# Cross-Modality Graph-based Language and Sensor Data Co-Learning of Human-Mobility Interaction

MAHAN TABATABAIE, University of Connecticut, USA
SUINING HE, University of Connecticut, USA
KANG G. SHIN, University of Michigan–Ann Arbor, USA

Learning the human–mobility interaction (HMI) on interactive scenes (e.g., how a vehicle turns at an intersection in response to traffic lights and other oncoming vehicles) can enhance the safety, efficiency, and resilience of smart mobility systems (e.g., autonomous vehicles) and many other ubiquitous computing applications. Towards the ubiquitous and understandable HMI learning, this paper considers both *"spoken language"* (e.g., human textual annotations) and *"unspoken language"* (e.g., visual and sensor-based behavioral mobility information related to the HMI scenes) in terms of information *modalities* from the real-world HMI scenarios. We aim to extract the important but possibly implicit HMI concepts (as the *named entities*) from the textual annotations (provided by human annotators) through a novel human language and sensor data *co-learning* design.

To this end, we propose CG-HMI, a novel **C**ross-modality **G**raph fusion approach for extracting important **H**uman-**M**obility **I**nteraction concepts from *co-learning* of textual annotations as well as the visual and behavioral sensor data. In order to fuse both unspoken and spoken "languages", we have designed a unified representation called the *human–mobility interaction graph* (HMIG) for each modality related to the HMI scenes, i.e., textual annotations, visual video frames, and behavioral sensor time-series (e.g., from the on-board or smartphone inertial measurement units). The nodes of the HMIG in these modalities correspond to the textual words (tokenized for ease of processing) related to HMI concepts, the detected traffic participant/environment categories, and the vehicle maneuver behavior types determined from the behavioral sensor time-series. To extract the inter- and intra-modality semantic correspondences and interactions in the HMIG, we have designed a novel graph interaction fusion approach with differentiable pooling-based graph attention. The resulting graph embeddings are then processed to identify and retrieve the HMI concepts within the annotations, which can benefit the downstream human-computer interaction and ubiquitous computing applications. We have developed and implemented CG-HMI into a system prototype, and performed extensive studies upon three real-world HMI datasets (two on car driving and the third one on e-scooter riding). We have corroborated the excellent performance (on average 13.11% higher accuracy than the other baselines in terms of precision, recall, and F1 measure) and effectiveness of CG-HMI in recognizing and extracting the important HMI concepts through cross-modality learning. Our CG-HMI studies also provide real-world implications (e.g., road safety and driving behaviors) about the interactions between the drivers and other traffic participants.

CCS Concepts: • **Human-centered computing** → *Ubiquitous and mobile computing*.

Additional Key Words and Phrases: Human-mobility interaction, cross-modality graph interaction fusion, language and sensor data co-learning, named entity recognition, human-mobility interaction concept extraction.

Authors' addresses: Mahan Tabatabaie, Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA, mahan.tabatabaie@uconn.edu; Suining He, Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA, suining.he@uconn.edu; Kang G. Shin, Department of Electrical Engineering and Computer Science, University of Michigan–Ann Arbor, Ann Arbor, MI, USA, kgshin@umich.edu.


2474-9567/2023/9-ART125 $15.00
https://doi.org/10.1145/3610904

**ACM Reference Format:**
Mahan Tabatabaie, Suining He, and Kang G. Shin. 2023. Cross-Modality Graph-based Language and Sensor Data Co-Learning of Human-Mobility Interaction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3, Article 125 (September 2023), 25 pages. https://doi.org/10.1145/3610904

## 1 INTRODUCTION

*Human-mobility interaction* (HMI) learning refers to the tasks of understanding how a mobility system user (e.g., a vehicle driver or a micromobility rider) *interacts* with her/his mobility system and the traffic environments (e.g., the driver slows down her/his car in response to a pedestrian crossing the street). HMI learning has become essential for many emerging and ubiquitous smart mobility applications. For instance, by learning the vehicle driver's decisions and responses at an intersection, one may devise safe car-maneuvering strategies and improve artificial intelligence (AI) designs of self-driving autonomous vehicles (AV). Effective and accurate HMI learning can serve as a basis for human behavioral learning for connected AV development [26], advanced driver assistance system (ADAS) designs [43, 44], and many other emerging AI-assisted mobility and micromobility systems [16].

Existing HMI studies [4, 10, 17, 30, 36, 47] largely focus on analyzing and interpreting the visual sensor (e.g., video frames of the vehicle's dash-view cameras) and the behavioral sensor data (e.g., on-board inertial measurement units (IMUs)), which can be considered as an "*unspoken language*". An interesting but largely under-studied way to understand the HMI lies with *analysis of the human's textual descriptions and thoughts about how the mobility system users are interacting with certain traffic and mobility conditions*. Specifically, a vehicle driver or a passenger can provide a textual description regarding the scenes of certain interaction events, such as abrupt braking due to the sudden appearance of a pedestrian, as the post-event *feedback* or even the *explainable* AV system designs [1, 2]. For instance, the passengers may express their concerns about the road conditions (e.g., slippery or an object) that may also inform additional, precautionary, or responsive measures taken by the human driver or the AV system. Such an additional verbal and textual modality, i.e., the *spoken language*, can help the core AI models and systems (e.g., ADAS) account for the latent *human factors*, such as the attention levels of the human perception upon the traffic environments in the HMI scenes. One can further derive the *semantic* decision-making processes and the *causal* relationships in the HMI scenes. This way, we can enable safer and more pleasant interactions with the mobility environments and other traffic participants. In other words, incorporating the human intelligence from the spoken language may help convey and express the implicit and subtle interactions that may not be easily revealed by existing visual and behavioral sensor analysis.

Motivated by the above-mentioned scenarios, our *goal* is to fuse the spoken and unspoken languages in terms of different *modalities* from the real-world HMI scenarios, and extract and learn the important HMI concepts from the human textual annotations towards understanding the HMI scenes. Such a novel cross-modality HMI language and sensor co-learning design will help the existing smart mobility applications to more effectively capture the *semantic dependencies* between the human decision-making (e.g., driving behaviors) and the interaction outcomes (e.g., prevention of a traffic accident). Furthermore, such a co-learning design can be beneficial for downstream human-computer interaction (HCI) in many ubiquitous computing applications, such as bridging the *semantic* and *physical* aspects of HCI in conducting multi-modal human behavior analysis [37].

Towards this goal, we focus on two case studies of emerging smart mobility systems for concrete insights, i.e., car driving [24, 58] and micromobility (e-scooter) riding [19]. Integrating the human intelligence, the car driving data can provide us the HCI insights on the AI-driven AV designs, while the micromobility riding enhances our understanding of smart micromobility systems. For each of these case-studies, we have prepared the "unspoken language" from the scenes, i.e., the driving/riding videos and the behavioral sensor measurements from on-board IMUs, as well as the "spoken language", i.e., textual annotations from the human annotators regarding the HMI scenes. We will focus on the understanding and learning of the following two sets of important HMI concepts from these modalities: (a) how the mobility system users interact with the traffic environment (such as road

structures or traffic signals), and (b) how the mobility system interacts with other mobility or traffic participants, such as other vehicles and pedestrians encountered.

The key of extracting the HMI concepts lies in the *co-learning* of multi-modal data (e.g., textual, visual, and behavioral sensor data) to construct the semantic correspondence across these modalities and enhancing the downstream HMI learnability, thereby semantically identifying and extracting the above-mentioned concepts from the human textual annotations. To this end, we need to carefully address the following two important research challenges:

(1) *How to design a unified feature representation to bridge modalities from spoken and unspoken languages for the HMI learning?* In particular, deriving the latent HMI concepts requires fusing the knowledge from heterogeneous modalities from the above-mentioned unspoken and spoken language. Existing behavior learning methods [31, 39, 44] and concept extraction approaches in the natural language processing (NLP) largely considered *representations* dedicated to the individual modality [13, 27, 28, 45], or aggregating the hand-crafted feature vectors from different modalities without further differentiation of their mutual relations. Such single-modal methods or handcrafted feature aggregation designs cannot represent the interactive relations across the HMI observations (e.g., videos regarding the pedestrian on the crosswalk), decision-making (e.g., deceleration of the vehicle), and the human textual annotations. Existing learning approaches, therefore, may not provide unified characterization and representation of their inter-dependencies for HMI, leading to poor interpretation of the complex HMI scenes.

(2) *How to formulate and capture the semantic interactions across modalities in the language and sensor data co-learning?* Language and sensor data co-learning should carefully account for how different modalities are *jointly* associated with the HMI scenes. For instance, simply modeling the detected existence of pedestrians in the video frames or human textual annotations does not fully reflect the pedestrians' interactions with the smart mobility system users in specific contexts (e.g., the driver yields to the pedestrians at the crosswalk). Such interactions lie within the semantic correspondence of certain visual objects, maneuver behaviors, and textual words, and these correspondence relations should be carefully extracted and grouped in order to provide a holistic understanding of the HMI scenes. Without modeling such a correspondence, prior behavior learning [31, 39, 44] and concept extraction approaches [13, 27, 28, 45] may not recognize the intrinsic and implicit interactions between the mobility system users (e.g., car drivers or e-scooter riders) and the mobility environments, yielding low accuracy in retrieving the concepts of the interaction events.

To address the above-mentioned research challenges, we propose `CG-HMI`, a novel cross-modality graph interaction fusion approach to fuse heterogeneous unspoken and spoken language for human-mobility interaction learning. Specifically, as illustrated in Fig. 1, we have designed a novel unified representation called the *human–mobility interaction graph* (HMIG) for each modality related to the HMI scenes, i.e., textual annotations, visual video frames, and behavioral sensor time-series (based on on-board inertial measurement units). Each node of the HMIG corresponds to the words related to HMI concepts, the detected traffic participant/environment categories, or the vehicle maneuver behavior types, in these modalities, while the edges represent their semantic correspondence. This way, the HMIG bridges the spoken and unspoken language, and formulates and captures the semantic interactions across the modalities. Towards HMI learning, we have formulated the HMI concepts as the *named entities* [29], and designed a named entities recognition (NER) approach based on the interaction graphs to extract the concepts. In order to extract the inter-modality and intra-modality semantic correspondences, we have designed a novel approach based on differentiable pooling-based graph attention. Our fusion design groups the graph nodes based on their differentiated semantic correspondence via the differentiable pooling. The resulting graph embeddings are further processed to detect and identify the HMI concepts within the human annotations, which benefits the further downstream HMI learning applications.

In summary, this paper makes the following four major technical contributions:

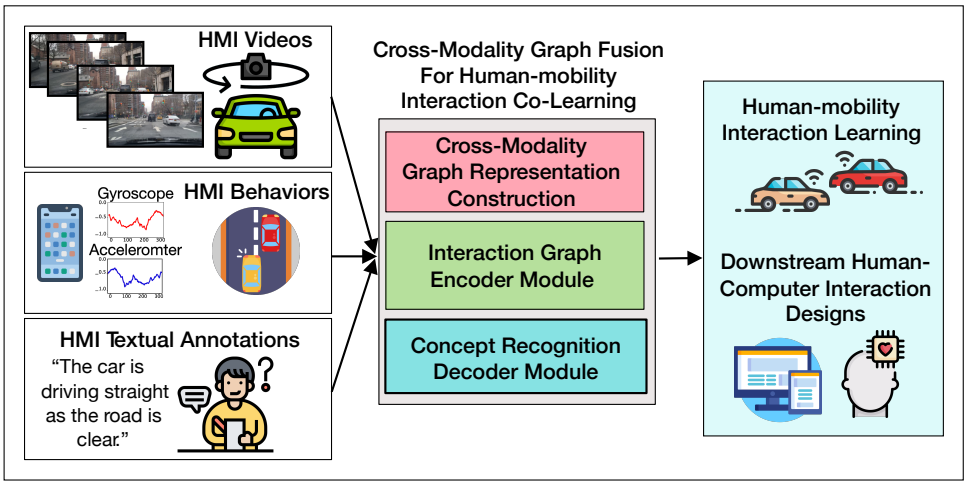


Fig. 1. Motivations of CG-HMI and the enabled potential HMI applications.

