2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

# Understanding Dynamic Scenes using Graph Convolution Networks

Sravan Mylavarapu*[1], Mahtab Sandhu*[2], Priyesh Vijayan[3],
K Madhava Krishna[2] , Balaraman Ravindran[4], Anoop Namboodiri[1]

***Abstract*— We present a novel Multi-Relational Graph Convolutional Network (MRGCN) based framework to model on-road vehicle behaviors from a sequence of temporally ordered frames as grabbed by a moving monocular camera. The input to MRGCN is a multi-relational graph where the graph's nodes represent the active and passive agents/objects in the scene, and the bidirectional edges that connect every pair of nodes are encodings of their Spatio-temporal relations.**

**We show that this proposed explicit encoding and usage of an intermediate spatio-temporal interaction graph to be well suited for our tasks over learning end-end directly on a set of temporally ordered spatial relations. We also propose an attention mechanism for MRGCNs that conditioned on the scene dynamically scores the importance of information from different interaction types.**

**The proposed framework achieves significant performance gain over prior methods on vehicle-behavior classification tasks on four datasets. We also show a seamless transfer of learning to multiple datasets without resorting to fine-tuning. Such behavior prediction methods find immediate relevance in a variety of navigation tasks such as behavior planning, state estimation, and applications relating to the detection of traffic violations over videos.**

## I. INTRODUCTION

We consider dynamic traffic scenes consisting of potentially active participants/agents such as cars and other vehicles that constitute the traffic and passive objects such as lane markings and poles (see example in Fig. 1). In this work, we propose a framework to model the behavior of each such active agents by analyzing the Spatio-temporal evolution of their relations with other active and passive objects in the scene. By relation, we refer to the spatial relations an agent/object possesses and enjoys with other agents/objects, such as between the vehicle and lane markings, as shown in Fig. 1(c).

Here, we model both objects and agents, and thus for convenience, we commonly refer them as objects and specifically as agents when referring to active vehicles. The evolution of the spatial relationship between all pairs of objects in a scene is essential in understanding their behaviors. To this end, we propose an Interaction graph that models different agents and objects in the scene as nodes. This graph captures the Spatio-temporal evolution of relations between all-pair of objects in the scene with appropriate bi-directional asymmetric edges with annotations reflecting their evolution (see Fig. 1(c)).

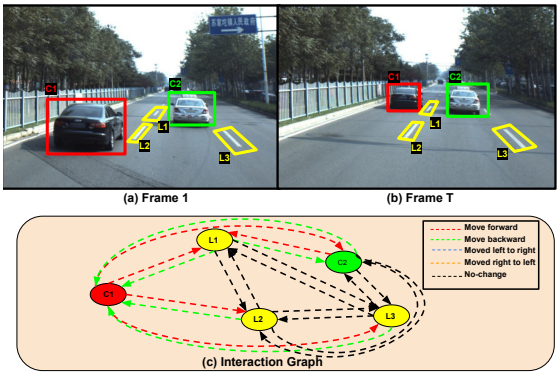


Fig. 1. The figures (a) and (b) show two cars and three lane-markings from two different time frames. The evolution of the whole scene is captured in the Interaction graph (c). Our proposed model infers over such Interaction graphs to classify objects' behaviors. Here, the car on the left is moving ahead of lane markings and the car on the right. With the Interaction graph (c), our model can predict that the car on the left is overtaking the right-side car.

The dynamic traffic scene modeled as an Interaction graph is then inputted to a Multi-Relational Graph Convolutional Network (MRGCN), which outputs the overall behavior exhibited by all the agents and objects in the scene. While the MRGCN maps the input graph to an output behavior for every graph node, we are only interested in active and not the passive objects. Hitherto by behavior, we denote the overall behavior of an active agent in the scene. For example, in Figure 1, the behavior of the car on the left is *Overtaking*, and the car on the right is *moving ahead*.

The choice of Graph Convolutional Networks (GCN) [1] and the use of its Multi-Relational Variant as a choice for this problem stems from the recent success of such models in learning over data that does not present itself in a regular grid like structure and yet can be modeled as a graph such as in social and biological networks. Since a road scene can also be represented as a graph with nodes sharing multiple relations with other nodes, a GCN based model is apt for inferring overall node (object) behavior from a graph of interconnected relationships.

*The decomposition of a dynamic on-road scene into its associated Interaction graph and the classification of the agent behavior by the MRGCN supervised over labels that are human-understandable (Lane Change, Overtake, etc.) form the main thesis of this effort.* Such behavior classification of agents in the scene finds immediate utility in downstream modules and applications. Recent research showcase results that understanding on-road vehicle behavior leads to better behavior planners for the ego vehicle [2]. In [2], belief states

[1]Authors are with the Center for Visual Information Technology, KCIS, IIIT Hyderabad
[2]Authors are with the Robotics Research Center, KCIS, IIIT Hyderabad
[3] School of Computer Science, McGill University and Mila
[4] Dept. of CSE and Robert Bosch Center for Data Science and AI, IIT Madras

over driver intents of the other vehicles where the intents take the labels "Left Lane Change", "Right Lane Change", "Lane Keep" are used for a high-level POMDP based behavior planner for the ego vehicle. For example, the understanding that a car on the ego-vehicle's right lane is executing a lane change behavior into the current lane of the ego-vehicle can activate its "Change to Right Lane" behavior operation option for path planning. Similarly, modeling an agent's behavior such as a car in a parked state can make the ego vehicle use feature descriptors of the parked car to update its state accordingly, which would not be possible if the active object was engaged in any other behavior. Understanding on-road vehicle behaviors also lend itself to very pertinent applications such as detecting and classifying traffic scene violations such as "Overtaking Prohibited", "Lane Change Prohibited."

