Supplementary Material for RS2G: Data-Driven Scene-Graph Extraction and Embedding for Robust Autonomous Perception and Scenario Understanding

1. Additional Ablation Studies

1.1. Downstream Component Analysis

We analyze each component of our downstream model as demonstrated in TABLE 1. As elaborated in the main submission, our downstream model consists of a spatial model (either MLP or MR-GCN) and a temporal model (either "mean", i.e., a linear layer, or LSTM). Here we compare the results of using different downstream models for the SOTA rule-based graph extraction and RS2G with the edge encoder based on the Transformer. In general, regardless of the downstream model, RS2G (Transformer) significantly outperforms the rule-based graph learning (GL) method in terms of accuracy, MCC, and AUC, indicating our proposed data-driven graph extraction approach provides more expressive and dynamic graph representations. Additionally, using MR-GCN for the spatial model provides considerably better performance than using MLP for both the rule-based graph extraction method and RS2G, indicating explicitly modeling relations among road users effectively enhances model performance for risk assessment.

For the rule-based graph extraction method, using LSTM as the temporal model provides better accuracy than using a linear layer ("mean"), as LSTM offers better performance in modeling temporal patterns in graph embeddings. However, it is noteworthy that for RS2G (Transformer), using 'mean' as the temporal model provides considerably better performance than using LSTM. This indicates the extraordinary capability and robustness of our Transformer edge encoder in extracting scene graphs, as even a simple mean operation can achieve adequate performance. In particular, as autonomous vehicles are embedded devices with limited resources, our model also provides a more resource-efficient solution for scenario understanding and risk assessment. On the other hand, using a more complicated model such as LSTM can potentially introduce unnecessary complexity, causing overfitting and degrading model performance.

Graph Extraction	Spatial Model	Temporal Model	Accuracy	MCC	AUC
Rule-Based	MLP	mean	52.15%	0.0000	0.4973
Rule-Based	MLP	LSTM	62.90%	0.2741	0.6811
Rule-Based	MRGCN	mean	63.44%	0.2696	0.6867
Rule-Based	MRGCN	LSTM	**75.27%**	**0.5197**	**0.8248**
RS2G	MLP	mean	81.74%	0.1857	0.9228
RS2G	MLP	LSTM	81.45%	0.402	0.9472
RS2G	MRGCN	mean	**87.80%**	**0.5403**	**0.9468**
RS2G	MRGCN	LSTM	84.15%	0.402	0.9362

Table 1. Analysis of each component of the downstream model. Models are trained and evaluated on *271-dash*. We demonstrate the impacts of different spatial and temporal models using rule-based graph extraction [3] and RS2G (Transformer).

1.2. Cosine Relation Similarity

We compare the cosine similarity between the data-driven relations learned by RS2G and the set of relations extracted by the SOTA rule-based graph extraction method [3] for the dataset *1043-carla* and *620-dash*, and present our result as heatmaps demonstrated in Figure 1. Cosine similarity is a widely-used metric for comparing sparse vectors, and is often employed in the computation of document similarity using term-frequency vectors [1]. Specifically, we employ cosine similarity to compare the adjacency matrices extracted by the rule-based graph extraction method with the data-driven adjacency matrices produced by RS2G. As the rule-based graph extraction method defines edges by physical meaning, e.g., left, very close, and in front of, this evaluation determines if the relations established by our data-driven approach can reflect similar information. Here our extracted graphs are 3-dimensional binary adjacency matrices; specifically, an $n \times n$ matrix for each relation $r \in \mathcal{R}$, where n denotes the number of nodes. Therefore, for each of the learned relations of our RS2G, we calculate the cosine similarity of its adjacency matrices with those relations extracted by the rule-based method, and then average across all the graphs in the dataset.

