Ordered Atomic Activity for Fine-grained Interactive Traffic Scenario Understanding

Nakul Agarwal\textsuperscript{1} \quad Yi-Ting Chen\textsuperscript{2}
\textsuperscript{1}Honda Research Institute USA \quad \textsuperscript{2}National Yang Ming Chiao Tung University
nakul\_agarwal@honda-ri.com \quad ychen@cs.nycu.edu.tw

Abstract

We introduce a novel representation called Ordered Atomic Activity for interactive scenario understanding. The representation decomposes each scenario into a set of ordered atomic activities, where each activity consists of an action and the corresponding actors involved and the order denotes the temporal development of the scenario. This design also helps in identifying important interactive relationships, such as yielding. The action is a high-level semantic motion pattern that is grounded in the surrounding road topology, which we decompose into zones and corners with unique IDs. For example, a group of pedestrians crossing in front is denoted as \(C_1 \rightarrow C_4: P^+\), as depicted in Figure 1. We collect a new large-scale dataset called OATS\textsuperscript{1} (Ordered Atomic Activities in interactive Traffic Scenarios), comprising 1026 video clips (\(\sim 20\)s) captured at intersections in San Francisco Bay Area. Each clip is labeled with the proposed language, resulting in 59 activity categories and 6512 annotated activity instances. We propose three fine-grained scenario understanding tasks, i.e., multilabel atomic activity recognition, activity order prediction, and interactive scenario retrieval. We also propose a Graph Convolutional Network based framework that models both appearance and motion of traffic participants to tackle the above tasks, that performs favorably against state-of-the-art methods. However, we find that the methods cannot achieve satisfactory performance, indicating rising opportunities for the community to develop new algorithms for these tasks towards better interactive scenario understanding.

1. Introduction

Intelligent transportation systems (ITS) have made significant progress in addressing traffic fatalities, with advancements in perception [20, 55, 70, 26, 19], prediction [90, 48, 16], hazard identification [54, 71, 39, 31], and planning [5, 1]. To deploy ITS at scale, extensive research has been conducted on scenario-based simulation assessment [15, 5, 85, 1] to validate ITS in challenging scenarios and identify the causes of failure. Effective scenario understanding and retrieval of real-world data, therefore, can enhance scenario-based assessment [65, 21] and help improve ITS deployment.

In this work, we focus on intersection scenarios, which involve the highest number of interactions among traffic participants and account for approximately 40\% of all crashes [53]. We specifically examine scenarios where traffic participants are crossing at the intersections, rather than

\textsuperscript{1}https://usa.honda-ri.com/oats
static participants who do not directly interact with the ego car. This design choice is motivated by the practice of scenario-based safety assessment [77, 63, 21, 13], which focuses on activities relevant to the ego vehicle. Our primary research objective is to identify an effective way to describe an interactive scenario involving traffic participants, in order to achieve a better scenario understanding and retrieval.

According to [77, 13], an interactive scenario representation should include information about the activities of road users (including their location, moving direction, and goal), the static environment, and the temporal evolution of these activities. We can use natural language [43, 34, 33], road scene graphs [89, 74], and attributes [58, 42, 48, 65] to represent interactive scenarios. While it is intuitive and explainable for humans using natural language, it often lacks explicit information about the motion directions and goals of road users, leading to verbose descriptions that are difficult to label and control for quality, as shown in Figure 1. Additionally, it is challenging to retrieve videos efficiently with lengthy natural language descriptions. Road scene graphs have also become a popular representation of traffic scenes. Each road user is represented as a node with low-level states such as location, speed, and direction, whereas edges capture pair-wise relationships between pairs of road users such as "following." However, these approaches do not represent the high-level semantics between actors’ actions and the underlying road topology, which enables efficient tagging for scenario-based assessment [21]. Attribute-based representations are promising options. However, existing works have not explored bridging the gap between activity and underlying road scene structures. Moreover, both road scene graphs and attribute-based approaches have not tackled the temporal development of a scenario and also lack activity order information.

