# On the Road to Clarity: Exploring Explainable AI for World Models in a Driver Assistance System

Mohamed Roshdi*, Julian Petzold*, Mostafa Wahby*, Hussein Ebrahim*, Mladen Berekovic*, Heiko Hamann†
*Institute of Computer Engineering
University of Lübeck, Lübeck, Germany
Email: petzold@iti.uni-luebeck.de
†Department of Computer and Information Science
University of Konstanz, Konstanz, Germany



***Abstract*—In Autonomous Driving (AD) transparency and safety are paramount, as mistakes are costly. However, neural networks used in AD systems are generally considered black boxes. As a countermeasure, we have methods of explainable AI (XAI), such as feature relevance estimation and dimensionality reduction. Coarse graining techniques can also help reduce dimensionality and find interpretable global patterns. A specific coarse graining method is Renormalization Groups from statistical physics. It has previously been applied to Restricted Boltzmann Machines (RBMs) to interpret unsupervised learning. We refine this technique by building a transparent backbone model for convolutional variational autoencoders (VAE) that allows mapping latent values to input features and has performance comparable to trained black box VAEs. Moreover, we propose a custom feature map visualization technique to analyze the internal convolutional layers in the VAE to explain internal causes of poor reconstruction that may lead to dangerous traffic scenarios in AD applications. In a second key contribution, we propose explanation and evaluation techniques for the internal dynamics and feature relevance of prediction networks. We test a long short-term memory (LSTM) network in the computer vision domain to evaluate the predictability and in future applications potentially safety of prediction models. We showcase our methods by analyzing a VAE-LSTM world model that predicts pedestrian perception in an urban traffic situation.**

## I. Introduction

Methods based on artificial intelligence (AI) have been shown to have increasing successes when applied to a vast variety of application fields (e.g., healthcare, farming, autonomous vehicles, etc.). Many symbolic AI approaches (e.g., rule-based methods) can represent problems in an easily interpretable and human-readable format. However, machine learning (ML) techniques that have shown more promising performance, such as artificial neural networks (ANNs), follow the subsymbolic paradigm. These techniques are considered 'black boxes' that are difficult to interpret and explain.

When designing an ML system, its interpretability and explainability are essential factors because they influence the user's trust and their ability to improve or re-adapt the system. A system is interpretable when represented in a human-understandable format and is considered explainable if humans can comprehend the decisions and predictions of the ML system [1]. This led to the emergence of the Explainable Artificial Intelligence (XAI) field, which aims to provide means to help interpreting and explaining ML systems.

In the context of autonomous driving, which is the main focus of this work, XAI is of particular interest due to the continuous increase of automation levels and the necessity to explain, at least retrospectively, the decisions made by large ANNs in dangerous situations. Here, we present an XAI approach that we showcase in an AD-specific application. We use XAI to explain previously trained ANN models that predict the behavior of vulnerable road users (VRUs; e.g., pedestrians, bicycles) [2] in urban traffic, as their safety is of highest concern. These prediction models can be used to develop an advanced driver assistance system (DAS), for example, as an early warning system that tries to anticipate dangerous situations on second-timescale. The line of sight between a human driver and pedestrians on the sidewalks can provide information on whether a pedestrian is planning, for example, to enter the road or perform any dangerous behavior [3]. Some prediction models exploit the same visual information of observing close-by VRUs to anticipate dangerous situations. We collected data from the pedestrian perspective at road crossing scenarios in simulations to train ANN [4]. This work is based on a synthetic environment using the CARLA traffic simulator [5]. We trained variational autoencoders (VAE) and long short-term memory (LSTM) networks that can be used to predict the positions and trajectories of VRUs in the near future (e.g., one second). With today's technology, a system on a vehicle equipped with a 360-degree camera could reconstruct relevant features of the perspective of surrounding VRUs to use them as input for prediction models. The feasibility and efficiency of this approach are increased if multiple cars share their perception (e.g., vehicle-to-X approaches [6]).

In this paper, we present four methods and tools to explain both the VAE and LSTM networks of the prediction model from [4] as shown in Figure 1. To explain the internal functionality of the convolutional VAE, we develop a tool based on activation visualization that visualizes the feature maps and principal components of the convolutional filters. This provides a visual understanding of the evolution of features through the convolutional layers and filters, enabling

Partially funded by the Federal Ministry of Education and Research (BMBF), projects 'NEUPA', grant 01IS19078, and 'KI-IoT', grant 16ME0091K.

We thank Robert, Ellen, and Anita de Mello Koch for their insights and advice, and for providing their source code.

the user of the tool to understand the internal functionality of the ConvVAE layers and its generated features. Second, we develop a visualization of the learned data manifold. We visualize the range of features encoded by the latent vector in a grid as done by [7] and utilized in an interactive tool by [8] to explain the mapping between the image features and encoded latent space values. As the challenge of fully explaining VAEs is vast, we present an alternative approach based on Renormalization Groups (RG) [9] from statistical physics as a third method. It transforms the model to an inherently transparent and interpretable model. In a fourth approach, we explain LSTMs by correlating the memory cells' hidden states with input features and implementing a custom feature relevance technique to assign relevance scores for input features. In an application-focused effort, we test how the LSTM reacts to input frames with domain-specific varying features (e.g., increasing number of pedestrians). We empirically evaluate the interpretability by comparing LSTM relevance heat maps to human visual attention maps. Our analysis yields a mean Normalized Scanpath Saliency (NSS) [10] of 0.53, comparing our heat maps to ground truth human visual attention, and reveals abnormal prediction behavior with potential implications for traffic safety. Our explanation and evaluation methods offer a basis for assessing the explainability and predictability of pedestrian perception prediction models that can be applied on different architectures.