The similarity in relations highlighted across datasets supports the fact that RS2G can effectively transfer knowledge from training domains to real-world scenarios. We analyze our results from the following three perspectives:

- ***Edges with Varying Degree of Importance:*** As shown in Figure 1, it is notable that some rule-based relations, such as "isIn," which reflects the direction information

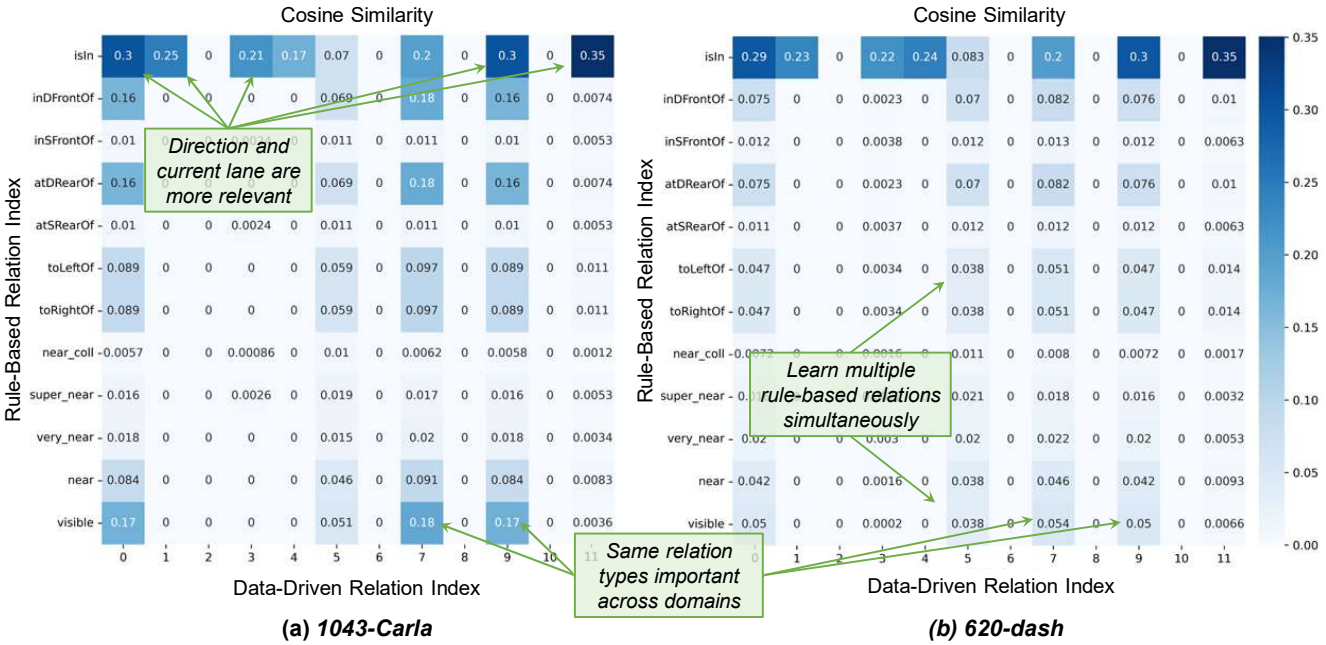

Figure 1. Cosine similarity between relations learned by RS2G (2D MLP) and the set of rule-based relations used in [3] for the synthetic *1043-carla* dataset and the real-world *620-dash* dataset.

and the current lane of the vehicle, exhibit high similarity scores with multiple data-driven relations. This observation suggests that the "isIn" relation plays a more critical role than other relations in risk assessment. In contrast to the rule-based graph extraction model, where all relation types carry equal weight, our proposed data-driven graph extraction approach exhibits a remarkable ability to discern and assign varying degrees of importance to each relation type. This capability enables a more nuanced representation of the underlying data, and thereby significantly enhances the adaptability of our model in capturing intricate relations among road users in real-world scenarios.