To this end, we propose a novel representation of an interactive scenario called Ordered Atomic Activity. In Figure 1, we see an ego vehicle turning right and yielding to a group of crossing pedestrians in the front and a crossing vehicle from the left side, while another car in front is crossing in the opposite direction of the ego vehicle. We represent this detailed description in a concise yet interpretable manner using our proposed representation, where each activity comprises an action and the corresponding actors, such as a group of pedestrians or the ego vehicle. The action is a high-level semantic motion pattern grounded in the surrounding road topology. Specifically, we decompose a road scene, such as a 4-way intersection, into a set of regions that represent corners and zones. For example, a right-turn action is denoted as $Z_1 \rightarrow Z_2$, representing a motion pattern from an initial position to the destination. "A group of pedestrians crossing in front" is denoted as $C_1 \rightarrow C_4$: $P+$, where $C_1$ and $C_4$ are the two corners, $\rightarrow$ is the motion direction and $P+$ are the group of pedestrians performing the action. Moreover, we denote temporal development in a scenario by tagging the order in which the activities occur. This also helps in representing yielding relationships. In Figure 1, the order of the group of pedestrians, crossing vehicle in the same direction as ego vehicle, and ego vehicle activities represent yielding. Ordered Atomic Activity is designed at a video level, making it highly scalable and easing the burden of annotation.

We collect a large-scale dataset OATS and annotate monocular videos with Ordered Atomic Activity. The OATS dataset comprises 1026 clips, each approximately 20 seconds long, of real human driving in the San Francisco Bay Area captured using an instrumented vehicle. We propose three new and challenging fine-grained scenario understanding tasks, including multilabel atomic activity recognition, activity order prediction, and interactive scenario retrieval. We design a graph convolutional network-based algorithm that models both appearance and motion of traffic participants for the above tasks. We also implement multiple baselines (including recent state-of-the-art video understanding algorithms [6, 79, 87, 18, 76, 2, 83]) and evaluate them on the proposed dataset.

Our extensive experiments demonstrate that the methods perform inadequately on the proposed three tasks. Specifically, the best-performing algorithms for multilabel activity recognition, activity order prediction, and interactive scenario retrieval achieve 26.7% mAP, 16.1% matching score, and 16.6% Recall@top50. To successfully recognize multiple Atomic Activities in a scenario, a model should detect and track moving road users, associate their spatio-temporal action with respect to the underlying road topology, and capture the concept of groups. We believe the proposed Ordered Atomic Activity and the dataset introduce new challenges to the video scene understanding community. We hope to encourage the research community to work collectively on this important and challenging area.

This paper makes the following contributions. First, we introduce a novel representation called Ordered Atomic Activity for interactive scenario understanding. Second, we construct a large-scale dataset and propose three essential fine-grained scenario understanding tasks: multilabel atomic activity recognition, activity order prediction, and interactive scenario retrieval. Third, we propose a graph convolutional based network that models both appearance and motion of traffic participants and performs favorably against state-of-the-art methods on the above tasks. We also conduct extensive experiments on the three tasks and establish a comprehensive benchmark suite for future research.

2. Related Work

Traffic Scene Datasets. In recent years, many traffic scene datasets have been proposed to stimulate progress in detection [20, 84, 55, 70, 26], tracking [20, 84, 55, 70, 26, 60, 59],
All these unique aspects mentioned above introduce new challenges for understanding traffic scenes from videos.

**Activity Order Prediction.** The task of generating an ordered list of actions is commonly referred to as transcript prediction within the action understanding community. While many studies utilize this activity order as weak supervision to address complex tasks like action segmentation [29, 61, 62, 14, 7, 41, 3, 67], only a few studies explicitly focus on generating this transcript using the ordered list of actions as the ground truth [67, 3]. Existing works have concentrated on cooking activities [37, 4, 17, 68] where object movements are limited or only a single action occurs at a given point in time. The community has not yet explored this aspect in traffic scenes, where the order of actions can help to identify critical interactions, such as yielding. In traffic scenes, individual actions begin at different times, and their progression happens simultaneously, making it extremely challenging to identify the multiple actions and corresponding orders. Note that this order cannot be automatically generated even if timestamps [16] are available. Timestamps alone cannot tell when a pedestrian starts crossing. Moreover, a car yielding to a crossing pedestrian might start crossing the zones before the pedestrian based on timestamps. The complicated interactive patterns make automatic generation infeasible. Thus, we label the activity orders manually and release the dataset. To the best of our knowledge, our work is the first to introduce and tackle this problem in driving scenes.