(a) **Unified Graph Representation:** We have provided a novel and unified graph representation design, named *human-mobility interaction graph* (HMIG), to model the interactions and correspondences across textual annotations, video frames, and behavioral sensor measurements;
(b) **Human–Mobility Interaction Concept Formulation:** We have formalized the HMI concepts from the human textual annotations to characterize the scenes when the mobility system users (e.g., car drivers) are interacting with complex traffic environments;
(c) **Cross-Modality Graph Fusion:** We have designed a cross-modality graph fusion approach to capture the semantic correspondences and interactions across the modalities in the HMIG for HMI concept learning and extraction. To the best of our knowledge, this is the *first* attempt to study the semantic interactions of multiple modalities for human-mobility interaction (HMI);
(d) **System Implementation and Experimental Studies:** We have developed and implemented the system prototype of CG-HMI. We have conducted extensive experimental studies with three real-world mobility system datasets (two of them are collected on our own on a university campus), i.e., two car driving datasets and one e-scooter riding dataset. We have performed the textual annotations of the HMI scenes with the HMI concepts. Our experimental studies have corroborated the effectiveness and accuracy in identifying and extracting the important HMI concepts (on average over 13% higher accuracy than other baseline approaches).

• **System Overview.** Fig. 2 provides a brief system overview on the information flow, which comprises the following three important stages:

(1) *HMI Data Pre-processing* (Sec. 3): We first prepare the HMI dataset for the CG-HMI model training. In particular, each record of the HMI dataset studied consists of an HMI textual annotation, recorded HMI video frames, and HMI behavioral sensor data (i.e., IMU measurements upon the HMI scenes of the mobility system status). We first pre-process each HMI textual annotation by dividing them into a series of smaller units called *tokens* [52]. For the recorded video of the HMI scene, we perform the object detection [38, 46] to obtain the types of the important HMI objects (e.g., cars, pedestrians, and traffic lights). For the HMI behavioral sensor data, we process the IMU time-series and identify a set of maneuver behaviors [8, 39, 44] (e.g., left turn or acceleration) in the HMI scenes.

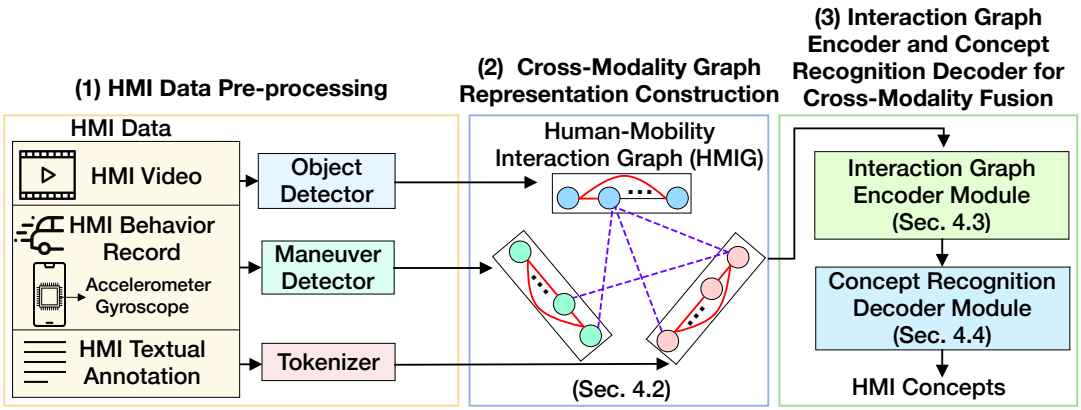


Fig. 2. System overview of CG-HMI that consists of three major stages. The returned HMI concepts can be further fed to other downstream applications such as inferring the causality and accountability of the interaction events.

(2) *Cross-Modality Graph Representation Construction* (Sec. 4.2): In this stage, given the tokenized HMI textual annotation, detected important HMI objects, and identified HMI behaviors, we form the cross-modality graph representation of HMIGs. Each node of HMIG corresponds to a node in textual, visual, and behavioral modalities. To support the HMI concept learning and extraction from the textual annotations, we associate a textual word for each graph node. We then formulate the semantic correspondences and interactions within the HMIG, namely the *inter-modality edges* and *intra-modality edges*.

(3) *Interaction Graph Encoder and Concept Recognition Decoder for Cross-Modality Fusion* (Sec. 4.3): In this stage, CG-HMI leverages the constructed cross-modality HMIGs, and encodes the nodes as well as the edges based on the differentiable pooling-based graph attention network. An interaction graph encoder module first extracts the dependencies within and across the modalities through the graph interaction attention learning. Then, CG-HMI conducts the node cluster assignment to determine the relevance of each node in the HMIG to a certain HMI concept. Furthermore, a concept recognition decoder module converts the graph embeddings into the word embeddings, and identifies our targeted HMI concepts within the textual annotations.

• **Contributions to UbiComp Community.** To the best of our knowledge, this is the *first* work on designing and implementing a language and sensor data co-learning system for understanding ubiquitous human–mobility interactions. Our insights gained from the model designs, system prototyping, and experimental studies will benefit the UbiComp community in the following two perspectives: (1) understanding the human–mobility interactions through cross-modality fusion for emerging smart mobility systems that are equipped with ubiquitous sensing capability; and (2) stimulating more human-centered computing designs and understandable modeling, through a novel language and sensor data co-learning approach, for HMI learning systems (such as ADAS). Our HMI framework can serve as an important building block to enable safer (or more responsible) operations and more enjoyable experience with other important mobility systems (e.g., AVs and other emerging smart micromobility systems). The thus-derived HMI concepts by CG-HMI can further serve as inputs to other downstream ubiquitous computing applications, such as explaining *causality* in the HMI scenes, inferring *accountability*, and enabling *responsibility* analysis in a traffic conflict or an accident event (e.g., auto insurance policy designs and traffic accident analyses from big mobility system data).

The rest of the paper is organized as follows. We first review the related work in Sec. 2. Then, we present the HMI data processing and preparation in Sec. 3, followed by the unified graph representation construction and the

core cross-modality graph interaction fusion designs in Sec. 4. We demonstrate the results of our experimental studies in Sec. 5, discuss the deployment in Sec. 6, and conclude the paper in Sec. 7.

## 2 RELATED WORK

We briefly review the related work in the following two categories.

• **Ubiquitous Human–Mobility Interaction.** Towards smart mobility system designs (such as the emerging AV systems), "spoken language", such as human textual annotations, has started to demonstrate the potential to be fused with the "unspoken language" that is often embedded within the sensor measurements (e.g., visual and behavioral sensors) towards better decision understanding and interpretation of the entire mobility system and its interactions with complex traffic environments/participants. For instance, Kim et al. [24] and Ben et al. [5] studied video-to-text models that generate textual explanations and justifications for the driving decisions. In particular, Kim et al. [24] augmented the BDD100k driving video dataset [58] with the human-generated textual annotations to describe the driving scenarios, and formed the BDD-X dataset that can be used for explanation of the AV modeling process. In addition, to enhance the interpretability of the AV system, Kim et al. [23] designed a model that learns to describe and summarize the visual observations and the actions of the AV system in response to the observations in natural language. Xia et al. [53] leveraged the driver gaze supervision to guide the AI model in finding the prominent parts of the video stream for car speed estimation. Similarly, Kim et al. [22] leveraged human textual advice and guidance to improve the learning and operation of the AV systems. Cao et al. [6] proposed a question-answering framework that can be used to explain the visual inputs in a textual form. Similarly, Acer et al. [3] introduced a model that can provide detailed and purposeful textual information about the surrounding environment (e.g., a driving scene). Furthermore, Zhan et al. [60] combined word contextual representations with temporal dynamics to enhance the subsequent human activity prediction.

`CG-HMI` differs from these prior studies in the following aspects. Aiming to understand the interactions between mobility system users and the mobility environments, `CG-HMI` focuses on identifying and extracting the essential HMI concepts from the human textual annotations by constructing the semantic correspondence across the unspoken (visual and behavioral sensors) and spoken (textual) language. Instead of conventional feature integration, `CG-HMI` provides a novel and unified cross-modality graph representation to bridge these modalities. Our novel designs of fusing the spoken and unspoken language yield accurate identification of the HMI concepts, and can serve as the key enabler for various downstream human-centered and ubiquitous applications in smart mobility systems [23, 42, 44] and HCI [60].

• **Named Entity Recognition (NER).** NER refers to the task of identifying various entity types (e.g., person names, locations, organizations, or other more fine-grained types in specific domains) in the texts [29, 40]. For instance, given the text "Albert Einstein was born in Germany", "Albert Einstein" is recognized as a named entity of "person" type, while "Germany" is identified as the "location" type. NER is important as it could be considered one of the main pre-processing steps for other downstream applications related to texts, such as user input understanding and question-answering models [29]. Existing NER studies largely focus on single-modal (text-only) named entity recognition [13, 27, 28, 45], which may not necessarily adapt to recognizing the HMI concepts that have complex inter-dependencies with multiple modalities. With the increasing association of texts with other important modalities, multi-modal NER [11, 34, 36, 49, 61, 62] often considers compressing the additional modalities (such as images or video frames) into feature vectors for extracting named entities within the texts. Chen et al. [11] analyzed the textual representations with the attention mechanism over the visual modality to reduce the sensitivity of the named entity recognition upon unrelated objects and improve the accuracy of NER. Chen et al. [9] studied the image description generation based on a pre-trained generative model in order to support accurate multi-modal NER. Moon et al. [32] proposed a contrastive learning approach that transforms the IMU sensor readings and the textual annotations upon the videos of human activities into a shared embedding space to enhance information retrieval. Chan et al. [7] investigated the AV trajectory planning

and control by encoding human textual commands with other sensor data fusion. These multi-modal feature integration approaches may not necessarily capture the interactive correspondence across these modalities, and may not scale to other heterogeneous modalities, such as behavioral data in complex HMI scenes.

Unlike the above-mentioned studies, we propose a novel approach of co-learning language and sensor data for ubiquitous mobility sensing, which is based on cross-modality graph interaction fusion to identify and extract the HMI concepts within the human textual annotations. Furthermore, unlike other NER tasks [27, 61] that focus on coarse-grained named entities, we focus on the subtle and implicit HMI concepts that require a model to capture the complex semantic correspondence including the inter- and intra-modality relations. The extracted fine-grained HMI concepts can also deepen our understanding of the HMI scenes and provide the accountability implications within the interactions.