- ***Transfer learning Capability:*** We observe that the relations deemed as important by RS2G in *1043 Carla* are also considered relevant in *620 dash*. In other words, relations learned from the simulation dataset, e.g., *1043 Carla*, will remain useful in handling more complex real-world scenarios in *620 dash*. For instance, in Fig. 1, the Relation 7 and Relation 9 of our data-driven GL model exhibit higher importance than other relations in both datasets. Additionally, the differences in the relative weights of these relations across datasets demonstrate the adaptability of our model. Specifically, RS2G can adapt to different datasets by adjusting the density of different relation types depending on the data, thereby providing enhanced performance for transfer learning.

- ***Comprehensive Representations of Relations:*** It is noteworthy that a relation learned by RS2G can include information from multiple rule-based relations, and thereby provide more expressive and complete graph representations. For instance, Relation 5 from RS2G exhibits some degrees of similarity to all the rule-based relations. This evaluation further shows that our data-driven graph extraction method can effectively enhance the quality of graph representations for road scenes.

2. Discussion

In this section, we discuss the practicality of deploying RS2G in a real-world autonomous system, the limitations of our research scope, and potential future research directions.

2.1. Practicality

Fundamentally, RS2G leverages existing deep-learning and graph-learning libraries, e.g., `PyTorch`, `scikit-learn`, and `NetworkX`, for implementation and execution; thus, RS2G is readily compatible with standard procedures for training, validation, and model compilation that are typically employed in autonomous computing platforms. The complexity of deploying RS2G is approximately equivalent to that of deploying a rule-based graph model, with the primary difference being a few additional layers for node and edge encoding. Additionally, the input and output requirements RS2G well align with the standard components found in most autonomous driving and Advanced Driver Assistance Systems (ADAS) pipelines, e.g., camera inputs, object detection models, and

ADAS control systems. In terms of practical utility, the output of RS2G can integrate with the ADAS system and offer valuable insights to inform critical tasks such as driver control handoff, emergency braking, and dynamic driving profile adjustments.

2.2. Limitations and Future Work

RS2G has demonstrated considerably improved generalization capabilities compared to the rule-based graph extraction method. However, we believe the following three areas are worth further exploration:

- ***Node Preprocessing:*** For our proposed RS2G, we combine Kullback-Liebler (KL) divergence and the Transformer to construct an effective data-driven *edge encoder* and deliver expressive graph representations. Nevertheless, the model performance can potentially be further improved by introducing a learned self-supervision component between the node and edge encoder. Specifically, the self-supervision module can be added between inputs and the edge encoder as a pre-processing step to further enhance the performance of our edge encoder. Along with our data-driven edge encoder, this self-supervision component can potentially model an invertible function mapping from the inputs to the outputs and produce a more general graph representation.

- ***Edge Encoder:*** In this work, we exhaustively studied the performance of various edge encoders, including 1D MLP, 2D MLP, and the Transformer. However, it is likely that some more sophisticated deep learning models can better capture intricate relations between nodes and provide more expressive graph representations. Additionally, it is also possible to integrate rule-based graphs with learned graphs, e.g., by combining their adjacency matrices, to leverage the benefits of both methods.

- ***Applications:*** In application domains, this paper primarily studies subjective risk assessment for AVs. However, the operational landscape for AVs demands a multifaceted array of tasks encompassing perception, planning, and safe maneuvering. Tasks such as localization, motion prediction, and path planning inherently require a nuanced comprehension of the semantic scene, and there is substantial evidence to suggest that the integration of graph-based modeling approaches can notably enhance performance in these areas [2]. Our data-driven graph extraction method can potentially benefit these applications by providing expressive and dynamic graph representations.

References

[1] Jiawei Han, Micheline Kamber, Jian Pei, et al. Getting to know your data. In *Data mining*, volume 2, pages 39–82. Morgan Kaufmann Boston, MA, USA, 2012. 1

[2] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pages 683–700. Springer, 2020. 3

[3] 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*, 2021. 1, 2