## II. Related Work

To provide explainability for black box models, XAI techniques rely on *interpretations*, a mapping from an abstract domain into a human-understandable domain [11]. [12] provide a taxonomy of XAI methodologies based on the insights they provide. Model Simplification techniques aim to compare complex and simplified models to gain insights. Deep Network Representation techniques aim to interpret the representation of data in the model. Deep Network Processing techniques provide insights on why certain inputs lead to their observed outputs. In this paper, we cover all three of these techniques.

*a) XAI in Autonomous Driving:* Some advantages of XAI, such as accessibility, confidence, or fairness [12], are beneficial for almost all uses of artificial intelligence, but certain application areas, such as autonomous driving (AD), are under more scrutiny [13]–[16]. An error in an AD system may cause high costs. AD system failure modes may be difficult to comprehend for humans, which reduces trust. For example, in a fatal accident [17], an autonomous car did not recognize a pedestrian pushing a bicycle. To increase safety and to alleviate trust issues, AD models require insights in causality to build confidence in their mechanisms. Hence, we focus on explaining a model used for AD. In the domain of AD, vehicle-to-X (V2X) [6] allow cars to communicate with traffic signs, other cars, or even pedestrians. Data obtained this way enables new approaches, such as perceiving traffic participants hidden from the car's line of sight, but visible by the camera of a traffic light. V2X approaches might also pose different challenges for XAI, for example, in the domain of machine behavior [18]. A relevant approach [4] uses action and camera data from the perspective of a pedestrian to predict what the pedestrian will see in future time steps.

*b) Model-Specific XAI Approaches:* Numerous approaches have focused on developing explainability techniques designed for particular Deep Learning (DL) architectures. Karpathy et al. [19] analyzed the functionality of LSTM memory cells in language models, detecting functionalities, such as maintaining the state of long-term dependencies, while Bach et al. [20] proposed Layer-wise Relevance Propagation to assign relevance scores to input features by propagating the model's outputs backwards using redistribution rules.

To interpret convolutional neural networks (CNNs), [21] visualized a network's learned features by optimizing random images to maximize the activations of convolutional filters, entire layers, or individual channels.

Selvaraju et al. [22] introduced GradCAM, a gradient-based technique that generates a localization map highlighting the relevant areas of an image that influence a classification. Lie et al. [23] extend this gradient-based visualization from classification networks to generative models like the VAE. Muhammad and Yeasin [24] augmented CAM methods by visualizing the principal components of feature maps. Cetin et al. [25] introduce the Attri-VAE, a VAE based on the $\beta$-VAE [26]. By introducing an attribute regularization term to their loss function, the authors disentangle latent space variables, compelling them to align with predefined image attributes for enhanced interpretability.

*c) DL Interpretation through Renormalization Groups:* In statistical physics, Renormalization Groups (RGs) are used to transform complex systems with a high order of parameters into simpler ones with a smaller set of parameters that can describe the general behavior of the system [27]. Metha and Schwab [28] experimentally demonstrated a mapping between RG and Restricted Boltzmann Machines (RBMs). The authors used this mapping to coarse-grain an Ising model, a 2D grid representing the spins of a magnet. Koch et al. [29] explored this connection further by comparing stacked RBMs to a flow of RG operations, where each layer performs a step of RG-like coarse graining. The authors also compared downsampling by convolutional pooling layers to the reduction in dimensionality performed by RG. Koch et al. [9] present an RG-inspired interpretation of unsupervised learning of RBMs and generative models. They compare between the weight matrix of an RBM trained on Ising data and an RG using Singular Value Decomposition (SVD) [30]. Koch et al. [9] noted that the singular vectors with the highest singular values of an SVD of image datasets have their support in low frequencies, with little information encoded in high frequencies. This phenomenon is similar to Momentum Space RG, another type of RG that eliminates high momentum modes represented as high frequencies. Koch et al. [9] introduced the RG Machine (RGM) that uses singular vectors of a training set to estimate weights and biases of an RBM. The RGM relies on low frequency modes of singular vectors, produces images similar to those of the RBM, and improves its performance

within a few epochs. Hence, the authors interpret the RBM learning process as a process similar to momentum space RG, keeping relevant features represented by low frequency modes of singular vectors.

## III. Methods

In this section, we present the interpretability methods that we applied to the Convolutional VAE and the LSTM of the pedestrian perception prediction approach in traffic situations [4]. Our goal is to create an explainable AI framework that provides human-understandable explanations of the internal processing within the models and the mapping between their inputs and outputs.