## 3 HMI DATA PREPARATION AND CO-LEARNING PROBLEM DEFINITION

We first overview the HMI datasets used for our `CG-HMI` system development in Sec. 3.1, and then present how we process and obtain the unspoken language – HMI participants and HMI behaviors – within the visual and behavioral sensor data in Sec. 3.2. Then, we present the HMI concepts in the spoken language – human textual annotations – in Sec. 3.3.

### 3.1 HMI Datasets under Study

To gain concrete insights from the HMI studies, we have prepared and harvested the following three datasets.

• **HMI Dataset 1** (DS1): We have processed and prepared the open-sourced BDD-X [24] dataset, which consists of car-driving behaviors (based on the open-source large-scale HMI dataset BDD100K [58]) annotated with the texts. Specifically, each record (sample) in the processed BDD-X dataset corresponds to an HMI video (recorded by the dash-view camera), the IMU measurements with Apple iPhone 5 during driving (i.e., accelerometer and gyroscope), and human-generated textual annotations which, describe the driving behaviors within the video frames.

• **HMI Dataset 2** (DS2): We have also developed our HMI data collection application on an Apple iPhone XR to harvest the visual and behavioral sensor data (Fig. 3(a)) during our daily commute (see Fig. 3(b)) when driving to and from a university campus (situated in a rural area) in North America, and performed the textual annotations of the HMI scenes. Similar to DS1, each record (sample) in DS2 consists of the HMI video (recorded with our smartphone's or dash-view camera), smartphone IMU readings, and the textual annotations regarding the interactions in the video frames.

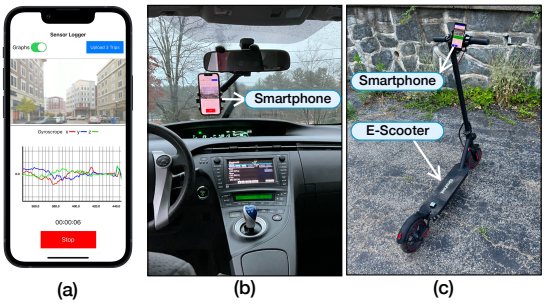

Fig. 3. (a) Our HMI data collection application. Setup for (b) car driving for DS2 and (c) e-scooter riding for DS3.

• **HMI Dataset 3** (DS3): To evaluate the HMI for the micromobility systems, we have collected our own HMI dataset during e-scooter riding similar to DS2 (Fig. 3(c)). We attached a Google Pixel 3 to the handlebars of the e-scooter and the smartphone's main (back-view) camera is used to record the HMI scenes during the daily commute rides on our university campus. Similar to DS1 and DS2, each record in DS3 consists of the HMI video, smartphone IMU readings, and the textual annotations of the video frames. In our e-scooter riding data collection, we have followed the local regulations and riding ethics (e.g., wearing a scooter helmet, avoiding conflicts with pedestrian crowds) during the e-scooter riding.

We further show the records (samples) of the video frames, and IMU sensor readings (i.e., accelerometer and gyroscope), along with the textual annotations of two HMI scenes from DS1 and DS3 in Fig. 4. Specifically, the

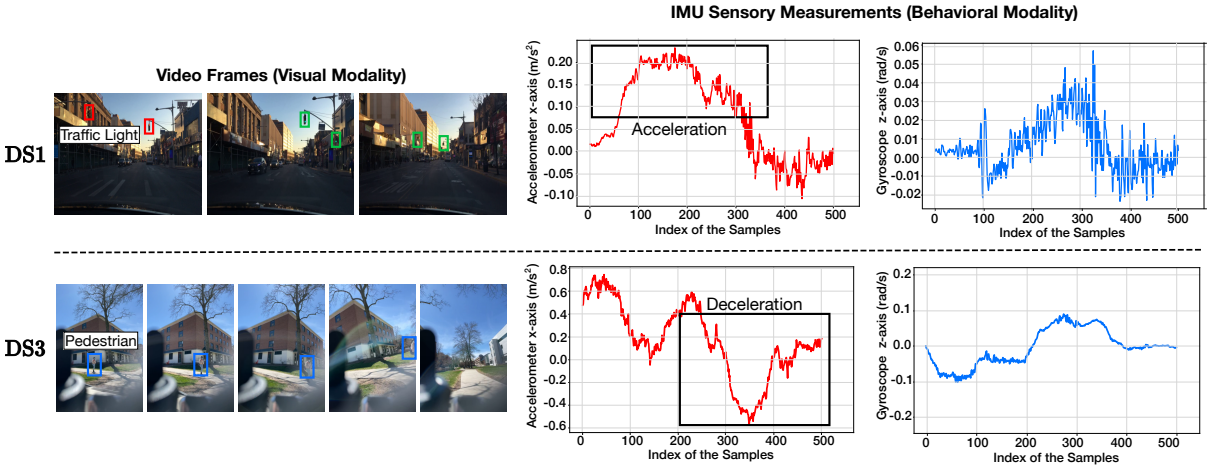


Fig. 4. Visualization of the visual and behavioral sensor modalities of two examples from DS1 (top) and DS3 (bottom) (the reading order of the frame/sensor records is from left to right), where the sentences "The car accelerates because the light has turned green" and "The rider slows down to let the pedestrians cross" are their corresponding textual annotations.

Table 1. Statistics of the detected HMI participants/objects and identified maneuver behaviors from the three datasets.

| Datasets | HMI Participants and Objects | | | | Detected Maneuvers and Behaviors | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Pedestrians | Cars | Traffic Light | Traffic Signs | Left Turn | Right Turn | Straight Driving | Stop | Acceleration | Deceleration |
| DS1 | 701 | 2,611 | 1,133 | 140 | 124 | 94 | 611 | 530 | 271 | 411 |
| DS2 | 379 | 620 | 75 | 84 | 84 | 78 | 260 | 63 | 70 | 188 |
| DS3 | 131 | 791 | 199 | 55 | 87 | 65 | 203 | 137 | 45 | 119 |

first HMI scene shows that as the traffic light turns from red to green (highlighted in the frames), the car starts to accelerate, which is reflected by the accelerometer time-series along the x-axis (i.e., forward direction) and depicted in the textual annotations. In the second HMI scene, the e-scooter rider starts to decelerate as s/he sees a pedestrian on the left, which is similarly reflected in the accelerometer time-series as well as the textual annotations. Note that we collected DS2 and DS3 in the naturalistic driving/riding settings, i.e., the data collection was performed during the daily commutes without experimental control, and followed the local traffic rules and social norms (e.g., minimum interference with the other traffic participants). Our smartphone-based data collection was unobtrusive, and did not interfere with the naturalistic vehicle driving and e-scooter riding as well as the nearby local road traffic.

## 3.2 Unspoken Language within Visual and Behavioral Sensor Data of HMI Scenes

• **HMI Participants and Objects from Visual Data.** For the given HMI video, we have implemented the YOLOv5 object detection algorithm [38, 46] to identify a total of $V$ different objects classes within each HMI video frame, i.e., pedestrians, motorized vehicles (e.g., cars, buses, and trucks), traffic lights, and traffic signs. In our data preparation, we process and extract each HMI scene, and each scene lasts 9s on average. We have further shown the statistics of the detected HMI participants and objects of the three datasets in Table 1.

• **HMI Behaviors from IMU Sensor Data.** The behavioral sensor data can serve as another unspoken language to express the HMI scenes. In this prototype HMI study, we focus on the IMU measurements (e.g., accelerometer and gyroscope) collected from the smartphones during car driving or e-scooter riding. We have identified and labeled the HMI behaviors of the mobility system users based on the on-board IMU measurements from DS1, DS2, and DS3. We have implemented an efficient IMU-based maneuver behavior identification algorithm

to process the time-series data of the accelerometer and gyroscope readings, and identify multiple different driver/rider maneuver classes [8, 39, 44]. We have identified the following $\mathcal{X} = 6$ maneuver classes: left turns, right turns, straight driving, stops, acceleration, and deceleration. We further illustrate the statistics of the detected maneuvers of DS1, DS2, and DS3 in Table 1. We note that this pilot study focuses on leveraging the IMUs for determining the interaction behaviors, since the mobile/vehicle-equipped IMUs can provide fine-grained maneuver behavior data and enable the ubiquitous computing applications, especially under GPS-less urban environments. The design of our CG-HMI is also general, and can be extended to other sensing modalities (e.g., LiDAR) for the more fine-grained analysis.

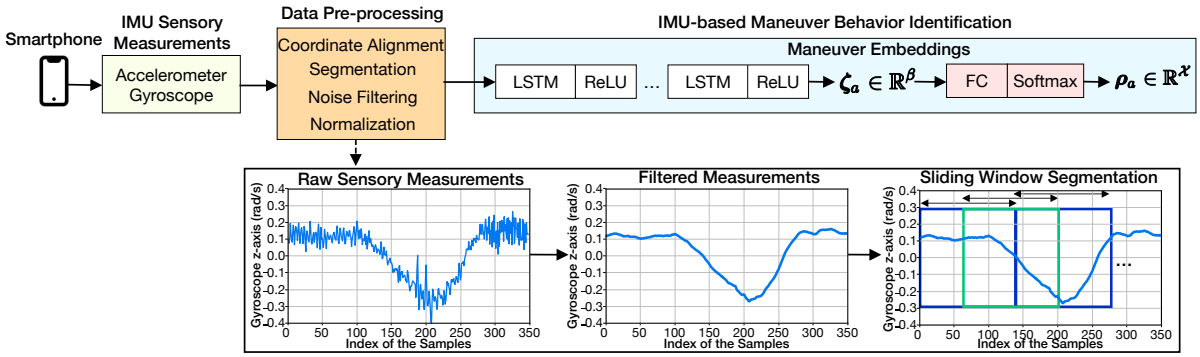


Fig. 5. Overview of behavioral sensor data processing and behavior recognition from the HMI scenes.

We illustrate the IMU-based maneuver identification in Fig. 5 for both the car driving and e-scooter riding. In particular, given the raw time-series of the accelerometer and gyroscope, we first perform the coordinate alignment to convert the sensor readings from the smartphone's coordinate system (local coordinate system) to that of the vehicle/e-scooter (global coordinate system). Then, we segment them with a sliding window of size $W$ (10s in this study) and 50% overlap. We further filter the sensor noise caused by the vibrations or the inherent imprecision of the sensors, and normalize the data to the range of [-1,1] [8]. For each $a$-th sample in our harvested HMI dataset, let $\mathbf{A}_a \in \mathbb{R}^{3\times W}$ and $\mathbf{R}_a \in \mathbb{R}^{3\times W}$ be the pre-processed accelerometer and gyroscope time-series segments with length $W$ along the $x$, $y$, and $z$ axes. Then, to capture the temporal information within each of the sensor time-series segments, we process the segments based on a total of $B_0$ consecutive long short-term memory (LSTM) layers [31] with $\beta$ hidden units and the rectified linear units (ReLU) activation function. Such a design adds non-linearity and generates the $\beta$-D maneuver embeddings $\boldsymbol{\zeta}_a \in \mathbb{R}^{\beta}$, i.e.,

$$\boldsymbol{\zeta}_a = \mathtt{LSTM}_{B_0}(\ldots(\mathtt{LSTM}_1(\mathbf{A}_a, \mathbf{R}_a))). \tag{1}$$

$\boldsymbol{\zeta}_a$ is further processed by a fully connected (FC) layer with $\mathcal{X}$ hidden units and the Softmax activation function to output the probabilities $\boldsymbol{\rho}_a$ for each of the $\mathcal{X}$ maneuver classes in our study, i.e.,

$$\boldsymbol{\rho}_a = [\rho_a[1], \ldots, \rho_a[\mathcal{X}]] \in \mathbb{R}^{\mathcal{X}}, \tag{2}$$

where

$$\rho_a[i] = \frac{\exp(\rho_a[i])}{\sum_{k=1}^{\mathcal{X}} \exp(\rho_a[k])}, \tag{3}$$

and $\rho_a[i]$ represents the probability of the $i$-th maneuver class being detected in the time-series segment. We denote the set of the detected maneuvers in the $a$-th sample through the above procedures as $\mathbf{B}_a$.

