".

To summarize, here we present:

- three different autoencoder networks (*Net1, Net2, Net3*) with increasingly sophisticated approaches to learning internal representations of the driving scenario;
- a recurrent neural network (*Net4*) which performs predictions and imagery, working exclusively with the latent representations created by the previous networks.

A. NET1: VARIATIONAL AUTOENCODER

The first model we present is essential. When talking about representation learning, the first architecture that comes to mind is the *autoencoder*. This is the simplest model of the family, composed of two sub-networks:

$$g_\Phi : \mathcal{X} \to \mathcal{Z},$$
$$f_\Theta : \mathcal{Z} \to \mathcal{X}.$$

The first sub-network is called *encoder* and computes the compact representation $\mathbf{z} \in \mathcal{Z}$ of a high-dimensional input $\mathbf{x} \in \mathcal{X}$. This network is determined by its set of parameters Φ. The second sub-network is the *decoder* (often called the *generative* network) which reconstructs the high-dimensional data $\mathbf{x} \in \mathcal{X}$ from the low-dimensional compact representation $\mathbf{z} \in \mathcal{Z}$. This network is determined by the set of parameters Θ. When training the autoencoder, the parameters Θ and Φ are learned by minimizing the error between input samples $\mathbf{x}_i$ and the outputs $f(g(\mathbf{x}_i))$.

A substantial improvement in the architecture of autoencoders comes with the integration with variational Bayesian methods. We refer to Appendix VI for a detailed mathematical definition. The variational autoencoder can learn a more ordered representation compared to the standard autoencoder. However, there is a much space for improvements, especially in our case where we want to focus only on learning representations of driving scenarios. Therefore, we present here our implementation of variational autoencoder mainly as a comparison with the next models.

Fig. 2 depicts the architecture of our variational autoencoder (*Net1*), while Table 6 shows the numbers of layers and the parameters adopted in the final version of the model. The input of the network is a single RGB image of 256 × 256 pixels. The encoder is composed of a stack of 4 convolutions and 2 fully-connected layers, converging to a latent space of 128 neurons. The decoder has a structure symmetric to the encoder, mapping the 128 neurons back to an image of 256 × 256. The network is trained to optimize the loss function in equation (15) in a totally unsupervised way.

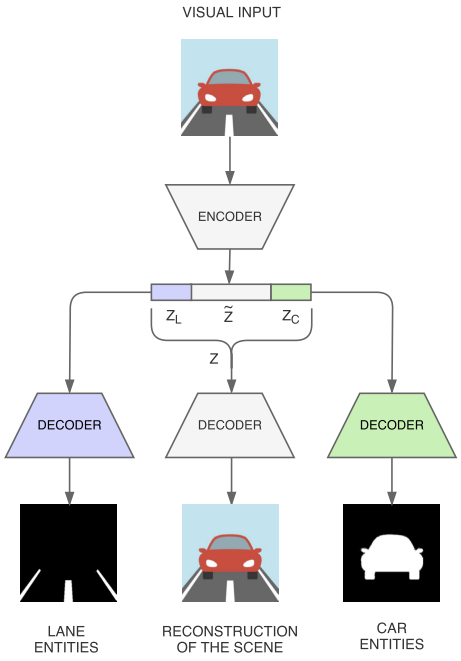

FIGURE 3. Architecture of our topological autoencoder (*Net2*), where the green color denotes the `cars` concept, and violet the `lanes` concept.

B. NET2: TOPOLOGICAL AUTOENCODER

The next model we present shares most of its architecture with the previous one. The crucial improvement is the introduction of a semantic organization in the latent spaces. As discussed in §I, the human brain projects sensory information – especially visual – into compact representations through the CDZ structures. Some of these representations constitute the *conceptual space*, where neural activations encode the entities in the environment that produced the perceptual stimuli. We can take inspiration from this theory and use the hierarchical architecture of CDZs as a "blueprint" to design a more sophisticated neural network, which can learn representations that are not only in terms of visual features but also in terms of useful *concepts*.

In the driving context, the entire road scenario is informative. However, from a conceptual point of view, it is not immediately necessary to infer categories for every entity present in a scene. Within the aims and limits of this paper, it is useful to project in conceptual space the entities mostly relevant to the driving task. Therefore, for simplicity in this model, we choose to consider the two main concepts of `cars` and `lanes`.

Fig. 3 presents the architecture of the topological autoencoder (*Net2*), composed of one shared encoder and three independent decoders. The choice of parameters is similar to *Net1*, as Table 7 shows. The encoder and each of the three decoders maintain the same structure as in *Net1*, and the size of the latent space remains unchanged. Still, the internal organization of the latent space is forcefully partitioned. The grey decoder of Fig. 3 works in the visual space – just like the decoder of Fig. 2 – mapping all the 128 neurons of the latent vector $\mathbf{z}$ altogether back into an RGB image. This decoder learns to reconstruct the input image and is trained in an unsupervised way. Instead, the decoder colored in green takes only a sub-vector $\mathbf{z}_C$ of 16 neurons from the latent space and produces a matrix $\mathbf{x}_C$ of 256×256 probability values. The sub-vector of 16 neurons is trained to represent the `cars` concept, and the output matrix can be interpreted as a semantic segmentation of the input image, where values indicate the probability of the presence of `cars` entities. Similarly, the violet decoder maps only a sub-vector $\mathbf{z}_L$ of 16 neurons representing the `lanes` concepts into a probability matrix $\mathbf{x}_L$ for `lanes` entities. These two decoders require supervised learning: their output is converted into binary images by applying a threshold, and trained to minimize the reconstruction error with semantic segmentation of the input images. As we mentioned in §II, the segmentation here can be considered a mere byproduct of the network, and the goal remains the meaningful latent representations.

We already discussed in §III that the idea of partitioning the latent vector into semantic components is not new. However, our approach is different: while we keep the two segments $\mathbf{z}_C$ and $\mathbf{z}_L$ disjointed, the entire $\mathbf{z}$ learns representations in the visual space. That is why the grey decoder of Fig. 3 takes as input the entire latent space. In this way, we try to adhere to the CDZ theoretical idea, as we explicitly force the network to pay attention to the `cars` and `lanes` entities in the environment. Another advantage of our approach in partitioning the latent space concerns the well-known crucial issue of lack of transparency in deep neural networks. In most models, no information is available about what exactly makes the models arrive at their predictions [75], [76]. We can mitigate the issue by explicitly assigning semantic meaning to the components of the inner representation.

To give a mathematical description of the model, it is composed of four sub-networks:

$$
\begin{aligned}
g_\Phi &: \mathcal{X} \to \mathcal{Z}, \\
f_{\Theta_V} &: \mathcal{Z} \to \mathcal{X}, \\
f_{\Theta_C} &: \mathcal{Z}_C \to \mathcal{X}_C, \\
f_{\Theta_L} &: \mathcal{Z}_L \to \mathcal{X}_L,
\end{aligned}
$$

with $\mathcal{Z} = \mathbb{R}^{N_V}$, $\mathcal{Z}_C = \mathbb{R}^{N_C}$ and $\mathcal{Z}_L = \mathbb{R}^{N_L}$. The subscript V denotes the visual space, and the subscripts C and L refer to the `cars` and `lanes` concepts respectively. For each latent vector $\mathbf{z}$ we have:

$$
\mathbf{z} \in \mathcal{Z} = [\mathbf{z}_C, \tilde{\mathbf{z}}, \mathbf{z}_L],
$$

where $\mathbf{z}_C$ and $\mathbf{z}_L$ are the two sub-vectors representing the `cars` and `lanes` concepts, respectively. The segment in between, $\tilde{\mathbf{z}}$, encodes the remaining generic visual features, while the entire latent vector $\mathbf{z}$ is a representation in the visual space. The final version of the model has $N_V = 128$ and $N_C = N_L = 16$. We will discuss this choice in §V-B, while other parameters and learning rate are included in Tables 7 and 8.

By calling $\Theta = [\Theta_V, \Theta_C, \Theta_L]$ the vector of parameters of all decoders, the loss functions of the model can be derived from the basic equation (15). At each batch iteration b, a random batch $\mathcal{B} \subset \mathcal{D}$ is presented, and the following loss is computed:

$$\mathcal{L}(\Theta, \Phi|\mathcal{B}) = E_K + \lambda_V E_V + \lambda_C E_C + \lambda_L E_L, \tag{1}$$

where

$$E_K = \left(1 - (1 - k_0)\kappa^b\right) \sum_{\mathbf{x}}^{\mathcal{B}} \Delta_{KL}\left(q_\Phi(\mathbf{z}|\mathbf{x}) \| p_{\Theta_V}(\mathbf{z})\right),$$

$$E_V = -\sum_{\mathbf{x}}^{\mathcal{B}} \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log p_{\Theta_V}(\mathbf{x}|\mathbf{z})\right],$$

$$E_C = -\sum_{\mathbf{x}}^{\mathcal{B}} \mathbb{E}_{\mathbf{z}_C\sim \Pi_C(q_\Phi(\mathbf{z}|\mathbf{x}))}\left[\log \widetilde{p}_{\Theta_C}(\mathbf{x}_C|\mathbf{z}_C)\right],$$

$$E_L = -\sum_{\mathbf{x}}^{\mathcal{B}} \mathbb{E}_{\mathbf{z}_L\sim \Pi_L(q_\Phi(\mathbf{z}|\mathbf{x}))}\left[\log \widetilde{p}_{\Theta_L}(\mathbf{x}_L|\mathbf{z}_L)\right].$$

Few observations are due for the differences between this loss function (1) and the basic one (15). First of all, we apply a delay in the contribution of the Kullback-Leibler divergence in the term E_K. This strategy is called *KL annealing* and was first introduced in the context of variational autoencoders for language modeling [77]. The reason is the encoder at the beginning of training is unlikely to provide any meaningful probability distribution $q_\Phi(\mathbf{z}|\mathbf{x})$. Therefore, there is a cost factor for the KL component, which is set initially at a small value k_0 and gradually increased up to 1.0 with a time constant κ.

A second difference in the loss function are the terms E_V, E_C, E_L. They represent the reconstruction errors of the visual scenario and the conceptual entities. The term E_V computes the error in the visual space using the entire latent vector $\mathbf{z}$, and it corresponds precisely to the second component in the basic loss (15). The other two terms E_C and E_L compute the error in the conceptual space and are slightly different. Only the relevant portion of the latent vector is considered, as symbolized by the projection operators Π_C, Π_L.