*a) Feature Maps Visualization Tool:* We developed a tool that visually explains the internal functionality of the ConvVAE model presented in [4]. The model encodes pedestrian perception into an abstract 50D latent vector $z$. This way, the perception is compressed and can be used to predict the next frame with an LSTM model, a type of recurrent neural network for time series prediction [31]. The input frames are represented in 24 channels corresponding to the 24 semantic classes of a custom CARLA environment. The VAE consists of four convolutional layers that sequentially extract lower-dimensional features from the input image. We introduce a Deep Network Representation [12] tool that interprets the functionality of the convolutional filters and their role in the feature extraction process in two phases. Our feature visualization tool feeds an image to the first convolutional layer and saves the output feature maps at each layer. We apply Min-Max normalization [32] to the feature maps and use them as weighted masks that act on the input image to map the grey-scale features to the RGB features, producing more legible feature maps. Next, the tool uses the feature maps at each layer to compute the SVD [30] of the images. This allows us to visualize the top singular vectors $v$ of the feature maps representing the principal components and gain a visual description of the layer's functionality, similar to EigenCAM [24] but in a regression rather than classification task. We conducted an experiment to evaluate the effectiveness of our approach in producing interpretable visualizations and generating insightful outputs at each layer. The experiment utilized pedestrian perception frames from a traffic scenario as input images for two different VAE networks. The first network, $VAE_1$, was trained on a dataset of approximately 805k frames extracted from a fixed pedestrian route in the traffic scenario. The second network, $VAE_2$, was an enhanced version trained on a larger dataset of approximately 4m frames, extracted from different pedestrian routes covering a wider range of pedestrian perspectives. Our objective was to determine if our tool could justify the improved performance of $VAE_2$. To evaluate the functionality of the two networks, we used the correlation distance metric $r(u,v) = 1 - \frac{(u-\bar{u})\cdot(v-\bar{v})}{||u||_2 \, ||v||_2}$ to group between filters $u$ and $v$ in the same layers similar filters of $VAE_1$ and $VAE_2$ and identify differences in extracted features between them. For each filter $u$ in a convolutional layer $L \in \{1,2,3,4\}$ in $VAE_2$, we paired it with the filter $v$ in the same convolutional layer in $VAE_1$ that minimized $r(u,v)$. This enables us to compare filters performing the same functionality across both VAEs.

*b) Latent Space Interpretation:* Besides our convolutional feature maps visualization tool that explains the internal functionality of the convolutional VAE, here, we present a complementary approach to explain the mapping between image features and latent vector values (Deep Network Processing [12]). Inspired by [8] and [7], we observe how systematic changes to latent values influence the visual features of the decoded latent vectors. Such mapping between latent vector changes and decoded visual features will later help us to explain the LSTMs (see Section III-0d). To approach this, we design an experiment where we systematically manipulate the 50D latent vector values of 30 encoded traffic scenario frames, extracted from the pedestrian scenario mentioned in Section I. Our initial experiments indicate that manipulating the values at one or a few positions of the latent vector results in minute changes at the decoded frames. Therefore, we divide each latent vector into five equally sized regions of size ten. We arrived at this division through qualitatively experimenting several division setting and choosing the most interpretable one. We test the influence of iteratively interpolating the values of a region by increments of $\{1,2,3\}$. After analyzing the influence of one region, we restore the values of the original vector, then we switch to the next region. This results in three different analyses per region and 15 frame analyses in total. Finally, the manipulated latent vectors are decoded back to $45 \times 85$ pixel images that are stored in a $3 \times 5$ grid that we call the *latent grid*. We apply the latent grid visualization to $VAE_1$ introduced in Section III-0a.

*c) RG-inspired Interpretable Autoencoder Architecture:* Most NNs are inherently non-transparent, meaning interpreting their functionality requires external analysis [12]. So far, we have presented two external analysis tools that visualize the internal functionality of convolutional layers and examine the mapping between latent encoding and image features. However, those techniques provide a limited interpretation of the mapping between the latent and feature space of the ConvVAE, a common issue in interpreting black-box models [33].

Here, we present a different approach, where instead of explaining such a non-transparent network, we reconstruct it into a model that is designed to be transparent and interpretable by nature that we can use as a backbone for an interpretable VAE architecture. First, we apply the SVD on a sample of the training data to extract interpretable singular vectors $U$ that represent the most relevant features. Each singular vector corresponds to a principal component with the same dimensionality as the dataset frames. Consequently they can be visualized. We use the 2D Fast Fourier Transform (FFT) on the singular vectors $U$, filter out higher frequency modes with a low pass filter, which is equivalent to applying an RG transformation [9] to remove irrelevant information, then apply the Inverse 2D Fourier transform to return to the singular vector space. The resulting filtered singular vectors matrix $\alpha$ is used to encode an input image into a latent

