2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Exploring the Potential of World Models for Anomaly Detection in Autonomous Driving

Daniel Bogdoll[†‡], Lukas Bosch[‡], Tim Joseph[†], Helen Gremmelmaier[†], Yitian Yang[†], and J. Marius Zöllner[†‡]
[†]FZI Research Center for Information Technology, Germany
bogdoll@fzi.de
[‡]Karlsruhe Institute of Technology, Germany

***Abstract*—In recent years there have been remarkable advancements in autonomous driving. While autonomous vehicles demonstrate high performance in closed-set conditions, they encounter difficulties when confronted with unexpected situations. At the same time, world models emerged in the field of model-based reinforcement learning as a way to enable agents to predict the future depending on potential actions. This led to outstanding results in sparse reward and complex control tasks. This work provides an overview of how world models can be leveraged to perform anomaly detection in the domain of autonomous driving. We provide a characterization of world models and relate individual components to previous works in anomaly detection to facilitate further research in the field.**

***Index Terms*—world model, anomaly, vision, corner case, autonomous driving, prediction, reconstruction, latent space**

I. Introduction

The detection of anomalies, or corner cases, is a challenging task with applications in autonomous driving. Neural networks, used in tasks like semantic segmentation, tend to be overconfident when confronted with the unseen. Therefore, an anomaly detection system can increase the reliability of autonomous systems which build on such components [1]. In perception systems of autonomous vehicles, anomalies can range from sensor failures due to bad lighting conditions, over pedestrians suddenly crossing the street, to abnormal driving patterns from other vehicles. The different contexts in which anomalies arise, and their inherent unpredictability, pose a challenge to their detection. Many recent approaches focus on unsupervised learning of models describing normality. The idea is to detect anomalies by their deviation from the learned model of normality [1]–[5]. Typically, the learning of normality is achieved by exploiting that models learn their training data's underlying patterns, which implicitly makes them learn a notion of normality corresponding to their training data [5].

At the same time, world models emerged in the field of Reinforcement Learning (RL) as a way to enable agents to predict the future conditioned on actions. This led to outstanding results in sparse reward and complex control tasks. Based on these successes, the question arises whether such world models, which learn a compact representation of the world and predict future changes, can be used outside of RL.

In this work, we introduce anomaly detection methods for autonomous driving in Sec. II and world models in Sec. III. In Sec. IV, we describe our concept for the detection of anomalies with world models and conclude our thoughts in Sec. V.

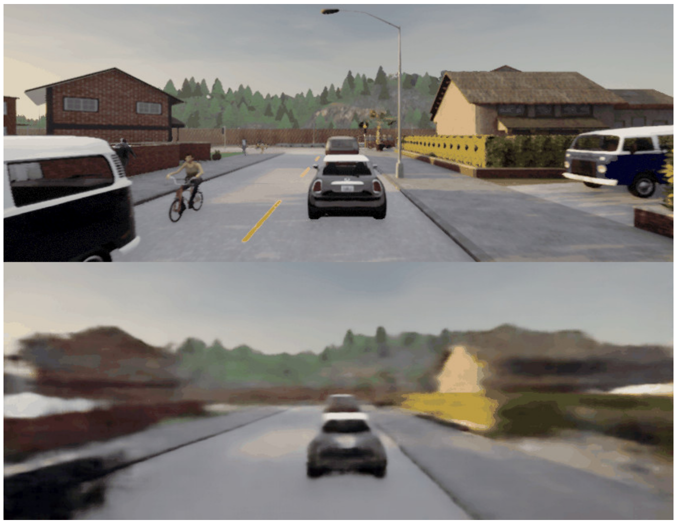

Fig. 1. The bottom row shows a scene reconstruction of a world model [6] Compared to the ground truth, the model cannot recover all scene components, such as the bicyclist. This phenomenon can be exploited through a clear definition of normality and targeted training to detect anomalies.

II. Anomaly Detection in Autonomous Driving

There are endless possibilities of deviations from normality, which makes it hard to formally define anomalies. Termöhlen et al. [1] were the first to formulate a definition in the domain of autonomous driving: "A corner case is given, if there is a non-predictable relevant object/class in [a] relevant location". This section provides an overview of works on defining and detecting anomalies.

Anomaly Types. Breitenstein et al. [7] developed a five-level systematization of corner cases, focussing on camera data. Heidecker et al. generalized the work and extended it by a method layer, which describes corner cases "due to uncertainty inherent in the methodology or the data" [8]. The first three layers describe anomalies in the surrounding environment. The fourth layer describes anomalies that stem from the utilized software stack itself. The *sensor layer* encompasses anomalies introduced by sensor attributes. The *content layer* describes anomalies detectable within a single frame, such as changing weather conditions or unknown objects. The *temporal layer* includes anomalies that become apparent only when analyzing multiple frames focusing on behavior.