Another difference is the use of a variant of the cross entropy in E_C, E_L, indicated with the symbols $\widetilde{p}_{\Theta_C}$ and $\widetilde{p}_{\Theta_L}$. This variant takes into account the large unbalance between the number of pixels belonging to a concept and all the other pixels, which is typical in ordinary driving scenes. Following the method first introduced in the context of medical image processing [78], we compensate this asymmetry by weighing the contribution of true and false pixels with P, the ratio of true pixels over all the pixels in the dataset, computed as follows:

$$P = \left(\frac{1}{NM}\sum_{j}^{M}\sum_{i}^{N} y_{i,j}\right)^{\frac{1}{s}}, \tag{2}$$

where M is the number of images in the dataset, N is the number of pixels in an image, and s is a parameter used to smooth the effect of weighting by the probability of ground truth: a value evaluated empirically as valid is 4. The term $y_{i,j}$ is the value of the i-th pixel (in a flatten order) of the j-th target image of the dataset. We use a set of target images for each semantic concept. Hence, we have a set of `car` labels composed of binary images where white pixels indicate the presence of cars in the scene, and a set of `lane` labels where white pixels correspond to lane markings.

Lastly, in the loss equation (1) the contributions of the terms E_V, E_C, E_L are weighted by the parameters $\lambda_V, \lambda_C, \lambda_L$. The purpose of these parameters is mainly to normalize the range of the errors, which varies widely from visual space to conceptual spaces. For this reason, typically $\lambda_V \neq \lambda_C = \lambda_L$.

C. NET3: TEMPORAL AUTOENCODER

The next model is the final step in our development of an autoencoder able to learn meaningful representations of the driving scenario. We made it clear in §I that our work aims to learn representations oriented to the driving task from a static and a dynamic perspective. In *Net2*, we include the static perspective, i.e., a conceptual organization of the latent representations. In our third model, *Net3*, we also include the dynamic perspective by forcing a temporal consistency in the representations.

We achieve representations consistent in the temporal dimension with the inclusion of a recursive module in the architecture of *Net2* and the use of self-supervision, as already mentioned in §II. In this way, the model learns how the concepts represented in the latent space will change in future driving scenarios. However, the predictions this model can make are still short-term, whereas longer-term predictions will be the subject of *Net4*.

Fig. 4 shows the architecture of *Net3*, and Table 9 describes the parameters of the final model. The model shares substantially the same architecture of *Net2*, except for an additional module based on a simple recursive neural network. The training procedure, however, is significantly different from the previous network. Let us introduce the notation $\mathbf{x}^{(t)}$ to indicate the frame t steps ahead of frame $\mathbf{x}$. Similarly, $\mathbf{z}^{(t)}$ refers to the latent representation of the image t steps ahead of that represented by $\mathbf{z}$. At each iteration of the training, the inputs of the model are two consecutive frames $\mathbf{x}$ and $\mathbf{x}^{(1)}$, which are fed to the common encoder. The encoder computes two latent representations $\mathbf{z}$ and $\mathbf{z}^{(1)}$, which are passed to a RNN trained to predict the latent vector $\mathbf{z}^{(2)}$ containing the representation of the successive frame in the sequence. Then, all three latent vectors are expanded using the same 3-decoders structure already seen in *Net2*, so that the overall model is trained to generate visual and segmented output images for the three frames $\mathbf{x}$, $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$.

The novel recursive sub-network of the model can be described by the function:

$$h_\Psi\left(\mathbf{z}, \mathbf{z}^{(1)}\right) \rightarrow \widetilde{\mathbf{z}} \approx \mathbf{z}^{(2)}. \tag{3}$$

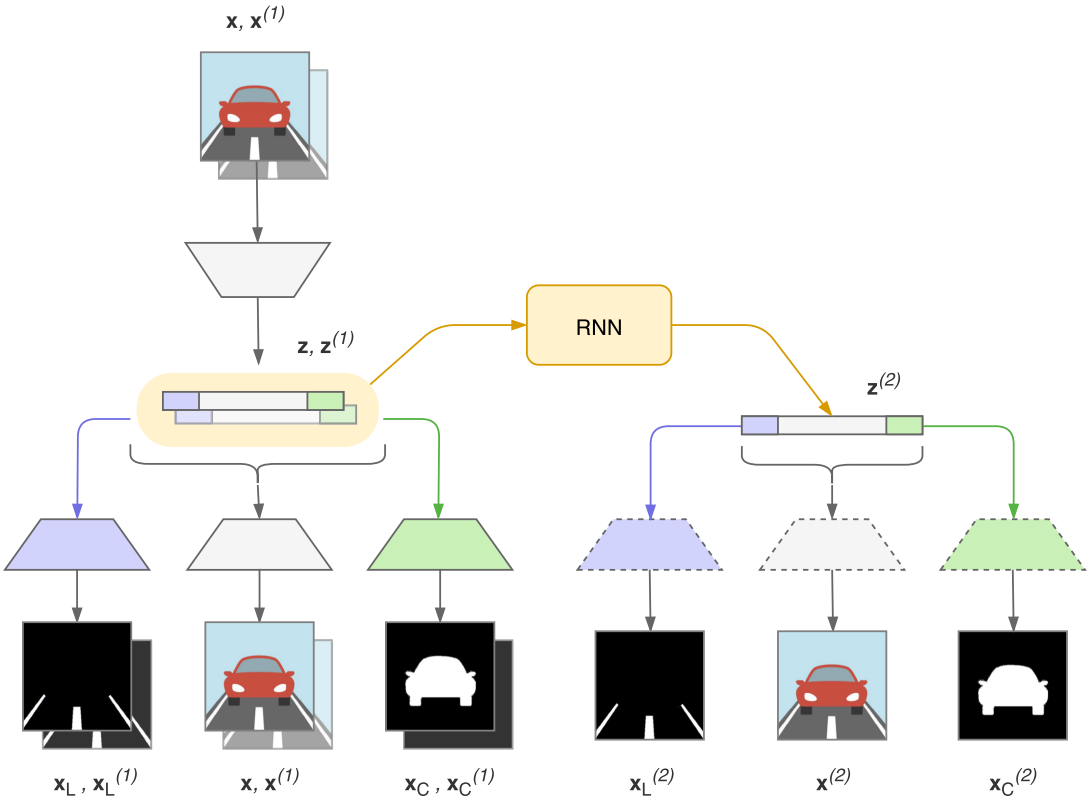

FIGURE 4. Architecture of our temporal autoencoder (*Net3*) in its final version, where the green color denotes the `cars` concept and violet the `lanes` concept. The decoders with dashed-line border are the same instances of the decoders with solid-line border.

This module is implemented using a basic recursive neural network (RNN) [79] with a time window of 2 and a set of parameters Ψ.

The formulation of the loss used for training the network is similar to equation (1) with additional terms for the recursive prediction:

$$\mathcal{L}(\Theta, \Phi, \Psi|\mathcal{B}) = \mathcal{L}(\Theta, \Phi|\mathcal{B}) + E' + E'', \tag{4}$$

where the first term is the same loss of equation (1) and the additional terms are:

$$E' = \lambda'_{\mathrm{V}} E'_{\mathrm{V}} + \lambda'_{\mathrm{C}} E'_{\mathrm{C}} + \lambda'_{\mathrm{L}} E'_{\mathrm{L}},$$
$$E'' = \lambda''_{\mathrm{V}} E''_{\mathrm{V}} + \lambda''_{\mathrm{C}} E''_{\mathrm{C}} + \lambda''_{\mathrm{L}} E''_{\mathrm{L}}.$$

For the sake of legibility, let $\tilde{\tilde{\mathbf{z}}} = h_\Psi\left(\mathbf{z}, g_\Phi(\mathbf{x}^{(1)})\right)$. The expressions of the remaining terms are the following:

$$E'_{\mathrm{V}} = -\sum_{\mathbf{x}}^{\mathcal{B}} \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x}^{(1)})}\left[\log p_{\Theta_{\mathrm{V}}}\left(\mathbf{x}^{(1)} \mid \mathbf{z}\right)\right],$$
$$E'_{\mathrm{C}} = -\sum_{\mathbf{x}}^{\mathcal{B}} \mathbb{E}_{\mathbf{z}_{\mathrm{C}}\sim \Pi_{\mathrm{C}}(q_\Phi(\mathbf{z}|\mathbf{x}^{(1)}))}\left[\log \widetilde{p}_{\Theta_{\mathrm{C}}}\left(\mathbf{x}^{(1)}_{\mathrm{C}} \mid \mathbf{z}_{\mathrm{C}}\right)\right],$$
$$E'_{\mathrm{L}} = -\sum_{\mathbf{x}}^{\mathcal{B}} \mathbb{E}_{\mathbf{z}_{\mathrm{L}}\sim \Pi_{\mathrm{L}}(q_\Phi(\mathbf{z}|\mathbf{x}^{(1)}))}\left[\log \widetilde{p}_{\Theta_{\mathrm{L}}}\left(\mathbf{x}^{(1)}_{\mathrm{L}} \mid \mathbf{z}_{\mathrm{L}}\right)\right],$$

$$E''_{\mathrm{V}} = -\sum_{\mathbf{x}}^{\mathcal{B}} \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log p_{\Theta_{\mathrm{V}}}\left(\mathbf{x}^{(2)} \mid \tilde{\tilde{\mathbf{z}}}\right)\right],$$
$$E''_{\mathrm{C}} = -\sum_{\mathbf{x}}^{\mathcal{B}} \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log \widetilde{p}_{\Theta_{\mathrm{C}}}\left(\mathbf{x}^{(2)}_{\mathrm{C}} \mid \Pi_{\mathrm{C}}(\tilde{\tilde{\mathbf{z}}})\right)\right],$$
$$E''_{\mathrm{L}} = -\sum_{\mathbf{x}}^{\mathcal{B}} \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log \widetilde{p}_{\Theta_{\mathrm{L}}}\left(\mathbf{x}^{(2)}_{\mathrm{L}} \mid \Pi_{\mathrm{L}}(\tilde{\tilde{\mathbf{z}}})\right)\right].$$

The contributions of the terms $E'_{\mathrm{V}}, E'_{\mathrm{C}}, E'_{\mathrm{L}}$ is similar to that of $E_{\mathrm{V}}, E_{\mathrm{C}}, E_{\mathrm{L}}$, as they represent the errors in the reconstruction of the frame successor of $\mathbf{x}$. The temporal coherence is measured by the terms $E''_{\mathrm{V}}, E''_{\mathrm{C}}, E''_{\mathrm{L}}$ representing the error between the frame 2 steps ahead of $\mathbf{x}$ and the images decoded from the latent vector predicted by the recursive sub-network h_Ψ.

