Graph(Graph): A Nested Graph-Based Framework for Early Accident Anticipation

Nupur Thakur, PrasanthSai Gouripeddi, Baoxin Li
Arizona State University
Tempe, AZ, USA
{nsthaku1, pgouripe, baoxin.li}@asu.edu

Abstract

Anticipating traffic accidents early using dashcam videos is an important task for ensuring road safety and building reliable intelligent autonomous vehicles. However, factors like high traffic on the roads, different types of accidents, limited angles of vision, etc. make this task very challenging. Using the early frames, a lot of existing methods predict a large number of false positives which poses a huge risk for all vehicles on the road. In this paper, we propose a novel end-to-end learning, nested graph-based framework named Graph(Graph) for early accident anticipation. It uses interactions between the objects in the same as well as the neighboring frames along with the global features to make precise predictions as early as possible. This way it is able to embed the local as well as global temporal information into the extracted features. Graph(Graph) outperforms state-of-the-art methods on different datasets by a large margin demonstrating its effectiveness. With empirical evidence, we highlight the importance of each component in Graph(Graph) and show their effect on the final performance. Our code is available at https://github.com/thakurnupur/Graph-Graph.

1. Introduction

Due to recent rapid progress in the self-driving cars and intelligent transportation industry [7, 22], early accident anticipation task has gained significant importance to ensure safety on the roads and avoid any casualties [1, 5, 18]. Human drivers learn to detect potential accidents from experience by observing the surrounding objects and their movements. The goal of the accident anticipation system is to predict potential hazards happening in the future given the current state of the surroundings by learning subtle cues like the human drivers do. Early anticipation will help in alerting the self-driving cars beforehand so that safety mechanisms to avoid accidents can be employed.

Nowadays, there are several kinds of sensors put on self-driving cars [19] to collect surrounding information. From the perception point of view, one of the ways to get such information is the videos recorded by dashboard cameras (shortly known as dashcams). Installing dashcams on cars has become popular in various countries, mainly to determine the responsible party in case of any accidents. Following existing literature [5, 16], we utilize these dashcam videos to predict if there is a possibility of an accident in the future.

This task of anticipating accidents using dashcam videos comes with several challenges. Firstly, the accidents are rare events (as compared to other events happening on the road) with diverse nature and happen suddenly. Second, these road scenes are generally cluttered as there are a lot of objects and events happening that are not relevant to the accident.

To address these challenges, there have been a number of methods [1, 5, 9, 18] proposed in the past. For example, [8] created a benchmark dataset called KITTI that contains videos covering a variety of driving scenarios to help learn the diverse nature of accidents. To address the second challenge, methods such as [1, 5] proposed converting the frames into object-based graph representation to eliminate the majority of background noise irrelevant to the anticipation task. However, the existing methods do not use temporal information while creating these graphs or while extracting the global frame features, which is necessary because the interaction of the objects in a frame is relative to their interactions in the previous frames. For example, consider the two frames in Fig. 1 where the bike (green bounding box) and car (red bounding box) collide with each other. Given the individual frame at time step $t+1$, it is difficult to determine if they will collide but given the reference of the previous frame, we can observe that these objects are moving closer meaning there may be a possibility of collision.

In this paper, we propose a novel end-to-end learning framework named Graph(Graph) (shortly referred to as GG) for early accident anticipation that incorporates tem-

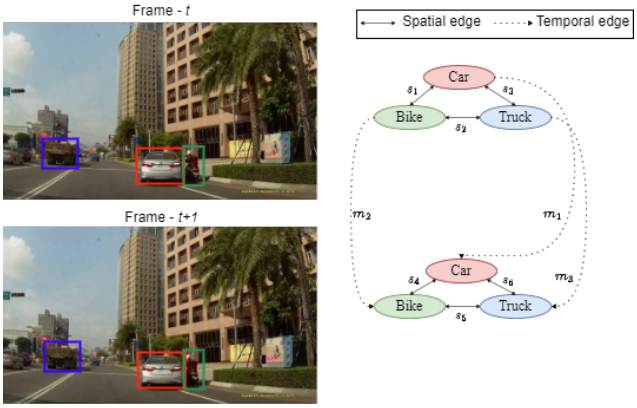

Figure 1. Illustration of spatio-temporal object graph representation of video frames (from DAD dataset). s_i and m_i denote the spatial edge weights and temporal edge weights respectively. The spatial edges provide the within-frame interaction information while the temporal edges help propagate the object information from previous frames to the next.

poral dependencies in local (object graph) as well as global (full frame) features. In particular, Graph(Graph) utilizes a nested graph representation of video frames which promotes learning subtle temporal cues by using rich spatio-temporal features. Our proposed method embeds temporal information at different stages of the framework by - 1) introducing temporal edges between objects across the frames in the object graph; 2) introducing the use of a pretrained feature extractor like I3D [4] that captures spatio-temporal global information from the frames; 3) introducing temporal connections from previous frames in the frame graph, built on top of the object graph. With extensive experiments on two dashcam video datasets - DAD [5] and CCD [1], we show Graph(Graph) is able to predict the accidents with the highest precision at the earliest, outperforming the state-of-the-art methods.