Our contributions are as follows:

1) We propose a novel yet simple scheme for spatial behavior encoding from a sequence of single-camera observations using straight forward projective geometry techniques. This method of encoding spatial behaviors for agents is better than previous efforts that have used end-to-end learning of spatial behaviors [3].

2) We demonstrate the aptness of the proposed pipeline that directly encodes Spatio-temporal behaviors as an intermediate representation into the scene graph $G$, followed by the MRGCN based behavioral classifier. We do this by comparing with two previous methods [3], [4] that are devoid of such intermediate Spatio-temporal representations but activate the behavior classifier on per frame spatial representations sequenced temporally. Specifically we tabulate significant performance gain of at-least $25\%$ on an average vis a vis [4] and $10\%$ over [3] on a variety of datasets collected in various parts of the world [5], [6], [7] and our own native dataset (refer Table IV-B).

3) We signify through a label deficient setup, the need for a neural-attention component that integrates with the MRGCN and further boosts its performance to nearly perfect predictions, as seen in Tables III and V. Critically incorporating the attention function leads to high performance even in a limited training set, which the MRGCN without attention function cannot replicate.

4) We also show seamless transfer of learning without further need to fine-tune across various combinations of datasets for the train and test split. Here again, we show better transfer capability of the current model vis a vis prior work, as shown in Table IV.

## II. Related Work

*1) Vehicle Behavior and Scene Understanding:* The problem of on-road vehicle scene understanding is an important problem within autonomous driving. Most earlier works relied on multiple sensor-based data to solve this task. Rule-based [8] and probabilistic modeling [9], [10], [11], [12], [13], [14], [15] where the goto approaches for classification of driver behavior with sensor data. Also, many of these works [11], [14] were concerned only with predicting future trajectories rather than classification. Herein, we chose a simpler and challenging set up to understand the scene and predict vehicle behaviors with observations from a single-camera in this work. While there are few works based on a single-camera data feed, they only focus on ego-centered predictions [16]. Here we focus on classifying other vehicles' behavior from an ego-vehicle perspective. Learning other vehicle behaviors can be helpful in behavior planning, state estimation, and applications relating to the detection of traffic violations over videos

*2) Graph based reasoning:* Graphs are a popular choice of data structures to model numerous irregular domains. With the recent advent of Graph Convolutional Networks (GCNs) [1] that can obtain relevant node-level features for graphs, there is a widespread adaption of graph-based modeling of numerous computer vision problems such as in situation-recognition tasks [17]. [18] encodes object-centric relations in an image using a GCNS to learn object-centric policies for autonomous driving. [4] and [3] models objects in a video as Spatio-temporal graphs to make predictions of Spatio-temporal nature. It is common to model the temporal context of objects with recurrent neural nets and spatial context with a graph-based neural net.

In this work, we focus on the task of on-road vehicle behavior and show that a proposed intermediate representation, called an Interaction graph, can yield better performance over working with a raw set of spatial graphs as done traditionally. This also portrays current models' incapacity to learn end-end and derive such useful features as with the Interaction graph from the raw spatial graphs. The closest works to ours are [16] and [19]. They generate an affinity graph that captures actor-objects relationships. A simple GCN is then used to reason over this graph to classify ego-car action and not other vehicles. In our work, we use a richer multi-relational graph and a corresponding multi-relational GCN to work on the same. Further, we propose an attention based model that can leverage different relation types depending on the scene context.

## III. Proposed Methodology

Dynamic scene understanding requires well modeling of the different Spatio-temporal relations that may exist between various active objects in a scene. Towards this goal, we propose a pipeline that first computes a time-based ordered set of spatial relations for each object in the video scene. Secondly, it generates a multi-relational interaction graph representing the temporal evolution of the spatial relations between entities obtained from the previous step. Finally, it leverages a graph-based behavior learning model to predict behaviors of vehicles in the scene. Fig 2 provides an overview of our proposed framework.

Our proposed pipeline leverages and improves the data modeling pipeline introduced in [3] (MRGCN-LSTM). Our pipeline's performance gains primarily stem from two of our contributions: (i) the *interaction graph* that provides useful and explicit temporal evolution information of spatial