**Traffic Scenario Retrieval.** The focus of most existing work in the analysis is detection and tracking of traffic participants [12, 73, 52], recognition of actions in traffic scenes [59, 60, 58, 46, 22], and traffic anomaly detection [52]. To enable fine-grained scenario retrieval, we need to analyze human driving data at multiple levels, ranging from object detection to fine-grained activity orders. Existing traffic scenario retrieval solutions describe traffic scenes using natural language [43], traffic scene attributes [42, 65], and latent representation [27]. Our work is closely related to [42, 65] as the proposed description language can be treated as traffic scene attributes. The paper [42] proposes attributes such as driver behavior/intention/attention. For [65], they consider vehicles’ speed and density, vehicles’ actions (e.g., braking, keeping lane), and vehicle-to-vehicle interactions (e.g., braking for another vehicle). However, we observe the following differences. First, we predict traffic scene attributes using monocular videos instead of multimodal data (e.g., GPS/IMU, LiDAR, and HD maps). Second, existing scene attributes do not jointly consider the relationship between action and road structures. While the work [65] offers HD maps to link actions and road structures, their attributes only describe the locations of vehicles. Third, *Ordered Atomic Activity* enables the development of predicting the activity order in a scenario.

### Table 1. Comparison of OATS with other datasets.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong># Clips</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>RGB Images</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>LiDAR Point Cloud</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Agent type</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Type of Intersection</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Atomic Activity</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Activity Order</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Semantic segmentation [11, 84, 88, 19], trajectory forecast [26, 90, 69], and long-term localization [47]. In addition to perception and prediction tasks, nuScenes [26], Argoverse [8], and Lyft [32] provide high definition maps that enable road topology understanding from monocular videos [64, 56]. Moreover, the datasets containing annotations for activity recognition and intention prediction of traffic participants [60, 58, 59, 46, 48, 22] are essential for the field. Some recent datasets [65] also provide HD maps, but lack a concept of spatial zones and only contain information in the form of semantic labels (road segments, lanes) or polylines/polygons. A dataset that jointly considers the relationship between behaviors of traffic participants, ego vehicle, and underlying road topology is under-explored.

To this end, we collect a large-scale dataset with diverse interactive scenarios labeled with the proposed representation *Ordered Atomic Activity*. Although *Ordered Atomic Activity* can be used to represent scenarios in existing datasets [16, 26, 19], these interactive scenarios are not predefined in the released annotations. Moreover, IP-related issues can make it challenging to augment and release data from such datasets. Therefore, we collect our own dataset to facilitate the new research direction. In addition, we propose and benchmark three novel fine-grained interactive scenario understanding tasks. We provide extensive discussion on the challenges posed by the proposed dataset and tasks.
3. The OATS Dataset

3.1. Data Collection Platform

The data is collected using an instrumented vehicle equipped with 3 Point Grey Grasshopper video cameras with a resolution of 1920 × 1200 pixels, a Velodyne HDL-64E S2 LiDAR sensor, and high precision GPS. All sensor data are synchronized and timestamped using ROS and customized hardware and software. The data is collected in the San Francisco Bay Area region and captures a diverse set of traffic scenarios at intersections, including different environments such as urban and suburban areas.

3.2. Annotation Methodology

Modeling both driver and traffic agent behavior is complex and involves different levels of cognitive processes, particularly in complicated interactive scenes, and is essential for the development of robust intelligent driving systems. Therefore the data selection and annotation protocol must be carefully designed. The first step involves the manual selection of short clips from hours of recording that include appropriate scenarios. Although in this work we only focus on 4-way intersections as mentioned in Section 1, we show the flexibility of Ordered Atomic Activity for different topologies in the supplementary material. We clip ~20s short clips around intersections. Each clip contains both the entering and exit of the ego vehicle from the intersection. We discuss the different aspects of our annotation in the following.

Annotation Consistency. We conduct the following quality control strategy. Given a video, it is first annotated by 2 independent human annotators who are experienced drivers residing in the USA to ensure familiarity with the rules of the road, signs etc. Finally, we have an internal expert annotator to review and obtain the final version. To demonstrate the consistency and validity of the annotations, we compute the intra-class correlation coefficient (ICC) [66], which is widely used [58] for the assessment of consistency made by different observers measuring the same quantity. The ICC for our annotations is 0.91, which indicates “excellent” consistency (ICC = 1 for absolute agreement) according to [10].

Atomic Activity. An atomic activity consists of an action and the corresponding actors. The action is a high-level semantic motion pattern that is grounded in the surrounding road topology. We first decompose the surrounding road topology into four corners (C1, C2, C3, C4) and four zones (Z1, Z2, Z3, Z4) with respect to the ego vehicle in an anti-clockwise manner, as shown in Figure 1. Actor consists of seven classes: car (C), bicyclist (K), pedestrian (P), ego (E), group of cars (C+), group of bicyclists (K+), group of pedestrians (P+). Given a clipped video, the annotator watches the full video to understand both the type of agents in the scenario and their direction of motion. After analyzing the video, the annotator annotates the Atomic Activity. For instance, we denote a group of pedestrians’ crossing activity from the near-right corner to the near-left corner with respect to the ego vehicle as C1 → C4: P+.