D. NET4: RECURRENT NETWORK

The last network we present is an example of how the results of the previous models can be exploited to perform long-term prediction of driving scenarios. The previous three sections (§IV-A to §IV-C) describe the steps we made towards the design of a model able to learn latent representations that are both conceptually organized and temporally consistent. *Net3* is the result of this development.

Once trained, *Net3* can be deployed in its encoding part to generate a latent representation of any visual driving scenario.

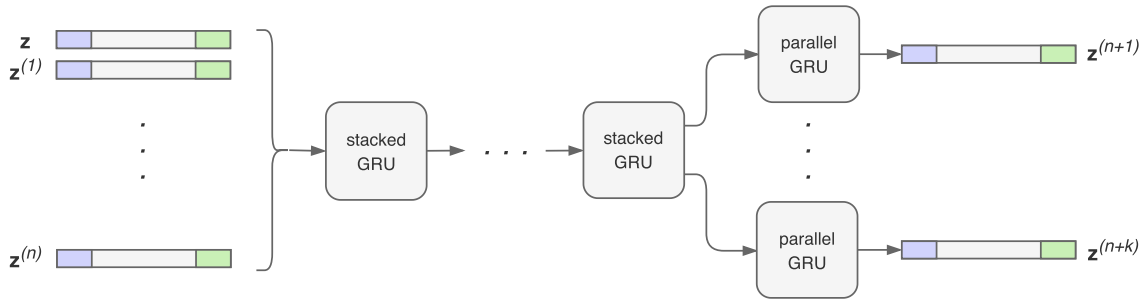

FIGURE 5. Architecture of our recurrent model (*Net4*), where the green color denotes the `cars` concept and violet the `lanes` concept.

FIGURE 6. Samples from one of the tracks in the SYNTHIA dataset. The images show the different results of rendering the same track using different environmental and lighting conditions.

Therefore, the long-term prediction can be realized by working entirely in the latent space. The advantage of having a compact latent representation allows the recurrent network to have a complex architecture with a limited number of parameters.

Fig. 5 shows the proposed recurrent network (*Net4*), which has a first module composed of multiple levels of stacked recurrent sub-networks, one for each latent vector in the input sequence. A second module is composed of multiple parallel recurrent sub-networks predicting successive latent vectors in the sequence. In the first module, each stacked sub-network sends its entire output sequence to the next sub-network input. In the second module, instead, the parallel sub-networks yield only the last output in the time sequence. All the sub-networks of the model share the same core architecture implemented with Gated Recurrent Units (GRUs) [80], and we will discuss this choice in V-C.

The overall model can be described by the function:

$$r_{\Xi} : \mathcal{Z}^{N_\mathrm{I}} \rightarrow \mathcal{Z}^{N_\mathrm{O}}, \tag{5}$$

$$r_{\Xi}\left(\mathbf{z}, \mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(N_\mathrm{I}-1)}\right) \rightarrow \left[\widetilde{\mathbf{z}}_1, \widetilde{\mathbf{z}}_2, \cdots, \widetilde{\mathbf{z}}_{N_\mathrm{O}}\right] \approx \left[\mathbf{z}^{(N_\mathrm{I})}, \mathbf{z}^{(N_\mathrm{I}+1)}, \cdots, \mathbf{z}^{(N_\mathrm{I}+N_\mathrm{O})}\right], \tag{6}$$

where N_I is the length of the input sequence, N_O is the length of the future predicted sequence, and Ξ is the set of parameters of the model. In the final version of the model, we choose $N_\mathrm{I} = 8$ and $N_\mathrm{O} = 4$. We use 2 stacked GRUs and 4 parallel GRUs, as described in Table 10.

Lastly, we want to note that this model does not use any odometry or other kind of information for the prediction, just the rich representation learned by the accompanying autoencoder.

V. RESULTS

In the last section of our paper, we present and discuss the results obtained by our models. We first spend a few words about the dataset adopted in this work. Then, we show qualitative and quantitative results for two of the autoencoder networks we implemented (*Net2* and *Net3*) and for the recurrent network (*Net4*). Lastly, we show further evaluation on the latent representations learned by the different autoencoder networks.

A. DATASET

The SYNTHIA dataset [81] consists of a large collection of photo-realistic video sequences rendered using the game engine Unity. It comprises about 100, 000 images of urban scenarios recorded from a simulated camera placed on the windshield of the ego car. Each video sequence is acquired at 5 FPS and comes with semantic annotations or several classes, including lane markings, which are not commonly found in other datasets.

Despite being artificially generated, this dataset offers a wide variety of reasonably realistic illumination and weather conditions, occasionally resulting even in very adverse driving conditions. The dataset features 5 sets of driving sequences. Each set contains about 10 recordings of the same track rendered under different environmental conditions: traffic, weather, season, and time of the day. Fig. 6 gives an example of the variety of data coming from the same driving sequence with different conditions. Moreover, the tracks are very diverse as well, including freeways, tunnels, congestion, ''NewYork-like cities'', and ''European towns'' – as the creators of the dataset describe it.

We randomly allocated 70% of the video sequences to the training set, 25% to the validation, and 5% to the test set, ensuring no overlap among the three sets. For a more interesting visualization of the results, we further organize

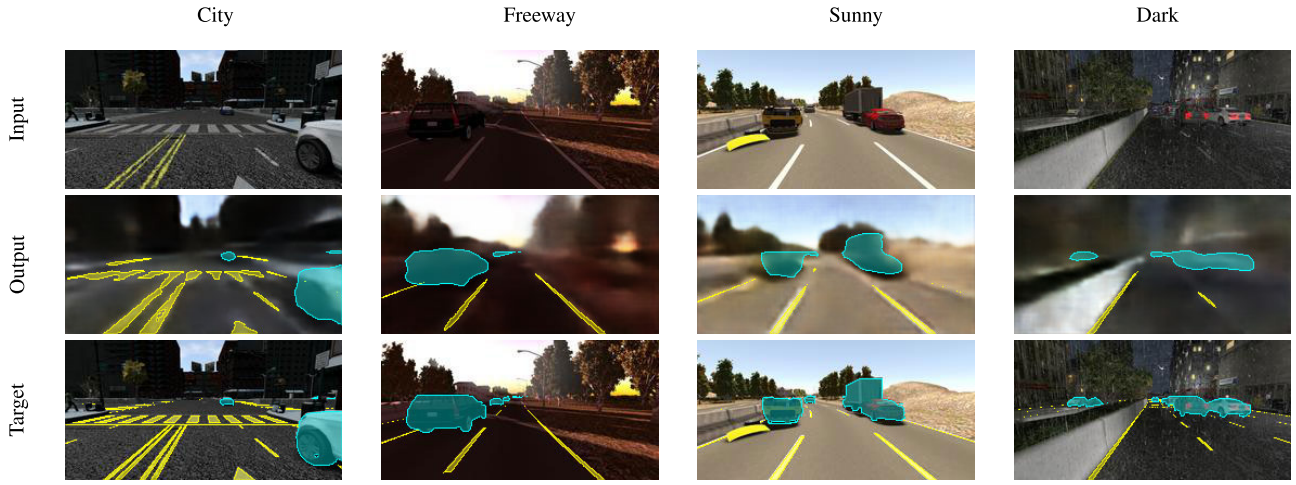

FIGURE 7. Results of *Net3* in reconstructing an image and its `cars` and `lanes` entities. In the first row, the input frames belonging to different categories of driving conditions. In the center row, the output of the network. In the last row, the same input frames plotted with a colored overlay showing the target `cars` entities in cyan and the `lanes` entities in yellow.

the test set into four (overlapping) categories, based on the driving scenarios: urban environments, freeways, sunny conditions, and darkness or adverse weather conditions.

B. RESULTS OF NET2 AND NET3

We present the results of the two autoencoders – *Net2* and *Net3* – we described in §IV-B and §IV-C. The networks are trained for 200 epochs in their final version. Note that here we omit the results of *Net1* since it lacks any conceptual information. However, in §V-D we will include a comparison of all three networks based on their latent representations.

First, we present some quantitative results obtained by the models when reconstructing an image and its `cars` and `lanes` entities, measured with the IoU *(Intersection over Union)* metrics. Table 1 displays the scores for the `cars` and `lanes` classes grouped into the four driving conditions mentioned above. The Table also includes the general scores on the entire test set. We compare the performance of our two autoencoders with two other well-known models[2] for pure semantic segmentation, FCN-8 [37] and U-Net [82] (both using VGG-16 as base model). The scores show how *Net3* can learn a more consistent latent representation compared to *Net2* and the FCN-8 model, in all the categories of driving sequences. The U-Net model outperforms all other models, although the scores are still comparable. However, for both *Net2* and *Net3*, it is evident how the task of recognizing the `cars` concept achieves better scores compared to the `lanes` concept. An explanation of why the latter task is more difficult can be the very low ratio of pixels belonging to the class of `lanes` over the entire image size, and consequently how easily the lane markings get occluded by other elements in the scene.

We would like to stress again that the purpose of our networks is not mere segmentation of visual input, as we discussed in §II. The segmentation operation must be considered a supporting task, forcing the model to learn a semantic organization of its internal representations, which is totally missing in the U-Net and FCN-8 models.

Second, we present two different qualitative results. Fig. 7 shows the images produced by *Net3* for four different images of the test set, one for each driving condition. Given an input image (showed on the top row of the Figure), the network produces its corresponding latent representation. The latent vector is passed to the three decoders to reconstruct the initial image and to extract the `cars` and `lanes` entities in the scene (center row of the Figure). For easy visualization, we show the output of the three decoders as a single image, having as background the reconstruction in the visual space, and as colored overlays the segmented entities of `cars` (in cyan) and `lanes` (in yellow). The images on the bottom row of the Figure are displayed as a reference, showing the target images with the colored overlays of the two classes.