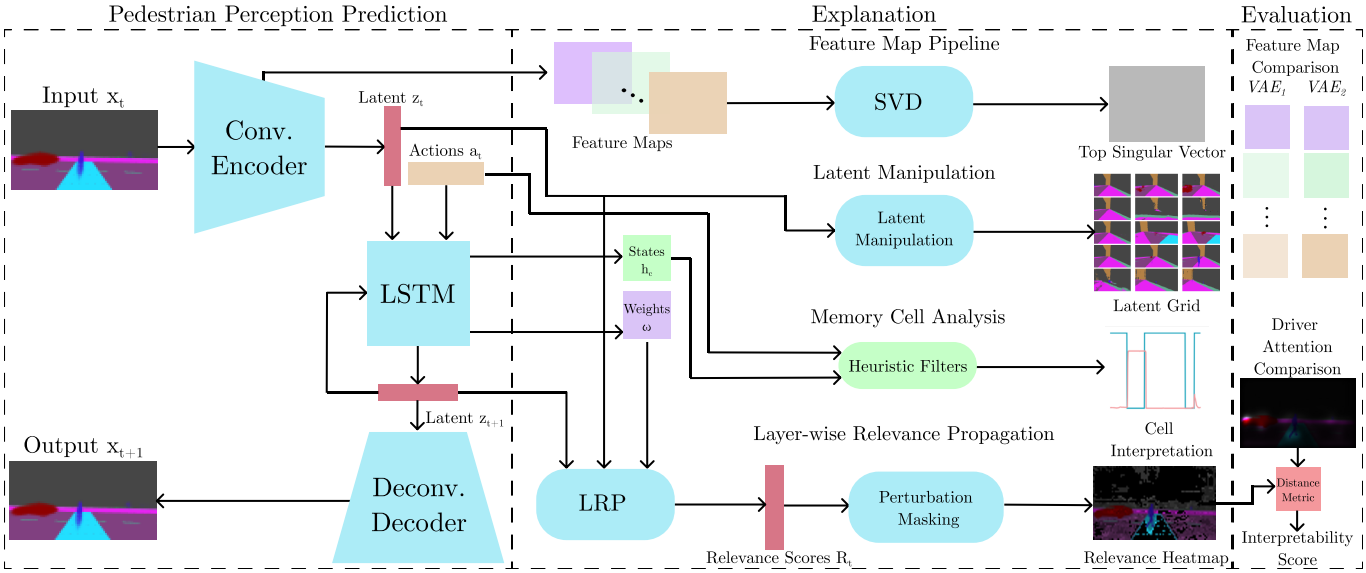


Fig. 1: An overview of our XAI system and its components.

vector $z$ by projecting it onto a lower dimensional vector space with the basis of singular vectors [30]. The values of $z$ indicate the degree of similarity between the input image and the singular vectors, where higher values indicate a stronger presence of features represented by the singular vectors in the input image. Similarly, $z$ can be decoded back to an image using the transpose of the singular vector matrix $\alpha^T$ via $z = \alpha^T x,\ \hat{x} = \alpha z$.

The hyper-parameters of the pipeline are the number of singular vectors used in the matrix $\alpha$ that ultimately determine the size of the latent vector $z$–since the input image is projected onto every single vector–and the cutoff frequency of the low pass filter. Since each singular vector can be visualized as a $45 \times 85$ frame, the features that $z$ encodes can now be mapped to a human understandable domain. We test the pipeline using a subset of the training dataset of $VAE_1$ consisting of 16,500 pedestrian perception frames. We use the Kullback-Leibler (KL) divergence [34] to evaluate the pipeline's performance at a range of hyperparameter settings and to compare against the baseline VAE. We showcase the pipeline's interpretability by visualizing singular vectors of $\alpha$ demonstrating abstract features encoded by $z$.

*d) Interpreting LSTM Dynamics and Feature Relevance:* The LSTM architecture's memory cell is a crucial component that stores state information from previous prediction steps. At each step, the memory cell decides whether to forget its current state and replace it with new input or maintain the current state [31]. Hidden state values $h_c$ of the memory cells determine whether to affect the network's output at each step. The memory cell's decisions are regulated by gates, which are simple multiplicative units that pass through a *tanh* or sigmoid activation function [31]. The LSTM's design enables it to store important information from prior states while protecting them from irrelevant inputs or noise [31]. To explain both the internal functionality and the feature relevance of the pedestrian perception prediction model from [4], we present two approaches for calculating the relevance of input features and for interpreting the functionality of LSTM memory cells in a street-crossing scenario, respectively. In the crossing scenario, a pedestrian approaches a zebra crossing, uses it to cross the street, and then continues on their way. The scenario is executed in CARLA [5] and the pedestrian's semantically segmented vision and action commands are recorded. At each time step, the LSTM uses a pedestrian action $a_t$ and latent vector $z_t$ to predict the next perception frame $z_{t+1}$, which is passed back to the LSTM in a feedback loop to predict again. The first approach is based on an input of 400 frames of the pedestrian's vision and storing the hidden states of the LSTM's 512 memory cells at each frame prediction. We use a sigmoid function to normalize the hidden state values between zero and one, where values closer to one indicate that the cell is triggered, that is, it passes its contents to the output. To visualize the activity of a hidden state with respect to the predicted frame, we plot the hidden state on the y-axis and the index of the generated frame on the x-axis in a 2D plot. We aim to identify cells with interpretable behavior (i.e., correlating the cells' hidden values and the features of the pedestrian perception) using two heuristic filters.

We qualitatively evaluate the interpretability of the top cells identified by the two filters based on a human user's ability to correlate the cell's hidden state behavior with the events in the traffic scene and the actions of the ego-pedestrian. Our investigations indicate that cells with noisy hidden state behavior tend to be uninterpretable, while interpretable cells tend to be active only at certain intervals.

We define the first filter $\kappa$ that returns cells that minimize the KL divergence between their hidden state values and a square pulse activated at a particular range when a certain event

occurs, such as when the pedestrian passes the crosswalk:

$$\kappa = \arg\min_{c}(\mathrm{KL}(h_c||\theta(t-r_1)-\theta(t-r_2)))\ , \qquad (1)$$

where $KL(\cdot||\cdot)$ is the KL divergence, $r_1$ and $r_2$ are the start and end points of the interval $I$ under investigation, $c$ is the index of a memory cell, $h_c$ is the hidden state values of cell $c$, and $\theta$ is the Heaviside step function. The LSTM takes in the ego-pedestrian actions $a = [a_0, a_1, a_2]$, where $a_0$ indicates movement (0 for stopped, 1 for in motion), $a_1$ is the pedestrian's body angle parallel to the ground, and $a_2$ is the head's rotation parallel to the ground. We introduce the second filter $\mu$ that identifies cells tracking action changes:

$$\mu = \arg\min_{c}(S_c(\nabla h_c, \nabla a))\ , \qquad (2)$$

where $S_c$ is the cosine similarity, $\nabla h_c$ is the gradient of hidden values $h_c$, and $\nabla a$ is the gradient of the action $a$.