The rest of the paper is organized as - Section 2 describes the related work followed by Section 3 which contains our proposed methodology, Graph(Graph). Next, we present the experiment results and ablation study in Sections 4 and 5 respectively. Finally, we conclude in Section 6.

2. Related Work

Anticipation of traffic accidents is dependent on many factors like the relative location of the vehicles, pedestrians, etc. Recurrent Neural Networks (RNN) and Convolution Neural Networks (CNN) have been used to model such spatial and temporal data for this task. [5] proposed Dynamic-Spatial-Attention (DSA) RNN with soft attention to learn subtle cues about the candidate objects. [6] proposed a two-stream approach to predict the traffic risk. One stream is for spatial information extraction analyzing the appearance features using RCNN and the second stream is for extracting the high-level motion features using LSTM. [9] utilized a dynamic spatial attention module to learn attention weights for aggregating object-level features from the frames. Dynamic temporal attention was then used to aggregate the learned object features.

The probability of risk is dependent on the regions around an ego-vehicle (or agent) and the behavior of the agent and [23] makes use of the holistic representation of the agent appearance and the region information to predict the accident probability. Lead time to the occurrence of an event proves handy in accident warning for autonomous driving systems. As the lead time to the occurrence of the accident is crucial for building safe autonomous systems, [17] predicted such lead time for an accident using CNN + RNN framework which was trained to find cues in the video that point to a future accident. The authors use survival analysis to predict the final accident occurrence probability.

Taking a different direction, [18] proposed a novel approach that considers the deviation from the future location of the object to predict the accident. The two-step process consisted of future object localization by predicting the immediate future bounding box of the object of interest, using GRU and RNN networks and the odometry change of the ego-vehicle. [16] proposed an adaptive exponential loss for early anticipation (AdaLEA) which gradually promotes earlier anticipation during training and uses Quasi-RNN [2] to get stable latent features. [11] models an event as either a discrete-time (such as heuristic heatmaps) or a continuous-time (such as Gaussian distribution) model. They proposed a Gaussian Mixture Model Heatmap (GMMH) combining Gaussian distribution and heuristic heatmaps for accident anticipation.

Many existing methods [1, 5] represent the frames as graphs of objects in the video. This has led to increased use of Graph Neural Networks (GNN) to process the graph input due to their expressive power of modeling dynamic state transition systems. GNNs have proven to be effective in many computer vision applications like action recognition in videos [12, 24] etc. For early accident anticipation, [1] proposed the use of graph convolutional recurrent network (GCRN) to extract relational latent features from the input graph created from objects and predict final scores using Bayesian Neural Networks. [21] used a combination of GCN and RNN for risk assessment in autonomous vehicle decisions. The scene graphs built from the objects in the frame are passed through graph attention and LSTM layers to get the final predictions.

These methods show that using such a graph representation of the frames is beneficial for precise predictions, but these graphs are built from individual frames only. We propose to embed temporal dependencies in such graph repre-

sentations to enhance the extracted features, thereby making precise predictions.

3. Graph(Graph): Methodology

In this section, we discuss the details of our proposed method, Graph(Graph). The aim of this task is, given a dashcam video, predict - 1) if there will be an accident in the future and 2) predict it as early as possible. For the first part, given a sequence of observed video frames denoted by $(X_1, X_2, .., X_N)$ where N is the number of frames, the expected output is $(p_1, p_2, ...p_N)$ where p_i denotes the probability of how likely an accident will occur in the future. For the second part, the time is known as Time-to-Accident (TTA) denoted by τ [1]. For videos containing accident (positive), it is defined as $\tau = a - t$ for $t < a$ where a marks the beginning of an accident and t is the first time step when the $p_t > \alpha$ where α is the threshold. Ideally, τ should be as high as possible so there is more time to prevent the anticipated accident.

We propose Graph(Graph) for this task and its overview is shown in Fig. 2. From a pretrained object detector, the objects, their features, and labels are used to create the spatio-temporal object graph. The objects are linked within the frames as well as across the frames to allow the flow of spatial and temporal information. It is processed using GCN and pooling layers to get a per-frame representation. The second graph, the frame graph is built on top of the object graph, using the extracted graph embedding and global features. The final probabilities are obtained after passing this graph through graph attention and fully-connected layers.

3.1. Spatio-Temporal Object Graph Learning

Usually, the road scenes contain numerous objects interacting with each other in various manners at all times. However, only a few of them, usually the ones nearby are relevant to the self-driving car. Modeling such object interactions explicitly is important as their interactions over time help in deciding if there will be a potential accident in the future. Therefore, instead of using only the global features extracted from the entire frame, using graph representation created using the objects in the frames helps in predicting the accident while getting rid of background noise.

