VRSTNN: Visual-Relational Spatio-Temporal Neural Network for Early Hazardous Event Detection in Automated Driving Systems

Dannier Xiao[1], Mehrdad Dianati[1,2], Paul Jennings[1] and Roger Woodman[1]

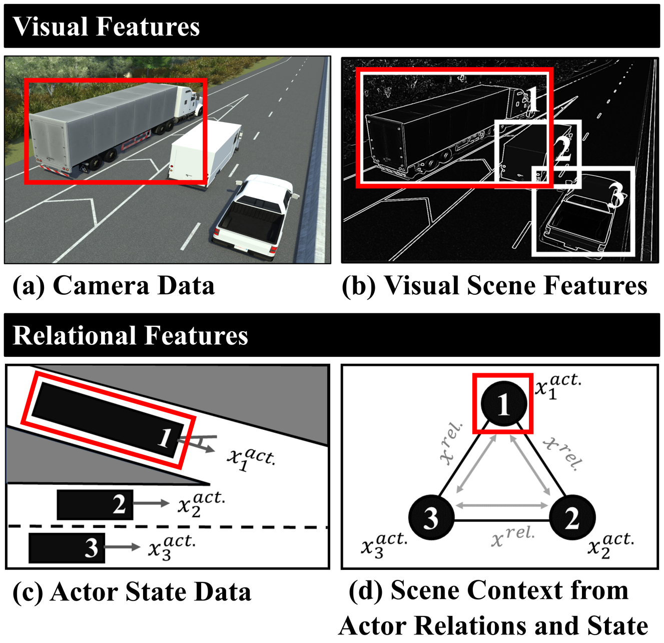
***Abstract*—Reliable and early detection of hazardous events is vital for the safe deployment of automated driving systems. Yet, it remains challenging as road environments can be highly complex and dynamic. State-of-the-art solutions utilise neural networks to learn visual features and temporal patterns from collision videos. However, in this paper, we show how visual features alone may not provide the essential context needed to detect early warning patterns. To address these limitations, we first propose an input encoding that captures the context of the scene. This is achieved by formulating a scene as a graph to provide a framework to represent the arrangement, relationships and behaviours of each road user. We then process the graphs using graph neural networks to identify scene context from: 1) the collective behaviour of nearby road users based on their relationships and 2) local node features that describe individual behaviour. We then propose a novel visual-relational spatio-temporal neural network (VRSTNN) that leverages multi-modal processing to understand scene context and fuse it with the visual characteristics of the scene for more reliable and early hazard detection. Our results show that our VRSTNN outperforms state-of-the-art models in terms of accuracy, F1 and false negative rate on a real and synthetic benchmark dataset: DOTA and GTAC.**

***Index Terms*—Hazardous event detection, spatio-temporal neural networks, visual and relational graph networks, visual convolutional networks, automated and autonomous vehicles**

I. Introduction

Automated driving systems (ADS) have the potential to greatly reduce the 1.35 million global road fatalities per year [1] and offer many other benefits [2]. However, to realise their full potential, higher levels of autonomy are required, i.e., SAE Level 3+ [3]. To enable the safe deployment of L3+ functions, the ADS must be able to detect scenarios that may lead to harm (i.e., hazardous events) and do so early, to enable a safe handover in L3 and a minimum-risk manoeuvre in L4.

Due to complex and diverse road environments, the accurate and timely detection of hazardous events is a challenging spatio-temporal pattern recognition problem. It requires understanding how scene objects like vehicles and pedestrians interact in space (spatial) and tracking how those interactions change in time (temporal). Interactions refer to how road users affect and respond to each other and provide the essential scene context to detect hazardous events early. For example in Fig. 1a, the difference between a safe and hazardous highway merge is difficult to assess visually without context or history: it may be safe if one car slows down to allow the other to merge but is hazardous if highway traffic fails to slow and the merging vehicle does not match the speed of highway traffic.


Visual Features

(a) Camera Data

(b) Visual Scene Features

Relational Features

(c) Actor State Data

(d) Scene Context from Actor Relations and State

Fig. 1: **Seeing is not the same as understanding:** context is required to detect hazardous events accurately and early. For example, in the motorway merge illustrated in **(a)**, the difference between a safe or hazardous merge requires context based on actor distances and speeds in relation to other road users. To provide scene context, our VRSTNN model fuses **(b)** visual scene features from image data with **(d)** context from graph-encoded actor data describing actor relationships $(x^{rel.})$ and state information (x^{actor}), e.g., speed and heading.

Herein referred to as hazard detection, traditional techniques propose physics-based methods to predict colliding trajectories [4] but struggle to capture the frequent interactions between drivers that cause sudden changes in trajectory (e.g., motorway merge or exit). Consequently, state-of-the-art (SOTA) hazard detection methods utilise deep learning models to capture complex interactions in both space and time using spatio-temporal neural networks (STNNs). SOTA works [5]–[7] propose classifying hazardous events by taking collision

*This research received funding support from Centre for Doctoral Training in Future Mobility Technologies at WMG, University of Warwick

[1] Intelligent Vehicles Research Group, WMG, University of Warwick, Coventry, UK {dannier.xiao, m.dianati, paul.jennings, r.woodman}@warwick.ac.uk

[2] School of Electronics, Electrical Engineering and Computer Science (EEECS), Queen's University, Belfast, UK m.dianati@qub.ac.uk

videos as input and processing each frame with a convolutional neural network (CNN) to learn visual features, followed by a recurrent neural network (RNN) to extract temporal patterns.

CNNs are widely used to extract visual features from images. Though, sole reliance on visual features may draw spurious patterns [8]–[11] and perception errors have been linked to 53% of Waymo incidents [12]. This highlights that seeing is not equal to understanding. Real-world driving often diverges from textbook guidelines and thus, context from actor interactions is crucial. However, identifying object relations is difficult for traditional CNNs, which use filters to extract visual features and lack explicit memory or attention [13]–[15].

Given the need for early hazard detection to enable a safe handover or minimum risk manoeuvre [16]–[18], this paper demonstrates the importance of context and proposes a real-time hazard detection system for the timely and accurate detection of hazardous events and benchmark against SOTA models and human response time to unexpected road hazards from the literature (1.1-1.8s [19]–[21], detailed in IV-C).

To achieve this, we first investigate different input encodings and feature extractors to understand what scene features are needed and how to process these features to identify hazards. From our results, we highlight the importance of scene context and propose a novel graph encoding of scene data that provides a framework to represent the arrangement, relationships and behaviours of each road user (i.e., actors). We achieve this by representing the road users as nodes and their relationships as edges, which is detailed in III-C2. At each node, we encode state data that describes actor acceleration, trajectory and other key kinematic indicators that precede accidents [16], [22].

We then investigate how to process the graphs to extract the context needed to capture early warning patterns. For this purpose, we use GNNs to extract scene context from the global graph structure and from the local node relationships. GNNs are considered to be versatile data-driven models. Our review of graph-based hazard detection methods revealed 19% higher accuracy [23] and 9x faster inference [18] of GNNs over CNN counterparts. In this study, we compare our GNN processing of actor state data against SOTA CNNs processing of image input to reveal a synergy between the networks. CNNs can extract visual features from semantically complex images but may lack contextual understanding. Thus, GNNs can provide this context by interpreting graph relations and structure.

Motivated by the above strength of the GNN and CNN-based approaches, we leverage the synergy between them to propose a novel visual-relational spatio-temporal neural network (VRSTNN) that fuses visual features with scene context to enable early detection, as illustrated in Fig. 1. Our VRSTNN model leverages CNN's ability to recognise hazardous visual patterns irrespective of position in the image (i.e., translation invariance). In addition, it leverages GNN's ability to recognise anomalies in graph structure that represent road user arrangement and relationships, irrespective of node order (i.e., permutation invariance). We then fuse the latent representation from each spatial block by concatenation along the feature dimension before temporal processing with a long-short-term memory (LSTM) network. Our study shows that the proposed VRSTNN approach helps achieve higher accuracy and earlier detection than the SOTA models in the literature.

In summary, the contributions of this paper are:

- Investigation of different visual and relational input encodings and spatial feature extractors to understand what scene features are necessary and how to process them.
- Proposition of a novel relational encoding of actor state based on leading kinematic indicators [16], [22]. 12 types of actor state and 8 spatial relation types are encoded in a bespoke relational graph structure to represent local and global context for GNN processing. In addition to a tensor structure to test CNN's ability to learn relational features.
- A novel VRSTNN architecture for hazard detection that utilises multi-modal input and processing to enable early detection through scene context. VRSTNN utilises a CNN and GNN respectively to extract visual features from image input, fused with scene context from graph input that represents road user arrangement and relationships.
- Evaluation of the proposed model that outperforms the SOTA on a real and synthetic dataset. In addition to the introduction of a new metric to evaluate the ability to predict using partial sequences for early detection. This also allows us to evaluate the minimum history required, to be used as a design parameter to tune detection.

The rest of the paper is organised as follows: Section II summarises key related works, III describes the methodology, IV explains our experimental setup V presents the results. VI discusses our key findings, and VII draws conclusions.

II. Related Work

To improve hazard detection, diverse scene representation and learning models are crucial, yet underexplored. To this end, this section presents the key literature on various input encodings and the subsequent models to extract spatial and temporal features for prediction. Thus, we give an overview of the SOTA and present the limitations we aim to address.

