Enhancing Vision-Language Models with Scene Graphs for Traffic Accident Understanding

Aaron Lohner[§*], Francesco Compagno[‡], Jonathan Francis[†§**], and Alessandro Oltramari[†**]
[§]School of Computer Science, Carnegie Mellon University; Pittsburgh, USA
[‡]University of Trento; Trento, Italy
[†]Bosch Center for Artificial Intelligence; Pittsburgh, USA
alohner@alumni.cmu.edu, francesco.compagno@unitn.it, {jon.francis, alessandro.oltramari}@us.bosch.com

***Abstract*—Recognizing a traffic accident is an essential part of any autonomous driving or road monitoring system. An accident can appear in a wide variety of forms, and understanding what type of accident is taking place may be useful to prevent it from reoccurring. The task of being able to classify a traffic scene as a specific type of accident is the focus of this work. We approach the problem by likening a traffic scene to a graph, where objects such as cars can be represented as nodes, and relative distances and directions between them as edges. This representation of an accident can be referred to as a *scene graph*, and is used as input for an accident classifier. Better results can be obtained with a classifier that fuses the scene graph input with representations from vision and language. This work introduces a multi-stage, multimodal pipeline to pre-process videos of traffic accidents, encode them as scene graphs, and align this representation with vision and language modalities for accident classification. When trained on 4 classes, our method achieves a balanced accuracy score of 57.77% on an (unbalanced) subset of the popular Detection of Traffic Anomaly (DoTA) benchmark, representing an increase of close to 5 percentage points from the case where scene graph information is not taken into account.**

***Index Terms*—Neuro-symbolism, Vision-Language Models, Foundation Models, Autonomous Driving, Perception**

I. Introduction

The task of understanding traffic scenarios is one with ever-increasing importance, particularly for the advancement of Autonomous Vehicle (AV) and road infrastructure systems [Xu et al., 2021, Yao et al., 2022, Qasemi et al., 2023, Francis et al., 2022]. A major aspect of this task is to efficiently and accurately recognize different types of traffic accidents, with the ultimate goal of preventing them. We approach this challenge by modeling traffic scenes using scene graphs. To improve classification performance, the graph representation is further aligned with representations from the vision and language modalities within contrastively-trained foundation models. Starting with video clips of traffic incidents, procured from the *Detection of Traffic Anomaly* (DoTA) dataset [Yao et al., 2022], we build Scene-Traffic-Graph Inference (STGi), a multi-stage, multimodal pipeline to pre-process videos, encode them as scene graphs, and align these representations with vision and language modalities to classify traffic accidents.

In this paper, we propose Scene-Traffic-Graph Inference(STGi) as a unified system for accident classification. First, we generate scene graphs using the `roadscene2vec` (rs2v) tool [Malawade et al., 2021]. Then, we use a scene graph encoder (SGE) from the same authors [Malawade et al., 2022] to obtain encodings that are aligned to encodings of textual and visual inputs, obtained from CLIP [Radford et al., 2021] and X-CLIP [Ni et al., 2022], respectively. X-CLIP directly expands upon CLIP's image encoder to include an attention mechanism to model inter-frame communication and to generate a new embedding representation from video frames. In essence, scene graphs are treated as a new modality (or 'view') that is aligned with text and video signals, and are all used together to classify traffic accident scenes.

A summary of contributions is as follows:

- We introduce Scene-Traffic-Graph Inference (STGi), a novel method for traffic accident classification which leverages scene graphs to capture the essential features of a traffic accident.
- We show that the added signal from a scene graph modality can enhance the performance of a video-language traffic accident classifier by nearly 5 percentage points.
- Experiments in this work demonstrate that aligning the scene graph modality with vision and language give similar results to omitting alignment training, although increasing the batch size and training time during alignment shows a trend of increasing scores and the potential in further improving classification results.