The first step of Graph(Graph) is using state-of-the-art object detectors like [14] to detect objects like cars, bikes, etc. in the observed video frames. For every frame X_i, we create a spatio-temporal object graph denoted by G_i^{obj} having S nodes where S is the number of detected objects in a frame. The initial node embeddings for this graph consists of two concatenated parts - 1) object features from the object detector $f_{obj}^e \in \mathcal{R}^{S \times d_1}$ where d_1 is the feature dimensionality and 2) the word embeddings of the detected object labels $f_{obj}^l \in \mathcal{R}^{S \times d_2}$ where d_2 is the dimension of the word embeddings. Inspired by [1], we define spatial adjacency matrix A_s^{obj} as,

$$A_s^{obj}(i,j) = \frac{e^{-d(c_i,c_j)}}{\sum_{ij} e^{-d(c_i,c_j)}} \tag{1}$$

where $d(c_i, c_j)$ is the Euclidean distance between the centers c_i and c_j of i^{th} and j^{th} object bounding boxes detected in a frame, respectively. This matrix is defined in a way that nearby objects contribute more as compared to far objects during message passing in the GCN layers. When using distance in pixel space, cases like object occlusions can lead to some false perception of how close the objects actually are physically. Despite that, this spatial adjacency matrix can get rid of irrelevant, far-away objects. Complex distance metrics can be used if camera intrinsics are known as mentioned in [1].

This spatial adjacency matrix A_s^{obj} is created from isolated, individual frames. Existing methods use this representation and depend solely on temporal models to learn the temporal dependencies. However, we propose to explicitly include the temporal information in the features to make them richer in information, thereby helping the final predictions. We introduce another adjacency matrix A_{tm}^{obj} that connects objects with the same label from previous frames. If O is the number of unique object labels detected in the current frame X_t, then temporal adjacency matrix A_{tm}^{obj} is defined as,

$$A_{tm}^{obj}(i,j) = \begin{cases} s(i,j), & l_i = l_j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $s(i,j)$ is the cosine similarity between the object features of i^{th} and j^{th} objects with labels l_i and l_j belonging to frames at time t and $t-u, u < t$, respectively. These edges ensure the flow of information between the objects of the same class across frames and avoid irrelevant information flow between objects from different classes. The node embeddings are updated using the message-passing mechanism by sending them through graph convolutional layers (GCN) [10],

$$\begin{aligned} f_{obj}^s &= \text{GCN}([\phi(f_{obj}^e), \phi(f_{obj}^l)], A_s^{obj}) \\ f_{obj}^{tm} &= \text{GCN}([\phi(f_{obj}^e), \phi(f_{obj}^l)], A_{tm}^{obj}) \\ f'_{obj} &= [f_{obj}^s, f_{obj}^{tm}] \end{aligned} \tag{3}$$

where ϕ represents fully-connected (FC) layer used to reduce the feature dimensionality and [.] represents the concatenation operator.

Fig. 1 shows an illustration of a spatio-temporal object graph. The spatial edges provide the relative position information of three objects in the frame. Once the node embedding is updated using these edges for the frame at time t, this information is propagated to the same class objects in the

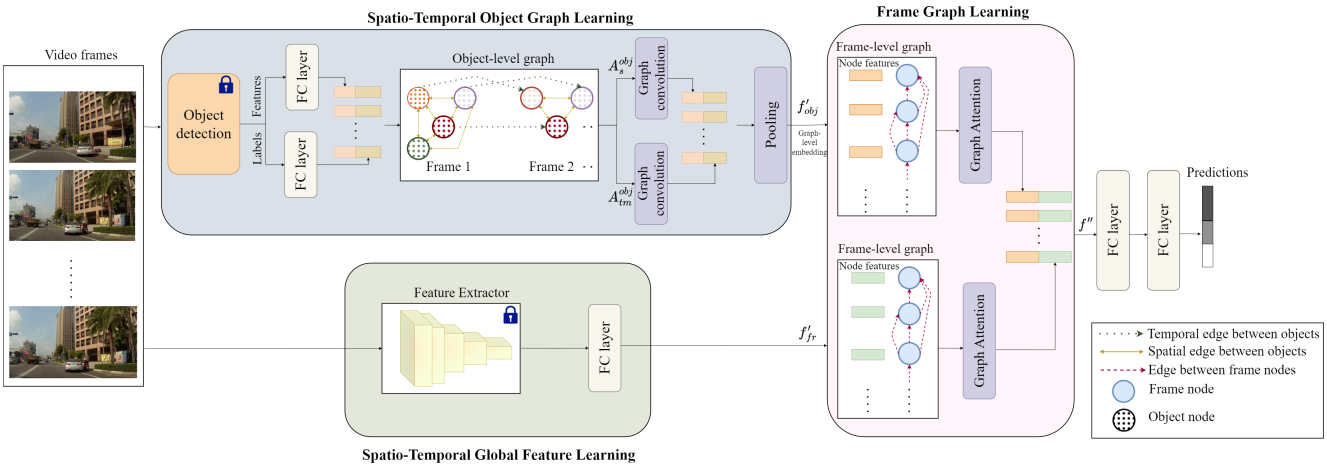

Figure 2. Overall framework of Graph(Graph). The objects detected in a frame form the object-level graph with spatial and temporal edges. This graph processed by graph convolution and pooling layers along with the extracted global spatio-temporal feature is used to create the frame graph and pass it through graph attention and FC layers to produce final softmax probabilities. The lock sign indicates that the module is frozen during the training.