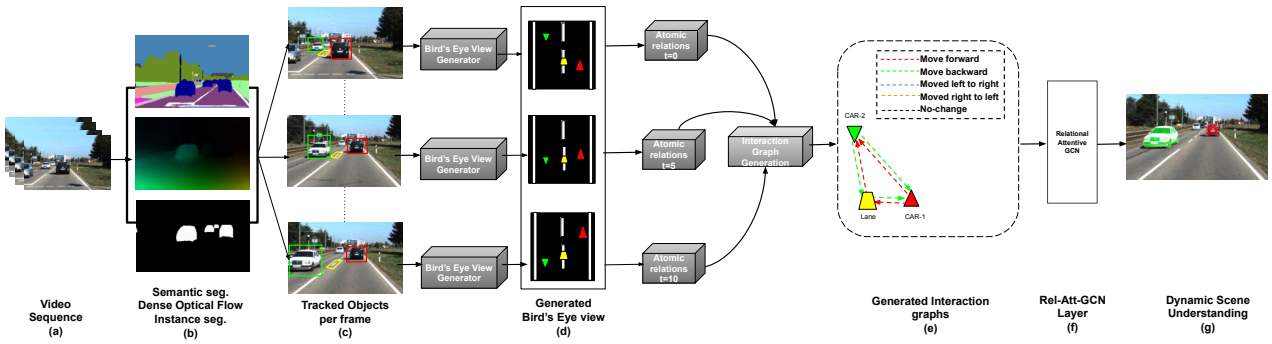


Fig. 2. **Overall pipeline of our framework:** The input (a) to the pipeline is monocular image frames. Various object tracking pipelines are used to detect and track objects, as shown in (b). (c) denotes the tracked objects. Tracklets for each object are projected to 3D space, Bird's eye-view at each time step, as shown in (d). Spatial relations from Bird's eye view are used to generate Interaction graphs (e). This graph is passed through a Rel-Att-RGCN (f) to classify objects in the scene as shown in (g).

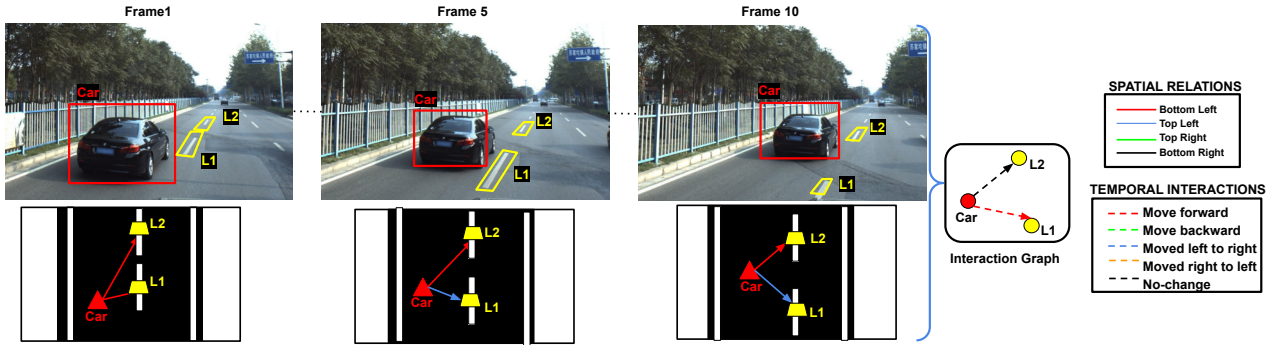


Fig. 3. **Temporal Interaction Graph Generation:** The top row contains image frames at t=0, t=5, t=10 time step with tracked objects. The bottom row shows the corresponding bird's eye view. In the scene, a car and two lane-markings are tracked at each time step. The thick red, blue edges between car, L1 in first, and the final frame denote their spatial relations. The car moves forward with respect to L1 and has no relational change with L2, as shown with thick edges in the Interaction graph. Temporal relations are represented with dotted lines, while spatial relations are represented with thick edges. Corresponding color coding is shown in legends.

relations. (ii) the proposed Multi-Relational Graph Convolutional Network (MRGCN) with our novel *multi-head relation attention function*, which we name Relation-Attentive-GCN (Rel-Att-GCN).

### A. Spatial Graph Generation

For dynamic scene understanding, we need to identify different objects in the scene and determine the atomic spatial relationships between them at each time-step. This phase of the pipeline closely follows the spatial scene graph generation step of [3]. refer to the same for mode details.

*1) Object Detection and Tracking:* Different *vehicles* in the video frames are detected and tracked through *instance segmentation* [20] and *per-pixel optical flow* [21] respectively. The instance-level segmentation of an object obtained from MaskRCNN are projected to the next frame and are associated with the highest overlapping instance using optical flows. Apart from tracking vehicles in the scene, static objects such as *lane markings* are also tracked with *semantic segmentation* [22] to better understand changing relations among static and non-static objects. [3] has shown that performance improvement can be obtained by leveraging a higher number of static objects in the scene.

*2) Bird's Eye View:* The tracklets of objects obtained in the image space are re-oriented in the Bird's eye view (Top View) by projecting the image coordinates into 3D coordinates as described in [23]. This reorientation facilitates determining spatial relations between different entities. Each object is assigned a reference point to account for the difference in heights. The reference is at the center for lane markings, and for vehicles, it is the point adjacent to the road.

*3) Spatial relations:* At all $T$ frames of the video, the spatial relations between different entities are determined using their 3D positional information in the Bird's eye view. Specifically, the spatial relations are the four quadrants, $\{$*top left, top right, bottom left, and bottom right*$\}$. For a subject entity, $i$, its spatial relation with an object entity, $j$ at time-step, $t$ is denoted as $S^t_{i,j}$.