II. Related Work

A. Scene Graphs for Representing Traffic Scenes

Several works have taken the approach of modeling traffic scenarios using scene graphs (SGs). Yu et al. [2020] propose one such representation that forms the basis of the SGs produced in our work. They define a graph structure with nodes representing entities such as vehicles and pedestrians while edges represent relations between the objects and are labeled by distance category (e.g., *visible, very near*) and orientation (e.g., *in front of, beside*). Zipfl and Zöllner [2021] also model these objects in a traffic scene, further accounting for relative speeds of objects in the graph edges. Guo et al. [2023] extracts similar semantic information from a traffic scene, but the focus is on generating a knowledge graph, accounting for added traffic information present in a scene such as road signs and lanes. Qasemi et al. [2023] also use a knowledge graph

*Work done while at CMU. **Equal advising.

Fig. 1: An overview of the STGi Architecture. Beginning with video and text inputs, video frames are sampled and used to generate scene graphs. Then, alignment training is performed on the three encoders before a prediction head is used to classif the accident.

to enhance a video-language model for the related task of Question Answering in the domain of traf c accidents. Their method is to inject domain-speci c knowledge by rst auto-generating video annotations and storing semantic information about a traf c scene in a knowledge graph. The use of a knowledge graph in this case is helpful in contextualizing the traf c scene in a broader traf c monitoring ontology and for answering general traf c accident-related questions, such as the impact of having more vehicles in a traf c scene, but is less effective in identifying accident types based on the interactions among vehicles in a speci c traf c scene. Instead, our method is better suited for the task of accident classi cation by leveraging scene graphs to capture relationships among objects that may be involved in a traf c accident in a given video example. Combining the use of scene graphs with knowledge graphs to model traf c settings, Dimasi [2023] makes use of a knowledge graph to boost the performance of the RelTR (Relation Transformer) model [Cong et al., 2022], an encoder-decoder model used for scene graph generation by framing it as a set prediction problem.

B. Leveraging Scene Graphs with Neural Networks

There are many possible designs and applications for leveraging the versatile structure of scene graphs to enhance the performance of neural networks [Li et al., 2019, Khademi and Schulte, 2020, Rana et al., 2023]. More generally, Zhang et al. [2019] describe the ability of scene graphs to enhance Visual Question Answering (VQA) systems as an alternative to solely focusing on vision and language features. Although their study is not speci c to our traf c domain, they claim that scene graphs derived from images can capture the essential visual features and may outperform images for VQA tasks.
Many works have focused on automatically generating scene graphs that can easily be encoded into neural networks for various downstream tasks. The rs2v tool [Malawade et al., 2021] generates scene graphs for individual frames of traf c videos and is applied to our work. Alternatively, Nag et al. [2023] design a tool speci cally for modeling videos with scene graphs which accounts for spatial and temporal relationships between frames based on sequence modeling with transformers. However, they train on the Action Genome [Ji et al., 2019] dataset and produce graphs that are unrelated to traf c scenarios, making this tool less applicable to our work.

C. Multimodal Alignment with Contrastive Representations

Radford et al. [2021] proposed a model that uses a contrastive training strategy to align text captions with images for the Contrastive Language–Image Pre-training (CLIP) model. From this, several works have focused on aligning encoders of additional modalities to those trained for CLIP, such as aligning audio [Wu et al., 2022] and haptic data [Tatiya et al., 2024]. Similar to our implementation, Koch et al. [2023] use CLIP's text encoder to contrastively train 3D scene graphs. The work by Huang et al. [2023] aims to enhance CLIP's performance by injecting scene graph information into the embedding generated by CLIP's text encoder before contrastive learning is performed between the image and text embedding spaces. To the best of our knowledge, however, our work is the rst to treat scene graphs as an additional modality that is grounded to a text and video encoder. There are also many possible strategies for fusing the outputs of a multimodal model before the downstream task at hand, such as early and late stage fusion involving concatenation [Sleeman et al., 2022, Pawłowski et al., 2023], merging [Sleeman et al., 2022], or sampling from a shared embedding space [Gao et al., 2020]. We experiment with several of these late fusion techniques on the three modality embeddings after training for alignment.