next frame through the temporal edges. This means while updating the node embeddings for objects in the frame at time $t+1$, there is explicit information of objects from previous frames. Also, connecting the 'car' object from frame t to 'truck' from frame $t+1$ would lead to the flow of unnecessary information which is why only objects from the same class are temporally connected.

3.2. Spatio-Temporal Global Feature Learning

While the graph representation of the frame helps in modeling the object interactions explicitly, the full frame feature is important to provide global information about the scene. Unlike the existing approaches [1, 5, 16] which use pretrained image models like VGG16 [15] to extract these global features, we propose to use pretrained video models like I3D [4] network for global feature extraction. This is beneficial as these models take a sequence of frames as input, having spatial and temporal knowledge while extracting the frame feature.

To extract this global frame feature f_{fr}^t for a frame at time t, a sequence of frames $X_{seq} = (X_{t-u}, ..., X_t)$ is passed to the pretrained network Q. For every frame (we drop superscript t for ease of reading), a $h-$dimensional feature is produced. To avoid high computation for the frame graph in the next phase of learning, we use a FC (fully-connected) layer to reduce the feature dimensions,

$$\begin{aligned} f_{fr} &= Q(X_{seq}) \\ f'_{fr} &= \phi(f_{fr}) \end{aligned} \tag{4}$$

where ϕ represents a FC layer.

3.3. Frame Graph Learning

The frame graph is created to learn the temporal dependencies between the frames using the features extracted from the object graph and the global features. In the frame graph, every node represents a frame at time step t and the edges connecting them are unidirectional (going from the previous frames to the current frame) to avoid the flow of information from future frames to past frames. For every node at time t, there are k edges from previous k nodes. Formally, the adjacency matrix A^{fr} for this graph is defined as,

$$A^{fr}(i,j) = \begin{cases} 1, & i - j <= k \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where i represents the current node.

This graph is passed through two parallel graph attention layers [3] (GAT) with f'_{obj} and f'_{fr} being the node features for first and second layer respectively,

$$\begin{aligned} f''_{obj} &= \text{GAT}(f'_{obj}, A^{fr}) \\ f''_{fr} &= \text{GAT}(f'_{fr}, A^{fr}) \\ f'' &= [f''_{obj}, f''_{fr}] \end{aligned} \tag{6}$$

The graph attention is used here to learn the importance of different frames (nodes) for predicting future accidents. These concatenated features are then passed through FC layers to yield the final frame-wise probabilities $(p_1, p_2, ..., p_N)$ of how likely an accident will occur in the future.

This entire system is trained in an end-to-end fashion with the only frozen components being the object detection

and the global feature extraction modules. We use the standard cross-entropy loss to train Graph(Graph),

$$L_(p, y) = -\sum_{m=1}^{M} y_m log\left(\frac{e^{p_m}}{\sum_j^M e^{p_j}}\right) \quad (7)$$

where M is the number of classes ($M = 2$ in this case), p denotes the softmax probabilities and $y \in \{0, 1\}$ represents the ground-truth labels for the frames.

4. Experiments and Results

In this section, we begin by describing the experimental settings like the datasets, network architecture, and baselines used for comparison and evaluation metrics followed by an explanation of different experiment results.

4.1. Experimental Settings

4.1.1 Datasets

We use two datasets of dashcam videos for our experiments. Dashcam Accident Dataset (DAD) [5] is a dataset containing dashcam videos from different cities in Taiwan. With fps (frames per second) of 20, each of these videos is 5 seconds long. It consists of 678 videos divided into 1750 clips of which 620 are accident clips (positive) with the accident in the last 10 frames and 1130 are normal (negative) videos. The training set has 455 positive and 829 negative videos making a total of 1284 videos. The test set has 165 positive and 301 negative videos, summing to 466 videos.

Car Crash Dataset (CCD) [1] consists of 4500 dashcam videos of which 1500 are positive videos with the accident in the last 2 seconds and 3000 normal videos are taken from BDD100K [20]. Each video is 5 seconds long with temporal and environmental annotations like weather, day/night, etc. provided. The train and test sets consist of 3600 videos and 900 videos respectively.