A. Input Encoding

The foundation of any learning model begins with the input. In this section, we discuss how scene data is encoded as it provides the essential information needed to learn the patterns used for prediction. The most common type of raw scene data comes from onboard cameras that capture a video of the collision. Thus, the most common input encodings are image-based, as seen in SOTA works [5]–[7]. These input encodings are created by splitting videos into image frames and processed with a grayscale transform and resized to reduce computational complexity and improve generalisation by allowing models to learn features independent of colour channel or aspect ratio.

Image encodings capture the visual characteristics of a scene through a grid of pixels that encode colour and intensity. While they represent visual features, they don't encode how the pixels relate or interact. This is a drawback as understanding how road users interact, gives the essential context needed to detect hazardous events early. Therefore, models using image input have the difficult task of identifying entities of interest and inferring their relationships, as this is a type of relational feature extraction that CNNs are not optimised for [13]–[15].

As a result, graph encodings have emerged as a promising solution to better represent relational problems [24]–[26]. A graph G is formally defined as a pair of sets $G = (N, E)$, where N is a set of nodes, connected by a set of edges E [27], [28]. Since traffic flow is highly relational, it is rational to model a traffic scene as a relational graph structure. This structure helps decompose a complex driving scene into entities of interest that can be represented as nodes and linked by edge relationships to represent how they interact. Road users are commonly encoded as nodes and edge weights are used to represent qualitative spatial relations (e.g. right, left, near, far) [18], [23], [29]. To generate a graph, monocular camera images are used to detect and project actors to a bird's eye view (BEV) map, which is used to derive spatial relations.

In the works of [18], [23], actors (e.g. vehicle and pedestrian) and static traffic objects (e.g. lane markings and traffic signs) are detected in image frames and mapped into BEV to calculate spatial relationships. The authors then represent vehicle actors and lanes as graph nodes with qualitative relations stored as edges to denote distance categories (e.g., ”NearCollision” (1.2m), ”Near” (4.9m) and directional categories composed of longitudinal and lateral pairs (e.g., ”Front Left”, ”Rear Right”). Though semantically descriptive, the node and edge features remain limited to five distance categories and eight directional relations in [18], [23]. This approach is also adopted in other related works of [29], [30]. One drawback of this approach is that categorical encoding requires discretization which may incur data loss. Yet, few authors explored unprocessed encodings of actor state, such as [31], but the features remain limited to location and heading.

In [18], [23], the authors also mixed vehicles, road lanes, traffic signs and traffic lights as nodes in one graph to capture the interaction between different traffic elements. However, vehicles have different effects on the scene than static traffic objects and thus, heterogeneous graphs may be difficult to scale for large traffic networks as their complex structure not only increases computational cost but adds model complexity. Thus, other works separate graphs by node type [32] to allow models to differentiate the unique characteristics of each type.

Given the aforementioned gaps, our work makes two key contributions to input encoding. Firstly, with limited studies of different input encodings, we investigate six different encoding schemes to contrast visual and relational scene encoding techniques. Furthermore, we study encodings of varying complexity to investigate if abstraction can help guide learning.

Secondly, the current relational encodings utilise limited actor features and may introduce complexity using heterogeneous graphs. The use of actor state remains generally limited in the literature to ego vehicle position, speed, acceleration in [5] and categorical spatial relations in [18], [23], [29], [30]. Conversely, our paper proposes a novel relational encoding using 12 types of actor state and 8 spatial relation types based on leading kinematic indicators from the literature [16], [22]. Actor state and inter-actor relations are encoded in a bespoke relational graph to represent local and global context and encoded both quantitively and qualitative to study the complexity against data loss trade-off.

B. Prediction Models

In this section, we discuss how the input encodings are then used by SOTA prediction models to identify hazardous events. While the specific model may vary, the hazard detection problem is often framed as a time-series anomaly detection task, where input encodings are processed sequentially to identify anomalous/hazardous event patterns over time. Due to domain complexity and event variety, STNN models have been proposed using different spatial and temporal blocks to extract spatial features and their temporal patterns for classification.

Traditionally, CNN spatial blocks have been used to process image data, as they can effectively extract visual features and are based on receptive field theory. As a well-established method for image-based anomaly detection, such models have been covered extensively in the literature [33]–[35]. CNNs use convolutional layers that consist of learnable filter matrices that scan over the image and apply a transformation. The output results in a new set of features that can be fed to the next layer to learn hierarchical features. Thus, this enables complex patterns to be generalised with translation invariance, i.e., the ability to detect irrespective of position within the image.

To classify collision scenes and anomalies, influential works such as [5], [36] propose pairing a CNN spatial block with an LSTM temporal block and acts as the benchmark for this study. Implementations such as [36] utilised monocular image input and integrated the ConvLSTM with an AutoEncoder to detect anomalies by predicting what the future frame should look like to compare for anomalies in appearance or motion. However, continuous frame generation leads to a high computational load that may be impractical for real-time operation.

In contrast, [5] directly utilised a three-layer ConvLSTM to assist hierarchical feature extraction and multi-camera input. This work also concatenated the extracted visual features with ego vehicle (EV) telemetry of position, speed and acceleration. However, with limited EV telemetry and no data on other actor states, this extension only yields a 0.29% accuracy gain.

To overcome the limitations of CNN spatial blocks, researchers have proposed the use of graph theory and GNNs to target relational reasoning [37]–[39]. Using a graph, complex road scenes can be decomposed into actors as nodes and their relationships as edges, as discussed in II-A. The graphs are then processed by specialised neural networks called GNNs, which extract features by iteratively aggregating and updating node features based on neighbouring nodes using permutation invariant aggregation functions to create an encoding that is independent of node or edge order, i.e., permutation invariant. As such, the resulting encoding aims to generalise features applicable across similar road scenes. This process gives GNNs their key advantage in capturing collective behaviour that considers both the global structure and the local features of nodes and edges. This helps model complex interactions, which can be difficult to generalise using other networks.

Our study builds upon previous work, where the use of GNNs for hazard detection was reviewed in our survey paper [40]. For this current paper, we highlight the iterative works of [18], [23], where the authors classified hazardous lane-change videos using a relational graph convolutional network

(RGCN). The RGCN was selected to directly learn actor relations, such that hazardous actor interactivity can be better generalised by understanding how different graph nodes interact and the importance of different relations. After spatial processing, an LSTM was then used to extract temporal patterns, followed by a fully connected layer for final prediction. To represent the scene, actors and lanes were modelled as nodes with categorical spatial relations as edges to describe distance and relation e.g., near, behind, right and left.

For hazard classification, [23] first proposes a model that performs binary sequence classification. In addition, accuracy and interpretability were enhanced by adding attention-based graph pooling to the RGCN and temporal attention to isolate hazardous actors and time frames for visualisation. Binary sequence classification was then extended in [18] to frame classification. Thus, allowing early detection by updating the prediction at each frame [18] and demonstrates 39% earlier detection than CNN counterparts. However, graph features were limited to categorical spatial relations. Moreover, testing on real-world complexity was limited to training on 1043 self-generated synthetic scenes and 571 real-world videos.

Motivated by the above gaps, our work makes three key contributions to prediction models. Firstly, given single processing methods of visual CNNs and relational GNNs, we propose the integration of both modalities. We build upon works such as [5] where authors concatenated image features with ego vehicle telemetry data without dedicated processing, resulting in only 0.29% accuracy gain. Whereas, we propose a method that uses dedicated spatial processing for each input type to optimally capture the unique features in image and graph data.

Secondly, given the importance of early detection, we introduce a new metric to evaluate a model's ability to capture early warning patterns and predict using partial sequences where the collision is not visible. This allows us to analyse the minimum sequence history as a design parameter to tune detection.

Thirdly, we address the issues of overfitting to a small dataset and applicability to real-world scenarios, as previous works relied on small or synthetic datasets [5], [18], [23]. To overcome this, we use a large real-world dataset of 4775 scenes to improve generalisability to real-world complexity.

III. PROPOSED METHOD

This section begins with the system model and problem definition in III-A, followed by an overview III-B and a description of the proposed VRSTNN model in III-C - III-F.

A. System Model and Problem Definition

The goal of runtime hazard detection is to identify all potential scenarios that could lead to harm, i.e., hazardous events. In this study, we define collision events as hazardous events as harm is materialised. To maximise impact, we focus on collisions caused by road users as 83-94% of accidents have been linked to human fault [41], [42].

We consider a system that consists of an ADS with a camera sensor and sufficient local processing to handle the automated driving functions. The ADS receives scene data, such as raw image data from egocentric camera sensors and object detections from computer vision (CV) algorithms. In addition, we assume actor state data such as their speed and heading can be obtained through CV-based state estimation or vehicle-to-vehicle cooperative awareness messages (CAMs) [43]. These data are then structured into input encodings for the runtime hazard detection module as time series data points.

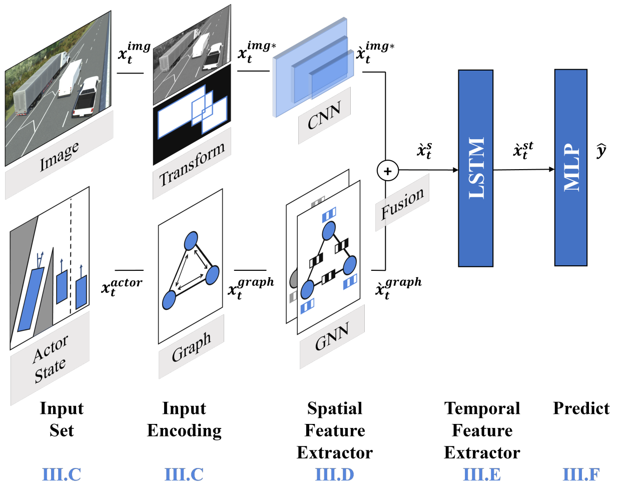


Fig. 2: Proposed VRSTNN pipeline: Multi-modal spatial features are extracted from image inputs via CNN for visual features and from actor state data via GNN for relational features. Fusion of both modalities is achieved by concatenating latent encodings and is processed by an LSTM for temporal pattern extraction and finally by an MLP for binary classification.