III. Methodology

To approach the problem of traf c accident classi cation, a multi-stage pipeline was developed. The pipeline, which is

represented in Figure 1, can be summarized in four stages and is described in the following sections:

1) Data pre-processing Sample video frames, generate captions, and use a scene graph generator to produce a set of scene graphs for each traf c accident example.
2) Scene graph encoder pre-training:Pre-train the scene graph encoder on the classi cation task.
3) Multimodal alignment: Align the scene graph encoder with frozen video and text encoders.
4) Fine tuning for downstream task: Train a classi cation head on top of the aligned multimodal model.

A. Data Pre-processing

In the rst stage, for extensibility and ease, we leverage and tune existing tools for generating traf c scene graphs that will later be fed into our modeling approach. Various scene graph generators (SGGs) are available for this, although not many t our speci c requirements for a generator that is capable of identifying and encoding features unique to traf c accidents, such as different types of vehicle collisions. Some do work for our needs — in particular, the roadscene2vec (rs2v) tool created by Malawade et al. [2021] which we leverage for generating scene graphs from traf c video frames. To use rs2v, we rst sample a xed amount of frames from each video, and text captions are generated manually for each class. Then, feeding in a series of frames from a traf c scene, the rs2v generator uses an object detector and keeps only the relevant entities relating to traf c (such as road, cars, pedestrians, etc.), ltering out other detected objects. Next, after generating a “bird's eye view” (BEV) projection of the image and approximating the relative location of each object in this projection, edges are connected between nearby entities in the scene, and vehicles are mapped to lanes. This forms the scene graph for a speci c frame from a video sample. The SGG is an important component of this pipeline as it de nes the elements in the scene graph modality.

An important step before employing this tool is to calibrate the BEV and adjust the proximity thresholds (which are used to create edges of varying attributes, such as near or visible, relating a pair of objects). The authors speci cally note the challenges they faced when adjusting these settings with regard to the DoTA dataset, and mention that for best results, they needed to calibrate their model for each traf c scene [Malawade et al., 2022]. In our case, we iteratively sample and adjust the BEV parameters and proximity thresholds based on the output quality of the SGs produced for various scenes in DoTA. We do this to select one con guration to generate the SGs for our task, although it remains an inherent challenge to generalize the parameter settings for the entire dataset. Some examples of the generated SGs can be seen in Appendix A.

For text, we manually compose captions describing each of the four accident classes (see Appendix B), pairing each caption with videos from its respective class to form the training examples. We also experiment with two sets of captions, however, we focus the scope of this work primarily on the scene graph generation process from video frames rather than caption generation. It should be noted that although these captions are used during training for aligning the SGE, they are not used during inference on the nal model because they are not from our dataset and provide a one-to-one mapping directly to the accident classes. Through ne-tuning the classi cation head using these captions, the model can achieve 100% accuracy this way. Instead, we ne-tune and test the classi cation head using the same caption as noted in Appendix B for all examples.

B. Scene Graph Encoder

In stage two, the SGE encodes scene graphs to xed-length embeddings. To do this, a multirelational graph convolutional network (MRGCN) as employed by Malawade et al. [2022] is used, which includes an attention mechanism along with LSTMs to model the spatial and temporal relations of the scene graphs generated for a given video. The SGE may be pre-trained for the classi cation task before aligning the scene graphs with the language and vision modalities.

C. Multimodal Alignment

For stage three, we use the CLIP model [Radford et al., 2021], but replace the CLIP image encoder with X-CLIP [Ni et al., 2022], a pre-trained video encoder, to accept videos rather than images. We leave CLIP's text encoder in place. We then freeze the weights from the video and text encoders and align them to the SGE encoder, which takes the new “scene graph modality” as input.