4.1.2 Networks and Baselines

We may use pretrained networks like Faster R-CNN [14] for the object detection module. For fair comparison, we use the detected object bounding boxes, their features and labels provided on the datasets' official code repositories[1, 2]. The top 19 objects are used with object feature vector dimension $d_1 = 4096$. We use the GloVe embedding [13] function from the spaCy library to get the word embeddings ($d_2 = 300$) for the object labels. As the videos are short (5 seconds), we set $u = 1$ for all the experiments, i.e. temporal edges between the objects from the previous frame only and $k = 20$. For all the datasets, we keep the same network architecture and configuration as shown in Fig. 3.

[1]https://github.com/smallcorgi/Anticipating-Accidents (DAD)
[2]https://github.com/Cogito2012/UString (CCD)

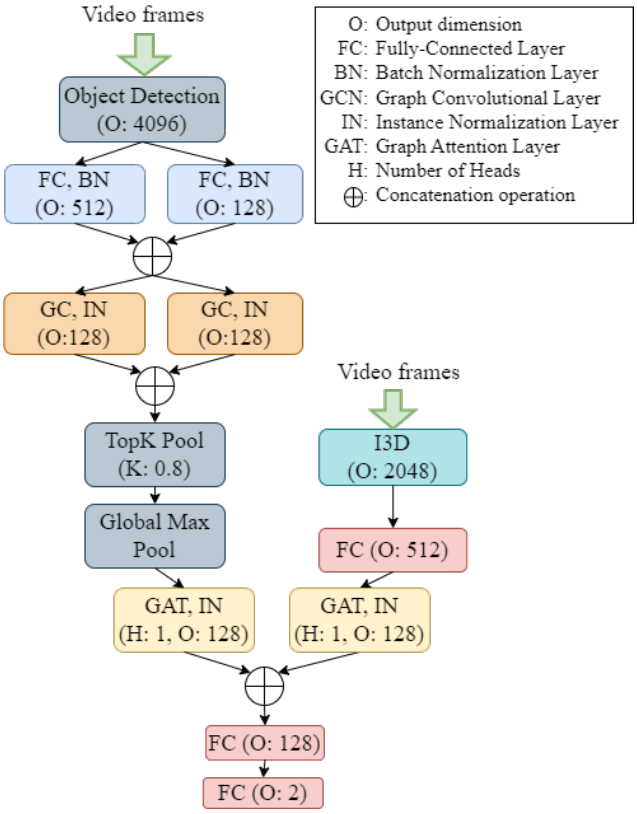

Figure 3. Graph(Graph) architecture configuration details

For global feature extraction, we use the I3D network[3] [4]. For every frame, the feature is h-dimensional where $h = 2048$. We use the PyTorch library for all our experiments.

We compare with three state-of-the-art methods - DSA [5], adaLEA [16] and Ustring [1]. We chose them for comparison as these are recent methods using a graph representation of the video frames as input. We quote the numbers for these baselines as reported in [1].

4.1.3 Evaluation Metrics

While anticipating accidents, we need to be 1) as precise as possible and 2) as early as possible. Following [1, 5], we use Average Precision (AP) and mTTA (mean Time-to-Accident) as evaluation metrics. Average precision helps in evaluating how correct the prediction is. For an accident video, if the prediction probability $p_t > \alpha$, then it is considered a correct prediction. Different values of the threshold α are used to get corresponding pairs of precision and recall values and calculate the Average Precision. The higher the AP, the more precise the model in predicting the accidents.

[3]We use the pretrained model and code provided at https://github.com/piergiaj/pytorch-i3d

To evaluate if the system is able to detect the accident early, we use the Time-to-Accident (TTA) metric. It is calculated by checking the difference between the start of the accident and the first frame where $p_t > \alpha$. For different values of α, different recall values and TTA values are obtained and mTTA is calculated by taking the mean of all of these TTA values. Even for low AP, a very high TTA value can be obtained where there are a lot of false positives. This is not meaningful as the technique is predicting accidents arbitrarily. Therefore, we report this metric corresponding to the highest AP achieved.

4.2. Evaluation Results

We discuss different evaluation results like comparison with baseline methods, visualization of the predictions of Graph(Graph), etc. in this section.

4.2.1 Comparison with State-of-the-art Methods

Tab. 1 summarizes our comparison results on DAD and CCD datasets. It is clear from the table that Graph(Graph) (shortly denoted as GG) outperforms all the baselines for both evaluation metrics. This implies it can predict the accident at the earliest and with the highest precision among all the compared methods. This also indicates that using spatio-temporal features (graph and global) instead of just spatial features significantly helps in modeling the behavior of the objects in the scene, thereby making precise predictions at the earliest.

Note that the videos in the DAD dataset are from various cities and generally have crowded traffic. Graph(Graph) results in a huge increase (almost $+10\%$ for AP and $+1$ second for mTTA) for this dataset as compared to the Ustring [1]. This increase in performance showcases the versatility of Graph(Graph) in handling videos with diverse conditions like varying traffic density, roads, and backgrounds. mTTA value of 4.45 seconds is extremely helpful to send early alerts to the drivers/autonomous cars and avoid several accidents and resulting casualties.