### 3.3 Spoken Language on HMI Concepts

We have defined and labeled the *HMI concepts* from the human textual annotations for our language and sensor data co-learning, which characterize the contexts and decision-making processes within the HMI scenes. Specifically, we focus on the following five important concepts related to HMI: (i) mobility system status (such as the left/right turn decisions by the drivers); (ii) traffic environment status (e.g., the contexts of clear way, or heavy traffic); (iii) interactions with other vehicles (for instance, passing a slow vehicle or allowing the front vehicle to merge); (iv) interactions with pedestrians (e.g., stop when a pedestrian is crossing, or signaled by a pedestrian); and (v) interactions with the road infrastructures (such as traffic lights, stop signs, and speed bumps).

Fig. 6 illustrates five examples of these different HMI concepts. Note that the extraction and recognition of the HMI concepts of CG-HMI cannot rely on simple word matching, since one word may relate to multiple different contexts and imply different semantics in different HMI concepts. Instead, CG-HMI aims to capture the sequences of multiple words inside the human textual annotations that are interconnected, thus enabling the comprehensive understanding of HMI scenes.

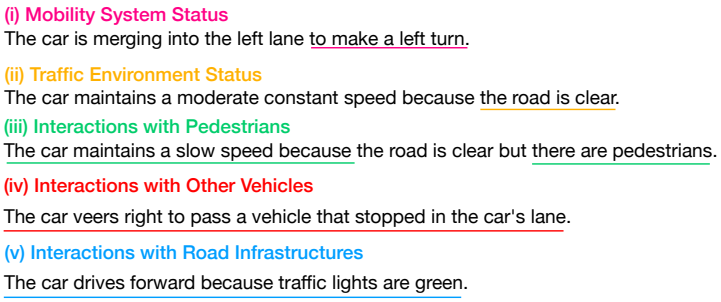


Fig. 6. Human textual annotation examples from DS1 with each of the 5 HMI concept classes.

In summary, we have identified and labeled a total of 5,793, 807, and 1,073 HMI concepts from DS1, DS2, and DS3, respectively. We further illustrate the distributions of HMI concepts in DS1, DS2, and DS3 in Fig. 7. Note that a car driver in DS1 and DS2 might encounter more vehicle traffic but fewer pedestrians than the e-scooter rider in DS3.

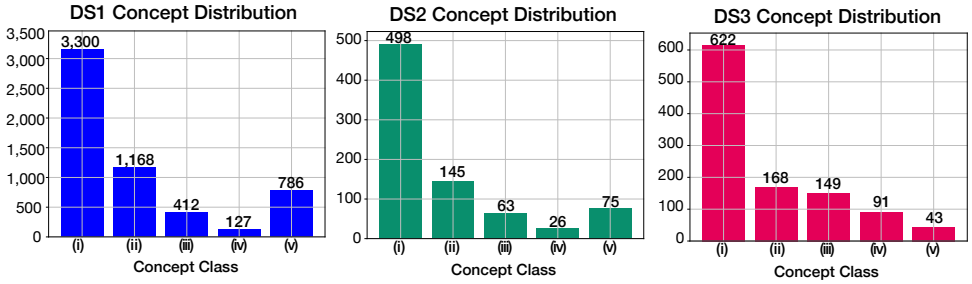


Fig. 7. Distribution of the different HMI concept classes (labeled in (i)–(v)) for DS1, DS2, and DS3.

Since the HMI annotations may be of different lengths, our pre-processing will break each sentence into smaller units. Then, CG-HMI learns the relations of different units, and assigns the labels of HMI concepts to the units in the HMI annotations, which follows the existing NLP practices [52]. Specifically, we convert each sentence regarding the HMI scenes of a sample $a$ into a series of tokens, i.e., $\mathrm{S}_a = [s_{a,1}, \ldots, s_{a,L}]$, with a maximum length

of $L$ ($L = 128$ in this study since the textual annotations are all shorter than 128 words). If a tokenized textual annotation has tokens of length shorter than $L$, CG-HMI pads it with the null tokens to the length of $L$ [52]. Here $s_{a,i}$ denotes the $i$-th token of the $a$-th textual annotation $\mathbf{S}_a$.

To ease the language and sensor data co-learning, for each HMI textual annotation $\mathbf{S}_a$, we provide the labels of HMI concepts, i.e., $\mathbf{Y}_a \in \mathbb{R}^{K \times L}$, where $K$ is the number of HMI concept classes and $L$ is the number of tokens in the textual annotations. Each element $\mathbf{Y}_a[i, j] = 1$ represents that the $j$-th token $s_{a,j}$ ($j \in \{1, \ldots, L\}$) belongs to the $i$-th HMI concept type, and $\mathbf{Y}_a[i, j] = 0$ otherwise. Fig. 8 illustrates a labeled example of $\mathbf{Y}_a$, where $\mathbf{Y}_a[2, 1] = 1$ denotes that the word "slowly" belongs to the HMI concept of "mobility system status".

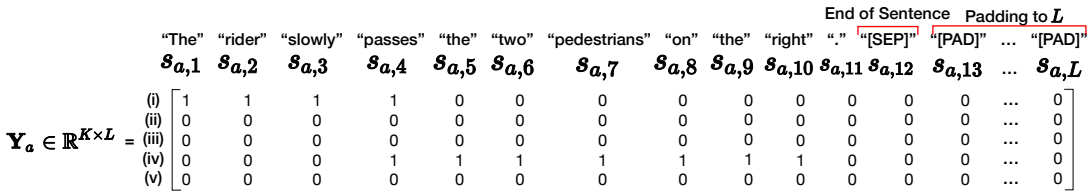


Fig. 8. Illustration of the labeled HMI concepts in a sentence of textual annotation.

• **Problem Statement.** We define the task of CG-HMI as: given a tokenized HMI textual annotation $\mathbf{S}_a$, the related HMI objects $\mathbf{P}_a$, and identified HMI behaviors $\mathbf{B}_a$ (e.g., maneuver behaviors) from the $a$-th HMI data sample, the goal of the HMI language and sensor data co-learning by CG-HMI is to determine the types and positions of a set of HMI concepts $\mathbf{Y}_a$ in $\mathbf{S}_a$.

## 4 CORE MODEL DESIGNS FOR CO-LEARNING LANGUAGE AND SENSOR DATA

We first overview the module designs of CG-HMI in Sec. 4.1. Based on the processed unspoken and spoken language, we present the designs of cross-modality graph representation in Sec. 4.2. Then, we discuss the detailed designs of the graph encoder and concept decoder in Secs. 4.3 and 4.4, respectively.

### 4.1 Model Overview of CG-HMI

The overall architecture of CG-HMI is further illustrated in Fig. 9. CG-HMI first constructs the human-mobility interaction graph (HMIG) to characterize the semantic correspondences among different modalities (Sec. 4.2). Based on all the HMIGs formed from the HMI scenes, CG-HMI aims to capture the semantic correspondences across these modalities, and capture their HMI concepts. CG-HMI first processes the HMIG with the cross-modality interaction graph encoder (GIA) module (Sec. 4.3) to generate the node embeddings. The key idea of this module is to leverage a novel graph attention network with differentiable pooling (DiffPool) to differentiate the HMIG edges, learn the latent semantic correspondence, and generate the node embeddings of the HMIG. Then, the concept recognition decoder module (Sec. 4.4) further decodes the node embeddings, and processes them through the bidirectional long short-term memory (BiLSTM). Finally, CG-HMI returns the scores of the words in the textual annotations and determines their relevance to each HMI concept, yielding the recognized HMI concepts.

### 4.2 Cross-Modality Graph Representation

• **Design Motivations.** Given the unspoken (i.e., visual and behavioral) and spoken (i.e., textual) languages of the HMI scenes, we propose the construction of a cross-modality human–mobility interaction graph (HMIG) in order to provide a unified representation to capture the semantic correspondence. Such a graph structure characterizes the *interactive* relations of the different modalities, which are further retrieved to determine the HMI concept types in the HMI scene.

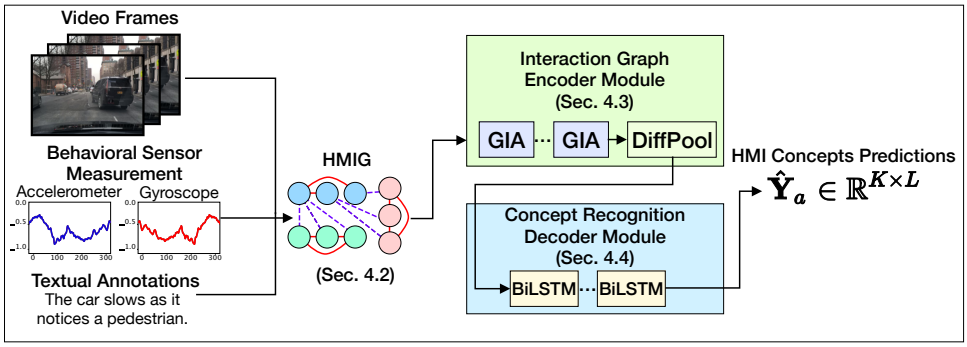


Fig. 9. Overview of the system workflow of CG-HMI.

• **Detailed Designs.** Our HMIG consists of modality nodes and intra-/inter-modality edges for the language and sensor data co-learning. We transform the $a$-th textual annotation $\mathbf{S}_a$, along with the set of HMI objects $\mathbf{P}_a$ and maneuver behaviors $\mathbf{B}_a$, into a graph representation $\mathbb{G}_a$. The formulation of $\mathbb{G}_a$ is detailed next.

**(a) Modality Nodes:** Each HMIG $\mathbb{G}_a$ consists of three types of nodes: visual, behavioral, and textual. Based on the pre-processed unspoken language in Sec. 3.2, each HMIG in this study consists of $V = 4$ *visual nodes* that correspond to the four types of traffic participants and objects of interest: pedestrians, cars, traffic lights, and traffic signs. In addition, we have a total of $M = 6$ *behavioral nodes* that represent the six maneuver classes (e.g., left/right turns) detected in the HMI scene, and $L$ *textual nodes* that represent the tokenized textual annotations. Thus, we form $\mathbb{G}_a$ that has a total of $T = (V + M + L)$ nodes for CG-HMI to learn their mutual semantic correspondences.