In our second approach, we propose an augmented Layer-wise Relevance Propagation [20] technique to visualize the relevance of the input latent features and map them to the RGB space. After the LSTM model predicts the future latent vector, we assign a relevance value of 1 for all values in $z_{t+1}$. Then we redistribute the relevance for lower layer neurons through two propagation rules. The two rules handle weighted connections and multiplicative connections such as in the cell state (see [35] for the formulas of our chosen rules). Similar to the latent space interpretation (see Section III-0b) experiment, in order to map the relevance scores of the input latent space to the RGB space, we perturb each latent value and assign its relevance score to the most affected pixel regions in the decoded image. We use the image relevance scores as a mask to visualize the most relevant visual features of the input image. To evaluate the LRP explanation and the differences between the decision making process of the LSTM and other human or machine models, we compare the resultant masks to human driver attention maps generated by a pretrained model proposed by [10] predicting the visual attention of human drivers. As metrics we use the mean NSS and Pearson's Coeffecient [10] over the whole scenario. As input to the LRP experiment we use the pedestrian crossing scenario shown in Table I and the latent grids produced in Section III-0b. We test the LSTM's response to varying input features and unforeseen scenarios that may lead to traffic hazards.

## IV. Results

We present the results of our adaptation of the convolutional feature map visualization [21] and latent space visualization [7], [8] to implement both Deep Network Representation and Deep Network Processing [12]. We also show our novel explanation by simplification [12] method to produce a transparent VAE, and our LSTM explainability framework combining memory cell analysis with latent space visualization.

*a) Convolutional VAE Feature Visualization:* We present the visualizations of the feature map comparison discussed in Section III-0a. Figure 2 shows that unlike $VAE_2$, $VAE_1$ was unable to effectively reconstruct the input, since the decoded image is noisy with missing features, such as the car and the crosswalk. In the bottom of Figure 2, each column represents the two pairs of feature maps belonging to filters in the same layer in the two VAEs. In the first layer, $VAE_2$ better captures the skyline feature as indicated by its brighter color.

In the visualization of the second layer and third convolutional layers, we observe similar trends, with $VAE_1$ poorly extracting the features of the crosswalk and skyline compared to $VAE_2$. These visualizations indicate that the filter of $VAE_1$ have not learned to properly extract the skyline and crosswalk. We exclude the $2 \times 2$ fourth layer's feature map comparison and singular vectors, as their low dimensionality lacks interpretability and hinders post-hoc interpretability in networks with chained convolutional layers. The singular vectors provide a method for users to see the effect of internal data transformations at each convolutional layer, especially in earlier layers with relatively larger filter dimensions. The first layer singular vectors of $VAE_2$ feature high activation values (marked by brighter colors) in patches corresponding to the crosswalk and skyline. Overall, the two-phase feature visualization pipeline has qualitatively demonstrated its ability to visualize internal data processes of the ConvVAE, visualizing internal causes of poor reconstruction by the ConvVAE, which could lead to dangerous traffic situations.

*b) Latent Space Interpretation:* We present here the $5 \times 3$ latent grids of $VAE_1$ for one example of the 30 input latent vectors we manipulate, shown in Figure 3. Each row in the grid, corresponding to a different latent vector slice of size ten, appears to control a set of semantic features of the decoded images. For example, the fourth row $z_4$ features the addition of a pedestrian, while the first row in features the addition of a car. By observing the different rows of the latent grid for a given image, a user can visualize a mapping between semantic features and their latent representation. The latent grid tool will be crucial in analyzing responses of the LSTM, since the latent vector is used by the LSTM to predict the next pedestrian perception frame.

*c) RG-inspired Interpretable Autoencoder Architecture:* For the SVD-based pipeline, we sample the data from the pedestrian vision dataset by [4] to provide an input space of semantically segmented images. We test two cutoff frequencies set at 150 and 175 frequency modes. We compute the SVD using Tensorflow 2.9.1 on three Nvidia A100 GPUs. We rely on the implementation of [9] for the low pass filter, but refactor it using Tensorflow. We compare the performance of SVD architectures to the baseline $VAE_1$ model using the KL Divergence reconstruction error of the test dataset of the $VAE_1$ model trained by [4]. The $VAE_1$ model, the SVD autoencoder with cutoff of 175 and 150 have a mean reconstruction error of 0.024, 0.179, and 0.521, respectively. In terms of interpretability, we can visualize and interpret features encoded by the latent vector as each latent value represents the strength of a principal component (the singular vector). For example, the top three singular vectors of the 175 frequency mode SVD in Figure 4 show different perspectives