D. Fine Tuning for Downstream Task

The nal stage involves training a classi cation head that accepts embeddings from the three modalities and outputs an accident class for a traf c scene. This requires the selection of a method to fuse the signal from the three modalities followed by training a small network. For this, we experiment with different late fusion techniques such as taking a weighted linear combination of the modality outputs [Kaliciak et al., 2014] and training various MLP classi ers [Kiela et al., 2018] with and without activations on top of the concatenated embeddings. Based on these experiments, we choose the approach of concatenating the vectors from the three modalities as described by Sleeman et al. [2022] and, following Wu et al. [2022], train a 2-layer MLP with ReLU activations as our classi cation head.

IV. Experimental Design

A. Dataset

Given the nature of our multimodal approach, we require data in the form of videos, text, and scene graphs. Regarding the videos, we focus on using the Detection of Traf c Anomaly (DoTA) dataset [Yao et al., 2022] for this task. DoTA is a dataset consisting of 4,677 curated videos of traf c anomalies from YouTube, and was speci cally designed to answer the following three questions about each video: (1) When does the anomalous event (i.e., accident) start and end; (2) Where are the anomalous regions of each video frame; and

(3) What type of anomaly is taking place. For our use case on this dataset, question one is irrelevant since we only sample frames from the anomalous portion of the video based on the annotations of the DoTA authors. Additionally, question two is less essential for us since we use a SGG, which is meant to extract the necessary objects from video frames and model their relevant spatial relationships for our task. We therefore prioritize providing an answer for the third question, that is, classifying different traf c anomalies. However, given the multitude of classes and the inherent challenge in distinguishing them even with the naked eye, we limit the problem to only a subset of four of the available classes.

In addition to the provided video classes, Yao et al. [2022a] also divides the videos into two types:ego (where the vehicle from which the video recording is captured is involved in the accident), and its opposite,non-ego. Every video class contains videos of both types. By default, rs2v is designed to only model relationships between the ego vehicle and other objects, not among other objects themselves, so we further restrict our task to the set of ego videos available in DoTA. In total, this leaves us with 4 classes and 2,163 videos. The classes are labeled as follows, describing the type movement of the vehicle as the accident occurs:moving_ahead_or_waiting, oncoming, turning, and lateral.

B. Baselines and Metrics

In order to evaluate the usefulness of representing traf c scenes graphically to classify them, we rst verify whether a SGE alone is superior to randomly guessing an accident class. We later evaluate the effectiveness of using SGs in a multimodal classi er by comparing the difference in the performance between using just text and video embeddings versus adding in the signal from the SG modality.

For this and other results, the primary metrics we focus on are accuracy and balanced accuracy. Accuracy is simply de ned as the proportion of the correctly classi ed predictions out of the total number of predictions and remains an interpretable metric that is relatively simple to compare across different benchmarks. There is, however, signi cant imbalance in the dataset, with theturning class appearing over twice as frequently as the remaining three. Since we want to evenly prioritize classifying each type of accident, we turn to balanced accuracy, which is computed as the (unweighted) average recall over all classes, as a more reliable metric for our use case.

C. Modality Alignment, Pre-training and Hyperparameters

The core TG model consists of three encoders for three different modalities: text, vision, and graph. For our classification task, a MLP classi er head is placed at the end of the model to output scores for each of the four classes. The text and vision encoders are both kept frozen throughout all of our experiments as we train the SGE to align its embedding space with those of text and vision using Symmetric Cross Entropy Loss [Wang et al., 2019]. To evaluate the effectiveness of this training regiment, we analyze the classi cation performance with and without training for modality alignment. We also experiment with the effect of pre-training the SGE before conducting the alignment training. As an additional pre-training baseline, we use a pre-trained model provided by Malawade et al. [2022], which was trained on a synthetic traf c dataset generated using the CARLA traf c simulator [Dosovitskiy et al., 2017]. Finally, we perform tests to evaluate the performance of different hyperparameter settings (batch size, training time) for the alignment training.