4.2.2 Visualizing the Predictions

In Fig. 4, we visualize predictions of different positive (accident) and negative (no accident) videos. The first two graphs are for positive videos and the last one is for a negative video from the DAD dataset. For positive videos, it can be observed that Graph(Graph) predicts an accident with a high probability from the very start ($\sim$4.5 seconds before the beginning of the accident). For a normal video, the scores are significantly lower than the threshold indicating fewer occurrences of false positives.

Based on the proximity of different objects, the confidence scores produced by Graph(Graph) change. For example, in the first row of Fig. 4, there is a dip in the confidence score near frame 40 where the objects (car, bike) seem to be moving away from each other. However, the score starts to go up after frame 50 where the objects are moving closer. Similarly, for the video shown in the second row in Fig. 4, the score starts to drop around frame 50 as the objects seem far away from each other. As we use pixel distance between the objects in the object graph, occlusions (like in frame 30 in the second row) can lead to high scores momentarily. Nonetheless, the performance of Graph(Graph) shows how the interactions of objects in pixel space can still help in eliminating irrelevant objects as well as in making precise predictions.

Dataset	Method	AP	mTTA (s)
DAD	DSA [5]	48.1%	1.34
	adaLEA [16]	52.3%	3.43
	Ustring [1]	53.7%	3.53
	GG (Ours)	**63.6%**	**4.45**
CCD	DSA [5]	99.6%	4.52
	Ustring [1]	99.5%	4.74
	GG (Ours)	**99.9%**	**4.96**

Table 1. Experiment results on DAD and CCD datasets. mTTA values are in seconds. Graph(Graph) (GG) outperforms all the baselines for both datasets. The numbers for baselines are quoted from [1].

4.2.3 Comparison of Different Global Features

Pretrained image networks like VGG16 [15] can be used for feature extraction for individual frames. However, these features are based on spatial information only and have no information about neighboring frames. As video frames are dependent temporally, using spatio-temporal frame features is very crucial, especially for accident anticipation task where relative movements of objects matter.

Feature type	AP	mTTA (s)
VGG-16 [15]	62.0%	4.21
I3D [4]	**63.6%**	**4.45**

Table 2. Results for different types of global frame features used in Graph(Graph) for DAD dataset. The spatio-temporal feature extracted from pretrained I3D performs better than only using the spatial feature extracted from VGG-16.

To extract a spatio-temporal global frame feature, we use a pretrained I3D [4] network. We present experiments using spatial (extracted using VGG16) and spatio-temporal (extracted using I3D) global features in the Graph(Graph) framework. All other modules remain the same, with only the type of global feature used changed. The results are pre-