### B. Temporal Interaction Graph Generation

Modeling object interactions as a time-based ordered set of spatial graphs is a popular approach in many Spatio-temporal problems. such as [24], [25], [26], [27] where spatial graphs are constructed over object interactions to classify actions in a video stream. [4], [3] use a similar approach for predicting on road object behaviors. We found that it is harder for models learned with such data to learn some of the simple temporal-evolution behaviors needed for the end task, specifically in our problem of interest. In our problem of focus,

behavior prediction, it is important for the model to learn the nature of some simple temporal evolution of interaction between entities such as *move-forward, move-backward, moved left to the right, moved right to the left, no-change*. However, we found that explicitly modeling such information was highly beneficial. A simple rule-based model with such interaction information outperformed learned models on the primitive information at the level of spatial relations. Having motivated by this insight, we propose a way to define an Interaction graph with temporal evolution information and a new model that can benefit from such information.

The Interaction graph summarizes the temporal evolution of spatial relations from $T$ frames into a *single multi-relational graph with temporal relations*, $R_d$ = {*move-forward, move-backward, moved left-to-right, moved right-to-left, no-change*}. The edge, $E_{i,j} \in R_d$, denotes the temporal relation between subject entity, $i$ and object entity, $j$. $E_{i,j}$ is computed by deterministic rule based pipeline, that takes in their spatial relations over $T$ frames, i.e, $\{S^1_{i,j}, S^2_{i,j}, ..., S^T_{i,j}\}$.

For example, an object entity, $j$ that is initially in the *bottom-left* quadrant with respect to subject entity $i$, changes it's atomic spatial relation to *top-left* with respect to $i$ at some time $t$, will have its temporal relation $E_{i,j}$ as *move-forward*. Similarly, an object entity, $j$ that is initially in *bottom-left* quadrant changes to *bottom-right* quadrant with respect to subject $i$ at some time $t$, will have temporal relation $E_{i,j}$ as *moved left-to-right*. Since these temporal relation annotated edges have a proper direction semantics, we can't treat the graph is undirected. Thus, we also introduce inverse edges for complementary relations suchs as *moved forward and backward* and *moved left-to-right and moved right-to-left*. An overview of the temporal interaction graph generation is presented in Fig. 3.

### C. Behavior Prediction Model

We propose a Multi-relational Graph Convolution Network (MRGCN) with a relation-attention module that conditioned on the scene, automatically learns relevant information from different temporal relations necessary to predict vehicle behaviors.

*1) Multi-Relational Graph Convolution Networks:* Recently Graph convolution Networks [1] has become the popular choice to model graph-structured data. We model our task of maneuver prediction by using a variant of Graph Convolutional Networks, Multi-Relational Graph Convolutional Networks (MRGCN) [28] originally proposed for knowledge graphs with multiple relation types. MRGCN is composed of multiple graph convolutional layers, one for each relation between nodes. A Graph Convolution operation for a relation $r$ here is a simple neighborhood-based information aggregation function. In the MRGCN, information obtained from convolving over different relations is combined by summation.

Let us formally define the temporal Interaction Graph as $G = (V, E)$ with vertex set $V$ and edge set, $E$, where $E_{i,j} \in R_d$ is an edge between node $i$ and $j$. The $i^{th}$ node feature obtained from a graph convolution over relation, $r$ in $l^{th}$ layer is defined as follows:

$$h^l_r[i] = \sum_{j \in \mathcal{N}_r[i]} \frac{1}{c_r[i]} W^l_r h^{l-1}[j] \tag{1}$$

where, $\mathcal{N}_r[i]$ denotes set of neighbour nodes for $v_i$ under relation $r$, $\mathcal{N}_r[i] = \{j \in V \mid E_{j,i} = r\}$ and $c_r[i] = |\mathcal{N}_r[i]|$ is a normalization factor. Here, $W^l_r \in \mathcal{R}^{d' * d}$ is the weights associated with relation $r$ in the $l^{th}$ layer of MR-GCN; $d', d$ are dimensions of $(l-1)^{th}$ and $l^{th}$ layers of MRGCN.

Neighborhood information aggregated from all the relations are then combined by a simple summation to obtain the node representation as follows:

$$h^l[i] = ReLU(W^l_s h^{l-1}[i] + \sum_{r \in R_d} h^l_r[i]) \tag{2}$$

where, the first terms correspond to the node information (self-loop) and $W_s \in \mathcal{R}^{d' * d}$ is the weight associated with self-loop. To account for the nature of the entity (active or passive), we learn entity embeddings, $\mathcal{E}_i \in \mathbb{R}^{|O| * d}$ for node $v_i$, where $O = \{Vehicles,\ Lane\ Markings\}$. The input to the first layer of the MRGCN, $h^0[i]$, is the embedding $\mathcal{E}_i$ based on type of node $i$.

*2) Relation-Attention MRGCN (Rel-Att-GCN):* The MRGCN defined in Eqn: 2 treats information from all the relations equally, which might be a sub-optimal choice to learn discriminative features for certain classes. Motivated by this, we propose a simple attention mechanism that scores the node information's importance along with individual neighborhood information from each relation.