**(b) Intra-/Inter-modality Edges:** Our HMIG aims to capture: (i) dependencies of nodes within the same modality, and (ii) interactions of nodes across different modalities in the HMI scene. For instance, when CG-HMI identified the traffic lights and pedestrians in the HMI scene (say, when the car stops at the intersection), CG-HMI learns the intra-modality edges between the related nodes within the visual and textual categories. In the meantime, CG-HMI associates the inter-modality edges between the above-mentioned nodes with the behavior nodes (say, the "left turn" node). With the above edges, CG-HMI formulates the HMIG into an adjacency matrix of $\mathbb{G}_a$, denoted as $\mathbf{C}_a$, to characterize these intra-/inter-modality edges, and hence derives the HMI concepts from the textual annotations. To this end, we consider $\mathbb{G}_a$ as a weighted graph, where each matrix element $\mathbf{C}_a[i, j] \in [0, 1]$ represents the *semantic correspondence*, i.e., interdependencies in the annotated HMI concepts, for each pair of $i$-th and $j$-th nodes. $\mathbf{C}_a[i, j]$ approaches 1 when two nodes have strong semantic correspondence, and 0 otherwise. In the model initialization, CG-HMI sets the edge weight between the nodes $i$ and $j$ as $\mathbf{C}_a[i, j] = 1$.

Based on the above, we illustrate a concrete example of the constructed HMIG of an HMI scene in Fig. 10. Note that the inter-modality edges are formed across the visual nodes, "car" and "traffic light", and the behavioral node "left turn", indicating their semantic correspondences in such an HMI scene. If certain objects or maneuvers are not detected in the HMI scene, their corresponding nodes in the HMIG are isolated without any edge.

## 4.3 Interaction Graph Encoder Designs

• **Design Motivations.** Our interaction graph encoder module aims to identify the most important neighboring nodes within and across the modalities in the HMIG to generate the node embeddings. Within this module, we will design a differentiable pooling-based graph attention network to determine the semantic correspondences between different nodes. Our differentiable pooling-based graph attention network performs node clustering assignment operations, which clusters the modality nodes into $L$ clusters. This way, this module ensures that the relevant features across different modalities are aggregated, and contribute to the final node embeddings and HMI concept recognition.

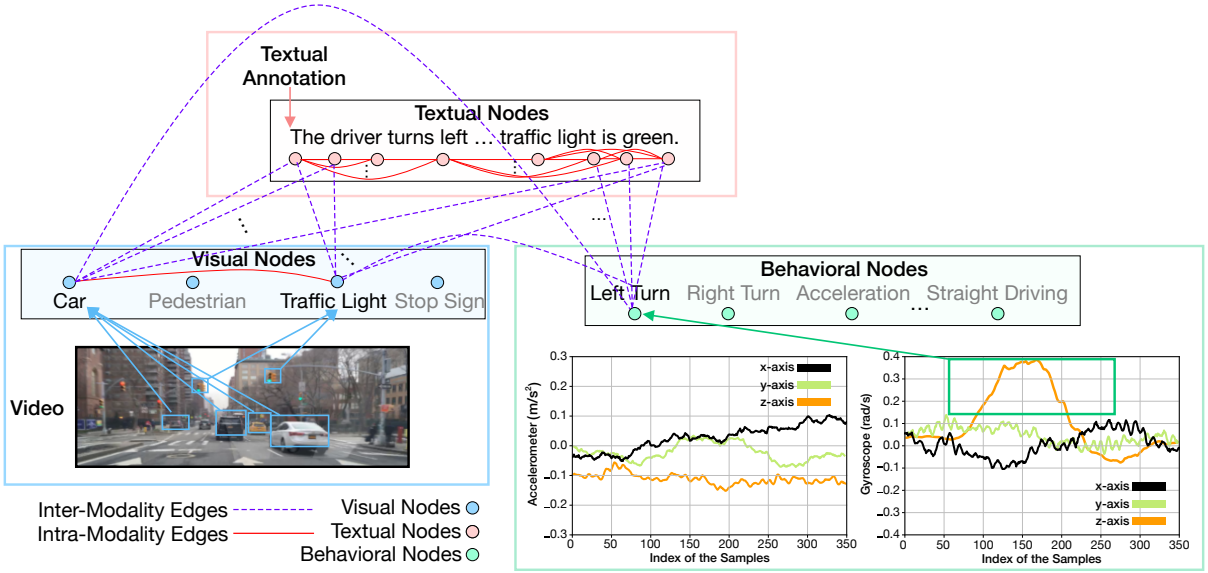


Fig. 10. Illustration of the cross-modality human-mobility interaction graph (HMIG).

• **Detailed Designs.** We have designed a differentiable pooling-based graph attention network to extract the node embeddings. As illustrated in Figs. 11 and 12, this module performs the following two major operations, i.e., (a) *graph feature encoding and interaction attention*, and (b) *node cluster assignment and embedding generation*. The details of each operation are given as follows.

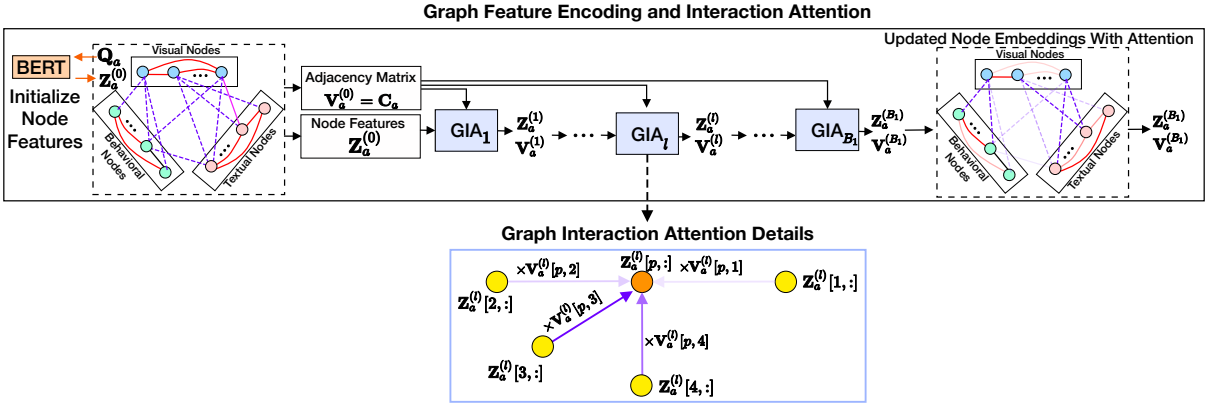


Fig. 11. Details of the graph feature encoding and interaction attention architecture in CG-HMI.

**(a) Graph Feature Encoding and Interaction Attention**. Fig. 11 illustrates the details of our architecture. In particular, given the HMIG $\mathbb{G}_a$, we associate its nodes with a set of words denoted by $\mathbf{Q}_a$, e.g., "traffic light" or "left turn" for visual and behavioral nodes, and the sentence tokens $\mathbf{S}_a$ for the textual nodes. Thus, to initialize the node features of $\mathbb{G}_a$, denoted by $\mathbf{Z}_a^{(0)}$, we feed words $\mathbf{Q}_a$ to the Bidirectional Encoder Representations from Transformers (BERT) [14] and use its output word embeddings as the initial node features, i.e., $\mathbf{Z}_a^{(0)} = \text{BERT}(\mathbf{Q}_a)$, where $\mathbf{Z}_a^{(0)} \in \mathbb{R}^{T \times \phi^{(0)}}$. Furthermore, since BERT is pre-trained on a large corpus, it can generate contextual embeddings that are useful to capture the semantic dependencies between the nodes.

Given the feature embeddings from all modalities, $\mathbf{Z}_a^{(0)}$, in order to extract the semantic correspondences, CG-HMI stacks a total of $B_1$ graph attention layers [48]. At each layer $l$, CG-HMI takes in the node features, $\mathbf{Z}_a^{(l)} \in \mathbb{R}^{T\times\phi^{(l)}}$, and the adjacency matrix, $\mathbf{C}_a \in \mathbb{R}^{T\times T}$. Let $\mathcal{N}_p$ be the neighborhood of a node $p$ in the HMIG according to the adjacency matrix $\mathbf{C}_a$, i.e., the nodes that have intra-/inter-modality edges from $p$ as well as $p$ itself. Each graph interaction attention layer (denoted as GIA) finds the graph interaction attention weights between two nodes $p$ and $q$ in the HMIG, denoted as $\mathbf{V}_a^{(l)} \in \mathbb{R}^{\phi^{(l)}\times\phi^{(l)}}$, based on a Softmax function, i.e.,

$$\mathbf{V}_a^{(l)}[p,q] = \texttt{Softmax}(\psi[p,q]) = \frac{\exp(\psi[p,q])}{\sum_{k\in\mathcal{N}_p}\exp(\psi[p,k])}, \quad p,q \in \{1,\ldots,T\}, \tag{4}$$

where the score of edge feature embeddings between nodes $p$ and $q$ is given by

$$\psi[p,q] \triangleq \sigma\left(\mathbf{W}_v^{\intercal}\cdot\left[\left(\mathbf{Z}_a^{(l)}[p,:]\cdot\mathbf{W}_g\right)\Big\|\left(\mathbf{Z}_a^{(l)}[q,:]\cdot\mathbf{W}_g\right)\right]\right), \tag{5}$$

and $\mathbf{Z}_a^{(l)}[p,:]$ and $\mathbf{Z}_a^{(l)}[q,:]$ represent the features of the $p$-th and $q$-th nodes, respectively, and $\mathbf{W}_v \in \mathbb{R}^{2\cdot\phi^{(l)}}$ and $\mathbf{W}_g \in \mathbb{R}^{\phi^{(l)}\times\phi^{(l)}}$ are trainable weight matrices. In addition, $\sigma(\cdot)$ denotes the activation function (LeakyReLU in our study), $\|$ represents the concatenation operation, and $(\cdot)^{\intercal}$ represents the matrix transpose operation. Eq. (4) characterizes how important the interdependencies between the nodes $p$ and $q$ are in the related HMI scenes, thus capturing their semantic correspondences. For instance, Fig. 11 shows that for a node $p$, the colors of the edges imply the relative stronger importance $\mathbf{V}_a^{(l)}[p,3]$ of its neighbor $q=3$ (with the feature $\mathbf{Z}_a^{(l)}[3,:]$).

We then conduct the graph convolution based on $\mathbf{V}_a^{(l)}$, and have

$$\mathbf{Z}_a^{(l+1)} = \mathbf{V}_a^{(l)}\cdot\mathbf{Z}_a^{(l)}\cdot\mathbf{W}_g + b_g, \tag{6}$$

where $\mathbf{Z}_a^{(l+1)}$ is the output of GIA, and $\mathbf{W}_g$ and $\mathbf{b}_g$ are trainable weight matrix and the bias, respectively.

We further denote all operations in Eqs. (4), (5), and (6) (see Fig. 11) as the function of $\texttt{GIA}(\cdot)$, i.e.,

$$\left[\mathbf{Z}_a^{(l+1)}, \mathbf{V}_a^{(l+1)}\right] = \texttt{GIA}\left(\mathbf{Z}_a^{(l)}, \mathbf{V}_a^{(l)}\right). \tag{7}$$

Given the above, we recursively leverage a total of $B_1$ GIA layers within CG-HMI to extract the node features and edge weights, i.e.,

$$\left[\mathbf{Z}_a^{(B_1)}, \mathbf{V}_a^{(B_1)}\right] = \texttt{GIA}_{B_1}\left(\ldots\left(\texttt{GIA}_1\left(\mathbf{Z}_a^{(0)}, \mathbf{V}_a^{(0)}\right)\right)\right). \tag{8}$$

Note that $\mathbf{V}_a^{(0)} = \mathbf{C}_a$ at the first GIA layer. In the subsequent GIA layers, the matrix $\mathbf{V}_a^{(l)}$ represents the weights of edges across all nodes and differs from the input $\mathbf{C}_a$.