Another qualitative result of *Net3* comes from interpolating between different latent spaces. In Fig. 8, each column shows what happens when taking the latent representation of a first frame (first row in the Figure) and linearly interpolate it with the latent representation of a second frame (last row). We generate 5 intermediate latent vectors, passed to the decoders of *Net3* to produce novel frames. The images prove to be a smooth and gradual shift from the first input to the second, and they successfully provide new plausible driving scenarios not seen before by the network.

C. RESULTS OF NET4

Here we show the results of our recursive model *Net4*, trained for 100 epochs on a corresponding dataset of latent representations computed by our most advanced autoencoder *Net3* over the initial SYNTHIA dataset.

Starting with the quantitative results, Table 2 contains the IoU scores obtained by the model in the different categories of driving sequences used before. As described in §IV-D, the network takes as input a sequence of 8 frames and predicts

[2]We used the Keras implementations available at https://github.com/divamgupta/image-segmentation-keras

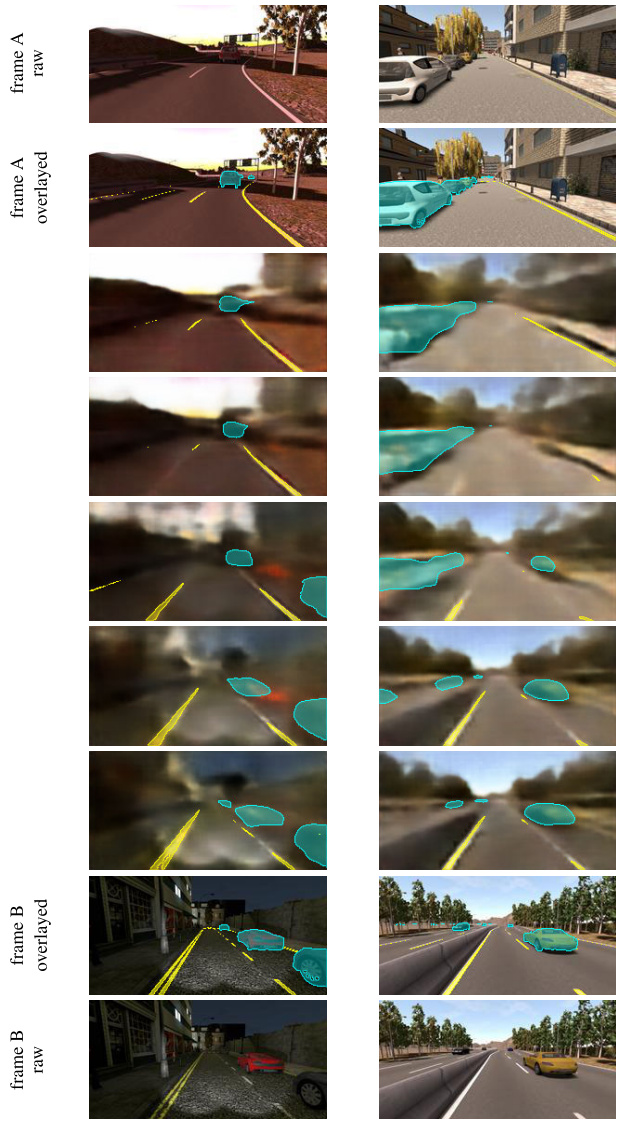

FIGURE 8. Two examples of interpolation between latent representations learned by *Net3*. The first 2 rows show the first input frame, with and without the colored overlay highlighting the `cars` and `lanes` entities. Similarly, the last 2 rows show the second input frame. The 5 central rows are the result of the linear interpolation between the latent representations of the two inputs.

the 4 subsequent frames. Since the SYNTHIA sequences are acquired at 5FPS, the network is predicting 0.8 seconds in the future. The table shows the scores for the 4 predicted frames, separated as usual in the `cars` and `lanes` classes. It is immediate to note the `cars` scores are always higher than the `lanes` scores, just like we saw in Table 1. However, the `cars` predictions worsen more significantly for the distant frames with a decay of 16%, while the `lanes` scores lose only 9%. This result can be explained by the fact that, generally, in a driving sequence, the lane markings change more smoothly and predictably compared to the cars, which can suddenly change their trajectory.

Another quantitative comparison is presented in Table 3, where we compare different implementations of *Net4* based on the type of internal recursive node: basic RNNs [79], GRUs [80] and LSTMs [83]. The results indicate the GRUs are the best choice in our case. While it is not surprising that the basic RNNs obtain the lowest score, the fact that GRUs outperform LSTMs might seem unexpected. We believe the reason is twofold: first, the number of parameters in the overall model increases by more than 30% when switching from GRUs to LSTMs. Second, although it is well known that LSTMs are the most powerful recursive node for long-term prediction because of their ability to keep track of events in the remote past, in the context of driving is not so crucial to memorize scenarios occurred several seconds before. While driving, the environment and the surrounding vehicles change continuously. It is often useless to try to draw a connection between the current scenario and, for example, the one seen 10 seconds before – note that the typical timescale of vehicle dynamics is less than one second. This situation is clearly opposite to Natural Language Processing, where LSTMs give their best.

As regards qualitative results, Fig. 9 shows four examples of visual predictions, one for each category of driving conditions. We include in the Figure the 4 predicted frames and their corresponding target frames, but we omit to show the 8 input frames to keep the Figure easy to read. The results in the "freeway" and "sunny" scenarios demonstrate that the model can predict an overtake maneuver from the left as well as from the right. Another interesting result is the different kind of predictions when facing a crosswalk: in the "city" scenario there is a car moving perpendicularly to the lane of the ego car, so the network correctly predicts to hold still at the crosswalk; in the "dark" scenario cars are driving in the same direction of the ego car, so the model predicts not to stop at the crosswalk and moves forward.

As a final qualitative evaluation, we try to replicate the phenomenon of mental imagery using *Net4*. To mimic this process, the network is called iteratively, and at each iteration, the output is fed back as the input of the next iteration. In our specific case, we choose to take the 1st of the 4 output vectors and use it as the 8th input vector of the next iteration. Fig. 10 presents the results of 9 iterations of imagery for two different scenarios, along with the corresponding reference frames (the input images are, again, omitted for practical reasons). Note that, while the imagery process must inevitably start with all input frames taken from the original dataset, the results provided in the Figure are obtained from forward iterations, that is when the network computes all input vectors as results of previous iterations. In both driving scenarios, it is possible to appreciate how the model can predict a quite plausible future from just its own representations of the world.

D. LATENT REPRESENTATIONS

We conclude our paper with a few more words on the latent representations learned by our autoencoders with a quantitative evaluation of their temporal consistency and a qualitative visualization of their conceptual organization.

First of all, let us justify the title of our paper. Table 4 shows the impact of the sizes N_C and N_L on the performance of

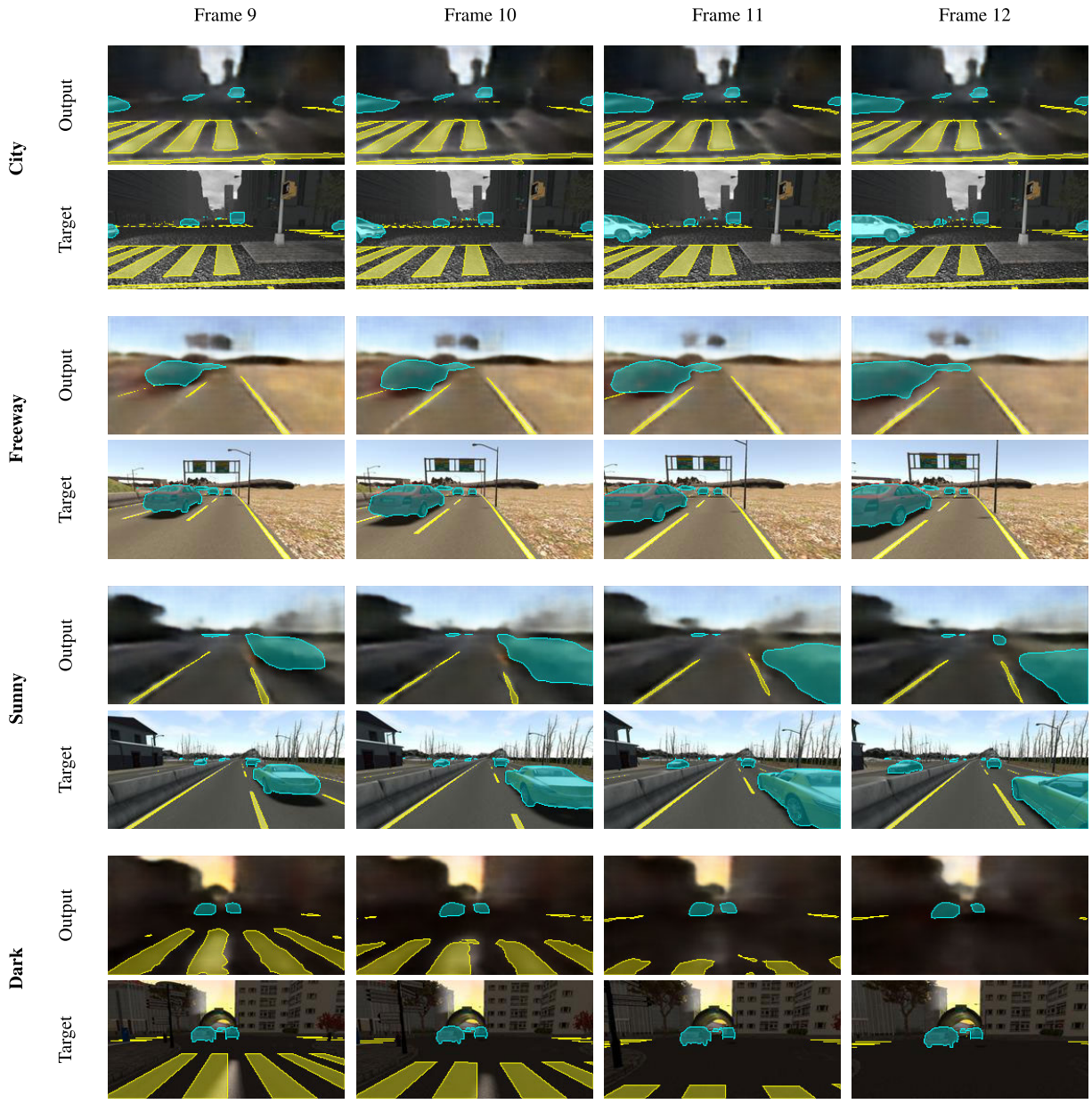

FIGURE 9. Results of *Net4* in predicting 4 future frames from an input sequence of 8 frames, under four different driving conditions. Odd rows show the output of the network, even rows the corresponding target frames.