The attention scores, $\alpha$ are computed by concatenating the information from the node ($h^{l-1}$) and its relational neighbors ($h^l_r$) and transforming it with a linear layer to predict scores for each component. The predicted scores are softmax normalized. The attention scores are computed as defined below.

$$\alpha^l[i] = softmax([h^{l-1}[i] \parallel h^l_1[i] \parallel h^l_2[i] ... \parallel h^l_{|R_d|}]W^l_u) \tag{3}$$

where, $\parallel$ represents concatenation and $\alpha^l[i] \in \mathcal{R}^{|R_d+1|}$ with $W^l_u$ being the linear attention layer weights. These probabilities depict the importance of a specific relation conditioned on that node and its neighborhood.

The attention scores are used to scale the node and neighbor information accordingly to obtain the node representation as follows.

$$h^l[i] = ReLU(\alpha^l_{node}[i] h^{l-1}[i] + \sum_{r \in R_d} \alpha^l_r[i] h^l_r[i]) \tag{4}$$

where, $\alpha^l_{node}$ is the attention score for self-loop. The node representation obtained at the last layer is used to predict labels, and the model is trained by minimizing the cross-entropy loss.

## IV. Experiment and Analysis

### A. Dataset

Numerous datasets have been released in the interest of solving problems related to autonomous driving. We choose

four datasets for evaluating our framework, of which three are publicly available: ApolloScapes [6], KITTI [29], Honda Driving dataset [7] and one is a proprietary Indian dataset. These datasets provide hours of driving data with monocular image feed in various driving conditions. We use the same dataset Train/Test/Val splits from [3] for Apollo Scape, KITTI, and Indian dataset and extend the setup to Honda dataset and manually annotated accordingly for our task.

1) Apollo Dataset: We choose Apollo-scapes as our primary dataset as it contains a large number of driving scenarios that are of interest. It includes vehicles depicting overtake and lane-change behaviors. The dataset consists of image feed collected from urban areas and contains various objects such as Cars, Buses, etc. The final dataset used here contains a total of 4K frames with multiple behaviors. 2) KITTI Dataset: It consists of images collected majorly from highways and wide open-roads, unlike Apollo-scapes. We select 700 frames from *Tracking Sequences* 4, 5, and 10 for our purpose. These chosen sequences contained a variety of driving behaviors compared to the rest. 3) Honda Driving Dataset: This dataset consists of multiple datasets in itself. From among them, we choose the H3D dataset for our task as it contained lane change behaviors. H3D comprises driving in urban city conditions where lane changes are prominent. We excluded overtaking behavior here due to its fewer occurrences in the dataset. It consists of a total of 1.5K frames. 4) Indian Dataset Although the datasets mentioned above are widely used and include wide vehicle behaviors, they mostly contain standard vehicle frames only. To showcase models' transfer learning capabilities on less standard vehicles, we also use an Indian driving dataset that includes vehicles such as auto-rickshaw, trucks, tankers, etc. This dataset contains 600 frames.

Class labels: The vehicle behaviors predicted in these datasets are: (i) Moving Away from Us (MAU), (ii) Moving Towards Us (MTU), (iii) Parked (PRK), (iv) Lane Change from Left-right (LCL), (v) Lane Change from left-Right (LCR) and (vi) Overtake (OVT).

### B. Experimentation details

All the models, both learning and rule-based, use the same pre-processing steps to identify and track objects in the scene, as explained in section: III-A.1 for $T = 10$ time-steps (frames). The Spatial graphs at each time frame are constructed by considering a maximum of 10 vehicles in the scene nearest to the ego-vehicle for classifying them. Then, an *Interaction graph* is independently generated from the set of $T$ temporally ordered spatial graphs. The MRGCN used is identical in both models MRGCN and Rel-Att-GCN. Note that the input and output dimensions of attention are equal. We empirically found that using 3 layers of MRGCN with dimensions 64, 32, and 6 (number of classes) respectively works best for our task. In the case of *Rel-Att-GCN*, simple attention is applied over the output of MRGCN for each node individually with 2 heads. Outputs of the heads are concatenated across relations and projected back to the MRGCN layer's output dimension with a linear transformation. We found adding skip connection from every layer $l$ to $(l+2)^{th}$ layer to be beneficial.

| Graph used | Methods | Moving away (MAU) | Moving towards (MTU) | Parked | Lane change L-R (LCL) | Lane change R-L (LCR) | Overtake |
|---|---|---|---|---|---|---|---|
| Temporally ordered set of Spatial graphs | St-RNN [4] | 76 | 51 | 83 | 52 | 57 | 63 |
| | MRGCN-LSTM [3] | 85 | 89 | 94 | 84 | 86 | 72 |
| Spatio-Temporal Interaction graphs | Rule Based | 90 | **99** | **98** | 81 | 87 | **90** |
| | MRGCN [28], [3] | 94 | 95 | 94 | 97 | 93 | 86 |
| | **Rel-Att-GCN** | **95** | **99** | **98** | **97** | **97** | 89 |

TABLE I

VEHICLE BEHAVIOR PREDICTION ON APOLLO SCAPE DATASET.

The inference time for the models: MRGCN-LSTM, MRGCN, and Rel-Att-GCN averaged over 1K graphs are 0.02, 0.03, and 0.04 seconds respectively. Note that the latter two models inference time also includes the creation of the Interaction graph. All the training and testing was done on a single Nvidia Geforce Gtx 1080 GPU. More details regarding the implementation can be found on our project website [1].