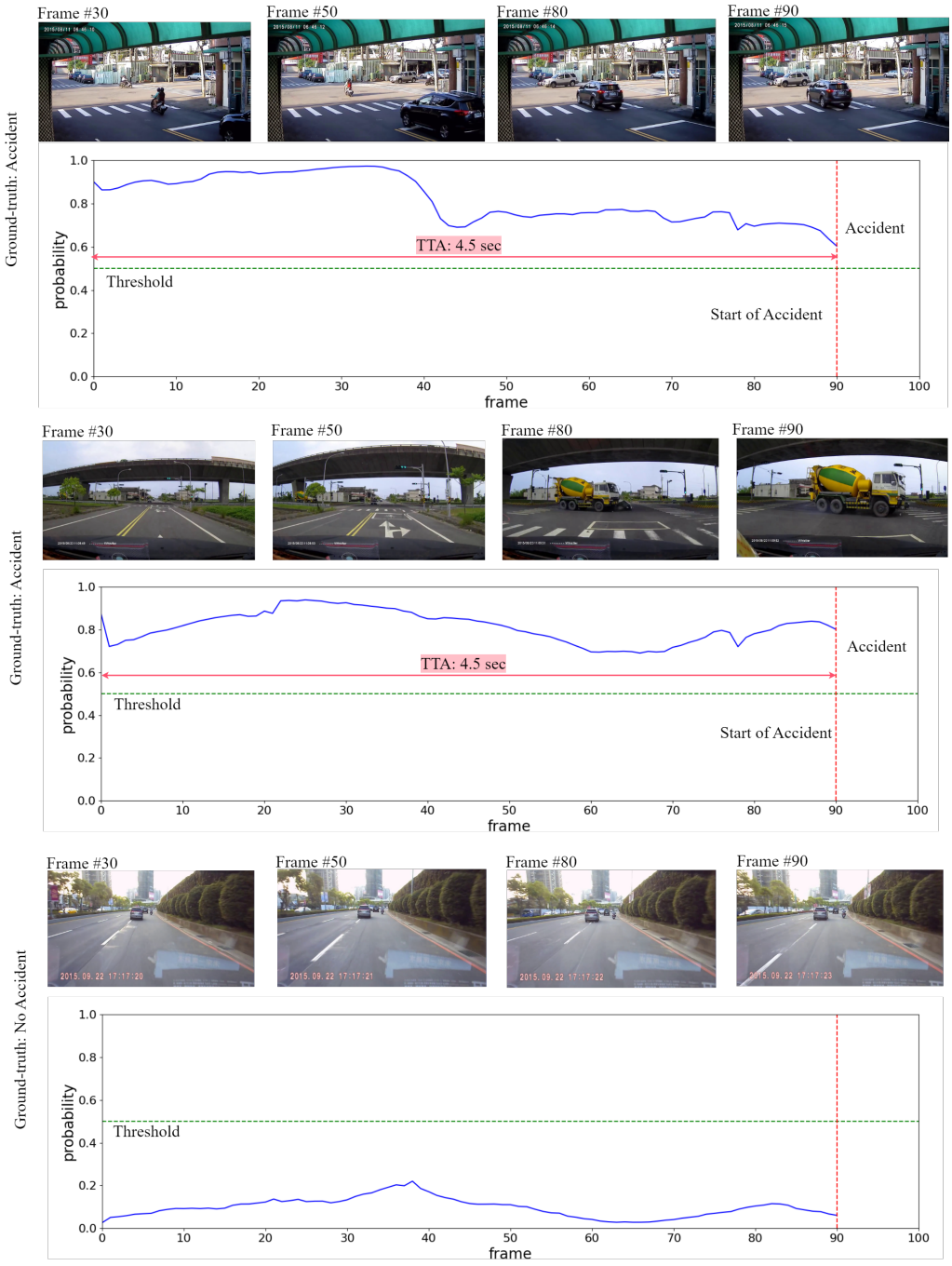

Figure 4. Prediction probability plots for different videos from the DAD dataset. The first two videos belong to the positive (accident) class and the last one belongs to the negative (no accident) class. Graph(Graph) is able to predict an accident very early for positive videos while the probability is lower than the threshold for all the frames for a negative video.

sented in Tab. 2. There is an increase in both precision and the mTTA when the I3D feature is used. As the I3D network takes a short sequence to generate the per-frame feature, the extracted feature has knowledge of previous frames making it richer in both spatial and temporal information than the one produced using spatial information only.

We also display the Precision-Recall curve and Time-to-Accident-Recall curve for both these cases in Fig. 5. For a high recall value i.e. Recall@80% the precision when using the I3D feature is higher than when the VGG16 feature is used, whereas the Time-to-Accident (TTA) for Recall@80% is similar for both cases. This indicates that there

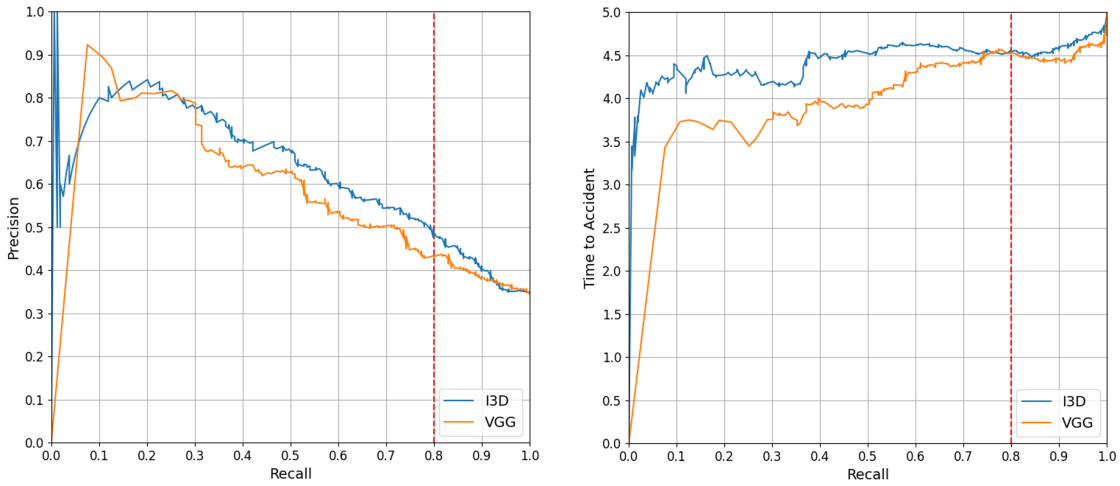

Figure 5. Precision vs. Recall curve (left) and Time-to-Accident (TTA) vs. Recall curve (right) for Graph(Graph) using I3D and VGG16 frame features for DAD. Precision when Recall@80% is higher for I3D features highlighting the importance of the temporal information.

are more false positives in the first case (VGG16 feature) as compared to the second one. Overall, including temporal information in the global feature enhances the performance as well as reduces the chances of false accident alerts.

5. Ablation Study

In this section, we present experiments to demonstrate the significance of different components used in Graph(Graph) - the spatial and temporal adjacency matrix for spatio-temporal object graph, A_s^{obj} and A_{tm}^{obj} respectively and the spatio-temporal global features, f_{fr}. Using experiments on the DAD dataset, we show how these components affect both the evaluation metrics.