TABLE 1. IoU scores of `cars` and `lanes` classes organized into four categories of driving conditions. We compare our two autoencoders with two other well-known models for pure semantic segmentation.

	Net2		Net3		FCN-8		U-Net	
	IoU car	IoU lane	IoU car	IoU lane	IoU car	IoU lane	IoU car	IoU lane
City	0.7834	0.6487	0.8305	0.7155	0.8033	0.6109	0.8552	0.7451
Freeway	0.7755	0.5840	0.7952	0.7490	0.7587	0.6959	0.7975	0.8666
Sunny	0.7736	0.6283	0.8077	0.6970	0.7741	0.6652	0.8351	0.8128
Dark	0.7682	0.6274	0.7943	0.7116	0.7450	0.6385	0.7914	0.7927
All	0.7702	0.6277	0.7992	0.7062	0.7558	0.6484	0.8076	0.8001

Net2, which are the number of neurons in the latent space representing the `cars` and `lanes` concepts, as defined in §IV-B. In the final version of the model, we choose to have $N_C = N_L = 16$, even if this does not correspond to the best IoU score. The reason we prefer having a latent representation of concepts as compact as possible is twofold: first, with

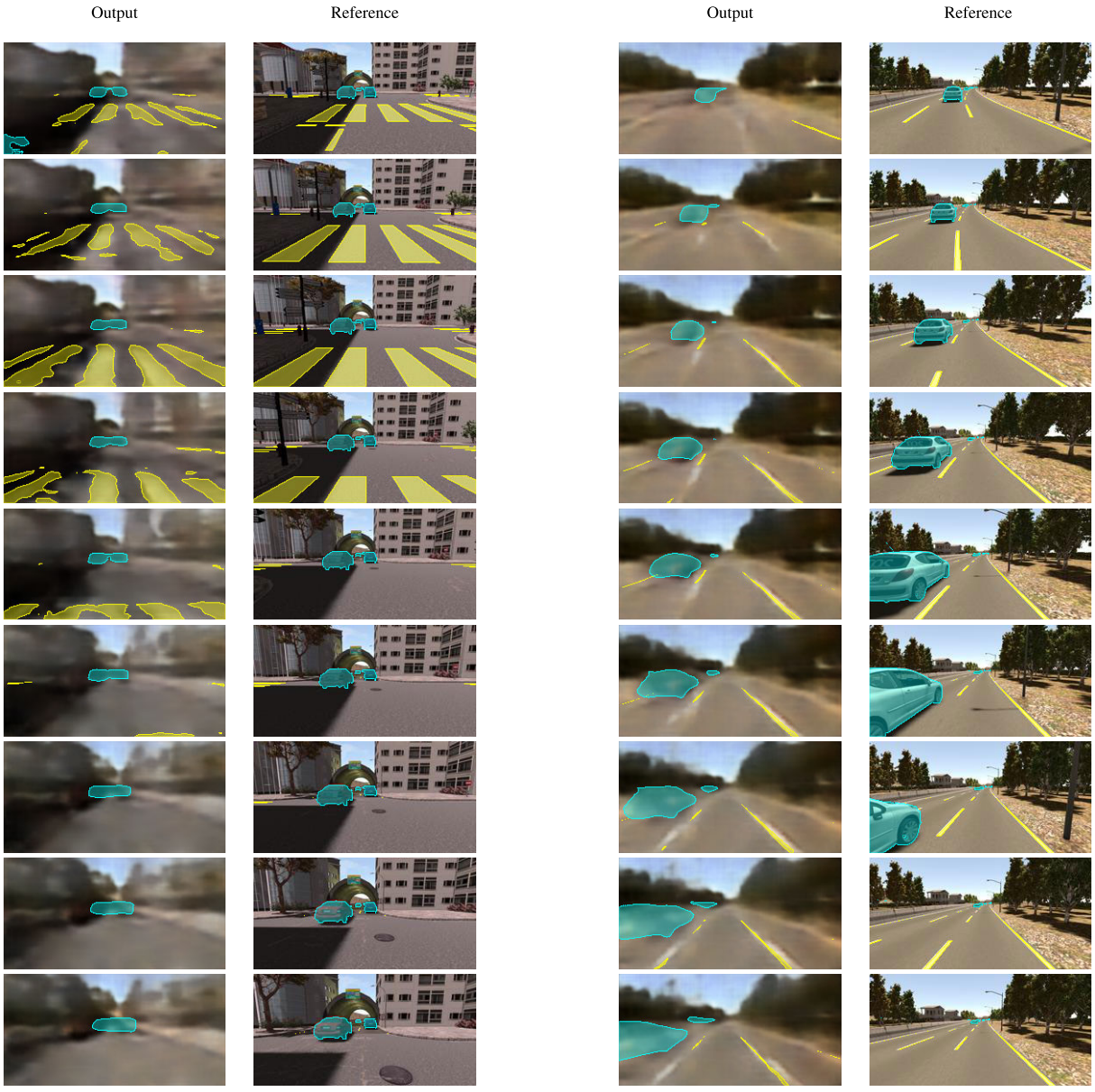

FIGURE 10. Examples of mimicking mental imagery with *Net4*, for two different driving scenarios. Odd columns show the result coming from the model, even columns are a reference of the corresponding frames.

TABLE 2. IoU scores of `cars` and `lanes` classes obtained by *Net4* when predicting 4 future frames, starting from an input sequence of 8 frames.

	Frame 9		Frame 10		Frame 11		Frame 12	
	IoU car	IoU lane	IoU car	IoU lane	IoU car	IoU lane	IoU car	IoU lane
City	0.7543	0.5692	0.7173	0.5472	0.6799	0.5421	0.6381	0.5220
Freeway	0.6928	0.5197	0.6336	0.4698	0.5967	0.4487	0.5589	0.4296
Sunny	0.7223	0.5338	0.6768	0.5001	0.6661	0.4831	0.6106	0.4693
Dark	0.7000	0.5226	0.6570	0.5120	0.6130	0.5014	0.5834	0.4832
All	0.7078	0.5268	0.6639	0.5075	0.6315	0.4946	0.5931	0.4782

a lower dimensionality, we force the model to capture the absolutely essential features from the data, discarding the non-relevant information; second, if the representation of a single concept occupies only a small fraction of the entire latent space, we can learn several different concepts at the same time. Here, we decide to assign 16 neurons to each concept with the idea that in the future, we can use the same architecture to learn more than two concepts, adding for

TABLE 3. Comparison of the performance of *Net4* using different implementations of recursive nodes.

	Frame 9		Frame 10		Frame 11		Frame 12	
	IoU car	IoU lane	IoU car	IoU lane	IoU car	IoU lane	IoU car	IoU lane
RNN	0.6836	0.4884	0.5963	0.4231	0.5100	0.3957	0.4598	0.3668
GRU	0.7078	0.5268	0.6639	0.5075	0.6315	0.4946	0.5931	0.4782
LSTM	0.6810	0.5196	0.6604	0.4911	0.6426	0.4696	0.6119	0.4623

TABLE 4. IoU scores of `cars` and `lanes` classes obtained by *Net2* using different numbers of neurons for the `cars` and `lanes` concepts in the latent space, while keeping the overall size $N_V = 128$. The final choice adopted in the model is marked in bold.

$N_C = N_L$	IoU cars	IoU lanes
48	0.7814	0.6460
32	0.7768	0.6334
24	0.7709	0.6440
16	**0.7702**	**0.6277**
12	0.7539	0.6139
8	0.7194	0.5965
4	0.6162	0.5123

TABLE 5. Statistics on the latent representations learned by our 3 autoencoder models. For both indicators, the lower the better.

	Temporal coherence $\xi_{\mathcal{Z}}$	Predictivity error $\rho_{\mathcal{Z}'}$
Net1	0.299	0.186
Net2	0.297	0.189
Net3	0.180	0.077

example `pedestrians` and `bikes`. Therefore, the final model adopts the most compact size not causing a severe drop in the performance, like in the cases of $N_C = N_L < 12$.

Then, we present a statistical evaluation of the latent representations measuring the consistency for the temporal dynamics and their predictability. Table 5 reports the results for all our 3 autoencoder models. A first indicator ξ evaluates the degree of temporal coherence by measuring the ratio between the difference of two latent vectors that are contiguous in time, and the variance over the entire dataset $\mathcal{Z}$ of latent vectors. The evaluation is done independently for each component of the latent vector and then averaged:

$$\xi_{\mathcal{Z}} = \frac{1}{N_V M} \sum_{i}^{N_V} \frac{\sum_{\mathbf{z}\in\mathcal{Z}} \left(z_i - z_i^{(1)}\right)^2}{\upsilon_i}, \tag{7}$$

where z_i is the i-th element of $\mathbf{z}$, $z_i^{(1)}$ is the i-th element of the successor of $\mathbf{z}$, υ_i is the i-th element of the variance vector of $\mathbf{z}$ over $\mathcal{Z}$, and M is the cardinality of $\mathcal{Z}$. A second indicator ρ measures the "predictability" of the representations, and it is computed as the mean square of the residual obtained when using two consecutive latent vectors to predict one neuron of a third vector by linear regression. In order to make computation time acceptable, this index is computed on a subspace $\mathcal{Z}'$ ten times smaller than $\mathcal{Z}$. By calling $\varepsilon(\mathbf{A}, \mathbf{b})$ the residual of the least squares approximation of the normal equation $\mathbf{Ax} = \mathbf{b}$, ρ can be written as follows:

$$\rho_{\mathcal{Z}'} = \frac{1}{N_V} \sum_{i}^{N_V} \varepsilon \left(\begin{bmatrix} \cdots & \cdots \\ \mathbf{z} & \mathbf{z}^{(1)} \\ \cdots & \cdots \end{bmatrix}_{\mathbf{z}\in\mathcal{Z}'}, \begin{bmatrix} \cdots \\ z_i^{(2)} \\ \cdots \end{bmatrix}_{\mathbf{z}\in\mathcal{Z}'} \right) \tag{8}$$

Therefore, Table 5 clearly shows how *Net1* and *Net2* have comparable scores, while *Net3* performs significantly better. In fact, only with *Net3* we introduce the temporal consistency inside the latent representations, and this is nicely reflected in the results.