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

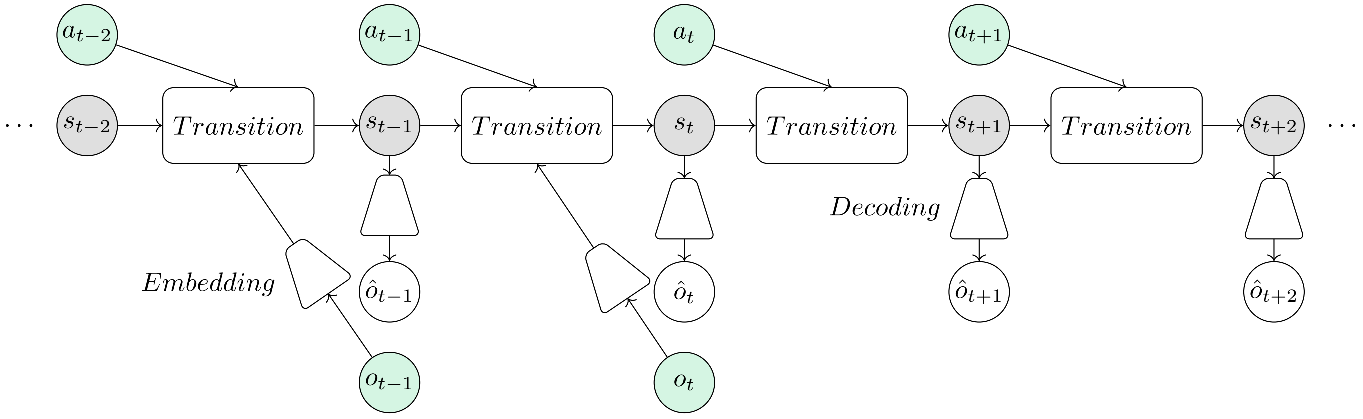

Fig. 2. A world model during inference, given high dimensional observations $o_t \ldots o_{t-i} \in \Omega$ and past and planned actions $a_{t-i} \ldots a_{t+j} \in A$ at time t. All state **transitions** up to $s_t \in S$ are computed with a *representation model* $p(s_t \mid s_{t-1}, a_{t-1}, o_t)$, where the observations are first being **embedded**. The embedding of actions is possible but optional. Future state **transitions** can be computed with the *prediction model* $p(s_t \mid s_{t-1}, a_{t-1})$ based on the Markov assumption, where each state only depends on its predecessor. With the *observation model* $p(o_t \mid s_t)$, reconstructions $\hat{o}_t \in \Omega$ can be **decoded** from state s_t.

The *method layer* describes anomalies introduced by the used methodology, including the underlying training data. These may be expressed through uncertainty or other means. Unlike the previous layers, the method layer defines anomalies depending on the system's capabilities. Zhou and Beyerer argue similarly, defining corner cases as "interpretation problem[s] in the networks" based on training and test data [9].

Detection Methods. As one of the early works, Breitenstein et al. [10] investigated concepts for detecting anomalies for autonomous driving based on camera data, classifying approaches into reconstructive, generative, predictive, confidence score, and feature extraction-based methods. Bogdoll et al. extended the work by including lidar- and radar-based methods [11]. Here, we provide an overview of these.

Reconstructive approaches aim to detect anomalies by reconstructing input frames from a compressed representation. Anomalies are identified when the reconstruction error or another suitable metric exceeds a predefined threshold, indicating the model's inability to represent the input accurately. Reconstruction-based methods build on the assumption that reconstructive models trained on normal input data fail to reconstruct anomalies. Similarly, *Generative methods* are also based on reconstructing methods, but "also regard the discriminator's decision or the distance between the generated and the training distribution" [10].

Predictive methods predict future frames based on preceding frames. Subsequently, anomalies are detected when a significant deviation between the prediction and the actual observation is encountered. The deviation can be measured with techniques from reconstruction-based approaches, e.g., reconstruction error.

Confidence score-based methods involve estimating the uncertainty associated with a model's prediction. Anomalies are detected when the confidence score of a model is low. Suggesting that the model is uncertain about the prediction might indicate that the observed data is potentially abnormal.

Lastly, *feature extraction based methods* detect anomalies by transforming the input data into a lower-dimensional feature space, which can emphasize meaningful patterns and separate anomalous data points from normal ones. Techniques such as deep feature extraction, clustering algorithms, and one-class support vector machines can be employed in this context. Contrary to previous methods, this approach only allows classification, not pixel or point-wise detection.

For training and evaluation of such such methods, various datasets are available [12]–[14]. Oftentimes, normality is represented by the classes and data provided by Cityscapes [15], while anomalies include categories such as costumed people, lost cargo, or animals.

III. World Models

This section provides a high-level characterization of world models as well as important examples from the literature. World models originate from the field of Reinforcement Learning. Here, for every state transition, an agent additionally receives a reward. Given the goal to maximize the expected cumulative reward, the question arises: Which actions to take? In model-free RL, the agent directly optimizes for this target based on its interactions with the environment. In model-based Reinforcement Learning, however, a dynamics model is learned first in order to predict the dynamics of the environment. This way, the costly policy learning can be done purely in the dynamics model.

While the term *world model* for such a dynamics model dates back to the early days of Artificial Intelligence [16], in modern days Ha and Schmidhuber coined the term [17]. However, they did not provide a clear definition. They were the first who modeled "dynamics observed from high dimensional visual data where [the] input is a sequence of raw pixel frames" [17]. LeCun broadly defines a world model as an "internal model of how the world works" [18]. Following Kendall, "A world model is a generative model that is able to predict the next state conditioned on an action" [19]. Chen et al. describe world models based on their capabilities to provide "abstract perceptual representations" and "explicit future predictions" [20]. Combining these concepts, we define a world model as follows:

> *A world model embeds sensory observations into a latent state, predicts action-conditioned state transitions, and is able to decode into observation space.*