To this end, we formulate the hazardous event detection task as a time series classification problem. The intended solution is expected to classify a sequence of input encodings as hazardous $\hat{y}_t = 1$ or non-hazardous $\hat{y}_t = 0$ at each time step t, given n number of previous input encodings to assess the temporal history. We denote the encoded input at time step t as x_t. Therefore, the problem can be formulated as a mapping function, denoted by f in (1), which produces a binary output for a given sequence of encoded inputs:

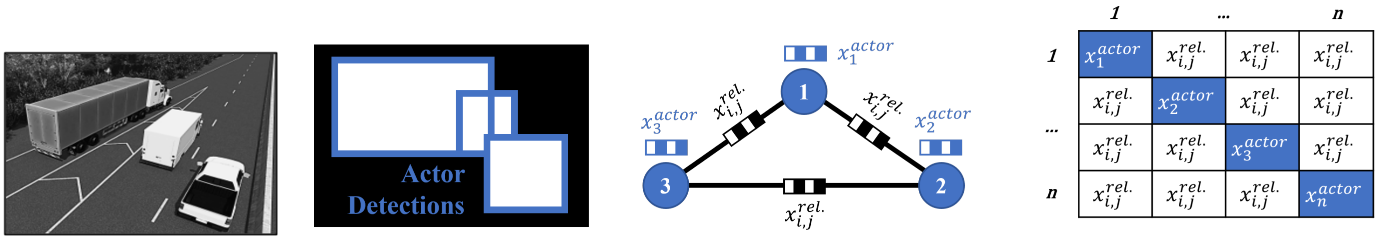
$$\mathbf{\hat{y}_t} = \mathbf{f}(\mathbf{x_0}..., \mathbf{x_{t-1}}, \mathbf{x_t}), \tag{1}$$

where $\hat{y}_t \in 0, 1$ is the prediction of a hazardous event at time step t and x_t is the input encoding at time t.

B. Method Overview

The processing pipeline of the proposed VRSTNN is illustrated in Fig. 2. The pipeline includes a multi-modal architecture to fuse visual scene features from image data with context from graph-encoded actor data describing road user arrangement, relationships and behaviour.

First, VRSTNN takes scene input in the form of raw image data from camera sensors and actor state data. To process the image data, a transformation is applied to improve generalisation and computational efficiency of the model. Concurrently, actor state data are encoded into a relational graph structure. The separate image and graph encodings are then passed to their dedicated spatial feature extractors: a CNN for images



	1	...		n
1	x_1^{actor}	$x_{i,j}^{rel.}$	$x_{i,j}^{rel.}$	$x_{i,j}^{rel.}$
	$x_{i,j}^{rel.}$	x_2^{actor}	$x_{i,j}^{rel.}$	$x_{i,j}^{rel.}$
...	$x_{i,j}^{rel.}$	$x_{i,j}^{rel.}$	x_3^{actor}	$x_{i,j}^{rel.}$
n	$x_{i,j}^{rel.}$	$x_{i,j}^{rel.}$	$x_{i,j}^{rel.}$	x_n^{actor}

(a) Visual Grayscale (b) Visual Bounding Box (c) Our Relational Graph Structure (d) Our Relational Adjacency Matrix

Fig. 3: Visual and relational input encodings used in this study to examine their capacity to generalise hazardous event patterns

and a GNN for graphs. The extracted spatial features are then fused and passed to an LSTM network to extract temporal patterns. Finally, a multi-layer perceptron (MLP) outputs a binary prediction to classify the scene as safe or hazardous.

Each step is described in detail in the following sections as follows: input encoding III-C, spatial processing III-D, temporal processing III-E and finally the output layer III-F.

C. Input Encoding

We investigated different visual and relational encoding schemes, as shown in Fig. 3. Among visual encodings, we contrast detailed input with an abstraction to guide learning, Fig. 3a, 3b. Then among relational encodings, we examine different graph and matrix encodings of actor state to describe how actors behave and interact in a scene, Fig. 3c, 3d.

1) Visual Encoding: We take raw RBG image input x_t^{img} of variable resolution and standardise it to 256x256 pixels through bilinear interpolation to minimise data loss and allow for efficient training. We study two types of image inputs. First, a conventional grayscale transform to represent visually rich input independent of colour channel, Fig. 3a. However, raw camera data can be noisy and complex, thus we compare this to a simplified binary map showing only actor bounding box (BBox) outlines, Fig. 3b. The BBox encoding encodes the background as 0 (black) and actors as 1 (white) to reduce noise and complexity by creating an abstract representation that guides learning towards actor interactions. The resulting encoding is denoted x_t^{img*} and is shown in Fig. 2.

2) Relational Encoding: We propose a novel graph encoding of scene data to provide a framework that represents the relationships between road users and information on each user, shown in Fig. 3c. The nodes encode the actors in a scene and their relationships as edges. For each actor, we encode state data based on key kinematic indicators that precede accidents [16], [22] and connect each actor using an edge to create a fully connected (FC) graph that traces their interactions.

TABLE I: ACTOR FEATURES

Type	Quantitative	Qualitative
Actor	speed, acceleration, heading, yaw_rate, pitch_rate, roll_rate	>avg_vel, >avg_acceleration, >avg_heading, >avg_yaw_rate, >avg_pitch_rate, >avg_roll_rate
Relations	time_to_collision, eucli_distance, eucli_longitudinal, eucli_lateral	time_to_col: (<1s, 1s, 2s, +4s), eucli_rel: (1m, 0.5$c*$, 1$c*$, +1$c*$), long_rel: (to_side, ahead, behind), lateral_rel: (same_lane, left, right)

c*: car lengths

In this study, a FC graph is used to model the complex interactions among road users as the influence of one road user on another is not known. This approach allows us to input all connections and utilise the GNN to learn the significance of each connection through training, as a popular approach when the node number is small [44]. This is motivated by similar works in trajectory planning [45]–[47], where a FC graph allows the GNN to discern significant connections among road users by learning from all potential interactions. We note the FC representation is a first step to build a platform for future work, as the effect of road users on others is not yet quantified.

To encode each graph x_t^{graph} at time t, we encode each node i with actor state data x_i^{actor} and encode each edge with relational features $x_{i,j}^{rel.}$ for each node i and neighbour j. For the relational features, we convert the quantitative values into categorical values and ranges which are each given a unique index e.g., {0: ahead, 1: behind, 2: left, 3: right}. We tabulate the features in Table I and study a quantitative (i.e., numerical) and simpler qualitative encoding (i.e., categorical) to study the complexity against data loss trade-off, see Table I.

The numerical features were then scaled using min-max normalisation to reduce the effect of varying magnitudes and units. Our qualitative actor features represent a novel discretization that compares the numerical average of the surrounding actors. The encoding is inspired by focal attention, where humans focus on abnormal behaviour. Similarly, our qualitative features represent a categorical discretization of the numerical equivalents, to simplify out noise and complexity.

Secondly, we propose a novel matrix equivalent using an adjacency matrix to test SOTA CNN's ability to learn relational features and perform non-visual numerical analysis. The rows and columns index n number of actors, such that matrix elements represent actor relations $x_{rel.}$ and the leading diagonal represents actor kinematics x_{actor}, shown in Fig. 3d.

D. Spatial Feature Extractor

After the inputs are processed, they are fed to their respective CNN and GNN spatial feature extractors.

1) Convolutional Neural Network: To process image input and extract the visual scene features, we utilise a SOTA CNN-based hazardous event classifier [5] as a suitable benchmark used by related works [18], [23]. The model consists of a 3-layer convolutional and recurrent LSTM network.

To allow feature fusion in our VRSTNN model, We isolate the CNN spatial block that takes as input, the processed images x_t^{img*} and extract the visual features to create an intermediate representation $\hat{x}_t^{img}$. The CNN block consists of 3 convolutional layers stacked sequentially. This stacking allows the initial layers to extract low-level patterns such as edges, which are refined by later layers into higher-order representations such as shapes. This is critical for reliable hazard detection as it enables CNNs to learn generalised patterns for use in diverse and unpredictable road conditions.

In the forward pass of the CNN spatial block, the 2D convolution plays a key role in feature extraction. This operation calculates the output feature map F from an 2D input matrix I using a convolution kernel K and is formulated in (2):

$$F[i,j] = (I * K)[i,j] = \sum_m \sum_n I[i-m, j-n] \cdot K[m,n] \quad (2)$$

where:

$F[i,j]$: value of the output feature map F at position i, j.
$I * K$: convolution operation $*$ between input matrix I and kernel K.
$\sum_m \sum_n$: double summation over indices m and n, representing the summing operation across the rows and columns of the convolution kernel K and the corresponding region of the input matrix I.
$I[...]$: value in the input matrix I at the adjusted position, correlating with the kernel's position at $i-m, j-n$.
$K[m,n]$: value at position $[m,n]$ in the convolution kernel.