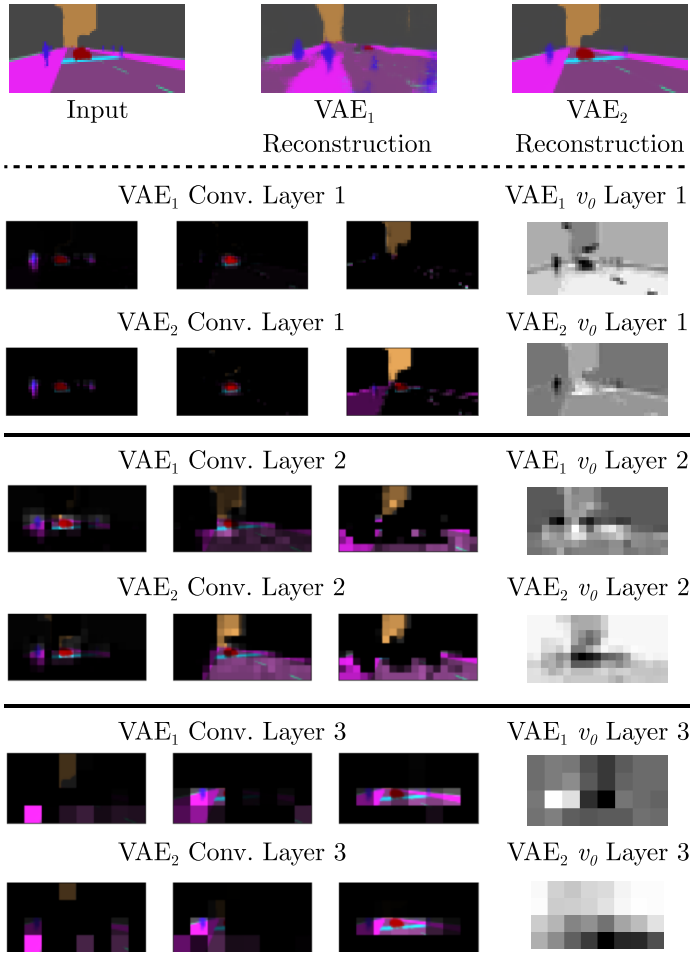


Fig. 2: Visualization of the RGB feature maps and top singular vector $v_0$ for layers 1-3 for *VAE*$_1$ and *VAE*$_2$. For each layer, each column represents the VAEs' most similar feature maps.

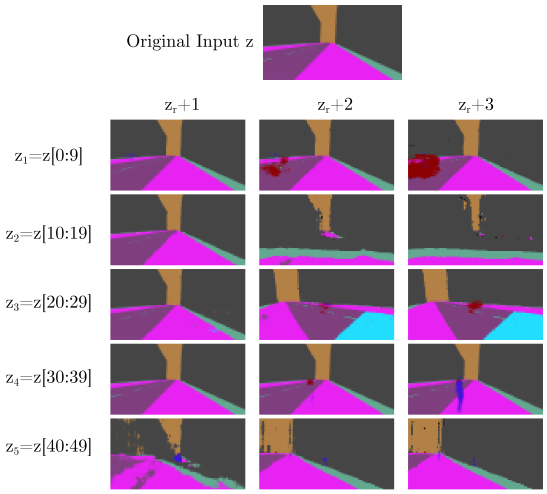


Fig. 3: Latent grids for $z$ (top). The rows represent regions of size ten in the latent vector. Each entry is the decoded vector taken after increasing the values of the region by $\{1, 2, 3\}$.

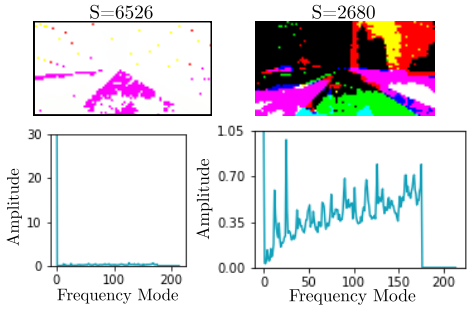


Fig. 4: Singular vectors of $\alpha$ with a cutoff frequency of 175 with the two highest singular values $S$ visualized (top). We show the frequency domains of the first (bottom left) and twelfth singular vector (bottom right). $\alpha[0]$ shows high support in only the lowest frequency, while $\alpha[11]$ has support throughout the spectrum, in agreement with [9].

of the ego-pedestrian. The first three values of the latent vector numerically correspond to each of these visualizable singular vectors, forming a transparent model [33].

*d) Interpreting LSTM Dynamics and Feature Relevance:* We do two LSTM memory cell tests. First we analyze the LSTM memory cells as the VAE-LSTM model predicts the perception of an ego-pedestrian crossing the street. Table I shows the three actions of the pedestrian and a textual description of the scene. We present a sample of memory cells with interpretable hidden state behavior, detected using filters $\kappa$ and $\mu$. Of the 512 cells, only 82 cells were found to meet the qualitative interpretation standard described in Section III-0d. Cell 134 (Figure 5-a) is active at range 80 to 159, corresponding to the interval where the ego-pedestrian stops walking while changing the angle of the head towards the street. Cell 134 was detected through the filter $\kappa$ using a Heaviside square pulse between 80 and 159. The interpretation of this behavior is that cell 134 keeps track of the periods in the sequence when the pedestrian performs the 'stop and scan' behavior before crossing the street, effectively acting as a movement detector. As for the filter $\mu$ that detects cells sensitive to the three action values, no interpretable cells were found to react to differences in the first two action values, corresponding to the flag of the ego-pedestrian movement and the angle of the body of the ego-pedestrian. However, cell 100 (Figure 5-c) was found to react to changes in the third action value that represents the angle of head of the ego-pedestrian. Cell 100 has high hidden cell values coinciding with high normalized values of the head angle, when the pedestrians looks sideways toward the street and passing cars.