Such a stochastic, generative *world model* $\mathcal{W}$, as shown in Figure II, can be described by three conditional probability distributions [21], [22], where the *representation model*

$$p(s_t \mid s_{t-1}, a_{t-1}, o_t) \tag{1}$$

describes the dependence of a latent state on the associated observation, the *prediction model*

$$p(s_t \mid s_{t-1}, a_{t-1}) \tag{2}$$

describes the transition from one latent state to the next. It is possible to predict multiple next states by sampling: $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$. Finally, the *observation model*

$$p(o_t \mid s_t) \tag{3}$$

describes the dependence of an observation on the associated latent state, which allows sampling reconstructions from it: $\hat{o}_t \sim p(o_t \mid s_t)$. In addition, a world model can have further connections or heads, e.g., an additional head for rewards is typical in Reinforcement Learning. Since a world model consists of multiple deep neural networks (DNN), Ha and Schmidhuber proposed a separation into a *Vision* and a *Memory* component [17]. Their vision model is responsible for compressing observations to compact representations, while the memory model is responsible for predicting latent states. As shown in Figure II, we find naming those components *Embedding* and *Transition* are more concise choices. Additionally, we explicitly introduce the *Decoding* model to emphasize the property of transforming abstract representations into explicit reconstructions, which aligns well with the three conditional probability distributions.

A. Problem Formulation

To model the interaction of an **agent** and a realistic **environment**, we adapt the well-known terminology from Partially Observable Markov Decision Processes (POMDP) and assume Markovian properties, as shown in Figure 3. It can be assumed that no **actor** within the environment has access to its latent **state**. In the domain of autonomous driving, any agent controlling an actor thus relies on sensory **observations** from the actor itself. The agent can be understood as a software component that controls the vehicle, given its sensor data as observations. With each **action**, the internal state of the environment gets updated.

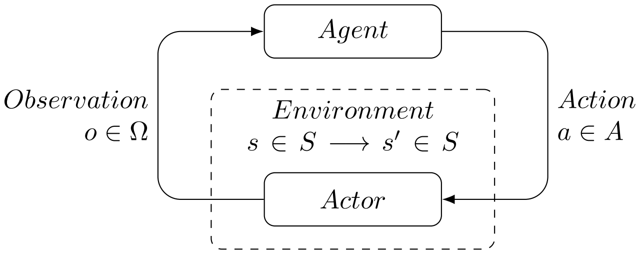

Fig. 3. Interaction of an agent with an actor in an environment, where A is a set of actions, S is a set of states, and Ω is a set of observations. Given an observation o, the action a of the agent results in a state transition $s \longrightarrow s'$ of the environment.

B. Embedding Models

While contrastive approaches exist [22], most recent works implement VAEs as the embedding model to embed observations in a latent state space and utilize a reconstruction loss during training [17], [21]–[26]. This also allows for decoding latent representations. Given a latent state s_t, the corresponding decoder can be used to obtain a reconstruction $\hat{o}_t$ of the original observation o_t. The state space structure is implicitly given by the embedding model. A common choice are Gaussian state spaces. Ha & Schmidhuber [17] use a convolutional Variational Autoencoder (VAE) to encode two-dimensional RGB image frames into low dimensional vectors $\mu, \sigma \in \mathbb{R}^{32}$, which represent the parameters of a Gaussian distribution $\mathcal{N}(\mu, \sigma I)$. Latent states $s \in \mathbb{R}^{32}$ are sampled from this distribution. The authors claim that the Gaussian prior makes the world model "more robust to unrealistic [s] vectors" predicted by their transition model. The locally linear latent state space dynamics model Embed to Control (E2C) [23] uses a similar VAE-type encoder, also resulting in a Gaussian state space. PlaNet, a model-based agent proposed by Hafner et al. [21], performs planning over imagined latent state trajectories, using the embedding model from Ha and Schmidhuber [17]. The reinforcement learning agent Dreamer, proposed by Hafner et al. [22], utilizes the previously introduced world model PlaNet. In contrast, Dreamers recent iterations DreamerV2 [25] and DreamerV3 [26] replace the Gaussian state space with a categorical state space, where image frames are mapped to a categorical distribution. From this distribution, a latent state s encoded in categoricals is sampled.

Variational Recurrent Kalman Networks (VRKN), proposed by Becker & Neumann [24], encode observations using a Neural Network with two output heads, which directly maps an observation to an intermediate representation w_t and its diagonal covariance σ^{w_t}. The latent state is then calculated using Bayes rule for Gaussian distributions (Kalman updates).

Hu et al. presented their Model-Based Imitation Learning (MILE) for predictions in urban environments [6], [16]. Compared to previous works, which apply world models in the context of RL, they use Imitation Learning (IL) to train their model. Inspired by Philion and Fidler [27], they encode high-resolution observations and lift the resulting features into 3D space based on a learned depth probability distribution for each feature. A pooling operation into Bird's-Eye View (BeV) space and mapping to a 1D vector follow.

C. Transition Models

A central idea of a world model is to be able to look ahead by predicting future latent states. Given a latent state s_t and an action a_t, a world model is able to predict future states. Therefore, in addition to an embedding model as described above, a *transition model* is needed.

Inspired by Stochastic Optimal Control algorithms, E2C [23] assumes locally linear dynamics at each time step to approximate global non-linear dynamics. Future latent states are sampled from a Gaussian distribution $\mathcal{N}(A_t\mu_t + B_t a_t + o_t, C_t)$, where μ_t and σ_t are the parameters of s_t computed by the embedding model. $A(s_t)$ is the local Jacobian with respect to s_t, $B(s_t)$ the local Jacobian with respect to a_t, $o(s_t)$ is an offset and $C_t = A_t \Sigma_t A_t^T + H_t$ with system noise H_t.

Ha & Schmidhuber [17] use a Recurrent Neural Network (RNN) with a Mixture Density Network (MDN) output layer, called MDN-RNN, to model the probability distribution $p(s_{t+1} \mid s_t, a_t, h_t)$, from which s_{t+1} can be sampled. The additional parameter h_t is the RNNs hidden state.