2) Graph Neural Network: To process graph input x_t^{graph}, we utilise a GNN to extract scene context from road user relations and state information. The GNN is based on the RGCN from the seminal work of [48] due to its effectiveness in graph classification using multi-relational data [48], [49]. This aligns with our task of hazard detection by graph classification, where the nodes represent road users with multi-relational edges that depict the relationships in Table I.

Our GNN is composed of two layers stacked sequentially to progressively refine the extracted features. The forward pass to calculate a hidden representation for each node i using neighbouring node j is formulated in (3) [48]:

$$\mathbf{h_i^{l+1}} = \sigma\left(\sum_{\mathbf{r}\in\mathbf{R}} \sum_{\mathbf{j}\in\mathcal{N}_\mathbf{i}^\mathbf{r}} \frac{\mathbf{1}}{\mathbf{c_{i,r}}} \mathbf{W_r^{(l)} h_j^{(l)}} + \mathbf{W_0^{(l)} h_i^{(l)}}\right), \quad (3)$$

where:

h_i^{l+1} : hidden representation h of node i in layer $l+1$
σ : non-linear activation function
$\sum_{r\in R}$: sum over all relation types r, r is a set of relations R
$\sum_{j\in\mathcal{N}_i^r}$: sum over all neighbour nodes j, j is a set of nodes $\mathcal{N}$ that neighbour the node i with relation type r
$\frac{1}{c_{i,r}}$: normalisation constant c for node i and relation r
$W_r^{(l)}$: weight matrix W_r shared by all neighbour nodes for layer l and relation type r
$h_j^{(l)}$: hidden representation of neighbour node j in layer l.
$W_0^{(l)}$: weight matrix W_0 for layer l, to learn individual importance of central node i
$h_i^{(l)}$: hidden representation of node i at previous layer l

The hidden representation h_i for each target node i is computed from information aggregated from its neighbours and from node i itself. In our case, the forward pass will calculate a representation h_i for each vehicle node i that represents information on nearby vehicles and vehicle i itself.

To calculate neighbouring node information, the network aggregates information from all neighbour nodes j with the same relation type r, where r represents a categorical index given to each qualitative relation e.g., {0: ahead, 1: behind, 2: left, 3: right}, see Table I. When aggregating neighbouring node information, the features are transformed using the relation-specific weight matrix W_r and normalized by a constant $\frac{1}{c_{i,r}}$ based on the number of neighbours. To exemplify the role of W_r, we consider nearby vehicle j and its relation r to target vehicle i. If relation r tells us that the nearby vehicle is behind the target vehicle, a dedicated weight matrix W_r for each relation r, is used to transform the features of the nearby vehicle. This adjustment is crucial because, e.g., if a vehicle is behind another, its acceleration may be weighted higher due to the increased risk of collision. Thus, W_r helps the network learn which features from nearby road users matter and in what contexts (over which relation r). After transforming the features of vehicle j, these features are aggregated using mean operation and this aggregation is performed over all neighbour nodes $\mathcal{N}$, expressed in $\sum_{j\in\mathcal{N}_i^r}$ and over all relation types R, expressed in $\sum_{r\in R}$, shown in Equation (3)

To calculate the features from node i, the network transforms the node's current features h_i using a weight matrix $W_0^{(l)}$ that learns which node features are most important for prediction. For example, the encoded yaw rate may signal an evasive manoeuvre and may be weighted higher.

Using both W_r and W_0, the network balances both local node and global neighbourhood information to combine how individual and collective behaviours influence prediction. The resulting aggregated neighbour and node i features are then passed through a non-linear activation function σ.

The resulting representation of the RGCN forward pass is then passed to a global pooling layer to better generalise collective behaviour that is more permutation invariant across similar unseen scenarios. Pooling is applied by summation across the node feature dimension to create an intermediate representation of the graph $\hat{x}_t^{graph}$, as formulated in (4).

$$\hat{\mathbf{x}}_\mathbf{t}^\mathbf{graph} = \sum_{\mathbf{i}\in\mathbf{N}} \mathbf{h_i}, \quad (4)$$

where:

$\hat{x}_t^{graph}$: intermediate representation of graph at time step t
$\sum_{i\in N}$: sum over nodes i, i is a set of all nodes N
h_i : hidden representation of node i

3) Spatial Feature Fusion: Fusion is then implemented to enrich learning with the extracted features from heterogeneous input and processing. With intermediate representation from both CNN $\grave{x}_t^{img}$ and GNN $\grave{x}_t^{graph}$ networks, a fused spatial encoding $\grave{x}_t^s$ is created at each time step t, by concatenating along the feature dimension to preserve the ordinal sequence.

Fusion is performed after feature extraction due to the heterogeneous image and graph inputs with different structures and semantically different scene encodings. This preserves input integrity and allows dedicated spatial processing.

E. Temporal Feature Extractor

For the temporal block shown in (5), an LSTM is used to encode the spatio-temporal encoding $\grave{x}_t^{st}$ at each time step. This study utilises the popular LSTM network for its ability to capture long-term dependencies, as seen in SOTA works in the area [5], [18], [23], [36]. The LSTM was selected for its ability to learn over long sequences due to feedback connections that selectively retain or forget information, as needed to capture complex event evolution of various types and sequence lengths. The LSTM takes the fused spatial encoding $\grave{x}_{t-1}^s$ and hidden state h_{t-1} at the previous time step $t-1$, and updates its current cell state c_t and hidden state h_t at each time step.

$$\mathbf{h_t}, \mathbf{c_t} = \mathbf{LSTM}(\mathbf{h_{t-1}}, \grave{\mathbf{x}}^{\mathbf{s}}_{\mathbf{t-1}}). \tag{5}$$

F. Output Prediction Layer

The spatio-temporal encoding $\grave{x}_{t+1}^{st}$ at each time step is then fed to a 3-layer MLP, to output the confidence values for each class to form the final prediction $\hat{\mathbf{y}}$ = 0: safe, 1: hazardous. The predicted confidence values $\hat{Y}$ from batch training are then compared with the ground-truth labels Y to calculate cross entropy loss and backpropagated. Once trained the model predicts the class from the highest confidence score.

IV. Experimental Setup

This section presents the experimental setup regarding dataset preparation, model configuration and metrics used to evaluate the performance of the proposed method.

A. Dataset Description and Preparation

Annotated datasets were used to train and test our model and were split 80% for training and 20% for testing. In this study, we define collision scenes as hazardous as the harm is materialised. To this end, we prepare two datasets: first a high-volume synthesised dataset with actor state and second from real-world scenes to capture real-world complexity. The datasets include scenes that vary across day and night and include diverse road environments such as urban and rural.

1) Synthesised Dataset: As real-world datasets lack actor state, researchers utilise simulation environments such as CARLA [50]; however it lacks environmental diversity and realism as actors require manual waypoints and do not consistently respond to EV actions. Therefore, the GTA Crash (GTAC) dataset [51] was used due to its realistic and randomizable actor models and scenarios that are popular for testing end-to-end control and computer vision [52]–[54].

We utilise 11298 GTAC scenes, with a 1:2.1 safe-to-hazardous scene ratio that we upsample to a 1:1 ratio to avoid bias. Scenes were filtered to contain a maximum of 20 unique actors per scene to capture 99.3% of the dataset while creating a standardised graph size that allows each actor to be assigned a unique node that retains actor history and allows interactions to be identified from the graph topology.

Scenes include a diverse range of environments and scenarios such as hazardous lane change, overtake and sudden braking, which reflect the most common crash events [41]. Each scene consists of 20 frames captured at 10 frames per second (FPS) and provides actor data (e.g., position, speed) for the ego vehicle and surrounding actors, with ego-centric camera data with actor bounding boxes, full details in [51].

To generate the BBox image encoding, the included actor bounding box annotations were utilised following the method described in III-C1. To generate the relational graph encoding, the included actor state data was encoded at the nodes and relationships (euclidean distance, longitudinal, lateral and time-to-collision) were calculated using actor position, heading and speed. The equivalent qualitative expressions were then derived to discretize absolute values into categories (e.g. in front, left, right) and boolean values comparing actor kinematics to the average from surrounding road users, as inspired by human focal attention that focuses on actors behaving unusually.

2) Real-World Dataset: A real-world dataset was also compiled to show generalisability for real domain complexity. The recent Detection of Traffic Anomaly (DOTA) dataset [55] is used as the largest public dataset of real collision scenes at the time of publication, with high resolution and detailed annotations across real-world dashcam videos. To provide scenes for the safe class, scenes were sampled from Berkeley Deep Drive (BDD) dataset [56]. The dataset was prepared following the steps in III-C1, resulting in 4775 videos with a 0.84:1 safe-to-hazardous scene ratio, which was upsampled. To match the synthesised dataset, the scenes were also filtered to a maximum of 20 unique actors per scene for consistency with GTAC. Each scene consists of 50 frames, at 10 FPS, with diverse locations and environments across day and night.

To generate the BBox image encoding, actor bounding boxes were generated using SOTA CV algorithms: Yolov8 [57] object detection and StrongSort tracking [58].

To generate the relational graph encoding, actor state was not available and thus, the qualitative actor relations such as distance were derived by comparing the spatial positions of actor-bounding box centroids to provide a rough estimation. Though, we are aware of the limitations of utilising pixel space and only use it as a pragmatic first approach, as seen in similar literature [17]. This approach was adopted as the study's focus is to demonstrate the viability of our proposed method on a real dataset and not on computer vision-based localization.