D. Implementation Details

We train the SGE for 200 epochs on the default hyperparameter settings as speci ed by Yu et al. [2020]. For the SGE pre-training experiments, we select the SGE with the lowest validation loss. We also experiment with the provided SGE pre-trained checkpoint that was trained on synthetic traf c scenes generated by CARLA. For the modality alignment training, we keep most of the default hyperparameter settings, varying batch size (32 to 128) and the number of epochs (8 to 20).

V. Results and Discussion

A. SGE Pre-training Results

As seen in Table I, pre-training the SGE on the DoTA dataset allows it to perform the classi cation task approximately 13 percentage points better than random classi cation, which would have an expected accuracy of 25%. This suggests that the spatial and temporal information encoded in the graph embeddings produced by the SGE contain useful information towards accident classi cation. Unsurprisingly, the model pre-trained on the synthetic CARLA dataset does not achieve as strong performance on our task as the model trained on DoTA. This is understandable, given that the former was trained to classify whether or not a traf c maneuver is deemed to be risky [Malawade et al., 2022] rather than to classify traf c accidents. Furthermore, the synthetic traf c scenes in the CARLA dataset used by the authors do not vary as much as those seen in DoTA, and do not contain examples from the four classes used for our task. However, by further training the CARLA model on the DoTA dataset, we are able to achieve similar results to training on DoTA alone (compare two rightmost columns of Table I). For the remaining experiments in the paper, when unspeci ed, we use the SGE pre-trained only on the DoTA dataset.

TABLE I: SGE Performance with varying pre-train data. Class-balanced accuracy scores are indicated with a (b).

Pre-training Data	CARLA	DoTA	CARLA+DoTA
Accuracy			
Train	-	62.72	64.09
Train (b)	-	64.78	61.42
Validation	-	42.49	41.91
Validation (b)	-	40.79	41.42
Test	17.32	37.88	36.03
Test (b)	26.33	38.05	37.81

TABLE II: Test Accuracy Comparison between three system settings. In the first setting, the SGE is not used and the inputs to the system are only from the video and text modalities (the SGG and SGE are unused and stage III-C from the pipeline is skipped). In the second, the SGE is used and aligned with the vision and text encoders before running classification (full architecture is used, but SGE pre-training done in stage III-B may be skipped). In the third, the SGE is used but not aligned with the other encoders (stage III-C is skipped).

Train Setting	No SGE	SGE with Alignment			SGE without Alignment
Pre-training data	None	None*	CARLA	DoTA	DoTA
Test accuracy	57.04	54.97	52.66	58.66	57.97
Test accuracy (b)	53.22	53.53	53.97	56.97	57.74
Column label	1	2	3	4	5

B. SGE with Alignment Results

The primary classification results after training for alignment can be seen in Table II. In Column 1, we verify the baseline accuracy of the model when the scene graph modality is not used at all. Looking at the balanced accuracy score, it is interesting to note that the performance under this setting is only slightly under that of the aligned model when no pre-training data is used or if only the CARLA dataset is used (Columns 2, 3). However, pre-training the model on the DoTA dataset (Column 4) shows a further boost in performance by about 3%. This boost suggests that the SG embeddings may be encoding some information about the traffic scenes that cannot be fully captured by either the language or vision modalities.