Activity Order. Besides labeling activities, we also assign labels to their corresponding order. The starting time of the activity, i.e. when the agent starts crossing, is the basis for annotating this order except for yielding actions. In such cases, the agent that yields is labeled later, regardless of when the action begins or ends. For instance, when a car yields to a pedestrian at a crossing, it may eventually cross the zones before the pedestrian, but the pedestrian should be labeled first. The strategy is challenging to be labeled automatically using timestamps. This is why human observation is necessary when annotating the activity order, and it cannot be inferred heuristically or programmatically using timestamps.

Type of Intersection. We further categorize the scenario into the following four types of intersections: Four-Way Unprotected Turn (4WUT), Four Way Protected Turn (4WPT), Four Way Two Stop (4W2S) and Four Way Four Stop (4W4S). These scenarios are differentiated based on the number of stop signs and traffic light types at intersec-
tions. This is particularly useful for traffic scene understanding and scenario retrieval based on a specific type of intersection.

3.3. Dataset Statistics

Our dataset comprises 1026 video clips and 59 activity categories. Figure 2 depict the label distribution of the different activities in our dataset, sorted in descending order. As shown in the figure, the label distribution is not uniform, which is typical of real-world data. For instance, activities comprising a group of actors (e.g., Z3-Z1:K+) are very rare compared to more frequent activities, such as Z3 → Z1:C or C2 → C3:P. More details regarding the frequency of individual activities, actors, actions and also the types of intersections can be found in the supplementary material.

4. Methodology

4.1. Model Description

In this section, we present the proposed model for multilabel atomic activity recognition. The framework is depicted in Figure 3. As atomic activity comprises both action and agent, our network considers both appearance and motion features. Given a video sequence, we begin by extracting tracklets of the actors in the scene using pre-trained Mask R-CNN [24] pre-trained on COCO dataset [44] and Deep SORT [81]. We then select a set of Z frames from the video and extract appearance features of N traffic agents using the Inception-v3 [72] backbone and RoIAlign [24]. We also extract motion features from tracklets using bounding boxes of agents. After completing the feature extraction process, we pass the motion and appearance features through separate graph convolution networks [35], with each node representing an actor. Finally, we fuse the learned features from both graphs for multilabel atomic activity recognition. We discuss the specifics of each module below.

Appearance Model. We utilize the graph structure to explicitly model pair-wise relations between different agents in the driving scene. Unlike prior graph-based algorithms [2, 83] that treat objects in the graph independently, we construct our graph by utilizing tracking. Given a set of N agents in the traffic scene with their corresponding tracklets, we construct a spatio-temporal graph \( G_t^a = (V_t^a, A_t) \), where \( V_t^a = \{v_t^a|v_t \in \{1, ..., N\}\} \) is the set of vertices of graph \( G_t^a \) and \( A_t = \{a_{ij}^a|v_t, j \in \{1, ..., N\}\} \) is the adjacency matrix \( v_t, j \in \{1, ..., Z\} \). In our graph, \( a_{ij}^a \) models the appearance relation between two agents at time \( t \) and is formally defined as:

\[
a_{ij}^a = \frac{f_a(v_t^i, v_t^j)\exp(f_a(v_t^i, v_t^j))}{\sum_{j=1}^{N} f_a(v_t^i, v_t^j)\exp(f_a(v_t^i, v_t^j))},
\]

where \( f_a(v_t^i, v_t^j) \) indicates the appearance relation between agents \( i \) and \( j \) at time \( t \), and \( f_a(v_t^i, v_t^j) \) is an indicator function that determines the presence of a tracklet. The softmax function is used to normalize the influence on agent \( i \) from other objects. The appearance relation is calculated as below:

\[
f_a(v_t^i, v_t^j) = \frac{\theta(v_t^i)^T \phi(v_t^j)}{\sqrt{D}},
\]

where \( \theta(v_t^i) = w_t^i v_t^i \) and \( \phi(v_t^j) = w_t^j v_t^j \). Both \( w_t \in \mathbb{R}^{D \times D} \) and \( w_t \in \mathbb{R}^{D \times D} \) are learnable parameters that map appearance features to a subspace and enable learning the correlation of two objects, and \( \sqrt{D} \) is a normalization factor. While [2, 83] can fill the graph with any random node at time \( t \), we need to take into account missing nodes due to both inconsistencies in tracking and agents entering and leaving the traffic scene at different times. To mitigate this issue, we set adjacency matrix values to zero when an object is missing using indicator function \( f_p \) as:

\[
f_p(v_t^i, v_t^j) = \|v_t^i = \text{present and } v_t^j = \text{present}\)
\]