Moreover, due to the dangerous nature of collision events, the dataset collects scenes from publicly available sources and thus, the scenes vary in quality, camera setup and software artefacts like telemetry overlay. Therefore, to avoid the detector utilising software artefacts to differentiate hazardous scenes, we mask the artefacts by padding the top and bottom 300 pixels in both classes, in line with artefact sizes.

TABLE II: MODEL PERFORMANCE

Dataset	Model	Input Features				Accuracy[3]		F1[3]		FNR[3]		FPR[3]	
		Actor Qual.	Actor Quan.	Img. BBox	Img. Gray	Scene 100%	Scene 50%	Scene 100%	Scene 50%	Scene 100%	Scene 50%	Scene 100%	Scene 50%
GTAC **11298**	CNN-LSTM	x				0.627	0.500	0.667	0.000	0.253	1.000	0.492	0.000
			x			0.500	0.500	0.000	0.000	1.000	1.000	0.000	0.000
				x		0.758	0.573	0.797	0.599	0.049	0.363	0.434	0.490
					x	0.648	0.500	0.615	0.000	0.439	1.000	0.266	0.000
	GNN-LSTM	x				0.877	0.814	0.881	0.819	0.086	0.159	**0.161**	**0.212**
			x			0.744	0.743	0.782	0.748	0.080	0.238	0.431	0.276
	FUSION-LSTM	x_1		x_2		**0.881**	**0.821**	**0.889**	**0.834**	**0.042**	**0.099**	0.196	0.259
		x_1			x_2	0.864	0.812	0.872	0.826	0.067	0.106	0.206	0.269
DOTA **4775**	CNN-LSTM	x				0.664	0.500	0.701	0.667	0.212	0.000	0.459	1.000
				x		0.716	0.500	0.747	0.667	0.161	0.000	0.408	1.000
					x	**0.809**	0.500	**0.821**	0.000	**0.128**	1.000	**0.253**	0.000
	GNN-LSTM	x				0.681	0.656	0.732	0.702	0.129	0.191	0.508	0.496
	FUSION-LSTM	x_1		x_2		0.732	0.681	0.730	0.665	0.274	0.367	0.263	**0.270**
		x_1			x_2	0.767	**0.730**	0.782	**0.755**	0.164	**0.168**	0.301	0.373

[1] Features processed with a GNN network

[2] Features processed with a CNN network

[3] We evaluate the ability to correctly predict initiating patterns for early detection by varying the number of frames removed from the end of collision Frame prediction evaluated at the last timestep when given i) 100% of the sequence ii) first 50%

B. Experimental Settings

Training was performed on an Intel i9-10980XE CPU, 128 GB RAM and Nvidia RTX 3090 GPU with 24 GB VRAM. Models were written with PyTorch, with deterministic model behaviour enabled to aid reproducibility. In addition, random seeds were set to 1 to reduce random variability and the model hyperparameters were found using a systematic grid search.

The models were trained using stochastic gradient descent with a maximum of 100 epochs with early stopping after a loss plateau of 20 epochs. Trained with a batch size of 8, a learning rate (lr) of $1x10^-4$ and a learning rate scheduler to reduce lr by a factor of 0.5 after a loss stagnation over 10 epochs. Moreover, to avoid the classifier developing a systemic bias from class imbalance, the minority class was randomly upsampled to match the majority class.

For visual learning, we utilise the publicly available architecture and hyperparameters from the SOTA implementation of [5]. Each of the 3 ConvLSTM layers has a hidden dimension of 8, spatial dropout of 0.75 and a final MLP with output 2. We enhanced the original input image from 64x64 to 256x256 to avoid excessive downscaling and used bilinear interpolation to avoid significantly affecting perceptual character.

For relational learning, a 2-layer RGCN model was used, with a hidden dimension of 256 and a pooling layer with ratio 1/20, equivalent to total nodes per graph. This is followed by an LSTM with temporal dropout of 0.25 and MLP of output 2. For fair comparison, relational feature learning using the SOTA CNN model was also tested with an equivalent relational Euclidean encoding as discussed in III-C2.

The fused VRSTNN model utilised the spatial layers from the ConvLSTM and RGCN to process each input modality sequentially before concatenating each latent representation and passing it to the LSTM layer and MLP with output 2. For convenience, the networks will be referred to as CNN-LSTM, GNN-LSTM and Fusion-LSTM respectively.

C. Model Evaluation Metrics

Hazardous event detection is critical for safety, thus it is essential to detect correctly and quickly. As such we evaluate using accuracy (ACC) and prioritise false negative rate (FNR) to highlight missed detections. To minimise false alarms, we also utilise the false positive rate (FPR). Furthermore, we evaluate the F1 score to represent the harmonic mean of precision and recall that ranges between 0 and 1, where 1 signifies a perfect ability to correctly detect hazardous events (precision) and detect all hazardous instances (recall).

Moreover, we introduce a new metric to evaluate the ability to predict early warning patterns using partial sequences. We start by evaluating the prediction at the last timestep given the entire sequence ($@scene_{100\%}$). We then remove the end frames where collision is visible and evaluate the prediction given the first 25% ($@scene_{25\%}$) and 50% ($@scene_{50\%}$) of the sequence for comparison. This allows us to evaluate the minimum history required for early detection, to be used as a design parameter to optimise the FNR and FPR trade-off.

Lastly, we review response time (RT), to measure how quickly the system identifies a hazardous event after the first signs of initiation, as provided by the datasets. RT is crucial as it impacts the ability to take mitigating action. For realistic RT, we report the first of three consecutive positive detections to simulate a warning at deployment that filters out transient anomalies by ensuring temporal consistency. As such, we report the average value and standard deviation to give a general overview. In addition, we provide the 90th percentile to represent the majority of cases where 90% of the values fall and the 99th percentile to represent the worst-case scenarios.

V. RESULTS

In this study, we evaluate the characteristics of visual and relational input encodings and spatial feature extractors for binary frame classification of collision scenes. We present our fused VRSTNN model and compare it against SOTA GNN-LSTMs and CNN-LSTMs. We present model performance in Table II given full and partial sequences to discuss results on input encoding, inference time, response time, transfer learning from simulation to real-world and present an ablation study. We also note that model comparisons are made in terms of absolute percentage difference and not relative difference so that readers can easily retrace the given calculations.

A. Full Scene Prediction

Given the full sequence $@scene_{100\%}$, our proposed fused model performed best on GTAC with 88.1% ACC, 88.9% F1 and 4.2% FNR. It outperforms the best CNN by 9.2% and GNN by 0.7% in F1. On DOTA, fusion achieved 76.7% ACC, 78.2% F1 and 16.4% FNR $@scene_{100\%}$, with an expected drop in the real-world dataset with mixed image quality, higher scene complexity and event variety. Moreover, DOTA lacked actor state information, which had to be estimated with computer vision, as described in III-C1. Thus, the CNN networks leveraged their advantage on DOTA and achieved the highest performance $@scene_{100\%}$, but drastically dropped in early detection. We also stipulate the visual networks were able to utilise extraneous image artefacts, as discussed in IV-A2. This is supported by the 9.4% ACC drop when trained on visual BBox input that limited extrinsic artefacts.

Of the individual models, the relational GNNs utilising qualitative features performed better than CNNs on GTAC given the entire sequence, with 88.1% F1, 8.6% FNR and outperformed the best CNN by 8.4% in F1. For fair input comparison, the CNN was trained on the same qualitative actor features but saw 21.4% lower F1 and 16.8% FNR. This is similarly reflected on DOTA, indicating CNN's preference for visual extraction over numerical analysis. This saw CNNs perform better on DOTA due to their advantage of only requiring image input, as opposed to GNNs reliance on accurate actor state which is not yet available in real hazard-focused datasets.

B. Early Detection

As shown in Table III, fusion particularly enhances early detection stability at both $@scene_{50\%}$ and $@scene_{25\%}$, showing the best performance across datasets. In contrast, the best CNN model saw -19.9% F1 drop on GTAC and -82.1% on DOTA $@scene_{50\%}$. Whereas, the fused network reduced F1 drop 3.6-fold to -5.5% on GTAC and 29.9-fold to -2.7% on DOTA. Compared to the best CNN models, fusion improved F1 drop by an average of 16.7-fold $@scene_{50\%}$ and 22.6-fold $scene_{25\%}$ across datasets.

Of the individual models, superior GNN performance becomes increasingly prominent. GNN maintains good stability across datasets, exhibiting a F1 drop $\leq$ 6.2% $@scene_{50\%}$ and $\leq$ 10.1% $@scene_{25\%}$. While almost all CNN models were unable to predict when the collision frames were removed and predicted all scenes as safe resulting in a FNR of 1 and F1 score of 0 or all scenes as hazardous, resulting in a FNR of 0 and F1 of 0.667. Moreover, this behaviour can also be evidenced by the near 0.5 ACC score on the balanced datasets.

C. Input Encoding

Regarding input encodings, we demonstrate synergy from multi-modal inputs that improve the detection of early warning patterns across datasets. In general, given accurate actor state data, the graph encoding performed better than visual encodings on GTAC but lacked this information on DOTA.

In addition, input abstraction to guide learning enhanced models with the simplified qualitative graph encoding of actor features enhancing F1 by 9.9% over the quantitative encoding. Similarly, the preference for abstraction is mirrored in the CNNs, with the visual BBox encoding achieving 18.3% higher F1 over grayscale images on GTAC. However, grayscale was 7.4% higher on DOTA as the bounding boxes needed to be estimated and highlight the importance of accurate actor data.