In our second LSTM approach, we visualize and evaluate the input relevance heat maps using LRP augmented with perturbation masking for the pedestrian scenario frames of Table I and the latent grids as the example shown in Figure 3. The empirical evaluation of the comparison between the LRP heatmaps and driver attention maps resulted in a mean NSS score of 0.53 and a Pearson Coeffecient of 0.46 for the pedestrian scenario of Table I. Using the latent grid as input to the prediction model, we were able to detect unpredictable

| Frames | $a_0$ | $a_1$ | $a_2$ | Scene description |
|---|---|---|---|---|
| 0-80 | 1 | 180 | 0 | Walk(Sidewalk), LookTo(Sidewalk) |
| 81-118 | 0 | [180 - 270] | [0 , -48.63] | Turn(Right), LookTo(Street) |
| 119-135 | 0 | 270 | [-48.63,-0.66] | Turn(Head,Right), LookTo(Crosswalk) |
| 136-158 | 0 | 270 | -0.66 | LookTo(Crosswalk) |
| 159-233 | 1 | 270 | [-0.66 - 85.54] | Walk(Crosswalk), LookTo(Street) |
| 234-337 | 1 | 270 | [85.54 - 0.24] | Walk(Crosswalk), LookTo(Crosswalk) |
| 338-377 | 0 | [270-180] | 0.24 | Turn(Body,Left), LookTo(Sidewalk) |
| 378-399 | 1 | 180 | 0.24 | Walk(Sidewalk), LookTo(Sidewalk) |

TABLE I: A pedestrian scenario description matching time frame ranges with pedestrian perception.

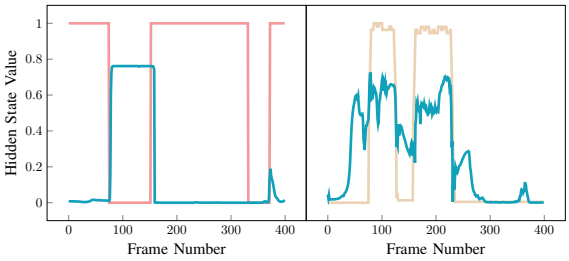


Fig. 5: Cell 134 (Left) and Cell 100 (Right) show the hidden state on the y-axis in blue relative to the pedestrian vision frames. The pink signal is movement action $a_0$, while the cream signal is head angle action $a_2$.

behavior of the LSTM. In the next frame prediction of $z_1 + 2$ and $z_1 + 3$ in Figure 6, the LSTM changes the car in the input frame to a cyclist in the output frame, which may cause dangerous effects in a DAS relying on future pedestrian perception. The LRP heatmaps for this scenario indicate that the latent manipulation of that region triggered the response of the LSTM even before the car becomes visible in frame $z_1+1$.

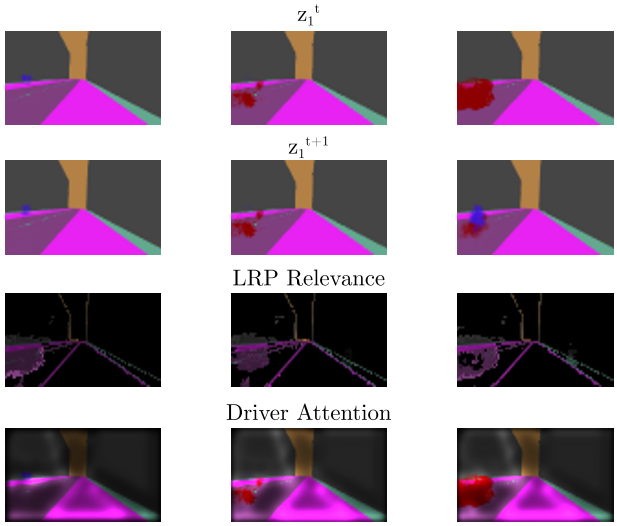


Fig. 6: LSTM predictions in second for latent grid $z_1$, with the LRP heatmap and human attention map in the third and fourth rows, respectively. The LSTM generates a cyclist instead of car, with the LRP detecting the presence of the car's region.

## V. Discussion

Our four methods apply several XAI techniques to ConvVAEs and LSTMs. The use case we address is a vision model that gives drivers recommendations based on pedestrian perception. We can use our feature map visualization tool to investigate internal deficiencies that cause poor model performance. The latent grid maps between latent space and feature space help to visualize the effect of changing different input features on the LSTM prediction in Section IV-0d. A drawback of our latent grid approach is its high granularity, as it does not interpret each latent vector value. Our prediction model explanation provides a baseline to examine the internal functionality and safety of the LSTM predictions prior to deployment in a DAS. Both the memory cell and LRP analysis can be applied to other prediction architectures such as Transformers [36] by analyzing the attention head activity and designing propagation rules for the different Transformer layers. By creating scenarios with rare or dangerous features, we can analyze prediction models' dynamics and the relevance of critical features. However, a drawback of our evaluation is the use of the driver attention model trained to simulate drivers' rather than pedestrians' point of view. Finally, we proposed a backbone for an interpretable VAE architecture. Inspired by the RGM [9], the ultimate goal is to fine-tune the SVD pipeline as an interpretable backbone of a VAE. The singular vectors of the reconstruction matrix $\alpha$ can be used without post-hoc explanation to provide a mapping between feature and latent space. However, calculating the SVD for large datasets seems computationally infeasible. Our use of SVD to encode and decode images is a linear Principal Component Analysis (PCA), which is sensitive to outliers and corrupted data [30].