1) Pre-trained SGE with Alignment: Regarding the case when the model is pre-trained on the DoTA dataset, the model actually has a slightly better performance without undergoing the alignment training (Column 5) versus with alignment (Column 4). This seems to suggest that the model is not learning further useful information during alignment training. One possible explanation for this is that the data domains of the frozen encoders are not similar enough to DoTA for any positive transfer of data domain knowledge to occur [Zhang et al., 2023]. Another possibility has to do with the alignment training procedure, which is based on CLIP's training algorithm and relies heavily on using a large batch size at each step [Radford et al., 2021]. This is because the algorithm performs contrastive training on each of the three pairs of embedding spaces (vision-graph, text-graph, and vision-text) for each batch of traffic accident examples. However, given the limited size of the DoTA dataset and compute constraints, our experiments in Table II use a batch size of 32 and are limited to 10 epochs of training. With a larger batch size (and possibly increasing the number of training epochs), it is possible that the hypothesized trend of higher accuracy when training for alignment would be present. To this end, we show some additional results in Table III, which indicate that both increasing batch size and the number of epochs give similar results to the "SGE without Alignment" case shown in Table II (Column 5) rather than worse results (Columns 2-4 of Table II). It is possible that further training of the larger batch sizes would have improved these results beyond the values in Column 5.

TABLE III: Pre-training on DoTA with different configurations. "e" is number of epochs, "bs" is batch size. Column 4 in Table II is the same as Column 1 in this table.

Pre-training on DoTA & SGE with Alignment				
	10e, bs32	20e, bs32	20e, bs64	10e, bs128
Test accuracy	58.66	58.2	57.51	60.51
Test accuracy (b)	56.97	57.77	57.53	57.72
Column label	1	2	3	4

2) No Pre-trained SGE with Alignment: Another surprising point that arises from the results presented in Table II is that there is very little improvement when training the SGE for alignment from scratch (Column 2) over the baseline "No SGE" case (Column 1). Similar to the case where we pre-trained on DoTA, it is once again possible that further increasing the batch size could be the solution. This is suggested by the results in Table IV, where larger batch sizes begin to show a significant increase in performance over the baseline.

TABLE IV: Training for alignment from scratch with different configurations. Column 2 in Table II is the same as Column 1 in this table.

No Pre-training & SGE with Alignment			
	10e, bs32*	10e, bs64	10e, bs128
Test accuracy	54.97	57.51	59.35
Test accuracy (b)	53.53	53.94	56.00
Column label	1	2	3

VI. Conclusion and Future Work

In this work, we have shown that encoding traffic information in the form of a scene graph is beneficial towards the goal of accident classification. This is made clear by the pre-training encoder results which show the SGE's ability to beat a random classifier at this task. It was further illustrated that the scene graph information can serve to enhance the performance of a vision-language classifier by fusing information from all three modalities. Finally, although we were not able to show an improved score for the classifier after alignment, it should be noted that alignment does not have a negative effect on performance, and we demonstrate that further training and fine-tuning may improve the score beyond the unaligned case.

There are several areas where future work can be done in this task. Firstly, although three modalities are used, little exploration has been conducted on the language modality. In principle, captions for each video should be derived from the video itself, as the SGs are. This would allow for the captions to be fed into the model to ne-tune the classi er head and run inference with a greater signal coming from the language modality. Next, although they provide some bene for the model, the generated SGs have a relatively limited structure: only categorical distance relations between a vehicle and objects in its surroundings are generated, along with mappings to a xed set of three traf c lanes (left, middle, and right). Enabling a semantic extension of the generated SGs may enhance the signal obtained from this modality. Similarly, by modeling relations between all objects in the scene, this pipeline can be expanded for non-ego use as well, and perhaps can be used on more classes in the DoTA dataset or similar use cases. Additionally, the MRGCN-based architecture of the SGE was taken directly from Yu et al. [2020], but perhaps it can be further improved by modifying either its spatial or temporal modeling components. This work has demonstrated that further ne-tuning and alignment training shows promise to improve results. Experiments with different modality fusion methods as well as classi cation heads may also lead to improvements.

References

Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. CoRR, abs/2201.11460, 2022. URL https://arxiv.org/abs/2201.11460.

Paolo Emmanuel Ilario Dimasi. Scene graph generation in autonomous driving: a neuro-symbolic approach. Master's thesis, Politecnico di Torino, 2023. URL http://webthesis.biblio.polito.it/id/eprint/29354.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017.