Regarding the encoding of actor state data, graph encodings were better than matrix encodings as graph structure provided a better encoding for actor features and relationships. This is shown as the CNN using the same qualitative actor state data exhibited 21.4% lower F1 when processed with the CNN on GTAC and 3.1% lower on DOTA, compared to the GNNs.

In addition, the GNN model was able to learn with the quantitative actor features and exhibited greater early prediction stability, with a maximum F1 drop of -5.9%, whereas the qualitative encoding showed almost double the drop of -10.1% across datasets. Regarding processing, CNNs struggled to learn with quantitative actor features with an F1 of 0, further indicating CNN's inclination towards visual encoding.

D. Real-Time Operation and Scalability

In real-time applications, at least 10 FPS is required for computer vision systems in ADS applications [59]. To assess performance, timing was measured per scene, simulating real-world continuous processing over extended sequences without interruptions. This mitigates inaccuracies and overhead due to timing functions and data transfer. Since the frames per scene are known, we calculate the average time per frame and FPS, as shown in Table IV. All models exceed the minimum 10 FPS requirement [59] and range between 277 to 3327 FPS.

GNNs were the most efficient and at least x2 faster on GTAC and 3.37x on DOTA. In contrast, the visual CNNs

TABLE III: EARLY DETECTION

Dataset	Model	Model Configuration			ACC[3]	F1[3]	FNR[3]
		Actor Qual.	Img. BBox	Img. Gray	Scene 25%	Scene 25%	Scene 25%
GTAC 11298	CNN		x		0.499	0.004	0.998
	GNN	x			0.773	0.780	0.194
	FUS	x_1	x_2		**0.784**	**0.802**	**0.126**
DOTA 4775	CNN			x	0.500	0.000	1.000
	GNN	x			0.648	0.695	0.199
	FUS	x_1		x_2	**0.734**	**0.760**	**0.158**

Features processed with a: [1]GNN network, [2]CNN network
[3]Prediction at the last timestep given the first 25% of sequence frames

TABLE IV: MODEL INFERENCE TIME

Dataset	Model	Model Configuration		Time[3] (ms)	FPS[4]	No. Param.
		Actor Data	Img. Data			
GTAC 11298	CNN		x	0.67	1498	1.68E+07
	GNN	x		**0.30**	**3327**	**4.36E+05**
	FUS	x_1	x_2	1.60	625	3.68E+08
DOTA 4775	CNN		x	0.76	529	1.68E+07
	GNN	x		**0.21**	**1885**	**4.36E+05**
	FUS	x_1	x_2	1.44	277	3.68E+08

Features processed with a: [1]GNN network, [2]CNN network
[3]Average time to process one frame, [4]Average frames per second

exhibited slower inference due to their larger input sizes and computationally expensive matrix multiplications from multiple convolutional layers. Conversely, graph encodings are compact and enable efficient processing using highly parallelisable GNN message-passing algorithms.

Our fused VRSTNN model surpassed the minimum FPS requirement at 625 FPS and was positioned as the third most efficient after the CNN and GNN-based models. This is expected as the VRSTNN model fuses the spatial feature extracted from both CNN and GNN models to leverage the advantages of both architectures. This synergy bolsters performance, justifying a reasonable trade-off in inference speed due to a larger intermediate representation at each step. However, future optimisation is possible by parallelising spatial blocks.

Regarding scalability, the models are designed to process entire video clips of up to 5 seconds for hazard detection. In deployment, the model will utilize a fixed sliding window corresponding to the video lengths tested, ensuring practical real-time performance. This sliding window method ensures scalability to continuous data streams. To further improve real-time performance, trends in computational power are also predicted to continue improving, as driven by the increase in AI development and adoption. Recent studies have found that the computing power for training AI models has been doubling every 10 months from 2015 to 2022 [60] and are expected to continue with AI-specialized hardware and more efficient algorithms that will only speed up inference times.

E. Response Time

It is important for models to predict both correctly and quickly for mitigating to be taken. Across datasets, the fusion models had the best response time across metrics. In particular, the fusion model with features least reliant on accurate actor data: grayscale image and qualitative actor features.

Table V tabulates RT and in general, the average RTs across models and datasets range from 0.21-0.57s and the majority of cases seen in the 90th percentile range from 0.2-1.6s, in line with human driver performance of 1.1-1.8s [19]–[21]. The only models to outperform humans across datasets were the leading fusion model and GNNs. However, as a safety-critical task, it is also important to evaluate the 99th percentile for the worst-case scenarios. Given accurate actor data on GTAC, the 99th percentile shows human-level performance from the GNN and fusion models but on DOTA, we see models fall short with a range of 3.4-4.4s. Thus, the results indicate a need to further improve detection given increased scene complexity and primarily image-only real-world datasets.

TABLE V: RESPONSE TIME (RT)

Dataset	**Model Configuration**			**AVG (s)**	**STD (s)**	**90TH (s)**	**99TH (s)**
	Actor Qual.	Img. BBox	Img. Gray				
GTAC 11298	x_2			0.283	0.380	0.8	1.8
		x_2		0.294	0.372	0.8	1.7
			x_2	0.571	0.603	1.6	1.9
	x_1			0.250	0.362	0.7	1.8
	x_1	x_2		0.214	0.313	0.4	1.7
	x_1		x_2	**0.210**	**0.295**	**0.4**	**1.6**
DOTA 4775	x_2			0.400	0.875	0.7	4.4
		x_2		0.438	0.845	1.3	3.7
			x_2	0.306	0.769	0.4	4.3
	x_1			0.373	0.879	0.7	4.2
	x_1	x_2		0.643	0.879	1.7	4.0
	x_1		x_2	**0.245**	**0.568**	**0.2**	**3.4**

Features processed with a: [1]GNN network, [2]CNN network

TABLE VI: TRANSFER LEARNING

Model	**Model Configuration**			**ACC**[1]		**F1**[1]	
	Actor Qual.	Img. BBox	Img. Gray	Base	Transfer	Base	Transfer
CNN	x			0.664	**0.665**	0.701	**0.711**
		x		0.716	**0.744**	0.747	**0.753**
			x	0.809	**0.847**	0.821	**0.839**
GNN	x			0.681	**0.687**	0.732	**0.736**
FUS	x_1	x_2		0.732	**0.776**	0.730	**0.746**
FUS	x_1		x_2	**0.767**	0.707	**0.782**	0.712

Features processed with a: [1]GNN network, [2]CNN network
[3]Pre-trained model on GTAC transferred on a real-world DOTA dataset

F. Transfer Learning

Starting with a pre-trained model on the simulation-based GTAC, we show fine-tuning on a real-world DOTA dataset in Table VI. We show up to 4.4% enhanced ACC and 1.8% F1 across leading models. The grayscale encoding enhanced learning the most, indicating good generalisability of such input as it did not rely on accurate actor data. The only model not showing enhancement was the fused model utilising qualitative actor features and grayscale images, indicating a more advanced fusion technique may be necessary to generalise features between datasets.

G. Ablation Study

Through fusion, we give networks a more comprehensive scene input to better generalise hazardous events. Table VII shows all fused model permutations, that led to the selection of the optimal VRSTNN configuration. Qualitative actor features enhanced performance across datasets and image encodings varied depending on the availability of accurate actor data on GTAC. In addition, the fusion of both types of actor data saw improvements on base quantitative actor models but led to longer inference times and marginal or lower performance than the qualitative model as we stipulate the latent representation becomes too large and introduces complexity and noise.

TABLE VII: ABLATION STUDY

Dataset	**Model Configuration**				**ACC**	**F1**	**FNR**
	Actor Qual.	**Actor Quan.**	**Img. BBox**	**Img. Gray**	Scene 100%	Scene 100%	Scene 100%
GTAC 11298	x_1		x_2		**0.881**	**0.889**	0.042
	x_1			x_2	0.864	0.872	0.067
		x_1	x_2		0.863	0.858	0.171
		x_1		x_2	0.784	0.812	0.070
	x_1	x_1	x_2		0.876	0.886	0.038
	x_1	x_1		x_2	0.818	0.843	**0.027**
DOTA 4775	x_1		x_2		0.732	0.730	0.274
	x_1			x_2	**0.767**	**0.782**	**0.164**

Features processed with a: [1]GNN network, [2]CNN network

VI. DISCUSSION

Key Contribution: This study proposes a multi-modal hazard detection model VRSTNN, as an independent warning system to enhance the safe operation of ADSs. Applied as a warning system to enable a safe handover in L3 and below, and a future trigger for minimum-risk manoeuvres above L3.

To achieve this, this, our model tackles the challenge of early and reliable hazard detection to enable the safe deployment of L3+ ADS. Our results highlight the critical role of scene context for early detection and we demonstrate how SOTA CNN accuracy drops to ∼50% once the frames showing the collision are removed. This may indicate an over reliance on prominent visual cues, e.g., vehicles directly in contact and a lower proficiency to capture early warning patterns.

To address these limitations, we propose a novel multi-modal VRSTNN model to fuse: 1) visual scene features from image data with 2) scene context from graph-encoded actor data that describes the arrangement, relationships and behaviours of road users. Our results demonstrate that VRSTNN outperforms SOTA models and matches human RT. VRSTNN achieves 88.1% ACC, 4.2% FNR, 0.21s average RT in synthetic GTAC with accurate state data. Even in the real-world DOTA dataset which lacks state data, VRSTNN achieves 76.7% ACC, 16.4% FNR, 0.25s average RT and predicts up to 3.75s before collision with an F1 of 76%.