PlaNet [21] uses a Recurrent State Space Model (RSSM) [21] to model latent dynamics. RSSMs combine State Space Models (SSM) and Recurrent Neural Networks by introducing a deterministic path to SSMs, consisting of the hidden states of an RNN. More precisely, the transition model is an RSSM consisting of a deterministic state model $h_t = f(h_{t-1}, s_{t-1}, a_{t-1})$ implemented as an RNN, and a stochastic state model $s_t \sim p(s_t \mid h_t)$. The sequence of hidden states $h_{1:T}$ is called the deterministic path, while the sequence of stochastic latent states $s_{1:T}$ is called the stochastic path of the RSSM. While purely deterministic transitions "prevent the model from capturing multiple futures", the stochastic transitions in state space models make "it difficult to remember information over multiple time steps" [21]. The RSSM of PlaNet is reused in all iterations of Dreamer.

VRKN [24] improve on RSSMs by eliminating the need for a deterministic path with a more principled modeling of aleatoric and epistemic uncertainty. This is achieved by using an inference scheme consistent with the generative model, i.e., by imposing the same assumptions on World Model learning and inference. Latent state estimates are forwarded using closed-form Gaussian marginalization, and then updated based on the respective observation using Kalman updates, thereby incorporating information from previous states. Epistemic uncertainty is captured using Monte Carlo dropout layers. Aleatoric uncertainty is captured in two ways: First, the latent state is a distribution capturing multiple possible futures, which provides information about aleatoric uncertainty. Second, Kalman updates enable the update of state estimates based on the respective observation. RSSMs can not correct latent state estimates and thus explain differences between estimates and observations implicitly by the transition model. VRKN can correct differences between state estimates and observations with Kalman updates. This eliminates the overestimation of aleatoric uncertainty present in RSSMs.

TABLE I
APPLICABILITY OF ANOMALY DETECTION APPROACHES WITH WORLD MODELS FOR CORNER CASE LEVELS

Layer	Sensor		Content			Temporal
Level	Hardware	Physical	Domain	Object	Scene	Scenario
Reconstructive	—	—	✓	✓	✓	—
Generative	—	—	✓	✓	✓	—
Predictive	—	—	—	—	—	✓
Confidence Score	—	—	✓	—	—	—
Feature Extraction	—	—	✓	—	—	—

MILE [6] follows a similar approach as Hafner et al [21]. The difference is that the deterministic history h_t in MILE is obtained only from the previous history h_{t-1} and stochastic state s_{t-1}, i.e., $h_t = f(h_{t-1}, s_{t-1})$ where f is modeled as an RNN. Actions are later introduced in the stochastic state model $s_t \sim p(s_t \mid h_t, a_{t-1})$.

In addition to this academic progress, world models are also finding their way into the industry. GAIA-1 by Wayve, trained on fleet data, "leverages video, text, and action inputs to generate realistic driving videos" [28]. Similarly, Elluswamy presented early results of a "general world model" developed by Tesla, also trained on fleet data [29].

IV. Anomaly Detection with World Models

After presenting anomalies in the domain of autonomous driving and recent world models, we now introduce our concept of utilizing world models to detect multiple forms of anomalies. Following the approach presented by Breitenstein et al. [10], we map the previously introduced detection methods to corner case levels [7], [8], as shown in Table I. We mark a method only as applicable if it is the primary detection approach, since oftentimes multiple methods are being used, supplementing each other. While *sensor layer* corner cases can affect sensory observations and might thus be detectable, we focus on anomalies in the surrounding environment. Thus we deem anomalies on the *sensor layer* out of scope.

World models are especially interesting for anomaly detection, as they are able to perform all detection approaches from the literature: As the *embedding model* is typically implemented as a stochastic VAE, it has both reconstructive and generative capabilities. The core purpose of the *transition model* is prediction. For both the embedding model and the transition model, epistemic uncertainty estimates are possible. Finally, since we have access to the model, features are available. This allows world models to detect a wide variety of anomalies in a single pass, compared to a plethora of models which would be necessary otherwise. In addition, the end-to-end training approach of world models allows for the manifestation of a common definition of normality.

In general, the field of anomaly detection, especially for autonomous driving, faces several challenges at the moment [11], [14], [30]. We want to address them systematically, introducing several assumptions in the following.

Normality. Since we aim to detect anomalies based on the capabilities of a world model, the learned normality of this world model is of utmost importance. However, real-world data collected at scale will contain unlabeled anomalies [31]. Thus, achieving a clear disjunction between normality and anomalies, represented by training and evaluation data, is hard. This is especially important for un- and self-supervised approaches, which learn from all patterns included in the training data. If they pick up unknown anomalies from the training data, which are defined as anomalies in the evaluation data, this can lead to a situation where these human-defined anomalies are no longer anomalies for the detecting system.

Mapping. Typically, a detection system provides "metric-based assessments of situations" [30]. To provide more context, it is often of interest to map detections to a corner case category, as shown in Table I. While the Table only shows the applicability of methods, the actual assignment during inference is challenging. For example, if a latent representation of an input deviates from learned representations of normal samples, a domain shift or a sufficiently large unknown object could be the reason.