### C. Baseline Comparisons

In Table: IV-B, we compare our models with Spatio-temporal approaches as well as a rule-based method on the Interaction graph. Table IV-B reports class-wise Recall scores of these methods. The results reported here in all the tables are averaged over 5 runs.

*1) Spatio-Temporal approaches:* To depict the importance of encoding Spatio-temporal information as an Interaction graph, we compare our model with Structural-RNN [4] and MRCGN-LSTM [3] that processes a time based ordered set of spatial graphs. Structural-RNN (St-RNN) encodes the spatial representation for each frame in a graph and then reasons over the temporal evolution of these graphs by feeding it to a Recurrent Neural Network. We adapt St-RNN's pipeline to our problem by replacing humans and objects in their model with vehicles and stationary landmarks, respectively. A similar methodology is employed for MRGCN-LSTM [3].

We show a quantitative comparison with our pipeline/model variations in Table IV-B. St-RNN that doesn't have any GCN components fare the worst. In Table IV-B, we observe that our method outperforms the traditional temporal based approach, St-RNN, by a significant margin. The gap is even more prominent when comparing the harder classes such as lane changing and overtaking, where we observe an average difference of 40% and 26%, respectively. A comparison between MRGCN-LSTM that uses a set of spatial graphs vs. MRGCN that uses the proposed interaction graph clearly shows the benefit of the proposed interaction graph. MRGCN outperforms its counter that learns in an end-end manner. This shows how such simple inherent behaviors are still hard for GCNs to learn. Further, with the addition of the attention mechanism, the Rel-Att-GCN model achieves an additional absolute 3-4% improvement on a few hard classes.

*2) Rule Based Baseline:* To showcase the effectiveness of an information-rich Interaction graph over traditional Spatio-

[1] code:https://github.com/ma8sa/Undersrtanding-Dynamic-Scenes-using-MR-GCN.git.

| Model | Rel-Att-GCN | | | Rule-Based | | |
|---|---|---|---|---|---|---|
| perf. measure | precision | recall | F1 | precision | recall | F1 |
| MAU | 97 | 95 | **96** | 97 | 91 | 94 |
| MTU | 95 | 99 | 97 | 100 | 99 | **99** |
| PRK | 100 | 98 | **99** | 100 | 99 | **99** |
| LCL | 94 | 97 | **95** | 96 | 81 | 88 |
| LCR | 97 | 97 | **97** | 97 | 88 | 92 |
| OVT | 71 | 89 | **79** | 36 | 90 | 52 |
| micro avg | 97 | 97 | 97 | 95 | 95 | 95 |
| macro avg | 92 | 96 | 94 | 88 | 91 | 87 |

TABLE II

REL-ATT-GCN VS RULE-BASED MODEL ON APOLLOSCAPE DATASET. BOTH THE MODELS USE THE PROPOSED INTERACTION GRAPH.

temporal modeling with a set of spatial graphs, we propose a rule-based approach to infer over the Interaction graph as one of our baselines. We use the Interaction Graphs generated by our pipeline (ref section:III-B) and employ an expert set of rules carefully framed to classify between behaviors. The deterministic classification is a simple max function over different relations a vehicle is associated with.

Relational behavior with majority count decides to which class the object belongs to from the following, $\{$*moving away, moving towards us, and lane changes*$\}$. For example, a node having the highest count for behavioral relation *moved left to right* would have its class as Lane change (Left to right). To obtain classification for *overtake* behavior, we iterate over all pair of vehicles, $i$ and $j$, that are not classified as *parked* or *moving towards us* in the first iteration and then we observe if there exists $e_{i,j}$ = *move-forward*, in which

The quantitative comparison in Table IV-B, clearly depicts the advantage of a rich temporal Interaction graph over models that directly utilize a set of spatial graphs, especially in the complex behavior class overtaking where it shows an 18 % and 27 % over MRGCN and St-RNN respectively. Though the rule-based model performs well compared to the Spatio-temporal approaches, it falls short against the learning-based models trained on the Interaction graph. The rule-based model is not powerful enough, especially on lane-change classes. This clearly explains the need for a learnable model to learn complicated patterns. On a closer look in Table II, it is clear that the rule-based method is not consistently better, especially on precision and recall metrics of Overtake and lane-change classes, respectively. The proposed Rel-Att-GCN clear outperforms the rule-based model on aggregate Micro and Macro average scores.

| Train Ratio | 0.05 | | 0.1 | | 0.2 | |
|---|---|---|---|---|---|---|
| Method | MRGCN | Rel-Att GCN | MRGCN | Rel-Att GCN | MRGCN | Rel-Att GCN |
| MAU | 61 | **94** | 91 | **95** | 89 | **97** |
| MTU | 47 | **99** | 75 | **99** | 85 | **99** |
| PRK | 90 | **98** | 86 | **98** | 86 | **98** |
| LCL | 36 | **98** | 69 | **98** | 89 | **98** |
| LCR | 54 | **94** | 73 | **95** | 88 | **96** |
| OVT | 50 | **60** | 58 | **65** | 76 | **77** |

TABLE III

RECALL ON APOLLOSCAPE DATASET FOR DIFFERENT AMOUNT OF TRAINING DATA

### D. Analysis of Relation-Attention MRGCN (Rel-Att-GCN)

The Rel-Att-GCN model is the MRGCN model with an additional attention component. The proposed Attention function factors into account that different types of relations in the interaction graph may have different relevancy to predict different classes. The varying importance of the relations in classifying vehicle behaviors can be visualized by analyzing normalized attention scores across relations for each class. One such visualization for the Apollo Scape dataset is depicted in Fig. 4. Higher values in each class (row) denote higher importance given by the attention function to that particular relation (column) to predict that class. The attention map clearly shows how classes such as overtake and lane changes depend on *moving forward* and *moved left to right* or *right to left* respectively. Despite how both MVA and OVT classes have high probability mass on move-forward relation, they can distinguish themselves based on the attention score spread over other relations.