Moving to a more qualitative analysis, we present in Fig. 11 a visual inspection of the latent representations learned by *Net2* and *Net3*. For each model, the left column of the Figure shows four images depicting the same driving scenario under different lighting conditions. For each input image, we plot the values of the 128 neurons composing the latent representation computed by the model, separating the 16 neurons representing the `cars` entities (second column from the left), the 16 neurons representing the `lanes` entities (last column), and the remaining 96 neurons representing generic visual features (third column). Ideally, only the generic 96 neurons should change in the four cases, because the input images differ only in the lighting conditions while having almost identical `cars` and `lanes` entities. Comparing the performance of *Net2* (a) and *Net3* (b), it is immediately clear how the latter learns a more robust representation. In the case of (b), the variation in the neurons representing the `cars` and `lanes` concepts is minimal. The variation in the general 96 neurons is also very localized: the neurons exhibit a similar overall distribution. This fits with the fact that the four images have the same surrounding (the trees, the soil on the right). Conversely, the representations learned by *Net2* do not appear as consistent. The `cars` and `lanes` neurons visibly change for each input, and even the other 96 visual features do not share any particular pattern in the 4 cases. Therefore, we can conclude that forcing a semantic organization at once and a temporal coherence leads to more robust and disentangled representations.

Lastly, we include the interesting outcome of exchanging parts of latent representations of different images. Fig. 12 shows the imaginary scenarios created by swapping between two input images the neurons corresponding to the `cars` and `lanes` concepts. Fig. 12(c) is produced by the decoders of *Net3* from a latent vector composed of $\mathbf{z}_C$ and $\mathbf{z}_L$ taken from the representation of (a), and $\widetilde{\mathbf{z}}$ coming from the representation of (b). Similarly, Fig. 12(d) is the result of combining $\mathbf{z}_C$ and $\mathbf{z}_L$ from the representation of (b) together with $\widetilde{\mathbf{z}}$ from the vector representing (a). This is a nice example of how our

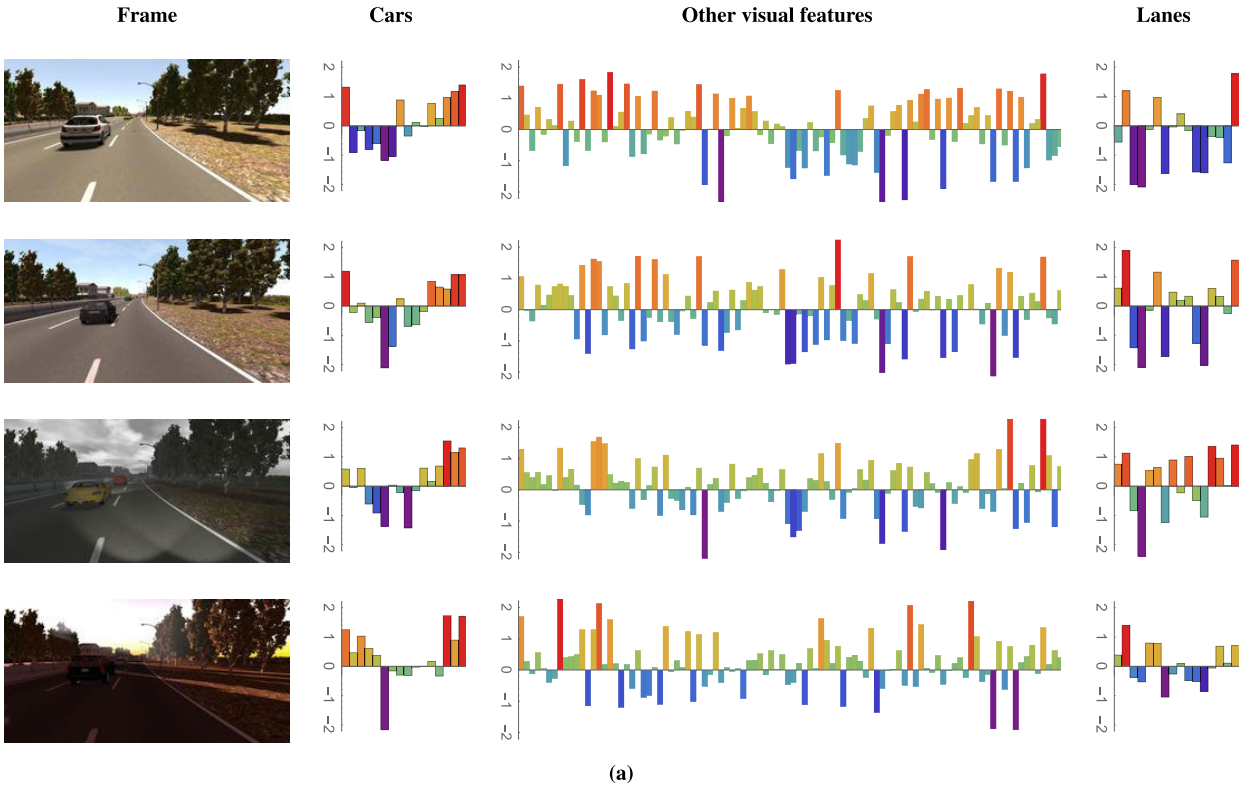

FIGURE 11. Visualization of the latent representations learned by *Net2* (a) and *Net3* (b). Each row depicts the values of the 128 neurons of the latent representation of the image on the left. The neurons corresponding to the `cars` and `lanes` concepts are plotted separately.

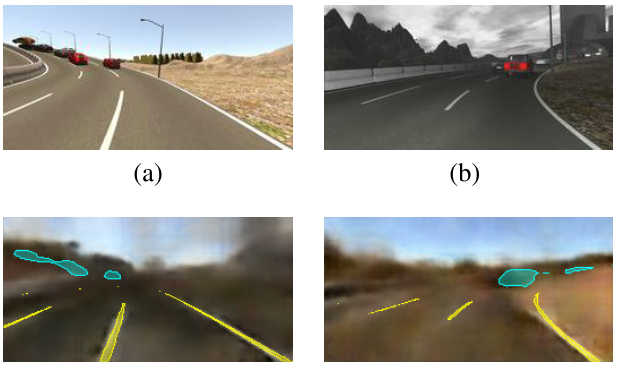

FIGURE 12. Result of swapping the conceptual parts of the latent spaces between two images using *Net3*. Image (c) is obtained by combining the `cars` and `lanes` neurons of (a) with the rest of the vector of (b). Image (d) is the opposite, combining the `cars` and `lanes` neurons of (b) with the rest of the vector of (a).

TABLE 6. Parameters describing the architecture of the variational autoencoder (*Net1*).

Encoder	convolution	$7 \times 7 \times 16$
	convolution	$7 \times 7 \times 32$
	convolution	$5 \times 5 \times 32$
	convolution	$5 \times 5 \times 32$
	dense	2048
	dense	512
Latent space		128
Decoder	dense	2048
	dense	4096
	deconvolution	$5 \times 5 \times 32$
	deconvolution	$5 \times 5 \times 32$
	deconvolution	$7 \times 7 \times 16$
	deconvolution	$7 \times 7 \times 3$
Total parameters		18 million

TABLE 7. Parameters describing the architecture of the topological autoencoder (*Net2*).

Encoder	convolution	$7 \times 7 \times 16$
	convolution	$7 \times 7 \times 32$
	convolution	$5 \times 5 \times 32$
	convolution	$5 \times 5 \times 32$
	dense	2048
	dense	512
Latent space		[16, 96, 16]
Each decoder	dense	2048
	dense	4096
	deconvolution	$5 \times 5 \times 32$
	deconvolution	$5 \times 5 \times 32$
	deconvolution	$7 \times 7 \times 16$
	deconvolution	$7 \times 7 \times 3$
Total parameters		35 million

model can perform another form of mental imagery, in the sense of creating artificial – although plausible – scenarios.

VI. CONCLUSION AND FUTURE WORKS

This paper presented a novel approach to the perception of driving scenarios loosely inspired by two theories on how the human brain works. We mimic the neurocognitive the-

TABLE 8. IoU scores of `cars` and `lanes` classes obtained by the topological autoencoder (*Net2*) using different values of learning rate. The final choice adopted in the model is marked in bold.

Learning rate	IoU car	IoU lane
1×10^{-2}	0.0000	0.0000
1×10^{-3}	0.7583	0.6568
5×10^{-4}	0.7391	0.6599
$\mathbf{1 \times 10^{-4}}$	**0.7702**	**0.6277**
5×10^{-5}	0.7584	0.6083
1×10^{-5}	0.7086	0.5502
1×10^{-6}	0.1187	0.1734

TABLE 9. Parameters describing the architecture of the temporal autoencoder (*Net3*).

Encoder	convolution	$7 \times 7 \times 16$
	convolution	$7 \times 7 \times 32$
	convolution	$5 \times 5 \times 32$
	convolution	$5 \times 5 \times 32$
	dense	2048
	dense	512
Latent space		[16, 96, 16]
Recurrent layer		$128 \times 2 \rightarrow 128$
Each of the 3 individual decoders	dense	2048
	dense	4096
	deconvolution	$5 \times 5 \times 32$
	deconvolution	$5 \times 5 \times 32$
	deconvolution	$7 \times 7 \times 16$
	deconvolution	$7 \times 7 \times 3$
Total parameters		35 million

TABLE 10. Parameters describing the architecture of the recurrent network (*Net4*).