Learning early warning patterns requires scene context and our VRSTNN achieves this through dedicated GNN processing that utilises our graph encodings of scene data to learn the global graph structure that describes road user arrangement and the local interactions from actor relationships. Thus, our VRSTNN fuses this scene context with visual features to interpret complex scene data and learn early warning patterns.

The VRSTNN architecture was built upon our findings studying different input encodings and feature extractors to understand what features and processing are necessary to identify hazardous event patterns. From our results on input encoding, we found that the model using accurate actor state data in GTAC outperformed the models using only visual data across all evaluation metrics. In addition, we found that unprocessed scene data can overload the model with complexity and noise, whereas simplified input encodings can guide learning. For example, the qualitative encoding of actor state showed 9.9% higher F1 and similarly, image BBox encoding showed 18.3% higher F1 over grayscale on GTAC.

To process the input encodings, our results showed that relational GNNs outperform visual CNNs in synthetic GTAC across all evaluation metrics, given accurate actor data. However, without state data in the real-world DOTA, GNN only outperforms in early detection and this reliance on actor state signifies a current limitation. To avoid input bias, we also processed actor state data using CNNs but showed poor performance which further indicates CNN's preference for visual analysis that remains invaluable for real-world datasets that lack actor state. Therefore, given the advantages and limitation of both networks, we propose the fused VRSTNN to leverage the advantages from both networks and represents a new multi-modal learning approach.

Comparison to Literature: Our findings are supported by other works [5], [61], which have highlighted the need for heterogeneous processing and multi-modal inputs for better scene understanding and more reliable detection. However, the optimal input encoding remains undefined in literature [62], [63] and thus, we expand early works that utilise ego position and speed [5]. In our work, we capture all scene actors with 12 telemetry and 8 relational encodings each, to learn advanced actor interactions encoded in a relational graph structure. The work in [5] concatenates telemetry with the extracted visual features, whereas we apply dedicated heterogeneous processing to optimise learning by input mode. As a result, we demonstrate fused visual relational learning. Whereas other authors have claimed superior performance by one learning paradigm over the other [18], [23], we demonstrate heterogeneous synergy from fusion.

Limitations: While our study presents a fused VRSTNN capable of real-time operation on a real dataset, a challenge remains in obtaining actor state data and manually annotated actor bounding boxes in large real-world collision datasets. Though ESTI standards for CAMs exist [43], the inclusion of actor state data in collision datasets remains limited.

In addition, this study focuses on establishing a foundational understanding to motivate further investigation. While the experiments conducted provide valuable insights, they are limited by the scope of public datasets. Recognizing the importance of practical validation, we acknowledge the necessity of incorporating case studies to identify the safe operating range and behaviour of the system before deployment. Therefore, future work should integrate case studies that offer a more granular understanding of the method's applicability and limitations.

Moreover, fusion was performed after feature extraction as the multi-modal inputs required dedicated spatial processing. To demonstrate fusion, the extracted features were concatenated along the feature dimension to preserve the ordinal sequence. However, as heterogeneous fusion remains a research question in itself, future work could explore different fusion schemes regarding how to fuse and when to fuse.

VII. CONCLUSION

In this study, we addressed the need for accurate and timely hazard detection for the safe deployment of L3+ ADSs. To achieve this, we investigated different input encodings to represent scene data and how to process that data to enhance early warning prediction and generalisation to unseen scenarios.

Our results on input encoding suggested that raw image and actor state data can overload the model with complexity and noise. Whereas, simplified encoding can improve performance by guiding learning towards areas of interest. From our findings, we proposed a relational graph encoding for scene data that provides a framework to represent the arrangement, relationships and behaviours of road users. We then processed the graph encodings with GNNs to learn essential scene context to detect hazardous events early using the global graph structure that describes road user arrangement and the local interactions from individual actor relationships.

Tests on individual models revealed superior GNN performance in early detection and processing of actor state data and indicated CNN preference for visual analysis that remains invaluable on real-world datasets that lack actor state.

Given the unique advantages of the features gained from image data and actor state data, we proposed a novel VRSTNN that leverages multi-modal processing by integrating a CNN to extract visual cues and a GNN to provide the essential scene context needed for early and reliable detection. Our evaluation results show that our VRSTNN outperforms SOTA models in terms of accuracy, F1 and FNR on a real and synthetic benchmark dataset and matches human-level and RT.

In conclusion, we present our proposed VRSTNN as a new learning approach that emphasises the fusion of visual cues with relational context to enrich scene understanding. As scene context is crucial for early and reliable hazard detection, we present a method to help extract this information for prediction. Ultimately, we present these innovations to help contribute to the safe development of L3+ driving systems.

References

[1] World Health Organization, “Global status report on road safety 2018,” WHO, Geneva, Switzerland, Tech. Rep., 2018.

[2] R. Hussain and S. Zeadally, “Autonomous Cars: Research Results, Issues, and Future Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2019.

[3] SAE International, “Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,” SAE International, Tech. Rep., 2018.

[4] S. Lefèvre, D. Vasquez, and C. Laugier, “A survey on motion prediction and risk assessment for intelligent vehicles,” *ROBOMECH Journal*, vol. 1, no. 1, p. 1, 12 2014.

[5] M. Strickland, G. Fainekos, and H. B. Amor, “Deep Predictive Models for Collision Risk Assessment in Autonomous Driving,” in *IEEE Int.Conf. on Robotics and Automation*. IEEE, 5 2018, pp. 4685–4692.

[6] E. Yurtsever, Y. Liu, J. Lambert, C. Miyajima, E. Takeuchi, K. Takeda, and J. H. L. Hansen, “Risky Action Recognition in Lane Change Video Clips using Deep Spatiotemporal Networks with Segmentation Mask Transfer,” in *IEEE Intelligent Transportation Systems Conference*. IEEE, 10 2019, pp. 3100–3107.

[7] C. Li, H. Chan, and T. Chen, “Who make drivers stop? Towards driver-centric risk assessment: Risk object identification via causal inference,” in *IEEE Conf. Int. Robots and Systems*, 2020, pp. 10 711–10 718.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you? Explaining the predictions of any classifier,” in *Int. Conf. Proc. ACM on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016. Association for Computing Machinery, 8 2016, pp. 1135–1144.

[9] G. Plumb, M. T. Ribeiro, and A. Talwalkar, “Finding and Fixing Spurious Patterns with Explanations,” *Transactions on Machine Learning Research*, pp. 1–26, 6 2022.

[10] S. Sagawa, A. Raghunathan, P. Wei Koh, and P. Liang, “An Investigation of Why Overparameterization Exacerbates Spurious Correlations,” in *Proceedings of Machine Learning Research*, 2020, pp. 13–18.

[11] H. Y. Yatbaz, M. Dianati, and R. Woodman, “Introspection of DNN-Based Perception Functions in Automated Driving Systems: State-of-the-Art and Open Research Challenges,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 1–19, 2023.

[12] S. S. Banerjee, S. Jha, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer, “Hands Off the Wheel in Autonomous Vehicles?: A Systems Perspective on over a Million Miles of Field Data,” in *IEEE Conf. Dependable Systems and Networks*. IEEE, 6 2018, pp. 586–597.

[13] G. Puebla and J. S. Bowers, “Can deep convolutional neural networks support relational reasoning in the same-different task?” *Journal of Vision*, vol. 22, p. 11, 9 2022.

[14] S. Stabinger, D. Peer, J. Piater, and A. Rodríguez-Sánchez, “Evaluating the progress of deep learning for visual relational concepts,” *Journal of Vision*, vol. 21, p. 8, 10 2021.

[15] J. Kim, M. Ricci, and T. Serre, “Not-So-CLEVR: learning same–different relations strains feedforward neural networks,” *Interface Focus*, vol. 8, pp. 1–13, 8 2018.

[16] D. Xiao, W. Geiger, H. Yatbaz, M. Dianati, and R. Woodman, “Detecting Hazardous Events: A Framework for Automated Vehicle Safety Systems,” in *IEEE Conf. Intell. Transp. Sys.* IEEE, 2022, pp. 641–646.

[17] W. Bao, Q. Yu, and Y. Kong, “Uncertainty-based Traffic Accident Anticipation with Spatio-Temporal Relational Learning,” in *ACM Int. Conf. on Multimedia*. ACM, 2020, pp. 2682–2690.

[18] A. V. Malawade, S. Y. Yu, B. Hsu, D. Muthirayan, P. P. Khargonekar, and M. A. Faruque, “Spatio-Temporal Scene-Graph Embedding for Autonomous Vehicle Collision Prediction,” *IEEE Internet of Things Journal*, vol. 4662, no. c, pp. 1–10, 2022.

[19] M. Green, “”How Long Does It Take to Stop?” Methodological Analysis of Driver Perception-Brake Times,” *Transportation Human Factors*, vol. 2, no. 3, pp. 195–216, 9 2000.

[20] P. L. Olson and M. Sivak, “Perception-Response Time to Unexpected Roadway Hazards,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 28, no. 1, pp. 91–96, 2 1986.

[21] J. M. Hankey, D. V. McGehee, T. A. Dingus, E. N. Mazzae, and W. R. Garrott, “Initial Driver Avoidance Behavior and Reaction Time to an Unalerted Intersection Incursion,” *Proceedings of the Human Factors and Ergonomics Society*, vol. 40, no. 18, pp. 896–899, 10 1996.