Once the nodes and adjacency matrix values are defined, we reason over the Graph Convolutional Network (GCN) [35]. GCN takes a graph as input, performs computations over the structure, and returns a graph as output. For a target node \( i \) in the graph, it aggregates features from all neighbor nodes according to values in the adjacency matrix. Formally, one layer of GCN can be written as:

\[
Z^{(l+1)} = \sigma A Z^{(l)} W^{(l)},
\]

where \( A \in \mathbb{R}^{NZ \times NZ} \) is the adjacency matrix for appearance model and \( Z^l \in \mathbb{R}^{NZ \times D} \) is the feature representations of nodes in the \( l \)th layer. The matrix \( W^l \in \mathbb{R}^{D \times D} \) is the layer-specific learnable weight matrix. The function \( \sigma(\cdot) \) denotes an activation function. We adopt ReLU as the activation function. Note that this layer-wise propagation can be stacked into multi-layers.

Motion Model. An essential aspect of our problem is how we represent the motion of agents, as it is a crucial part of our annotations. In this work, we model the motion of agents in 2D. Our motion model is inspired by recent success in the motion-based action recognition [86] and trajectory prediction [50, 36, 91]. We use the relative motion between tracklets at different times \( t \) as input to the graph rather than absolute coordinates in the image space. We construct another spatio-temporal graph \( G_t^m = (V_t^m, E_t) \), where \( V_t = \{v_t^m|v_t \in \{1, ..., N\}\} \) is the set of vertices of graph \( G_t^m \) and \( E_t = \{e_{ij}^t|v_t, j \in \{1, ..., N\}\} \) is the set of edges. We set \( e_{ij}^t = 1 \) if \( e_t \) and \( e_t \) are connected, and \( e_{ij}^t = 0 \) otherwise. We attach a value \( b_{ij}^t \) to model the relation between two nodes \( i \) and \( j \), which is computed by some kernel function for each \( e_t \). The \( b_{ij}^t \) are organized into the weighted adjacency matrix \( B_t \). We introduce \( b_{ij}^t_{sim,t} \) as a kernel function to be used within the adjacency matrix \( B_t \).
Given Z frames from an input video, we first extract tracklets of traffic agents and then construct two separate spatio-temporal GCNs $G^a$ and $G^m$ each to model both their appearance and motion in the scene respectively. Unlike previous methods, we utilize tracking to capture the position of the traffic agent in the graph. Finally, the extracted GCN features are then fused for multi-label classification.

Algorithm 1 Retrieval Algorithm

**Input:** $L, P, V$ and model I

**Output:** recall@topK

1. $t_p = 0$, $f_p = 0$, $f_n = 0$
2. top $K = \text{Hamming}(T(S_q), L)$
3. for $k \leftarrow 1$ to $K$
do:
4. $q \leftarrow 1$ to $q$
do:
5. if $M_{S_q} \subseteq M_{S_j}$ then:
6. $t_p + = 1$
7. else:
8. $f_p + = 1$
9. for $k \leftarrow 1$ to $(F - K)$do:
10. for $i \leftarrow 1$ to $q$
do:
11. if $M_{S_q} \subseteq M_{S_j}$ then:
12. $f_n + = 1$
13. return recall = $\frac{t_p}{t_p + f_n}$

The function $b_{sim,t}^{i,j}$ is defined as:

$$b_{sim,t}^{i,j} = \begin{cases} \frac{1}{||u_t^{i} - u_t^{j}||_2}, & ||u_t^{i} - u_t^{j}||_2 \neq 0 \\ 1, & u_t^{i} \text{ or } u_t^{j} \text{ missing} \\ 0, & \text{otherwise} \end{cases}$$  

where $u_t^i$ is 2D position of agent $i$ at time $t$ and can be defined as:

$$u_t^i = [c_x, c_y],$$

where $c_x$ and $c_y$ are centers of bounding box for agent $i$ at time $t$. Once the nodes, edges, and adjacency matrix for motion GCN are formed, we perform spatio-temporal graph convolution operation similar to [50, 86].

**Loss Function.** After constructing the motion and appearance graphs, we extract feature representations learned through the GCNs and combine them for recognition. As both our motion and appearance graphs are composed of tracklets, there are multiple ways to link information from the two graphs. We experimented with various approaches and discovered that late fusion works best. We conjecture that it is because the tracklet cues are noisy. We discuss the performance of different fusion schemes in the supplementary material. The entire model can be trained end-to-end with backpropagation. The final loss function is defined as:

$$
L = -\frac{1}{R} \sum_{i=1}^{R} y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i)),
$$

where $y$ is the label, $p(y)$ is the predicted probability and $R$ is the batch size.