Jonathan Francis, Bingqing Chen, Weiran Yao, Eric Nyberg, and Jean Oh. Distribution-aware goal prediction and conformant model-based planning for safe autonomous driving. arXiv preprint arXiv:2212.08729, 2022.

Jing Gao, Peng Li, Zhikui Chen, and Jianing Zhang. A Survey on Deep Learning for Multimodal Data Fusion. Neural Computation, 32(5):829–864, 05 2020. ISSN 0899-7667. doi: 10.1162/neco_a_01273. URL https://doi.org/10.1162/neco_a_01273.

Y. Guo, F. Yin, X. Li, X. Yan, T. Xue, S. Mei, and C. Liu. Visual traf c knowledge graph generation from scene images. In 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pages 21547–21556, Los Alamitos, CA, USA, oct 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.01975. URL https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.01975.

Yufeng Huang, Jiji Tang, Zhuo Chen, Rongsheng Zhang, Xinfeng Zhang, Weijie Chen, Zeng Zhao, Zhou Zhao, Tangjie Lv, Zhipeng Hu, and Wen Zhang. Structure-clip: Towards scene graph knowledge to enhance multi-modal structured representations, 2023.

Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as composition of spatio-temporal scene graphs. CoRR, abs/1912.06992, 2019. URL http://arxiv.org/abs/1912.06992.

Leszek Kaliciak, Hans Myrhaug, Ayse Goker, and Dawei Song. On the duality of speci c early and late fusion strategies. In 17th International Conference on Information Fusion (FUSION), pages 1–8, 2014.

Mahmoud Khademi and Oliver Schulte. Deep generative probabilistic graph neural networks for scene graph generation. Proceedings of the AAAI Conference on Arti cial Intelligence, 34(07):11237–11245, Apr. 2020. doi: 10.1609/aaai.v34i07.6783. URL https://ojs.aaai.org/index.php/AAAI/article/view/6783.

D. Kiela, E. Grave, A. Joulin, and T. Mikolov. Ef cient large-scale multi-modal classi cation, 2018.

Sebastian Koch, Pedro Hermosilla, Narunas Vaskevicius, Mirco Colosi, and Timo Ropinski. Lang3dsg: Language-based contrastive pre-training for 3d scene graph prediction, 2023.

Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering, 2019. URL https://arxiv.org/abs/1903.12314.

Arnav Vaibhav Malawade, Shih-Yuan Yu, Brandon Hsu, Harsimrat Kaeley, Anurag Karra, and Mohammad Abdullah Al Faruque. roadscene2vec: A tool for extracting and embedding road scene-graphs. CoRR, abs/2109.01183, 2021. URL https://arxiv.org/abs/2109.01183.

Arnav Vaibhav Malawade, Shih-Yuan Yu, Brandon Hsu, Deepan Muthirayan, Pramod P. Khargonekar, and Mohammad Abdullah Al Faruque. Spatiotemporal scene-graph embedding for autonomous vehicle collision prediction. IEEE Internet of Things Journal, 9(12):9379–9388, 2022. doi: 10.1109/JIOT.2022.3141044.

Sayak Nag, Kyle Min, Subarna Tripathi, and Amit K. Roy Chowdhury. Unbiased scene graph generation in videos, 2023.

Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition, 2022.

Maciej Paw owski, Anna W ojewska, and Sylwia Sysko-Roma czuk. Effective techniques for multimodal data fusion: A comparative analysis. Sensors, 23(5), 2023. ISSN 1424-8220. doi: 10.3390/s23052381. URL https://www.mdpi.com/1424-8220/23/5/2381.

Ehsan Qasemi, Jonathan M Francis, and Alessandro Oltramari. Traf c-domain video question answering with automatic captioning. arXiv preprint arXiv:2307.09636, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from