**(b) Node Cluster Assignment and Embedding Generation.** We illustrate the details of the node cluster assignment and embedding generation in Fig. 12. Specifically, given the node features $\mathbf{Z}_a^{(B_1)} \in \mathbb{R}^{T\times\phi^{(B_1)}}$ and the weighted adjacency matrix $\mathbf{V}_a^{B_1} \in \mathbb{R}^{T\times T}$, CG-HMI further performs the differentiable pooling [57] to determine the clusters of nodes that have strong relevance with each other. This way, CG-HMI differentiates and identifies the node clusters that demonstrate strong semantic interactions. For instance, Fig. 12 illustrates that the first through the fourth nodes are fused by the differentiable pooling operation due to their interdependencies and semantic correspondences.

In particular, given the outputs of GIA, we perform the Laplacian smoothing [25] to aggregate the local information of each node to generate the embeddings $\mathbf{H}_a^{(B_1)} \in \mathbb{R}^{L\times\omega}$, i.e.,

$$\mathbf{H}_a^{(B_1)} = \left(\hat{\mathbf{D}}_a^{(B_1)}\right)^{-\frac{1}{2}}\cdot\left(\mathbf{V}_a^{(B_1)} + \mathbf{I}\right)\cdot\left(\hat{\mathbf{D}}_a^{(B_1)}\right)^{-\frac{1}{2}}\cdot\mathbf{Z}_a^{(B_1)}\cdot\mathbf{W}_e + \mathbf{b}_e, \tag{9}$$

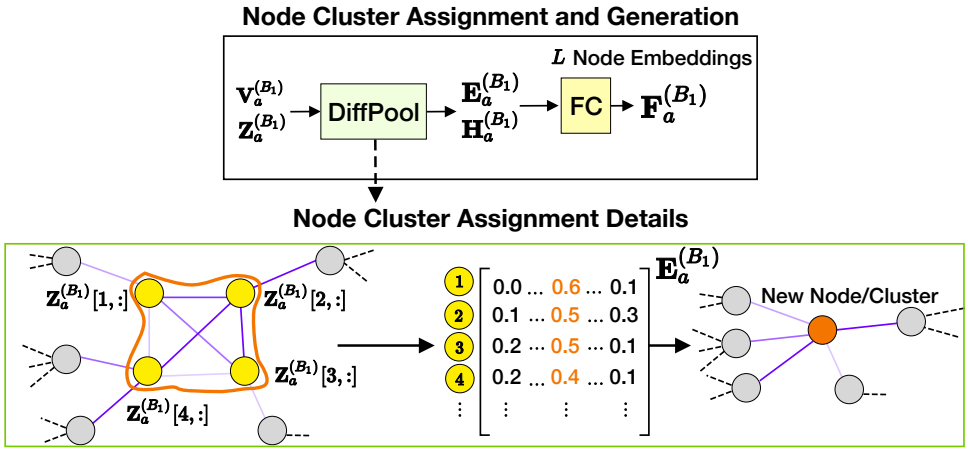


Fig. 12. Details of node cluster assignment and embedding generation in CG-HMI.

where $\hat{\mathbf{D}}_a^{(B_1)} \in \mathbb{R}^{T\times T}$ represents the diagonal degree matrix, i.e., $\hat{\mathbf{D}}_a^{(B_1)}[i,i] = \sum_j \mathbf{V}_a^{(B_1)}[i,j]$. In addition, $\mathbf{V}_a^{(B_1)} + \mathbf{I}$ imposes the self-loop operation [51] to each node with the identity matrix $\mathbf{I} \in \mathbb{R}^{T\times T}$, and $\mathbf{W}_e \in \mathbb{R}^{\omega\times\omega}$ and $\mathbf{b}_e \in \mathbb{R}^{\omega}$ are its trainable weight matrix and the bias with $\omega$ hidden units to generate the embeddings.

Then, for each node in HMIG, we find the assignment score that represents the importance of being assigned to a node cluster. As shown in Fig. 12, CG-HMI determines the assignment matrix $\mathbf{E}_a^{(B_1)} \in \mathbb{R}^{T\times L}$, where $\mathbf{E}_a^{(B_1)}[p,c]$ represents the assignment score that a node $p \in \{1,\ldots,T\}$ be assigned to the cluster $c \in \{1,\ldots,L\}$. In other words, $\mathbf{E}_a^{(B_1)}[p,c] = 1$ if the node $p$ should be assigned to cluster $c$, and 0 otherwise.

Specifically, we calculate $\mathbf{E}_a^{(B_1)}$ by applying the Softmax function in a row-wise manner on another Laplacian smoothing output to generate the probability of each node $p$ being assigned to each $c$ of the $L$ clusters, i.e.,

$$\mathbf{E}_a^{(B_1)}[p,c] = \texttt{Softmax}\left(\mathbf{X}_a^{(B_1)}[p,c]\right) = \frac{\exp\left(\mathbf{X}_a^{(B_1)}[p,c]\right)}{\sum_{k=1}^{L}\exp\left(\mathbf{X}_a^{(B_1)}[p,k]\right)}, \quad p \in \{1,\ldots,T\}, \quad c \in \{1,\ldots,L\}, \tag{10}$$

where the score of the feature embeddings $\mathbf{X}_a^{(B_1)}$ is formally given by

$$\mathbf{X}_a^{(B_1)} = \left(\hat{\mathbf{D}}_a^{(B_1)}\right)^{-1/2} \cdot \left(\mathbf{V}_a^{(B_1)} + \mathbf{I}\right) \cdot \left(\hat{\mathbf{D}}_a^{(B_1)}\right)^{-1/2} \cdot \mathbf{Z}_a^{(B_1)} \cdot \mathbf{W}_c + \mathbf{b}_c, \tag{11}$$

and $\mathbf{W}_c \in \mathbb{R}^{\omega\times\omega}$ and $\mathbf{b}_c \in \mathbb{R}^{\omega}$ are the trainable parameters.

After the node cluster assignment, we further combine the node cluster assignment scores $\mathbf{E}_a^{(B_1)}$ and the intermediate node embeddings $\mathbf{H}_a^{(B_1)}$ through an FC layer (with $\mu$ hidden units). Specifically, CG-HMI finds the updated node embeddings $\mathbf{F}_a^{(B_1)} \in \mathbb{R}^{L\times\mu}$, i.e.,

$$\mathbf{F}_a^{(B_1)} = \texttt{FC}\left(\left(\mathbf{E}_a^{(B_1)}\right)^{\mathsf{T}} \cdot \mathbf{H}_a^{(B_1)}\right), \tag{12}$$

which further differentiates the features of the nodes assigned. Given the node cluster assignment $\mathbf{E}_a^{(B_1)}$, we further calculate the updated matrix of GIA weights, denoted as $\mathring{\mathbf{V}}_a^{(B_1)} \in \mathbb{R}^{L\times L}$, i.e.,

$$\mathring{\mathbf{V}}_a^{(B_1)} = \left(\mathbf{E}_a^{(B_1)}\right)^{\mathsf{T}} \cdot \mathbf{V}_a^{(B_1)} \cdot \mathbf{E}_a^{(B_1)}. \tag{13}$$

### 4.4 HMI Concept Recognition Decoder Designs

• **Design Motivations.** We illustrate the module designs of our concept recognition decoder in Fig. 13. Based on the node embeddings, the concept recognition decoder further derives the *relevance score* of each word with respect to the HMI concept classes.

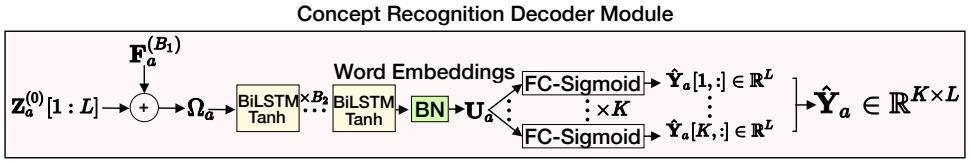


Fig. 13. Details of the concept recognition decoder module.

This module leverages the combination of the node embeddings and the initial textual word embeddings to ensure the critical information from the textual information can be retrieved. Furthermore, we note that HMI concepts within the textual annotations may exhibit sequential dependencies. For instance, the car decelerates when the driver observes the encounter with a pedestrian. Therefore, we have further designed the bi-directional long short-term memory (`BiLSTM`) to capture the sequential dependencies. The bi-directional design also helps capture the correlations when the cause (e.g., the encounter with pedestrians) and the effect (e.g., the car decelerates) appears in an arbitrary order inside the textual annotations.

• **Detailed Designs.** In particular, the concept recognition decoder module first takes in the initial textual node embeddings $\mathbf{Z}_a^{(0)}[1:L] \in \mathbb{R}^{L\times\mu}$ and feeds them to the node embeddings $\mathbf{F}_a^{(B_1)} \in \mathbb{R}^{L\times\mu}$ that is returned from the interaction graph encoder module (Eq. (12)). This way, we obtain $\mathbf{\Omega}_a \in \mathbb{R}^{L\times\mu}$, i.e.,

$$\mathbf{\Omega}_a = \mathbf{Z}_a^{(0)}[1:L] \oplus \mathbf{F}_a^{(B_1)}, \tag{14}$$

where $\oplus$ represents the element-wise addition operation. Then, `CG-HMI` stacks a total of $B_2$ `BiLSTM` layers, each of which has $\eta$ hidden units and `Tanh` as the activation function, as well as a batch normalization (`BN`) layer to further decode $\mathbf{\Omega}_a$. The resulting word embeddings $\mathbf{U}_a \in \mathbb{R}^{\eta}$ characterize the temporal dependencies across the input node embeddings, i.e.,

$$\mathbf{U}_a = \texttt{BN}(\texttt{BiLSTM}_{B_2}(...\texttt{BN}(\texttt{BiLSTM}_1((\mathbf{\Omega}_a))))). \tag{15}$$

Given the word embeddings $\mathbf{U}_a \in \mathbb{R}^{\eta}$, `CG-HMI` simultaneously feeds them through $K$ parallel fully-connected layers (`FC`) with the `Sigmoid` activation function. Each of the `FC` layer has a total of $L$ hidden units (i.e., one estimation per word) to generate the relevance score of each word for each of the $K$ HMI concept classes. Each row $\hat{\mathbf{Y}}_a[i,:] \in \mathbb{R}^L$ corresponds to the $i$-th concept class, and is given by

$$\hat{\mathbf{Y}}_a[i,:] = \texttt{FC}(\mathbf{U}_a), \quad i \in \{1,\ldots,K\}. \tag{16}$$

The model training of `CG-HMI` is presented as follows. Let $\hat{\mathbf{Y}}_a[i,:] \in \mathbb{R}^L$ be the predicted *relevance scores* of each of the $L$ tokens for the $i$-th concept, and $\mathbf{Y}_a[i,:] \in \mathbb{R}^L$ as their corresponding ground-truth labels. We adopt a relevance score threshold parameter $\gamma$ to convert the predicted relevance scores $\hat{\mathbf{Y}}_a$ to 1 if they are higher than $\gamma$, and to 0 otherwise. The training process of `CG-HMI` aims to minimize the binary cross-entropy loss between the prediction $\hat{\mathbf{Y}}_a[i,:]$ and the ground-truth $\mathbf{Y}_a[i,:]$, i.e.,

$$\mathcal{L} \triangleq -\sum_{i=1}^{K}\sum_{j=1}^{L}\left(\mathbf{Y}_a[i,j]\cdot\log\left(\hat{\mathbf{Y}}_a[i,j]\right)\right) + (1-\mathbf{Y}_a[i,j])\cdot\log\left(1-\hat{\mathbf{Y}}_a[i,j]\right). \tag{17}$$

# 5 EXPERIMENTAL STUDIES

We first present the experimental settings in Sec. 5.1, followed by the experimental results in Sec. 5.2.