Fig. 4. Figure shows attention scores between class labels (rows) and types of spatio-temporal interactions (columns). Higher scores for a particular relation indicates a higher dependence of the class on that particular relation.

Such reasoning helps the network to learn effectively under label scarcity, as given in Table III. Herein, we report results for models trained with 5% , 10% and 20% of training data. Rel-Att-GCN is able to show fidelity even when only 5% percent of data is present in contrast to a normal MR-GCN trained on the same interaction graph, which finds it difficult to learn from the smaller dataset. A similar trend is observed as we increase the size of training data to 10% and 20% of the actual dataset. In Table V, where the model is learned with 70% data, we observe that Rel-Att-GCN achieves superior performance across all datasets when compared with plain MR-GCN as well as temporal based methods.

### E. Transfer Learning

To showcase our proposed pipeline's generality, we trained the model only on the Apollo dataset and tested it on validation sets of Honda, KITTI, and Indian datasets. At the testing phase, we removed the classes which were not present in corresponding datasets. As the proposed pipeline does not rely upon any visually learned features, we can achieve

| Trained on | Apollo | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Tested on** | **Honda** | | | **KITTI** | | | **Indian** | | |
| Method | MRGCN LSTM | MRGCN | Rel-Att GCN | MRGCN LSTM | MRGCN | Rel-Att GCN | MRGCN LSTM | MRGCN | Rel-Att GCN |
| Moving away from us | 55 | 92 | **92** | 99 | 99 | **99** | 99 | 98 | **99** |
| Moving towards us | 79 | 60 | **92** | 98 | 98 | **98** | 93 | 94 | **97** |
| Parked | 91 | 65 | **99** | 99 | 98 | **99** | 99 | 95 | **99** |
| Lane-Change(L) | 65 | 73 | **94** | - | - | - | - | - | - |
| Lane-Change(R) | 87 | 83 | **92** | - | - | - | - | - | - |

TABLE IV

TRANSFER LEARNING RESULTS: WE TRAIN THE MODELS ON APOLLO SCAPES DATASET AND TEST ON HONDA, KITTI AND INDIAN DATASETS.

| Trained and Tested on | Apollo | | | Honda | | | KITTI | | | Indian | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | MRGCN LSTM | MRGCN | Rel-Att GCN | MRGCN LSTM | MRGCN | Rel-Att GCN | MRGCN LSTM | MRGCN | Rel-Att GCN | MRGCN LSTM | MRGCN | Rel-Att GCN |
| Moving away from us | 85 | 94 | **95** | 83 | 97 | **99** | 85 | 92 | **98** | 85 | 90 | **97** |
| Moving towards us | 89 | 95 | **99** | 79 | 86 | **90** | 86 | 91 | **98** | 74 | 91 | **97** |
| Parked | 94 | 94 | **98** | 85 | 88 | **99** | 89 | 95 | **99** | 84 | 93 | **99** |
| Lane-Change(L) | 84 | 97 | **97** | 75 | 91 | **91** | - | - | - | - | - | - |
| Lane-Change(R) | 86 | 93 | **97** | 60 | 81 | **85** | - | - | - | - | - | - |
| Overtake | 72 | 86 | **89** | - | - | - | - | - | - | - | - | - |

TABLE V

PERFORMANCE OF METHODS ON DIFFERENT DATASETS. THE MODELS HERE ARE TRAINED AND TESTED ON THE SAME DATASET.

fidelity across all datasets, as seen in Table IV. Evaluation results for the model trained and tested on validation sets of the same dataset can be found in Table V. From a comparison between Table IV and Table V, the transfer learning model though not better than models that are trained and tested on the same dataset, is on par with them. Notably, in the Honda dataset, the Rel-Att-MRGCN performs better in transfer for all classes than the model trained and tested on Honda. We attribute this behavior to high variation present in the Apollo dataset, which the other datasets lack.

### *F. Qualitative*

A video demonstration of the qualitative performance of our model on different datasets can be found here[2]. In Fig. 5 and 6 we showcase few qualitative results from different video snapshots. We follow a consistent convention for color-coding to depict behaviors. Red depicts vehicles *Moving Away From Us*, Green for *Moving towards Us* and Blue for *Parked* vehicles and Yellow and Orange depict *Lane Change Left to right* and *Lane Change Right to left* respectively while Magenta corresponds to *overtaking* vehicles.

In Fig. 5, sub-figures (a) and (b) show instances of vehicles *Moving Away From Us and Moving towards* on the KITTI dataset. On the same Fig. 5, sub-figures (c) and (d), showcase results from the Indian Dataset, wherein image (c), we see a bus and truck parked and in (d) we see *Lane change* behavior depicted by the car on the right.