[22] Tschodrich Sebastian, Matthies Marc, Monske Simon, Valerie Hergeth Antonia, Kohlhas Christian, and Kiefer Nicolas, “The vehicle data big bang; how to turn theory into reality,” Capgemini, Tech. Rep., 2022.

[23] S.-Y. Yu, A. V. Malawade, D. Muthirayan, P. P. Khargonekar, and M. A. A. Faruque, “Scene-Graph Augmented Data-Driven Risk Assessment of Autonomous Vehicle Decisions,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

[24] C. Michael, C. Madalina, M. Pierre, and R. Sebastian, *Graph Structures for Knowledge Representation and Reasoning*, ser. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021.

[25] M. M. Bronstein, J. Bruna, Y. Lecun, A. Szlam, and P. Vandergheynst, “Geometric Deep Learning: Going beyond Euclidean data,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[26] C. Michel and M. Marie-Laure, *Graph-based Knowledge Representation*. London: Springer London, 2008.

[27] M. Z. Al-Taie and S. Kadry, *Graph Theory*. Berlin, Germany: Springer, 2017.

[28] R. J. Trudeau, *Introduction to Graph Theory*. New York, NY, USA: Dover Pub., 2003.

[29] S. Mylavarapu, M. Sandhu, P. Vijayan, K. M. Krishna, B. Ravindran, and A. Namboodiri, “Towards Accurate Vehicle Behaviour Classification With Multi-Relational Graph Convolutional Networks,” in *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 10 2020, pp. 321–327.

[30] ——, “Understanding dynamic scenes using graph convolution networks,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 8279–8286, 2020.

[31] S. Casas, C. Gulino, R. Liao, and R. Urtasun, “SpAGNN: Spatially-Aware Graph Neural Networks for Relational Behavior Forecasting from Sensor Data,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2020, pp. 9491–9497.

[32] C. Li, Y. Meng, S. H. Chan, and Y. T. Chen, “Learning 3D-aware Egocentric Spatial-Temporal Interaction via Graph Convolutional Networks,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2020, pp. 8418–8424.

[33] R. Nayak, U. C. Pati, and S. K. Das, “A comprehensive review on deep learning-based methods for video anomaly detection,” *Image and Vision Computing*, vol. 106, p. 104078, 2 2021.

[34] K. Pawar and V. Attar, “Deep learning approaches for video-based anomalous activity detection,” *World Wide Web*, vol. 22, pp. 571–601, 2019.

[35] B. Kiran, D. Thomas, and R. Parakkal, “An Overview of Deep Learning Based Methods for Unsupervised and Semi-Supervised Anomaly Detection in Videos,” *Journal of Imaging*, vol. 4, p. 36, 2 2018.

[36] W. Luo, W. Liu, and S. Gao, “Remembering history with convolutional LSTM for anomaly detection,” in *IEEE Conf. on Multimedia and Expo*. IEEE, 7 2017, pp. 439–444.

[37] Z. Liu and J. Zhou, “Introduction to Graph Neural Networks,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, pp. 1–127, 3 2020.

[38] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A Comprehensive Survey on Graph Neural Networks,” *IEEE Trans. Neural Networks and Learning Systems*, vol. 32, pp. 4–24, 2021.

[39] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful are Graph Neural Networks?” in *7th International Conference on Learning Representations*, 2018, pp. 1–17.

[40] D. Xiao, M. Dianati, W. G. Geiger, and R. Woodman, “Review of Graph-Based Hazardous Event Detection Methods for Autonomous Driving Systems,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–19, 2023.

[41] UK Government, “Road accidents and safety statistics - GOV.UK,” 2020. [Online]. Available: https://www.gov.uk/government/collections/road-accidents-and-safety-statistics#road-casualty-annual-statistics-

[42] US Department of Transportation, “Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey,” NHTSA, Washington, DC, Tech. Rep., 2018.

[43] European Telecommunications Standards Institute, “EN 302 637-2 - V1.4.1 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service,” 2019.

[44] P. Veličković, “Everything is connected: Graph neural networks,” *Current Opinion in Structural Biology*, vol. 79, p. 102538, 4 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0959440X2300012X

[45] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, “STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 10 2019, pp. 6271–6280. [Online]. Available: https://ieeexplore.ieee.org/document/9010834/

[46] D. Cao, J. Li, H. Ma, and M. Tomizuka, “Spectral Temporal Graph Neural Network for Trajectory prediction,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2021-May. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1839–1845.

[47] W. Kong, Y. Liu, H. Li, C. Wang, Y. Tao, and X. Kong, “GSTA: Pedestrian trajectory prediction based on global spatio-temporal association of graph attention network,” *Pattern Recognition Letters*, vol. 160, pp. 90–97, 8 2022.

[48] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling Relational Data with Graph Convolutional Networks,” in *Lecture Notes in Computer Science*. Cham, Switzerland: Springer International Publishing, 2018, vol. 10843, pp. 593–607.

[49] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional Networks on Graphs for Learning Molecular Fingerprints,” in *NIPS’15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015, pp. 2224–2232. [Online]. Available: http://arxiv.org/abs/1509.09292

[50] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An Open Urban Driving Simulator,” in *1st Conference on Robot Learning (CoRL)*, 2017.

[51] H. Kim, K. Lee, G. Hwang, and C. Suh, “Crash to Not Crash: Learn to Identify Dangerous Vehicles Using a Simulator,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 978–985, 7 2019.

[52] A. O. Ly and M. Akhloufi, “Learning to Drive by Imitation: An Overview of Deep Behavior Cloning Methods,” *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 195–209, 2021.

[53] M. A. Martinez II, C. Sitawarin, K. Finch, L. Meincke, A. Yablonski, and A. Kornhauser, “Beyond Grand Theft Auto V for Training, Testing and Enhancing Deep Learning in Self Driving Cars,” in *Transportation Research Board 97th Annual Meeting*, 2018, pp. 1–15.

[54] R. Pfeffer, K. Bredow, and E. Sax, “Trade-off analysis using synthetic training data for neural networks in the automotive development process,” *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pp. 4115–4120, 2019.

[55] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, “Unsupervised Traffic Accident Detection in First-Person Videos,” in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 273–280.

[56] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning,” in *Proceedings IEEE Computer Vision and Pattern Recognition*. IEEE Computer Society, 2020, pp. 2633–2642.

[57] J. Glenn, C. Ayush, and Q. Jing, “YOLO by Ultralytics,” 2023.

[58] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, “StrongSORT: Make DeepSORT Great Again,” *IEEE Transactions on Multimedia*, pp. 1–14, 2023.

[59] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, “A Survey on 3D Object Detection Methods for Autonomous Driving Applications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 10 2019.

[60] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos, “Compute Trends Across Three Eras of Machine Learning,” in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2022-July. Institute of Electrical and Electronics Engineers Inc., 2022.

[61] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, “Survey on Scenario-Based Safety Assessment of Automated Vehicles,” *IEEE Access*, vol. 8, pp. 87 456–87 477, 2020.

[62] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A Survey of Autonomous Driving: Common Practices and Emerging Technologies,” *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020.

[63] J. Guo, U. Kurup, and M. Shah, “Is it Safe to Drive? An Overview of Factors, Metrics, and Datasets for Driveability Assessment in Autonomous Driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 3135–3151, 8 2020.

Dannier Xiao is a Ph.D. student at the University of Warwick, researching machine learning for time-series anomaly detection in automated vehicle systems. He is part of the Intelligent Vehicles Research Group and a Lab Teacher for WM919-15 Machine Intelligence and Data Science. He received his M.Sc. in Advanced Mech. Engineering from Imperial College London in 2020 and has worked in the industry for over two years in engineering and finance across Siemens, Rolls-Royce Motor Cars and Goldacre.

Mehrdad Dianati (Senior Member, IEEE) is a professor of connected and cooperative autonomous vehicles at WMG, the University of Warwick and the School of EEECS at the Queen’s University of Belfast. He has been involved in a number of national and international projects as the project leader and the work-package leader in recent years. Prior to academia, he worked in the industry for more than nine years as a Senior Software/Hardware Developer and the Director of Research and Development. He frequently provides voluntary services to the research community in various editorial roles; for example, he has served as an Associate Editor for the IEEE Transactions On Vehicular Technology. He is the Field Chief Editor of Frontiers in Future Transportation.

Paul Jennings received the B.A. degree in physics from the University of Oxford in 1985 and the Ph.D. degree in engineering from the University of Warwick in 1996. From 1985 to 1988, he was a Physicist with Rank Taylor Hobson. Since 1988, he has been focusing on industry-focused research for the Warwick Manufacturing Group (WMG), The University of Warwick, where he is currently the Research Director. His research interests include connected and autonomous vehicles, testing, human factors, mobility and user engagement in product and environment design, and focusing on automotive and healthcare applications.

Roger Woodman is an Assistant Professor and Human Factors research lead at WMG, University of Warwick. He received the Ph.D degree from Bristol Robotics Laboratory and has over 20 years of experience working in industry and academia. Among his research interests, are trust and acceptance of new technology with a focus on self-driving vehicles, shared mobility, and human-machine interfaces. He lectures on the topic of Human Factors of Future Mobility and is the Co-director of the Centre for Doctoral Training, training doctoral researchers in the areas of intelligent and electrified mobility systems. He is a Chartered Engineer (CEng) and a Fellow of the Higher Education Academy (FHEA).