**4.2. Activity Order Prediction**

The appearance and motion features are obtained from our spatio-temporal graphs $G^a_t$ and $G^m_t$, respectively. We fuse them to form a latent video representation $O \in \mathbb{R}^{T \times D}$ of an input video $X_t = \{x_t | t \in \{1, ..., Z\} \}$. To model the activity order, we incorporate the frame classification and segment generation branches (denoted as $FC$ and $SG$ respectively) proposed in [67] for calculating the mutual consistency loss. The branch $FC$ takes the shared latent video representation $O$ as input and predicts the class probabilities $FC(O) = Y \in \mathbb{R}^{T \times P}$, where $P$ is the number of activities in the dataset. The branch $SG$ predicts the segments $SG(O) = S$, where each segment $s_m$ consists of predicted ordered activity probabilities $a_m$ and the estimated relative log length $l_m$ of that segment. To calculate the mutual consistency loss for training the network, we first start with the estimated relative log length $l'_m$ for each segment $s_m$, which is then converted into the absolute length $l_m'$. For each segment, we compute its absolute starting position $p_m'$ within the video. Given $l'_m$ and $p'_m$, for each segment we generate a mask $w_m$ using the differentiable mask generation module. For each segment, we use the ground-truth action label
\[ L_\mu = \sum_{m=1}^{M} L_{\mu,m}(Y, w_m, \hat{a}_m) \]  

(8)

For more details regarding the loss function, please refer to [67].

### 4.3. Scenario Retrieval

We denote an intersection scenario as \( S_q = (V, P) \) in the query set \( Q \), where \( V \) is the set of agents in a scenario and \( P \) is the set of tracklets. Each activity \( l \) in video-level labels \( L \) has the format of Action:Actor where the former gives direction of motion and the latter denotes the type of agent. When retrieving scenarios for driver behavior understanding, not only is it important to match the configuration of the traffic scene based on activities, but also to independently focus on the type of agent and motion dynamics. This is because sometimes the focus for retrieving a certain scenario could be based on the number of type of agents in the scene, e.g. car or pedestrian or both, while other times the user might be more interested in the direction of motion of traffic participant, e.g. Z1-Z2 or C1-C2. Some users might be interested in both aspects. Therefore, we formulate a metric that takes into account all three factors: activities, action, and actor.

We denote the scenario database as \( F \) and our goal is to retrieve top \( K \) similar scenarios \( S'_f = (V, P) \forall i \in \{1, \ldots, K\} \) given \( S_q \). We first retrieve top \( K \) scenarios using the hamming distance between predicted binary labels (through model I in Figure 3) for \( S_q \) and ground truth labels for \( S'_f \forall f \in \{1, \ldots, F\} \}. Once the top \( K \) scenarios are retrieved, we convert predicted labels for \( S_q \) and \( L \) into dictionaries \( M_{S_q} \) and \( M_S \), respectively, where the key is noun, verb or action unit and values are their instances/frequencies in the output. We consider a retrieved scenario to be a match if \( M_{S_q} \) is a subset of \( M_{S_f} \). The algorithm of the proposed retrieval system is discussed in Algorithm 1.

### 5. Experiments and Analysis

#### Datasets & Metrics.

To account for the variability in the number of labels in each video, we divided our OATS dataset into three groups, namely \( s_1, s_2, s_3 \), with a sufficient number of samples for each class. The splits \( s_1 \) and \( s_2 \) consist of 350 video clips while split \( s_3 \) has 326 clips. These groups are permuted as splits for training and testing to ensure fair evaluation [38]. For further details on the dataset creation and experimental setup, refer to the supplementary material.

We also conduct action understanding experiments on the Collective Activity dataset [9], which contains 44 short video sequences from 5 group activities (crossing, waiting, queuing, walking, and talking) and 6 individual actions (NA, crossing, waiting, queuing, walking and talking). We follow the same evaluation scheme as in [83, 57]. We report mean average precision (mAP) for multi-label atomic activity recognition, matching score [67] for activity order prediction, and recall@topK for scenario retrieval.

#### Implementation Details.

Our implementation is based on PyTorch and we use a single Nvidia Quadro RTX 6000 GPU. We use 32 frames as input each having an image size of 224 x 224, and a batch size of 1 for experiments. We adopt stochastic gradient descent with ADAM to learn the network parameters and train the model for 50 epochs using a learning rate ranging from 0.0002 to 0.0001. All feature layers are jointly updated during training. More implementation details and qualitative results can be found in the supplementary material.