Stacked recurrency	GRU	$128 \times 8 \rightarrow 128 \times 8$
	GRU	$128 \times 8 \rightarrow 128 \times 8$
Parallel recurrency	GRU	$128 \times 8 \rightarrow 128$
	GRU	$128 \times 8 \rightarrow 128$
	GRU	$128 \times 8 \rightarrow 128$
	GRU	$128 \times 8 \rightarrow 128$
Total parameters		600.000

ories with the tools available within the deep learning framework. Specifically, we choose the autoencoders to emulate the theoretical idea of *convergence-divergence zones*, which code perceptual concepts using low-dimension representations. Then, we follow the theory of the *predictive brain* by forcing the probabilistic representation learned by variational autoencoders to capture information about the dynamics of the scenario.

We proposed a method to learn to represent visual scenarios into compact vectors that are at once semantically organized and temporally coherent. Our approach differs from other related works precisely in the learning of the representations: first, there is a semantic organization in the sense that distinct parts of the representation are explicitly associated with specific concepts useful in the context of driving; second, the temporal coherence that is achieved through self-supervision allows the representation to be exploited for mental imagery and prediction of plausible future scenarios.

Our work aims to learn compact and informative representations that can be useful for various downstream driving tasks. Here we presented the example of predicting long-term

future frames in a video sequence. However, once learned, the representations can be deployed in many possible contexts. For example, we are currently working on using the representations to predict future occupancy grids. Moreover, since we achieve to assign only 16 neurons to each concept in the representation, it is possible to include in future works more than two concepts inside the latent representations. It would be interesting, for example, to include concepts of vulnerable road users, such as `pedestrians` and `bikes`. One more future development we have planned is the adoption of a dataset of real-world video sequences. One of the reasons we adopted the SYNTHIA dataset at the beginning of our research, besides its large size and variety, was the availability of lane marking annotations, which are very rare among the classical datasets for autonomous driving. Recently, UC Berkeley introduced the Berkeley DeepDrive dataset [84], including several types of lane marking annotations from high-quality real video sequences. Hence, the adoption of this novel dataset could be an interesting future addition to our work.

VARIATIONAL INFERENCE

The variational inference framework takes up the issue of approximating the probability distribution $p(\mathbf{x})$ of a high dimensional random variable $\mathbf{x} \in \mathcal{X}$. This approximation can be performed by a neural network like the decoder part of *Net1*. The neural network by itself is deterministic, but its output distribution can be easily computed as follows:

$$p_\Theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}\left(\mathbf{x}|f_\Theta(\mathbf{z}), \sigma^2\mathbf{I}\right), \tag{9}$$

where $\mathcal{N}(\mathbf{x}|\mu, \sigma)$ is the Gaussian function in $\mathbf{x}$, with mean μ and standard deviation σ. Using this last equation it is now possible to express the desired approximation of $p(\mathbf{x})$:

$$p_\Theta(\mathbf{x}) = \int p_\Theta(\mathbf{x}, \mathbf{z})d\mathbf{z} = \int p_\Theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}. \tag{10}$$

It is immediate to recognize that the kind of neural network performing the function $f_\Theta(\cdot)$ is exactly the decoder part in the autoencoder, corresponding to the divergence zone in the CDZ neurocognitive concept. In the case when $\mathcal{X}$ is the domain of images, $f_\Theta(\cdot)$ comprises a first layer that rearranges the low-dimension variable $\mathbf{x}$ in a two dimensional geometry, followed by a stack of deconvolutions, up to the final geometry of the $\mathbf{x}$ images.

In equation (10) there is clearly no clue on what the distribution $p(\mathbf{z})$ might be, but the idea behind variational autoencoder is to introduce an auxiliary distribution q from which to sample $\mathbf{z}$, and it is made by an additional neural network. Ideally, this network should provide the posterior probability $p_\Theta(\mathbf{z}|\mathbf{x})$ – which is unknown – and should be a network like the decoder part of *Net1*. Its probability distribution is:

$$q_\Phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}|g_\Phi(\mathbf{x}), \sigma^2\mathbf{I}\right). \tag{11}$$

While the network $f_\Theta(\cdot)$ behaves as decoder, the network $g_\Phi(\cdot)$ corresponds to the encoder part in the autoencoder, projecting the high-dimensional variable $\mathbf{x}$ into the low dimensional space $\mathcal{Z}$. It continues to play the role of the convergence zone in the CDZ idea.

The measure of how well $p_\Theta(\mathbf{x})$ approximates $p(\mathbf{x})$ for a set of $\mathbf{x}_i \in \mathcal{D}$ sampled in a dataset $\mathcal{D}$ is given by the log-likelihood:

$$\ell(\Theta|\mathcal{D}) = \sum_{\mathbf{x}_i \in \mathcal{D}} \log \int p_\Theta(\mathbf{x}_i|\mathbf{z})p(\mathbf{z})d\mathbf{z}. \tag{12}$$

This equation cannot be solved because of the unknown $p(\mathbf{z})$, and here comes the help of the auxiliary probability $q_\Phi(\mathbf{z}|\mathbf{x})$. Each term of the summation in equation (12) can be rewritten as follows:

$$\begin{aligned}\ell(\Theta|\mathbf{x}) &= \log \int p_\Theta(\mathbf{x}, \mathbf{z})d\mathbf{z} \\ &= \log \int \frac{p_\Theta(\mathbf{x}, \mathbf{z})q_\Phi(\mathbf{z}|\mathbf{x})}{q_\Phi(\mathbf{z}|\mathbf{x})}d\mathbf{z} \\ &= \log \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\frac{p_\Theta(\mathbf{x}, \mathbf{z})}{q_\Phi(\mathbf{z}|\mathbf{x})}\right],\end{aligned} \tag{13}$$

where in the last passage we used the expectation operator $\mathbb{E}[\cdot]$. Being the log function concave, we can now apply Jensen's inequality:

$$\begin{aligned}\ell(\Theta, \Phi|\mathbf{x}) &= \log \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\frac{p_\Theta(\mathbf{x}, \mathbf{z})}{q_\Phi(\mathbf{z}|\mathbf{x})}\right] \\ &\geq \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log p_\Theta(\mathbf{x}, \mathbf{z})\right] + \\ &\quad - \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log q_\Phi(\mathbf{z}|\mathbf{x})\right].\end{aligned} \tag{14}$$

Since the derivation in the last equation is smaller or at least equal to $\ell(\Theta|\mathbf{x})$, it is called the *variational lower bound*, or *evidence lower bound* (ELBO). Note that now in $\ell(\Theta, \Phi|\mathbf{x})$ there is also the dependency from the parameters Φ of the second neural network defined in (11).

It is possible to rearrange further $\ell(\Theta, \Phi|\mathbf{x})$ in order to have $p_\Theta(\mathbf{x}|\mathbf{z})$ instead of $p_\Theta(\mathbf{x}, \mathbf{z})$ in equation (14), moreover, we can now introduce the *loss function* $\mathcal{L}(\Theta, \Phi|\mathbf{x})$ as the value to be minimized in order to maximize ELBO:

$$\begin{aligned}\mathcal{L}(\Theta, \Phi|\mathbf{x}) &= -\ell(\Theta, \Phi|\mathbf{x}) \\ &= -\int q_\Phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\Theta(\mathbf{x}, \mathbf{z})}{q_\Phi(\mathbf{z}|\mathbf{x})}d\mathbf{z} \\ &= -\int q_\Phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\Theta(\mathbf{x}|\mathbf{z})p_\Theta(\mathbf{z})}{q_\Phi(\mathbf{z}|\mathbf{x})}d\mathbf{z} \\ &= \Delta_{\mathrm{KL}}\left(q_\Phi(\mathbf{z}|\mathbf{x})\|p_\Theta(\mathbf{z})\right) + \\ &\quad - \mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log p_\Theta(\mathbf{x}|\mathbf{z})\right].\end{aligned} \tag{15}$$

where the last step uses the Kullback-Leibler divergence Δ_{KL}. Still, this formulation seems to be intractable because it contains the term $p_\Theta(\mathbf{z})$, but there is a simple analytical formulation of the Kullback-Leibler divergence in the Gaussian case (see Appendix B in [31]):

$$\Delta_{\mathrm{KL}}\left(q_\Phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right) = -\frac{1}{2}\sum_{i=1}^{Z}\left(1+\log\left(\sigma_i^2\right) - \mu_j^2 - \sigma_i^2\right), \tag{16}$$

where μ_i and σ_i are the i-th components of the mean and variance of $\mathbf{z}$ given by $q_\Phi(\mathbf{z}|\mathbf{x})$.

TABLES OF NETWORK PARAMETERS

See Table 6–10.

REFERENCES

[1] *Global Status Report on Road Safety: Summary*, World Health Org. (WHO), Geneva, Switzerland, 2018.

[2] J. Fleetwood, "Public health, ethics, and autonomous vehicles," *Amer. J. Public Health*, vol. 107, no. 4, pp. 254–280, 2017.

[3] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Tech. Rep. DOT HS 812, vol. 115, 2015.

[4] E. D. Dickmanns, "Vehicles capable of dynamic vision: A new breed of technical beings?" *Artif. Intell.*, vol. 103, nos. 1–2, pp. 49–76, 1998.

[5] S. Ingle and M. Phute, "Tesla autopilot: Semi autonomous driving, an uptick for future autonomy," *Int. Res. J. Eng. Technol.*, vol. 3, no. 9, pp. 369–372, 2016.

[6] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[7] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[9] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2010, pp. 253–256.

[10] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alexander Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[11] R. VanRullen, "Perception science in the age of deep neural networks," *Frontiers Psychol.*, vol. 8, p. 142, Feb. 2017.

[12] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2722–2730.

[13] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," 2017, *arXiv:1704.07911*. [Online]. Available: http://arxiv.org/abs/1704.07911

[14] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robot.*, vol. 37, no. 20, pp. 362–386, Apr. 2020, doi: 10.1002/rob.21918.

[15] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and R. S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–20.

[16] M. Mesulam, "From sensation to cognition," *Trends Cognit. Sci.*, vol. 121, no. 6, pp. 455–462, 1998.

[17] P. Jacob and M. Jeannerod, *Ways of Seeing: The Scope and Limits of Visual Cognition*. Oxford, U.K.: Oxford Univ. Press, 2003.

[18] S. M. Kosslyn, *Image and Brain: The Resolution of the Imagery Debate*. Cambridge, MA, USA: MIT Press, 1994.

[19] S. T. Moulton and S. M. Kosslyn, "Imagining predictions: Mental imagery as mental emulation," *Phil. Trans. Roy. Soc. B, Biol. Sci.*, vol. 364, no. 1521, pp. 1273–1280, May 2009.