Evaluation. For evaluation, concrete scenarios containing anomalies are necessary. However, deriving concrete scenarios from the rather broad corner case categories in a scalable fashion is hard [30], [32]. In most benchmarks, the anomaly type is defined as either unknown objects, domain shifts, or abnormal behavior [14]. Accordingly, the computer vision community focuses on "contextual anomalies on the scene level" [11]. Addressing these challenges, we define the following assumptions:

- To properly define anomalies, full control over both the training and evaluation data is necessary
- For a precise definition of normality, anomalies cannot be part of the training data
- For the evaluation, human-defined anomalies are required, which must be in agreement with the normality defined by the training data

In order to have full control over both training and test data, a controlled environment is necessary. The next sections will first introduce training and test data, followed by the general inference concept of the model and concrete methods for the detection of anomalies.

A. Training Data

As described earlier, the definition of an anomaly depends on the capabilities of the perception system, which depends on the data it was trained on. Following this concept, in order to detect anomalies, a concept of normality needs to be defined first. We can clearly define both the static and the dynamic environment of the training dataset $\mathcal{D}_{norm}^{train}$ based on several attributes: *Region, Weather, Time of Day, Objects, Actors, and Behaviors*. In a similar fashion, these can be found in the 6-Layer PEGASUS model [33]. Based on a simulation engine [34], [35], these can be set in a reproducible and deterministic way. For the behavior of the ego-vehicle during training, a driving model providing expert demonstrations is well suited [6], [34], [36]. Combining the static and dynamic attributes of the environment and a driving model for the ego vehicle allows for the assembly of a well-defined and large-scale dataset for training. If anomaly detection approaches shall be compared based on an evaluation dataset $\mathcal{D}_{ano}^{eval}$, they need to share a common definition of normality in the form of a prescribed training dataset $\mathcal{D}_{norm}^{train}$. Today, this is typically not the case, which makes it impossible to truly compare approaches [14]. For training details, we refer to common literature in the domain of world models [6], [17], [21], [22].

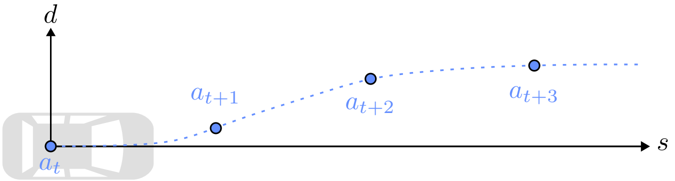

Fig. 4. Planned actions in the context of autonomous driving. Based on a vehicle model, the *planning module* determines a finite list of actions for the ego vehicle in order to reach planned future vehicle states.

B. Evaluation Data

For the evaluation dataset $\mathcal{D}_{ano}^{eval}$ we need to integrate anomalies purposefully. Since the training dataset is well defined, this is possible on all corner case levels separately: On the *sensor layer*, perturbations can be added to the sensor configuration or sensor data. On the *domain level*, different regions, weather conditions, or daytimes can be chosen. For both the *object* and *scene level*, unknown objects or known objects in atypical places can be inserted. Finally, for the *scenario layer*, atypical behaviors can be defined or generated. For all of these anomalies, ground truth is available, which allows us to use them for evaluation. In addition, it is possible to combine corner case levels.

C. Model Inference

As shown in Figure II, future actions are necessary for the rollout of a world model. In Reinforcement Learning or Imitation Learning, typically a policy is learned to select an action $a_t = (acc_t, \delta_t)$ during inference, as visible in Figure 5. Applied to the domain of anomaly detection for autonomous driving though, a set of planned actions is known beforehand which can be utilized for the rollout, similar to the known actions used for planning with the PLaNet model [21]. In autonomous driving, it is still common to divide the overarching task into a subset of modules, most importantly perception, prediction, planning, and control. Given information about the state of the actor and some vehicle model, the planning module is able to compute a list of planned actions, as shown in Figure 4. While the state of the actor is implicitly included in the environment state s, some attributes of it can be explicitly known or measured, as shown in Equation 4.

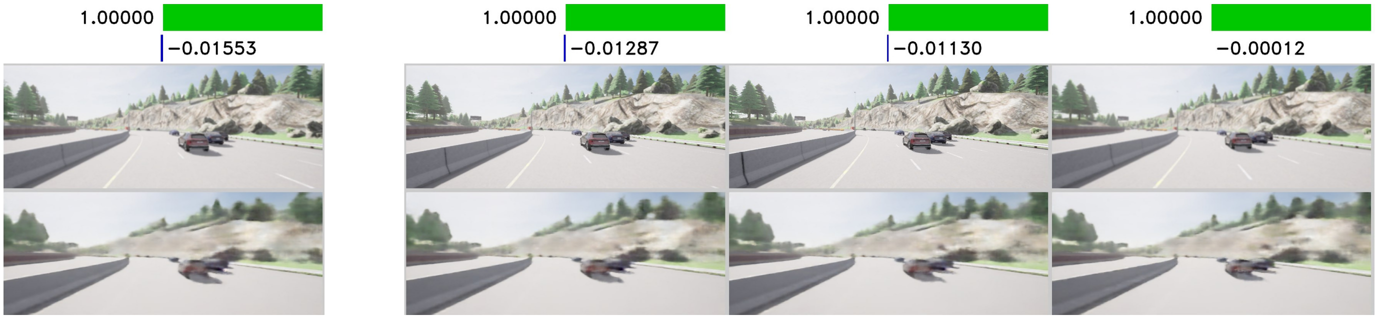

Fig. 5. On the left, the last input frame for the prediction and its reconstruction are shown. The bottom row on the right shows the predictions of a world model conditioned on actions, where each action consists of acceleration and steering angle values as shown in the top row [6]. Compared with the ground truth, the model is able to predict normal behavior. Under the hypothesis that it cannot predict atypical behavior unseen during training, differences between future observations and the predictions can be used for anomaly detection.