---

Table 2. Comparison of our framework with other state-of-the-art algorithms for multi-label atomic activity recognition.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Pretrain</th>
<th>Splits</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSN [76]</td>
<td>Resnet152</td>
<td>IG65M</td>
<td>( s_1 )</td>
<td>12.41</td>
</tr>
<tr>
<td>TPN [87]</td>
<td>Resnet50</td>
<td>ImageNet</td>
<td>( s_2 )</td>
<td>11.57</td>
</tr>
<tr>
<td>SlowOnly [18]</td>
<td>Resnet50</td>
<td>ImageNet</td>
<td>( s_3 )</td>
<td>11.89</td>
</tr>
<tr>
<td>SlowFast [18]</td>
<td>Resnet50</td>
<td>None</td>
<td></td>
<td>12.59</td>
</tr>
<tr>
<td>I3D (NL) [79]</td>
<td>Resnet50</td>
<td>ImageNet</td>
<td></td>
<td>12.95</td>
</tr>
<tr>
<td>ARG [83]</td>
<td>Inceptionv3</td>
<td>ImageNet</td>
<td></td>
<td>13.59</td>
</tr>
<tr>
<td>Ours</td>
<td>Inceptionv3</td>
<td>ImageNet</td>
<td></td>
<td>16.12</td>
</tr>
</tbody>
</table>

Table 3. Ablation studies showing the effect of motion and appearance modeling in our framework.

<table>
<thead>
<tr>
<th>Method</th>
<th>Splits</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion</td>
<td>( s_1 )</td>
<td>13.68</td>
</tr>
<tr>
<td>Appearance</td>
<td>( s_2 )</td>
<td>15.70</td>
</tr>
<tr>
<td>Motion + Appearance</td>
<td>( s_3 )</td>
<td>14.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26.70</td>
</tr>
</tbody>
</table>

---

\( \hat{a}_m \) instead of the estimated class probabilities. The final loss function is given by:

\[
L_\mu = \sum_{m=1}^{M} L_{\mu,m}(Y, w_m, \hat{a}_m)
\]
Multilabel Atomic Activity Recognition. Given the labeled \textit{Ordered Atomic Activity} in the OATS dataset, we formulate the problem as a multi-label recognition problem. We compare our method in Sec. 4 against the different state-of-the-art video understanding algorithms. We start with algorithms extracting features at the video-level \cite{6, 79, 87, 18, 76}. As shown in Table 2, given that our scene deals with multiple agents in the scene spanning approximately 20s of video, implicitly focusing on agents through video-level features cannot recognize activities successfully. For algorithms \cite{2, 83}, which explicitly models objects in videos, the results improve. While these object-aware models perform better than purely video-based modeling, they focus on the appearance modeling of the agents, instead of motion modeling. As atomic activity comprises both action and actor, both appearance and motion should be modeled explicitly. The efficacy of the proposed modeling is presented in Table 2. We also conduct ablation studies to diagnose the effects of appearance and motion modeling of our framework in Table 3. It is worth noting that the unsatisfactory performance demonstrated in Table 2 indicates the challenging nature of the task. In addition, also highlight the scope for improvement before moving on to even more complicated scenarios and descriptions.

We also provide results on the Collective Activity dataset in Table 4 to show the efficacy and generalizability of our proposed framework. Since we design our framework keeping in mind video-level supervision, we avoid individual actions and only use group activities in our experiments. We compare our method against \cite{83}, which is the state-of-the-art algorithm on the Collective dataset. To have a fair comparison, we run their publicly available code\footnote{https://github.com/wjchaoGit/Group-Activity-Recognition} using their own fixed setting but without individual action supervision. We observe that even though \cite{9} has much less object motion in the image space compared to OATS, our framework can perform favorably against the state-of-the-art.

Activity Order Prediction. Very few works focus on explicitly predicting activity order using only the ordered list of activities as ground truth, and none do so in traffic scenes to the best of our knowledge. Therefore, we adapt state of the art method MuCon \cite{67} to our task as a baseline for comparison. Table 5 demonstrates that our method outperforms MuCon \cite{67} for activity order prediction on the OATS dataset. We report results with both w/ and w/o start and end tokens, which help in correctly identifying the number of activities in the predicted sequence \cite{67}. The results indicate that while our method is much better when it comes to predicting the correct number of activities, it is only slightly better in predicting the actual activity order.