[20] A. R. Damasio, "Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition," *Cognition*, vol. 33, nos. 1–2, pp. 25–62, Nov. 1989.

[21] K. Meyer and A. Damasio, "Convergence and divergence in a neural architecture for recognition and memory," *Trends Neurosc.*, vol. 32, no. 7, pp. 376–382, Jul. 2009.

[22] J. S. Olier, E. Barakova, C. Regazzoni, and M. Rauterberg, "Re-framing the characteristics of concepts and their relation to learning and cognition in artificial agents," *Cognit. Syst. Res.*, vol. 44, pp. 50–68, Aug. 2017.

[23] M. Jeannerod, "Neural simulation of action: A unifying mechanism for motor cognition," *NeuroImage*, vol. 14, no. 1, pp. S103–S109, Jul. 2001.

[24] G. Hesslow, "The current status of the simulation theory of cognition," *Brain Res.*, vol. 1428, pp. 71–79, Jan. 2012.

[25] K. L. Downing, "Predictive models in the brain," *Connection Sci.*, vol. 21, no. 1, pp. 39–74, Mar. 2009.

[26] K. Friston, "The free-energy principle: A unified brain theory?" *Nature Rev. Neurosci.*, vol. 11, no. 2, pp. 127–138, 2010.

[27] K. Friston, T. FitzGerald, F. Rigoli, P. Schwartenbeck, and G. Pezzulo, "Active inference: A process theory," *Neural Comput.*, vol. 29, no. 1, pp. 1–49, Jan. 2017.

[28] E. Rolls, *Cerebral Cortex: Principles of Operation*. Oxford, U.K.: Oxford Univ. Press, 2016.

[29] R. B. Conway, "The organization and operation of inferior temporal cortex," *Annu. Rev. Vis. Sci.*, vol. 4, pp. 381–402, Sep. 2018.

[30] M. Tschannen, M. Lucic, and O. Bachem, "Recent advances in autoencoder-based representation learning," in *Proc. NIPS Workshop Bayesian Deep Learn.*, 2018, pp. 1–26.

[31] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[32] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Mach. Learn. Res.*, E. P. Xing and T. Jebara, Eds. Beijing, China: JMLR.org, 2014, pp. 1278–1286.

[33] A. Plebe, R. Donà, G. P. P. Rosati, and M. D. Lio, "Mental imagery for intelligent vehicles," in *Proc. 5th Int. Conf. Vehicle Technol. Intell. Transp. Syst.*, 2019, pp. 43–51.

[34] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, 1989.

[35] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[36] A. Mitiche and I. B. Ayed, *Variational and Level Set Methods in Image Segmentation*. Berlin, Germany: Springer-Verlag, 2010.

[37] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[38] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[39] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[40] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," 2020, *arXiv:2001.05566*. [Online]. Available: http://arxiv.org/abs/2001.05566

[41] G. H. Patel, D. M. Kaplan, and L. H. Snyder, "Topographic organization in the brain: Searching for general principles," *Trends Cognit. Sci.*, vol. 18, no. 7, pp. 351–363, Jul. 2014.

[42] J.-P. Thivierge and G. F. Marcus, "The topographic brain: From neural connectivity to cognition," *Trends Neurosc.*, vol. 30, no. 6, pp. 251–259, Jun. 2007.

[43] M. A. Sommer and R. H. Wurtz, "Composition and topographic organization of signals sent from the frontal eye field to the superior colliculus," *J. Neurophysiol.*, vol. 83, no. 4, pp. 1979–2001, Apr. 2000.

[44] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early aaccess, May 4, 2020, doi: 10.1109/TPAMI.2020.2992393.

[45] A. Newell and J. Deng, "How useful is self-supervised pretraining for visual tasks?" in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 7345–7355.

[46] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6874–6883.

[47] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 868–884.

[48] F. Chiaroni *et al.*, "Self-supervised learning for autonomous vehicles perception: A conciliation between analytical and learning methods," 2019, *arXiv:1910.01636*. [Online]. Available: https://arxiv.org/abs/1910.01636

[49] L. Chen, W. Tang, and N. John, "Self-supervised monocular image depth learning and confidence estimation," 2018, *arXiv:1803.05530*. [Online]. Available: http://arxiv.org/abs/1803.05530

[50] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*. [Online]. Available: http://arxiv.org/abs/1807.03748

[51] A. D. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1747–1756.

[52] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with PixelCNN decoders," in *Proc. Adv. Neural Inf. Process. Syst.*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 4790–4798.
[53] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2051–2060.
[54] D. Ha and J. Schmidhuber, "World models," 2018, *arXiv:1803.10122*. [Online]. Available: http://arxiv.org/abs/1803.10122
[55] M. Pasquier and J. Richard Oentaryo, "Learning to drive the human way: A step towards intelligent vehicles," *Int. J. Vehicle Auto. Syst.*, vol. 6, nos. 1–2, pp. 24–47, 2008.
[56] S. Chen, S. Zhang, J. Shang, B. Chen, and N. Zheng, "Brain-inspired cognitive model with attention for self-driving cars," *IEEE Trans. Cognit. Develop. Syst.*, vol. 11, no. 1, pp. 13–25, Mar. 2019, doi 10.1109/TCDS.2017.2717451.
[57] X. Zhang, M. Zhou, H. Liu, and A. Hussain, "A cognitively inspired system architecture for the Mengshi cognitive vehicle," *Cerebral Cortex*, vol. 12, pp. 140–149, Nov. 2019, doi: 10.1007/s12559-019-09692-6.
[58] A. Ofner and S. Stober, "Towards bridging human and artificial cognition: Hybrid variational predictive coding of the physical world, the body and the brain," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–4.
[59] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[60] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval," in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, 2011, pp. 489–494.
[61] G. E. Hinton, A. Krizhevsky, and D. S. Wang, "Transforming auto-encoders," in *Proc. Int. Conf. Artif. Neural Netw.* New York, NY, USA: Springer-Verlag, 2011, pp. 44–51.
[62] C. Zhang, J. Butepage, H. Kjellstrom, and S. Mandt, "Advances in variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 2008–2026, Aug. 2019.
[63] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. Lecun, "Off-road obstacle avoidance through end-to-end learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 739–746.
[64] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, and J. Zhao, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*. [Online]. Available: https://arxiv.org/abs/1604.07316
[65] H. M. Eraqi, M. N. Moustafa, and J. Honer, "End-to-end deep learning for steering autonomous vehicles considering temporal dependencies," 2017, *arXiv:1710.03804*. [Online]. Available: http://arxiv.org/abs/1710.03804
[66] M. Bansal, A. Krizhevsky, and A. Ogale, "ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst," 2018, *arXiv:1812.03079*. [Online]. Available: http://arxiv.org/abs/1812.03079
[67] D. Wang, C. Devin, Q.-Z. Cai, F. Yu, and T. Darrell, "Deep object-centric policies for autonomous driving," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8853–8859.
[68] G. Sistu, I. Leang, S. Chennupati, C. Hughes, S. Milz, S. Yogamani, and S. Rawashdeh, "NeurAll: Towards a unified model for visual perception in automated driving," 2019, *arXiv:1902.03589*. [Online]. Available: http://arxiv.org/abs/1902.03589
[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
[70] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767
[71] E. Santana and G. Hotz, "Learning a driving simulator," 2016, *arXiv:1608.01230*. [Online]. Available: http://arxiv.org/abs/1608.01230
[72] A. Amini, W. Schwarting, G. Rosman, B. Araki, S. Karaman, and D. Rus, "Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 568–575.
[73] D. T. Kulkarni, F. W. Whitney, P. Kohli, and B. J. Tenenbaum, "Deep convolutional inverse graphics network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2539–2547.
[74] J. Zhao, M. Mathieu, R. Goroshin, and Y. Lecun, "Stacked what-where auto-encoders," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–12.
[75] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," 2017, *arXiv:1708.08296*. [Online]. Available: http://arxiv.org/abs/1708.08296
[76] A. Plebe, M. D. Lio, and D. Bortoluzzi, "On reliable neural network sensorimotor control in autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 711–722, Feb. 2020.
[77] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," 2015, *arXiv:1511.06349*. [Online]. Available: http://arxiv.org/abs/1511.06349
[78] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, M. J. Cardoso, T. Arbel, G. Carneiro, T. Syeda-Mahmood, J. M. R. S. Tavares, M. Moradi, A. Bradley, H. Greenspan, J. P. Papa, A. Madabhushi, J. C. Nascimento, J. S. Cardoso, V. Belagiannis, and Z. Lu, Eds. Cham, Switzerland: Springer, 2017, pp. 240–248.
[79] L. J. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–221, 1990.
[80] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Stroudsburg, PA, USA: Association Computational Linguistics, 2014, pp. 1724–1734.
[81] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3234–3243.
[82] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Cham, Switzerland: Springer, 2015, pp. 234–241.
[83] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[84] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," 2018, *arXiv:1805.04687*. [Online]. Available: http://arxiv.org/abs/1805.04687

ALICE PLEBE received the B.Sc. and M.Sc. degrees in computer science from the University of Catania, Italy, in 2014 and 2016, respectively. She is currently pursuing the Ph.D. degree with the University of Trento. She works on deep neural networks inspired by the human-cognition of driving, as part of the EU Project Dreams4Cars.

MAURO DA LIO (Member, IEEE) received the Laurea degree in mechanical engineering from the University of Padova, Italy, in 1986. He was worked in underwater robotics (EUREKA Project) with Offshore Oil Research Company. He is currently a Full Professor of mechanical systems with the University of Trento, Italy. He was involved in several EU framework program six and seven projects, such as PReVENT, SAFERIDER, interactIVe, VERITAS, AdaptIVe, and No-Tremor. He is a Coordinator with the EU Horizon 2020 Dreams4Cars Research and Innovation Action, a collaborative project with the Robotics domain, which aims at increasing the cognition abilities of artificial driving agents by means of offline simulation mechanisms broadly inspired to the human-dream state. His research interests include modeling, simulation, and optimal control of mechanical multibody systems, in particular, vehicle and spacecraft dynamics, and modeling of human-sensory-motor control, in particular, drivers and motor impaired people.