$$s_{\text{actor}} = \left\{ \begin{array}{ll} \text{coordinates:} & (x, y, z), \\ \text{rotation:} & (\phi, \theta, \psi), \\ \text{velocity:} & v, \\ \text{acceleration:} & acc, \\ \text{steering\_angle:} & \delta, \\ \vdots & \end{array} \right\} \tag{4}$$

Now, given observations, a list of past and planned actions, and a world model $\mathcal{W}$, a sequence of future latent states, also referred to as a *rollout trajectory*, can be predicted. Based on

$$p(o_{1:T} \mid a_{0:T-1}) \triangleq \mathbf{E}_{p(s_{1:T} \mid a_{0:T-1})} \left[\prod_{t=1}^{T} p(o_t \mid s_t) \right] \tag{5}$$

which describes the next T observations conditioned on T given actions [21], we can sample N times from the distribution in order to derive multiple futures in observation space:

$$\{\hat{o}^n_{1:T}\}^N_{n=1} \sim p(o_{1:T} \mid a_{0:T-1}) \tag{6}$$

D. Anomaly Detection

To the best of our knowledge, no existing method for anomaly detection based on sensor data utilizes a world model as described in Section III. The anomaly detection method proposed by Chakraborty et al. [37] is the only approach known to us which utilizes embeddings, latent state transitions, and decodings to detect anomalies. However, their transition model is not action-conditioned and their input consists of map and trajectory data instead of sensor observations. In the following, we will introduce methods from the literature, which can also be implemented based on a world model, from the categories *reconstructive, generative, predictive, confidence score, and feature extraction*, as shown in Table I.

Reconstructive and Generative. Most world models leverage embedding models which make use of a reconstructive training objective [17], [21], [22], [25], [26], which form a reconstructive element inside a world model applicable to anomaly detection. While there are many anomaly detection approaches that use different forms of reconstructions [38]–[40], their concepts can be applied to world models. It is important to keep in mind that well-trained VAEs generalize well and are able to reconstruct the unseen [41], which is why special care is necessary during training and the design of anomaly detection methods.

Confronted with a *domain level* corner case, the reconstruction quality can be poor for the entire frame. This can be detected, especially in comparison to cases where only certain regions of the reconstruction have poor quality. This classification task can be performed based on methods such as reconstruction quality [42], reconstruction probability [43], or even combinations with extracted features [44], [45]. Some of these methods also provide pixel-wise anomaly scores, which can be used for *object level* or *scene level* corner cases, which are hard to distinguish from a detection point of view. For example, Vojir et al. [46] aim to detect anomalies on roads using a reconstructive approach. The idea is to reconstruct the surface of the road and the remaining environment from a latent representation in such a way that the road reconstruction shows minimal error while the reconstruction of the remaining environment shows maximal error. The resulting pixel-wise errors are combined with a semantic segmentation output by feeding both into the semantic coupling module, which outputs two maps, one for the road class and one for the anomaly class.

Predictive. Predictive capabilities form the core of a world model, enabling the prediction of future latent states and, based on the observation model, the reconstruction into the observation space, as shown in Figure 5. Given such predictions, either the epistemic uncertainty or the comparison to future observations, which are available after Δt, can be used to detect *scenario level* corner cases. When multiple futures are being predicted, distance metrics in their latent representations can be used to detect the prediction which generally aligns best with the ground truth.

Liu et al. [3] were the first to propose an approach to video anomaly detection involving the prediction of a future image frame and comparing the predicted frame with the ground truth frame. The approach adopts U-Net [47] to predict future frames based on the preceding image sequence. Intensity and gradient losses are directly computed between predicted and ground truth frames. To compute an optical flow loss, the authors leverage Flownet [48] to estimate the optical flow between the predicted or ground truth frames and their preceding frames. An optical flow constraint boosts anomaly detection performance as it imposes motion consistency for normal events through capturing temporal information. Based on the assumption that normal events are well predicted, the final anomaly score is based on the Peak Signal-to-Noise Ratio (PSNR) between predicted and ground truth frames. Similarly, Termöhlen et al. [1] use a convolutional Autoencoder to predict an image frame from a sequence of preceding frames, but in the domain of camera-based autonomous driving. An error map is built from pixel-wise prediction errors and weighted subsequently based on semantic information.

Chakraborty et al. [37] propose a Structural Attention-based Recurrent VAE (SABer-VAE) for detecting anomalies in vehicle trajectories. The environment is modeled as a road map consisting of equidistant nodes on each lane. Furthermore, vehicle positions are given in coordinates at each timestep. The authors detect anomalies in observed vehicle trajectories by predicting future vehicle states and calculating a prediction loss from which a final anomaly score is derived.

Their *embedding model* consists of two paths. The primary encoding path models vehicle-vehicle interactions with a self-attention module. It takes vehicle positions and their relative distance to each other as inputs and transforms them, analogously to [17], but using a different architecture, into the parameters μ and σ of a Gaussian latent state distribution. The secondary encoding path models lane-vehicle interactions and consists only of an attention module to calculate embeddings. Their *transition model* uses a stochastic Koopman operator, where two Neural Networks predict the Koopman matrices in order to predict the “one-step future states of vehicles”.