## 5.1 Experimental Settings

• **Baselines for Performance Comparison:** We compare our proposed CG-HMI with the following baselines and state-of-the-art approaches:

(1) BCRF [21, 40]: which takes in the textual modality and recognizes the HMI concepts based on LSTM and conditional random filed (CRF);
(2) BCNN [12]: which extracts the character-level features through convolutional neural networks (CNNs), and then adopts LSTM for HMI concept recognition;
(3) BTrans [14]: which leverages the bidirectional transformers architecture to find the word embeddings for HMI concept recognition;
(4) HAN [56]: which implements the hierarchical attention network (HAN) to recognize the HMI concepts;
(5) UMGF [61]: which adapts the graph neural network approach in [61] to integrate the textual and visual data for the HMI concept recognition;
(6) FEC [34, 36]: which leverages the concatenation of the visual and textual feature embeddings for the multi-modal HMI concept recognition;
(7) RPAtt [41]: which implements an attention mechanism based on the textual-visual relation propagation to recognize HMI concepts.
(8) DAT [55]: which integrates a direction-aware mechanism within multiple transformer encoders to process each of the visual, behavioral, and textural modalities.
(9) UMT [59]: which implements a multi-modal transformer network with multi-modal attention to capture and extract the interactions from the visual, behavioral, and textual modalities.

• **Parameter Settings and Evaluation Environment Setup:** Unless otherwise stated, we use the following parameter settings by default. To train our model, we use 90% of each dataset for model training and validation, and the rest 10% for testing. We leverage the Adam optimizer with a learning rate of 0.001. For maneuver behavior identification, we use $B_0 = 1$ layer of LSTM with $\beta = 64$ hidden units to generate the maneuver embeddings. In preparing the textual annotations, we set the maximum number of tokens $L = 128$, and the BERT encoder module produces word embeddings of size $\phi^{(0)} = 768$.

For the interaction graph encoder module (Sec. 4.3), we use $B_1 = 2$ consecutive GIA layers, and set the number of hidden units of all the GIA layers and the DiffPool layers as $\phi^{(i)} = \omega = 256$ (see Eqs. (6), (8), (9), and (10)). For the concept recognition decoder module (Sec. 4.4), we set $\mu = \phi^{(0)} = 768$ for the FC layer (Eq. (12)). In addition, we leverage $B_2 = 1$ BiLSTM layer with a total of $\eta = 64$ hidden units in Eq. (15). We empirically set the relevance threshold $\gamma = 0.3$.

We have performed our experiments of all evaluated approaches on an HPC server equipped with Linux Ubuntu 18.04.5 LTS, an AMD Ryzen Threadripper 3960X 24-Core CPU, 4×GeForce RTX 3090 with GDDR5 24GB, and 128GB RAM. With the above computing environment, the average computation time per sample for the data pre-processing stage, model training stage, and prediction stage of our CG-HMI system prototype is 86.31ms, 19.80ms, and 8.60ms, respectively.

• **Performance Metrics:** In evaluating the performance, we recall that our approach is to estimate whether a word in the textual annotation belongs to any of the HMI concepts or not (see Sec. 3.3), i.e., $\mathbf{Y}_a[i, j] = 1$ represents that the $j$-th token belongs to the $i$-th HMI concept, and 0 otherwise. We follow the common practices in the NER [52] and examine the true positive, false positive, and false negative. Then, we find the *Precision*, *Recall*, and *F1 measure* (i.e., each word could be assigned a label correctly or incorrectly according to a concept) to evaluate the performance of CG-HMI and other baseline approaches.

## 5.2 Experimental Results

Table 2. Overall performance in HMI language and sensor data co-learning (%) across the three datasets.

| Model | DS1 | | | DS2 | | | DS3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Recall | Precision | F1 | Recall | Precision | F1 | Recall | Precision | F1 |
| **CG − HMI** | **96.1** | **78.9** | **86.7** | **96.2** | **73.3** | **83.2** | **91.1** | **78.1** | **84.1** |
| BCRF | 81.2 | 68.6 | 74.4 | 78.0 | 65.8 | 71.4 | 80.7 | 55.5 | 65.7 |
| BCNN | 81.9 | 66.2 | 73.2 | 78.6 | 63.6 | 70.3 | 77.1 | 55.6 | 64.6 |
| BTrans | 82.9 | 52.9 | 64.6 | 79.6 | 50.8 | 62.0 | 76.1 | 50.4 | 60.6 |
| HAN | 82.7 | 76.4 | 79.4 | 79.4 | 73.4 | 76.2 | 76.7 | 62.7 | 69.0 |
| UMGF | 74.1 | 63.1 | 68.2 | 71.2 | 60.6 | 65.4 | 66.2 | 56.8 | 61.4 |
| FEC | 91.3 | 75.0 | 82.3 | 82.8 | 58.0 | 68.2 | 86.5 | 74.2 | 79.8 |
| RPAtt | 94.9 | 70.6 | 80.9 | 86.8 | 56.9 | 68.7 | 96.2 | 63.8 | 76.8 |
| DAT | 94.8 | 66.4 | 78.1 | 80.7 | 60.8 | 69.4 | 86.7 | 57.1 | 68.9 |
| UMT | 91.5 | 69.1 | 78.7 | 78.4 | 66.6 | 72.1 | 86.0 | 60.9 | 71.3 |

• **Overall Model Performance:** We first overview the evaluation result across all HMI concept classes of CG-HMI well as the baseline approaches in Table 2. One can observe that our proposed model outperforms all the baseline schemes for all the datasets. In particular, CG-HMI achieves 10.0%, 16.7%, and 11.5% higher recall; 11.3%, 11.5%, and 18.4% higher precision; and 11.1%, 13.9%, and 15.4% higher F1 measure on average for each of DS1, DS2, and DS3, respectively. Compared to other single-modality and conventional multi-modality approaches, CG-HMI fuses the visual and behavioral modalities with the textual modality through the novel graph interaction fusion upon the HMIGs, and thus yields higher accuracy in recognizing the HMI concepts.

In our experimental studies, we can observe that the single-modality approaches like BCRF, BCNN, and BTrans achieved generally lower recall, precision, and F1 measure than other multi-modality approaches. The single-modality approaches focus only on the textual modality and may thus not extract the semantic relations with other modalities. In addition, while the multi-modality approaches such as UMGF, FEC, and RPAtt considered both the visual and textual modalities, their feature integration designs may not suffice to accurately recognize all the HMI concepts.

In particular, UMGF and RPAtt focus on incorporating the explicit visual characteristics of the objects, which, however, might not necessarily help extract the implicit HMI concepts. While FEC leverages feature concatenation to fuse cross-modal information, it overlooks the possible relations across different modalities and thus may not capture the more subtle and implicit HMI concepts. While the transformer-based approaches like DAT and UMT can be adapted to account for the visual, behavioral, and textual modalities, their transformer designs have not comprehensively modeled the semantic correspondence across the modalities as our proposed cross-modality designs in CG-HMI. Therefore, they may not comprehensively capture the intrinsic and implicit interactions between the mobility system users and the mobility environments, and yield low accuracy in retrieving the concepts of the interaction events.

• **Detailed Performance on the Three Datasets:** We further demonstrate the performance of different approaches in identifying the HMI concepts with respect to the three different datasets. For each dataset, we show the recall, precision, and F1 measure regarding the HMI concepts of (i) mobility system status, (ii) traffic environment status, (iii) interactions with pedestrians, (iv) interactions with other vehicles, and (v) interactions with road infrastructures. We can see that CG-HMI achieves better performance for all the HMI concept classes compared with other baseline approaches, i.e., on average 11.1%, 13.9%, and 15.4% higher F1 measure, 11.3%, 11.5%, and 18.4% higher precision, and 10.0%, 16.7%, and 11.5% higher recall with respect to DS1, DS2, and DS3. Furthermore, we can also see that the other approaches, particularly the single-modality approaches, generally achieve much lower performance in identifying the HMI concepts related to the complex interaction behaviors,

Table 3. Detailed HMI language and sensor data co-learning results for different HMI concepts in DS1 (%).

| Model | Recall | | | | | Precision | | | | | F1 Measure | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | (i) | (ii) | (iii) | (iv) | (v) | (i) | (ii) | (iii) | (iv) | (v) | (i) | (ii) | (iii) | (iv) | (v) |
| **CG − HMI** | **95.5** | **95.7** | **96.5** | **95.2** | **97.9** | **76.9** | **77.1** | **78.1** | **80.0** | **82.8** | **85.2** | **85.4** | **86.3** | **86.9** | **89.7** |
| BCRF | 82.2 | 87.6 | 83.2 | 81.1 | 72.4 | 68.2 | 70.9 | 71.2 | 72.1 | 60.6 | 74.5 | 78.3 | 76.7 | 76.3 | 65.9 |
| BCNN | 86.7 | 81.7 | 81.8 | 78.3 | 81.1 | 71.7 | 66.7 | 66.0 | 57.9 | 68.9 | 78.4 | 73.4 | 73.1 | 66.6 | 74.5 |
| BTrans | 84.5 | 83.9 | 80.7 | 78.4 | 87.1 | 58.5 | 56.7 | 49.6 | 40.3 | 59.5 | 69.1 | 67.6 | 61.4 | 53.2 | 70.7 |
| HAN | 75.3 | 74.0 | 87.6 | 96.4 | 80.6 | 69.3 | 91.0 | 72.4 | 74.0 | 75.3 | 72.2 | 72.2 | 81.6 | 83.7 | 77.8 |
| UMGF | 74.5 | 73.7 | 76.2 | 75.2 | 71.3 | 63.2 | 57.3 | 58.6 | 75.1 | 61.6 | 68.3 | 64.4 | 66.2 | 75.1 | 66.1 |
| FEC | 90.7 | 90.9 | 91.6 | 90.4 | 93.0 | 73.0 | 73.2 | 74.1 | 76.0 | 78.6 | 80.9 | 81.1 | 82.0 | 85.6 | 85.2 |
| RPAtt | 90.3 | 94.0 | 97.1 | 100 | 93.1 | 69.1 | 66.3 | 78.8 | 66.4 | 72.2 | 78.2 | 77.7 | 86.9 | 79.8 | 81.3 |
| DAT | 94.6 | 93.5 | 95.4 | 95.2 | 95.5 | 64.9 | 62.3 | 58.9 | 76.9 | 69.2 | 77.0 | 74.8 | 72.8 | 85.1 | 80.2 |
| UMT | 92.7 | 94.2 | 91.2 | 90.2 | 89.3 | 58.2 | 69.5 | 70.3 | 74.2 | 73.1 | 71.5 | 79.9 | 79.4 | 81.4 | 80.3 |