A_s^{obj}	A_{tm}^{obj}	f_{fr}	**AP**	**mTTA (s)**
✓	✗	✗	55.8%	4.03
✓	✓	✗	60.5%	4.43
✗	✗	✓	48.8%	4.45
✓	✓	✓	**63.6%**	**4.45**

Table 3. Ablation study results on DAD dataset. mTTA values are in seconds. The effect of every component on the final performance is significant with the highest performance achieved when all components are used.

Tab. 3 summarizes the ablation study results. There is almost a 5% increase in AP from case 1 (first row) to case 2 (second row), which is a result of using A_{tm}^{obj} adjacency matrix that links objects across frames. This showcases the importance of our proposed A_{tm}^{obj} adjacency matrix and how temporal information flow across the frames is crucial in making confident predictions. The third row uses only the global spatio-temporal feature f_{fr} and the frame graph. There is a good amount of confidence in the predictions for this case showing how these features contain meaningful and relevant information for the task. However, there is a huge dip in the performance (almost 15% as compared to the one when the entire framework is used) which highlights the importance of the object graph. Also, even with low AP, the mTTA value is high for this case, indicating a large number of false positives. This means the model is overfitting on accident videos. Finally, the last row is the performance when all the components are used together which yields the highest AP and mTTA values.

6. Conclusion

As early accident anticipation is one of the major tasks towards building safe autonomous vehicles, we propose a novel end-to-end learning, nested graph-based framework named Graph(Graph) for handling this task. The graph representation of frames provides explicit information about the interactions between the objects while the global features provide the overall scene information. Our method introduces a simple way of incorporating the flow of temporal information in the features for object graphs, frame graphs, and global feature extraction. We evaluate Graph(Graph) on DAD and CCD datasets and show how it can increase performance from that achieved by the state-of-the-art methods by a large margin.

Acknowledgement

The work was supported in part by a grant from ONR (No. N00014-19-1-2119). The views and opinions expressed in this paper are those of the authors and do not necessarily reflect the views or opinions of ONR.

References

[1] Wentao Bao, Qi Yu, and Yu Kong. Uncertainty-based traffic accident anticipation with spatio-temporal relational learning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2682–2690, 2020. 1, 2, 3, 4, 5, 6

[2] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-recurrent neural networks. *arXiv preprint arXiv:1611.01576*, 2016. 2

[3] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. 4

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 4, 5, 6

[5] Fu-Hsiang Chan, Yu-Ting Chen, Yu Xiang, and Min Sun. Anticipating accidents in dashcam videos. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part IV 13*, pages 136–153. Springer, 2017. 1, 2, 4, 5, 6

[6] Gary-Patrick Corcoran and James Clark. Traffic risk assessment: A two-stream approach using dynamic-attention. In *2019 16th Conference on Computer and Robot Vision (CRV)*, pages 166–173. IEEE, 2019. 2

[7] D Dolgov. Google self-driving car project-monthly report-september 2016-on the road. *Technical report, Google*, 2016. 1

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1

[9] Muhammad Monjurul Karim, Yu Li, Ruwen Qin, and Zhaozheng Yin. A dynamic spatial-temporal attention network for early anticipation of traffic accidents. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):9590–9600, 2022. 1, 2

[10] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 3

[11] Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Future event prediction: If and when. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[12] Yangjun Ou, Li Mi, and Zhenzhong Chen. Object-relation reasoning graph for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20133–20142, 2022. 2

[13] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 5

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 3, 5

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4, 6

[16] Tomoyuki Suzuki, Hirokatsu Kataoka, Yoshimitsu Aoki, and Yutaka Satoh. Anticipating traffic accidents with adaptive loss and large-scale incident db. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3521–3529, 2018. 1, 2, 4, 5, 6

[17] Yoshiaki Takimoto, Yusuke Tanaka, Takeshi Kurashima, Shuhei Yamamoto, Maya Okawa, and Hiroyuki Toda. Predicting traffic accidents with event recorder data. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*, pages 11–14, 2019. 2

[18] Yu Yao, Mingze Xu, Yuchen Wang, David J Crandall, and Ella M Atkins. Unsupervised traffic accident detection in first-person videos. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 273–280. IEEE, 2019. 1, 2

[19] De Jong Yeong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6):2140, 2021. 1

[20] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 5

[21] Shih-Yuan Yu, Arnav Vaibhav Malawade, Deepan Muthirayan, Pramod P Khargonekar, and Mohammad Abdullah Al Faruque. Scene-graph augmented data-driven risk assessment of autonomous vehicle decisions. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7941–7951, 2021. 2

[22] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020. 1

[23] Kuo-Hao Zeng, Shih-Han Chou, Fu-Hsiang Chan, Juan Carlos Niebles, and Min Sun. Agent-centric risk assessment: Accident anticipation and risky region localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2222–2230, 2017. 2

[24] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020. 2