<table>
<thead>
<tr>
<th>Method</th>
<th>Group Action</th>
<th>ARG \cite{83}</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>MuCon</td>
<td>48.12/16.45</td>
<td>52.06/15.93</td>
<td>47.93/13.23</td>
</tr>
<tr>
<td>Ours</td>
<td>52.09/17.32</td>
<td>52.94/16.28</td>
<td>51.73/14.65</td>
</tr>
</tbody>
</table>

Table 4. Comparison with the state-of-the-art algorithm on the Collective dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Matching Score</th>
<th>s1</th>
<th>s2</th>
<th>s3</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>MuCon</td>
<td>49.37/15.20</td>
<td>52.00/15.93</td>
<td>47.93/13.23</td>
<td>49.37/15.20</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>52.25/16.08</td>
<td>52.04/16.28</td>
<td>51.73/14.65</td>
<td>52.25/16.08</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Activity order prediction results on OATS w/ (left) and w/o (right) start and end tokens.
Figure 5. Qualitative results of our method for scenario retrieval. On the left is the query input, middle contains some examples from the scenario database, and right shows the top two retrieved scenarios from the database.

<table>
<thead>
<tr>
<th>Method</th>
<th>Actor R@10</th>
<th>Actor R@30</th>
<th>Actor R@50</th>
<th>Action R@10</th>
<th>Action R@30</th>
<th>Action R@50</th>
<th>Activity R@10</th>
<th>Activity R@30</th>
<th>Activity R@50</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG [83]</td>
<td>0.60</td>
<td>1.91</td>
<td>3.22</td>
<td>0.32</td>
<td>1.49</td>
<td>2.31</td>
<td>0.32</td>
<td>1.49</td>
<td>2.31</td>
</tr>
<tr>
<td>Ours</td>
<td>6.89</td>
<td>12.92</td>
<td>18.87</td>
<td>1.53</td>
<td>6.33</td>
<td>16.56</td>
<td>1.53</td>
<td>6.33</td>
<td>16.56</td>
</tr>
</tbody>
</table>

Table 6. Scenario retrieval results on the OATS dataset

Scenario Retrieval. Given the low performance in Table 2 and Table 5, we select a subset of videos from the 3 splits $s_1$, $s_2$, and $s_3$, which comprises 13 activities and train the proposed model for multi-label atomic activity recognition. We then use those results for the task of scenario retrieval. More specifically, we use one split, e.g. $s_1$ as our query set and train our model using the database which comprises the other two splits, e.g. $s_1$ and $s_2$. We then use the trained model to predict binary labels for the query set which is then used for retrieval as described in Section 4.3. We calculate the recall@topK (for different Ks) for Actor, Action, and Activity individually. Due to the lack of existing methods in this space, we compare with the best-performing method in Table 2. Figure 5 and Table 6 show that although our method performs favorably against [83], the absolute numbers are quite low, indicating the challenging nature of the task and encouraging future research. This is also natural given the low performance of multi-label classification by state-of-the-art methods in Table 2. These low results are also at par with other retrieval applications [45] where the ground truth generally focuses on a single aspect, whereas we are trying to retrieve a complex traffic configuration that covers various aspects such as type of agent, motion information, and scene information.

Inference Speed. Although real-time inference is not the focus of this paper, our model is quite fast. While scenario retrieval has an offline use case, the model takes around 40 ms for multi-label classification in Table 2 and about 70 ms for Activity Order prediction in Table 5 using a single Nvidia Quadro RTX 6000 GPU, making it viable from an application point of view.

Limitations. One of the limitations of our work is that we do not explicitly model group actions in our framework, i.e. $P+$, $C+$, and $K+$. Distinguishing between individual and group actions can play an important role in downstream applications. Additionally, we primarily rely on [67] for Activity Order modeling whereas results in Table 5 show that a more sophisticated approach is required to solve this challenging task. We plan to address these issues in future work.

6. Conclusion

We introduce a novel representation called Ordered Atomic Activity for interactive scenario understanding. The representation decomposes each scenario into a set of ordered atomic activities, where each activity consists of an action and the corresponding actors involved and the order denotes the temporal development of the scenario. Given the lack of an appropriate dataset, we introduce a new large-scale dataset OATS along with three crucial fine-grained interactive scenario understanding tasks, i.e., multi-label atomic activity recognition, activity order prediction, and scenario retrieval. We also provide an algorithm to solve scenario retrieval and benchmark these tasks on our dataset with existing SOTA algorithms, as well as with our proposed framework that performs favorably against these methods, for enabling future research.
References


Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for