In Fig. 6 (a) we see a car changing lane and in Fig. 6 (b) we observe a car classified as overtaking. Fig. 6 (c) and (d) show fidelity of our pipeline in traffic scenarios. In Fig.6 (c) we observe a car *changing lane* and merging into the road on the right and two cars *coming towards us*, in (d) we see a car changing a lane (on the right), a car parked on the left and two pickup trucks *moving away* from us. The qualitative results validates the proposed model's near-perfect classification and generalizability across datasets even in the presence of less (or not) observed test vehicles.

[2]`video`: https://youtu.be/TT4J-uH4xqI

## V. CONCLUSIONS

This paper proposed a novel pipeline for on-road vehicle behavior understanding and classification. It decomposed an evolving dynamic scene into a multi-relational Interaction graph whose nodes are the agents/actors in the scene, and edges are Spatio-temporal encodings that signify the agents' spatial behaviors. The interaction graph was further acted upon by a Multi-Relational Graph Convolution Network (MRGCN) to learn and classify the vehicle's overall behavior. The key takeaway is this two-stage classification that showed much-improved performance over end-end learning frameworks. The improved performance is attributed to edge encodings of the interaction graph being an accurate intermediate representation of spatial behaviors between agents that are difficult to characterize in an end-end learning framework. The MRGCN is integrated with an attention layer that further improved the performance, often near-perfect performance. Significant performance gain on various datasets that are consistent across several metrics confirms the efficacy of the proposed framework. Seamless data transfer across datasets further showcases its reliability. Future directions include integrating the proposed framework with a behavior planner.

## REFERENCES

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.


Fig. 5. Sub-figures (a) and (b) showcases prediction on standard vehicles from KITTI dataset. Whereas in (c), we can see Petrol Tanker and Bus's behavior predictions from the Indian dataset that shows object class-agnosticism. In (d), we can see a complex lane changes behavior prediction.


Fig. 6. The figure shows multiple scenarios depicting various behaviors. (a), (b) are samples from ApolloScapes dataset while (c) and (d) are from Honda dataset. In (a), (c), and (d), we observe the model accurately classifying lane change, both left to right and right to left. (b) depicts a case where the magenta car overtakes the red car.


[2] M. Meghjani, Y. Luo, Q. H. Ho, P. Cai, S. Verma, D. Rus, and D. Hsu, "Context and intention aware planning for urban driving," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2891–2898.

[3] 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," *arXiv preprint arXiv:2002.00786*, 2020.

[4] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 5308–5317.

[5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[6] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 954–960.

[7] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes," in *International Conference on Robotics and Automation*, 2019.

[8] X. Geng, H. Liang, B. Yu, P. Zhao, L. He, and R. Huang, "A scenario-adaptive driving behavior prediction approach to urban autonomous driving," *Applied Sciences*, vol. 7, no. 4, p. 426, 2017.

[9] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," *CoRR*, vol. abs/1803.07549, 2018. [Online]. Available: http://arxiv.org/abs/1803.07549

[10] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[11] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka, "Interaction-aware probabilistic behavior prediction in urban environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3999–4006.

[12] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena, "Car that knows before you do: Anticipating maneuvers via learning temporal driving models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3182–3190.

[13] D. Mitrovic, "Reliable method for driving events recognition," *IEEE transactions on intelligent transportation systems*, vol. 6, no. 2, pp. 198–205, 2005.

[14] R. Sabzevari and D. Scaramuzza, "Multi-body motion estimation from monocular vehicle-mounted cameras," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 638–651, 2016.

[15] A. Narayanan, I. Dwivedi, and B. Dariush, "Dynamic traffic scene classification with space-time coherence," 2019.

[16] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu, "Learning actor relation graphs for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9964–9974.

[17] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler, "Situation recognition with graph neural networks supplementary material."

[18] D. Wang, C. Devin, Q.-Z. Cai, F. Yu, and T. Darrell, "Deep object-centric policies for autonomous driving," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8853–8859.

[19] C. Li, Y. Meng, S. H. Chan, and Y.-T. Chen, "Learning 3d-aware egocentric spatial-temporal interaction via graph convolutional networks," *arXiv preprint arXiv:1909.09272*, 2019.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[21] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan, "Learning features by watching objects move," in *CVPR*, 2017.

[22] S. Rota Bulò, L. Porzi, and P. Kontschieder, "In-place activated batchnorm for memory-optimized training of dnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5639–5647.

[23] S. Song and M. Chandraker, "Robust scale estimation in real-time monocular SFM for autonomous driving," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014, pp. 1566–1573. [Online]. Available: https://doi.org/10.1109/CVPR.2014.203

[24] S. Geng, P. Gao, C. Hori, J. L. Roux, and A. Cherian, "Spatio-temporal scene graphs for video dialog," *arXiv preprint arXiv:2007.03848*, 2020.

[25] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles, "Action genome: Actions as compositions of spatio-temporal scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 236–10 247.

[26] R. Herzig, E. Levi, H. Xu, H. Gao, E. Brosh, X. Wang, A. Globerson, and T. Darrell, "Spatio-temporal action graph networks," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[27] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 399–417.

[28] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*. Springer, 2018, pp. 593–607.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.