Confidence- and Feature-based. The process of embedding environment observations into a latent space is one form of feature extraction. Based on these features or uncertainties w.r.t. to latent states, a whole observation or state can be classified as an anomaly [49], enabling the detection of *domain level* corner cases. In single cases, where a large part of an image consists of *object level* or *scene level* corner cases, these can also be detected, but this dependence does not allow for the general detection of these levels. For example, Norlander & Sopasakis [50] propose the Conditional Latent Space Variational Autoencoder (CL-VAE) to detect anomalies in class-labeled data. The authors train a VAE using a Gaussian mixture model such that for each class in the data set, a Gaussian prior is fitted in the latent space. The authors show that anomalies lie between the clusters, which can be visually confirmed. The favorable structure of the latent space can be used for anomaly detection by employing techniques such as Isolation Forests [51]. Another approach for anomaly detection in surveillance videos is proposed by Park et al. [5], who propose a memory module, which represents a storage of latent representations of normal image frames. A U-Net [47] based encoder extracts query features from a video frame. The queries are fed into the memory model to update the latent memory items. To reconstruct image frames, the memory items are concatenated with the query features and fed into the decoder, which outputs a reconstructed frame. Since “prediction can be considered as a reconstruction of the future frame using previous ones” [5], the model can be adapted to predict the next frame with minimal effort using the same underlying architecture and loss functions. For training, the authors extend the reconstruction loss by adding a feature compactness as well as a feature separateness loss to condition the latent memory and query space. During inference, an anomaly score is obtained by combining the Euclidean distance between the extracted query and the closest memory item with the PSNR between input and reconstruction.

V. Conclusion

We presented a detailed characterization of *world models* together with anomaly detection approaches from the literature which build on ideas already present in world models or applicable to our characterization. We show that world models hold great potential for the task of anomaly detection in the context of autonomous driving. Along with a clear definition of anomalies and the categorization of anomaly detection approaches into predictive, generative, confidence score, and feature extraction-based methods, we showed that world models can be used to implement existing ideas from current approaches in a unified anomaly detection framework.

VI. Acknowledgment

This work results from the jbDATA project supported by the German Federal Ministry for Economic Affairs and Climate Action of Germany (BMWK) and the European Union, grant number 19A23003H.

References

[1] J.-A. Termöhlen, A. Bär, D. Lipinski, and T. Fingscheidt, “Towards Corner Case Detection for Autonomous Driving,” in *Intelligent Vehicles Symposium (IV)*, 2019.

[2] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning Temporal Regularity in Video Sequences,” *arXiv:1604.04574*, 2016.

[3] W. Liu, W. Luo, D. Lian, and S. Gao, “Future Frame Prediction for Anomaly Detection – A New Baseline,” in *Computer Vision and Pattern Recognition Conference (CVPR)*, 2018.

[4] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh, “Learning Regularity in Skeleton Trajectories for Anomaly Detection in Videos,” in *Computer Vision and Pattern Recognition Conference (CVPR)*, 2019.

[5] H. Park, J. Noh, and B. Ham, “Learning Memory-Guided Normality for Anomaly Detection,” in *Computer Vision and Pattern Recognition Conference (CVPR)*, 2020.

[6] A. Hu, G. Corrado, N. Griffiths, Z. Murez, C. Gurau, H. Yeo, A. Kendall, R. Cipolla, and J. Shotton, “Model-Based Imitation Learning for Urban Driving,” in *Neural Information Processing Systems (NeurIPS)*, 2022.

[7] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt, “Systematization of Corner Cases for Visual Perception in Automated Driving,” in *Intelligent Vehicles Symposium (IV)*, 2020.

[8] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt, and B. Sick, “An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving,” in *Intelligent Vehicles Symposium (IV)*, 2021.

[9] J. Zhou and J. Beyerer, “Corner Cases in Data-Driven Automated Driving: Definitions, Properties and Solutions,” in *Intelligent Vehicles Symposium (IV)*, 2023.

[10] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt, “Corner Cases for Visual Perception in Automated Driving: Some Guidance on Detection Approaches,” *arXiv:2102.05897*, 2021.

[11] D. Bogdoll, M. Nitsche, and J. M. Zöllner, “Anomaly Detection in Autonomous Driving: A Survey,” in *Computer Vision and Pattern Recognition Conference (CVPR) Workshop*, 2022.

[12] D. Bogdoll, F. Schreyer, and J. M. Zöllner, “AD-Datasets: A Meta-Collection of Data Sets for Autonomous Driving,” in *International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, 2022.

[13] D. Bogdoll, J. Hendl, F. Schreyer, N. Gowda, M. Färber, and J. M. Zöllner, “Impact, Attention, Influence: Early Assessment of Autonomous Driving Datasets,” in *International Conference on Control and Robotics Engineering (ICCRE)*, 2023.

[14] D. Bogdoll, S. Uhlemeyer, K. Kowol, and J. M. Zöllner, “Perception Datasets for Anomaly Detection in Autonomous Driving: A Survey,” in *Intelligent Vehicles Symposium (IV)*, 2023.

[15] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] A. Hu, “Neural World Models for Computer Vision,” Ph.D. dissertation, University of Cambridge, 2022.

[17] D. Ha and J. Schmidhuber, “World Models,” *arXiv:1803.10122*, 2018.

[18] Y. LeCun, “A Path Towards Autonomous Machine Intelligence,” *OpenReview:BZ5a1r-kVsf*, 2022.

[19] A. Kendall. (2023) Frontiers in Embodied AI for Autonomous Driving. CVPR Workshop on End-to-End Autonomous Driving. [Online]. Available: https://www.youtube.com/watch?v=Rn9mvSDdSqo

[20] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, “End-to-end Autonomous Driving: Challenges and Frontiers,” *arXiv:2306.16927*, 2023.

[21] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, “Learning Latent Dynamics for Planning from Pixels,” in *International Conference on Machine Learning (ICML)*, 2019.