## VI. Conclusion and Future Work

We introduced an end-to-end XAI framework for explaining the internal functionality and feature-output mapping of DL models in a perception prediction network for autonomous driving [4]. We uncovered the internal feature extraction process of the Convolutional VAE with convolutional feature map visualization and adopted a Feature Relevance explanation approach relating the feature space with the latent space through the latent grid analysis. Using the latent grids as inputs helped us to detect significant prediction errors by the LSTM that may affect the safety of VRUs, with our LRP explanation providing explanations of the relevant features

impacting predictions. We have detected a significant space of interpretable LSTM memory cells and proposed methods to infer their functionality, such as keeping track of pedestrian movement, the presence of cars, and street crossing. Our empirical comparison of the LRP and driver attention maps presents an interpretation baseline for pedestrian prediction models and can be improved by training a pedestrian attention model to compare to the prediction model feature relevance. Our XAI methods may be beneficial in evaluating the transparency and trust-worthiness of state-of-the-art DL models in AD, enabling their certification in the future. Our autoencoder architecture with an inherently interpretable latent space is based on the relationship between autoencoders, PCA, and RG [9]. For future work, we plan to integrate the SVD pipeline into a VAE and combine convolutional layers with the SVD to create hybrid VAE models with a basis of interpretability. We plan to use incremental and randomized SVD [30] to increase dataset size and Robust PCA [30] to enhance generalization and noise handling.

## REFERENCES


[1] A. Erasmus, T. D. P. Brunet, and E. Fisher, "What is interpretability?" *Philosophy & Technology*, vol. 34, no. 4, pp. 833–862, Dec 2021.

[2] G. Yannis, D. Nikolaou, A. Laiou, Y. A. Stürmer, I. Buttler, and D. Jankowska-Karpa, "Vulnerable road users: Cross-cultural perspectives on performance and attitudes," *IATSS research*, vol. 44, no. 3, pp. 220–229, 2020.

[3] L. Lévêque, M. Ranchet, J. Deniel, J.-C. Bornard, and T. Bellet, "Where do pedestrians look when crossing? a state of the art of the eye-tracking studies," *IEEE Access*, vol. 8, pp. 164 833–164 843, 2020.

[4] J. Petzold, M. Wahby, F. Stark, U. Behrje, and H. Hamann, ""if you could see me through my eyes": Predicting pedestrian perception," in *8th Int. Conf. on Control, Automation & Robotics (ICCAR)*, 2022, pp. 184–190.

[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[6] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2X-Sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 914–10 921, 2022.

[7] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd Int. Conf.e on Learning Representations, ICLR*, 2014.

[8] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS 2018)*. USA: Curran Associates Inc., 2018, pp. 2455–2467.

[9] A. De Mello Koch, E. De Mello Koch, and R. De Mello Koch, "Why unsupervised deep networks generalize," vol. abs/2012.03531, 2020. [Online]. Available: https://arxiv.org/abs/2012.03531

[10] Y. Xia, D. Zhang, J. Kim, K. Nakayama, K. Zipser, and D. Whitney, *Predicting Driver Attention in Critical Situations*, 05 2019, pp. 658–674.

[11] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[12] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[13] S. Atakishiyev, M. Salameh, H. Yao, and R. Goebel, "Towards safe, explainable, and regulated autonomous driving," *arXiv preprint arXiv:2111.10518*, 2021.

[14] D. Omeiza, H. Webb, M. Jirotka, and L. Kunze, "Explanations in autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[15] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of deep vision-based autonomous driving systems: Review and challenges," *International Journal of Computer Vision*, pp. 1–28, 2022.

[16] J. Dong, S. Chen, S. Zong, T. Chen, and S. Labi, "Image transformer for explainable autonomous driving system," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2732–2737.

[17] N. T. S. Board, "Collision between vehicle controlled by developmental automated driving system and pedestrian in tempe, arizona, on march 18, 2018," 2019.

[18] I. Rahwan, M. Cebrian, N. Obradovich *et al.*, "Machine behaviour," *Nature*, vol. 568, p. 477–486, 2019.

[19] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding recurrent networks," 2015. [Online]. Available: https://arxiv.org/abs/1506.02078

[20] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, vol. 10, no. 7, pp. 1–46, 07 2015.

[21] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017, https://distill.pub/2017/feature-visualization.

[22] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[23] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, and O. Camps, "Towards visually explaining variational autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8642–8651.

[24] M. B. Muhammad and M. Yeasin, "Eigen-cam: Class activation map using principal components," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2020.

[25] I. Cetin, M. Stephens, O. Camara, and M. A. G. Ballester, "Attri-VAE: Attribute-based interpretable representations of medical images with variational autoencoders," *Computerized Medical Imaging and Graphics*, vol. 104, p. 102158, 2023.

[26] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *International conference on learning representations*, 2017.

[27] S. Iso, S. Shiba, and S. Yokoo, "Scale-invariant feature extraction of neural network and renormalization group flow," *Phys. Rev. E*, vol. 97, p. 053304, May 2018.

[28] P. Mehta and D. J. Schwab, "An exact mapping between the variational renormalization group and deep learning," *ArXiv*, vol. abs/1410.3831, 2014.

[29] E. De Mello Koch, R. De Mello Koch, and L. Cheng, "Is deep learning a renormalization group flow?" *IEEE Access*, vol. 8, pp. 106 487–106 505, 2020.

[30] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, nov 1997.

[32] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, 2021.

[33] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, May 2019.

[34] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[35] L. Arras, G. Montavon, K.-R. Müller, and W. Samek, "Explaining recurrent neural network predictions in sentiment analysis," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2017, pp. 159–168.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.