[22] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, “Dream to Control: Learning Behaviors by Latent Imagination,” in *International Conference on Learning Representations (ICLR)*, 2020.

[23] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller, “Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images,” in *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[24] P. Becker and G. Neumann, “On Uncertainty in Deep State Space Models for Model-Based Reinforcement Learning,” *Transactions on Machine Learning Research (TMLR)*, 2022.

[25] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, “Mastering Atari with Discrete World Models,” in *International Conference on Learning Representations (ICLR)*, 2021.

[26] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, “Mastering Diverse Domains through World Models,” *arXiv:2301.04104*, 2023.

[27] J. Philion and S. Fidler, “Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D,” in *European Conference on Computer Vision (ECCV)*, 2020.

[28] WAYVE, “Introducing GAIA-1: A Cutting-Edge Generative AI Model for Autonomy,” https://wayve.ai/thinking/introducing-gaia1/, 2023, accessed: 2023-06-19.

[29] A. Elluswamy, “Foundation models for autonomy,” https://www.youtube.com/watch?v=6x-Xb_uT7ts, CVPR Workshop on Autonomous Driving, 2023, accessed: 2023-06-30.

[30] D. Bogdoll, J. Breitenstein, F. Heidecker, M. Bieshaar, B. Sick, T. Fingscheidt, and J. M. Zöllner, “Description of corner cases in automated driving: Goals and challenges,” in *International Conference on Computer Vision (ICCV) Workshop*, 2021.

[31] X. Du, X. Wang, G. Gozum, and Y. Li, “Unknown-Aware Object Detection: Learning What You Don’t Know from Videos in the Wild,” in *Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

[32] D. Bogdoll, S. Guneshka, and J. M. Zöllner, “One Ontology to Rule Them All: Corner Case Scenarios for Autonomous Driving,” in *European Conference on Computer Vision (ECCV) Workshop*, 2022.

[33] M. Scholtes, L. Westhofen, L. R. Turner, K. Lotto, M. Schuldes, H. Weber, N. Wagener, C. Neurohr, M. H. Bollmann, F. Körtke, J. Hiller, M. Hoss, J. Bock, and L. Eckstein, “6-layer model for a structured description and categorization of urban traffic and environment,” *IEEE Access*, vol. 9, 2021.

[34] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Conference on Robot Learning (CoRL)*, 2017.

[35] F. Mütsch, H. Gremmelmaier, N. Becker, D. Bogdoll, M. R. Zofka, and J. M. Zöllner, “From model-based to data-driven simulation: Challenges and trends in autonomous driving,” in *Computer Vision and Pattern Recognition Conference (CVPR) Workshop*, 2023.

[36] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, “End-to-end urban driving by imitating a reinforcement learning coach,” in *International Conference on Computer Vision (ICCV)*, 2021.

[37] N. Chakraborty, A. Hasan, S. Liu, T. Ji, W. Liang, D. L. McPherson, and K. Driggs-Campbell, “Structural Attention-Based Recurrent Variational Autoencoder for Highway Vehicle Anomaly Detection,” in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023.

[38] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena, “Pixel-wise Anomaly Detection in Complex Driving Scenes,” in *Computer Vision and Pattern Recognition Conference (CVPR)*, 2021.

[39] M. Grcić, P. Bevandić, Z. Kalafatić, and S. Šegvić, “Dense anomaly detection by robust learning on synthetic negative data,” *arXiv:2112.12833*, 2021.

[40] M. Grcić, J. Šarić, and S. Šegvić, “On advantages of mask-level recognition for outlier-aware segmentation,” in *Computer Vision and Pattern Recognition Conference (CVPR) Workshop*, 2023.

[41] D. Bogdoll, J. Jestram, J. Rauch, C. Scheib, M. Wittig, and J. M. Zöllner, “Compressing Sensor Data for Remote Assistance of Autonomous Vehicles using Deep Generative Models,” *Conference on Neural Information Processing Systems (NeurIPS) Workshop*, 2021.

[42] H. S. Vu, D. Ueta, K. Hashimoto, K. Maeno, S. Pranata, and S. M. Shen, “Anomaly Detection with Adversarial Dual Autoencoders,” *arXiv:1902.06924*, 2019.

[43] P. Munjal, A. Paul, and N. C. Krishnan, “Implicit Discriminator in Variational Autoencoder,” in *International Joint Conference on Neural Networks (IJCNN)*, 2020.

[44] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, “Latent Space Autoregression for Novelty Detection,” in *Computer Vision and Pattern Recognition Conference (CVPR)*, 2019.

[45] L. Wang, D. Zhang, J. Guo, and Y. Han, “Image Anomaly Detection Using Normal Data Only by Latent Space Resampling,” *Applied Sciences*, vol. 10, 2020.

[46] T. Vojir, T. Šipka, R. Aljundi, N. Chumerin, D. O. Reino, and J. Matas, “Road Anomaly Detection by Partial Image Reconstruction with Segmentation Coupling,” in *International Conference on Computer Vision (ICCV)*, 2021.

[47] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[48] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, “FlowNet: Learning Optical Flow with Convolutional Networks,” in *International Conference on Computer Vision (ICCV)*, 2015.

[49] S. Klaus, “Anomaly Detection in the Latent Space of VAEs,” Abschlussarbeit - Bachelor, Karlsruher Institut für Technologie (KIT), 2022.

[50] E. Norlander and A. Sopasakis, “Latent space conditioning for improved classification and anomaly detection,” *arXiv:1911.10599*, 2019.

[51] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in *International Conference on Data Mining (ICDM)*, 2008.