Table 4. Detailed language and sensor data co-learning results for different HMI concepts in DS2 (%).

| Model | Recall | | | | | Precision | | | | | F1 Measure | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | (i) | (ii) | (iii) | (iv) | (v) | (i) | (ii) | (iii) | (iv) | (v) | (i) | (ii) | (iii) | (iv) | (v) |
| **CG − HMI** | **96.1** | **95.4** | **93.1** | **100** | **96.5** | **70.6** | **74.2** | **70.1** | **82.6** | **69.2** | **81.4** | **83.5** | **80.1** | **90.5** | **80.6** |
| BCRF | 78.9 | 84.1 | 79.9 | 77.8 | 69.5 | 65.4 | 68.1 | 68.3 | 69.2 | 58.1 | 71.5 | 75.2 | 73.6 | 73.2 | 63.3 |
| BCNN | 83.2 | 78.4 | 78.5 | 75.1 | 77.8 | 68.8 | 64.0 | 63.3 | 55.5 | 66.1 | 75.3 | 70.5 | 70.1 | 63.9 | 71.5 |
| BTrans | 81.2 | 80.5 | 77.4 | 75.2 | 83.6 | 56.1 | 54.4 | 47.6 | 38.6 | 57.1 | 66.3 | 64.9 | 58.9 | 51.1 | 67.8 |
| HAN | 72.2 | 71.1 | 84.1 | 92.5 | 77.3 | 66.5 | 87.3 | 69.5 | 71.0 | 72.2 | 69.2 | 78.3 | 76.1 | 80.3 | 74.7 |
| UMGF | 71.5 | 70.7 | 73.1 | 72.1 | 68.4 | 60.6 | 55.0 | 56.2 | 72.1 | 59.1 | 65.6 | 61.9 | 63.6 | 72.1 | 63.4 |
| FEC | 87.7 | 92.3 | 89.4 | 58.3 | 86.3 | 67.4 | 63.9 | 64.1 | 33.3 | 61.3 | 76.2 | 75.5 | 74.6 | 42.3 | 71.6 |
| RPAtt | 93.1 | 81.1 | 78.5 | 100 | 81.5 | 68.3 | 59.3 | 52.3 | 47.3 | 57.4 | 78.7 | 68.5 | 62.7 | 64.2 | 67.3 |
| DAT | 86.8 | 80.7 | 93.5 | 77.2 | 65.6 | 64.5 | 59.7 | 78.3 | 60.7 | 41.1 | 74.0 | 68.6 | 85.2 | 68.0 | 50.6 |
| UMT | 84.2 | 80.1 | 89.2 | 66.8 | 71.9 | 69.2 | 63.3 | 75.4 | 58.2 | 67.2 | 75.9 | 70.7 | 81.7 | 62.2 | 69.4 |

Table 5. Detailed language and sensor data co-learning results for different HMI concepts in DS3 (%).

| Model | Recall | | | | | Precision | | | | | F1 Measure | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | (i) | (ii) | (iii) | (iv) | (v) | (i) | (ii) | (iii) | (iv) | (v) | (i) | (ii) | (iii) | (iv) | (v) |
| **CG − HMI** | **89.3** | **93.3** | **93.6** | **93.7** | **85.5** | **72.2** | **81.5** | **72.7** | **78.7** | **85.5** | **79.8** | **87.0** | **81.8** | **85.5** | **85.5** |
| BCRF | 81.8 | 83.0 | 89.6 | 81.2 | 68.3 | 61.3 | 61.3 | 62.1 | 69.9 | 20.5 | 70.1 | 71.0 | 78.5 | 71.4 | 31.5 |
| BCNN | 79.3 | 80.2 | 81.1 | 76.8 | 68.2 | 62.1 | 61.9 | 65.3 | 63.0 | 25.7 | 69.6 | 69.8 | 72.3 | 69.2 | 37.3 |
| BTrans | 78.2 | 79.8 | 76.0 | 67.4 | 79.2 | 49.1 | 52.0 | 49.1 | 42.7 | 59.2 | 60.3 | 62.9 | 59.6 | 52.2 | 67.6 |
| HAN | 75.1 | 84.1 | 89.2 | 73.5 | 61.6 | 57.5 | 75.9 | 75.4 | 53.8 | 51.2 | 65.1 | 79.9 | 81.7 | 62.1 | 55.9 |
| UMGF | 66.7 | 67.3 | 67.5 | 68.9 | 64.1 | 56.9 | 51.5 | 52.7 | 67.5 | 55.35 | 61.4 | 58.3 | 59.1 | 68.2 | 59.4 |
| FEC | 84.8 | 88.6 | 88.9 | 89.0 | 81.2 | 68.6 | 77.4 | 69.1 | 74.7 | 81.2 | 75.8 | 82.6 | 77.7 | 81.2 | 81.2 |
| RPAtt | 91.2 | 100 | 99.0 | 96.0 | 95.0 | 56.4 | 73.4 | 62.2 | 59.6 | 67.8 | 69.6 | 84.6 | 76.4 | 73.5 | 79.1 |
| DAT | 85.9 | 87.2 | 81.8 | 85.1 | 93.7 | 58.3 | 60.9 | 54.9 | 54.0 | 57.7 | 69.5 | 71.8 | 65.7 | 66.1 | 71.4 |
| UMT | 86.4 | 82.9 | 83.3 | 86.3 | 91.2 | 62.6 | 66.2 | 56.1 | 56.2 | 63.8 | 72.6 | 73.6 | 67.1 | 68.1 | 75.2 |

such as interactions with pedestrians or other vehicles. It is mainly because these baseline approaches do not account for the semantic interactions and behavioral information of the textual, visual, and behavioral sensor modalities, thus yielding lower accuracy in the HMI learning.

• **Model Ablation Studies:** Taking DS1 as an example, we have performed the model ablation studies on CG-HMI to evaluate the importance of different modules, and we show the corresponding F1 scores in Fig. 14. Specifically, we have compared the full version of CG-HMI (labeled as (1)) with the following variations of CG-HMI:

(2) *w/o V*: which implements CG-HMI without the visual modality (i.e., video frames of the HMI scenes).

(3) *w/o B*: which implements CG-HMI without the behavioral modality (i.e., maneuvers performed by the car drivers or e-scooter riders in the HMI scenes).
(4) *w/ T*: which implements CG-HMI without both behavioral and visual modalities.
(5) *w/ PMI*: which leverages the point-wise mutual information [50] of the words, instead of our proposed HMIG representation design, to calculate their co-occurrence within the textual annotation to construct the edges between them.
(6) *w/ sim*: which constructs the graph by connecting every pair of nodes only based on the cosine similarity of their corresponding embeddings [15, 20].
(7) *w/o GCN*: which only leverages the graph convolutional network (GCN) without the differentiable pooling-based graph attention network.

We can observe the performance degradation of variations (2), (3), and (4), i.e., removals of visual, behavioral, or both modalities, compared with (1), which corroborates the importance of our proposed cross-modality fusion designs. Furthermore, in terms of the interaction graph design, we can see that the point-wise mutual information (5), the similarity across the node embeddings (6), and the graph convolution operation (7) cannot extract sufficient semantic interaction knowledge, and hence yield lower accuracy than (1).

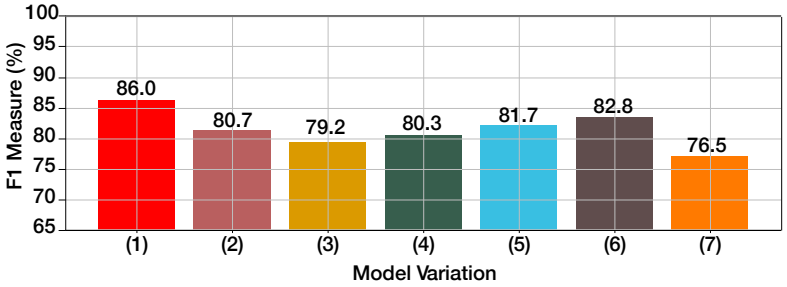


Fig. 14. Ablation studies of CG-HMI.

• **Model Sensitivity Studies:** We further focus on DS1 and perform the model sensitivity studies regarding the important parameters of CG-HMI. Fig. 15(a) shows its F1 scores with different numbers of GIA layers ($B_1$) in the interaction graph encoder module. Fig. 15(b) also illustrates the performance of CG-HMI for different numbers of hidden units ($\phi^{(i)}$) in each GIA layer. We can observe that the small number of GIA layers or the number of hidden units does not suffice to provide accuracy estimation, while further increasing the number of GIA layers or hidden units makes it hard for the model to generalize. Therefore, we set $B_1 = 2$ and $\Phi^{(i)} = 256$ by default. We further show in Fig. 15(c) the impact of the number of the BiLSTM layers. We can observe that in general one BiLSTM layer suffices to capture the sequential dependencies within the textual annotations. The inclusion of two or more layers of recurrent layers may render the model training more challenging, and therefore, we observe a drop in the performance (up to 11.3% in our experimental studies).

• **Result Visualization and Case Studies:** We have further visualized the relevance scores for the words in the textual annotations along with the video frames and the IMU sensor measurements in Fig. 16 to illustrate the HMI learning results. The warmer colors imply the larger relevance scores for each word (the tokenized unit). We also highlighted the ground-truth HMI concepts in green boxes. We can observe from the high relevance scores that CG-HMI accurately identifies the underlying HMI concepts.

We further visualize two examples of the HMIGs learned by CG-HMI from DS1 in Figs. 17(a) and (b), and one from DS3 in Fig. 17(c). In particular, we have visualized the average graph interaction attention weights, $\mathbf{V}_a^{(l)}$, across all the nodes to show the semantic interaction and correspondence captured by CG-HMI. We note that the words in blue, green, and red fonts represent the visual, behavioral, and textual nodes, respectively. Fig. 17(a) shows the semantic correspondence between the words related to the traffic lights and the textual nodes related to

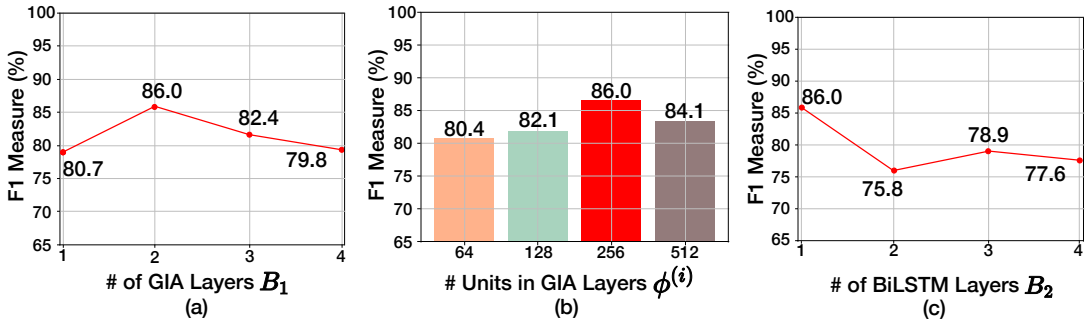


Fig. 15. Sensitivity studies results of CG-HMI: (a) number of GIA layers; (b) number of units in GIA layers; and (c) number of BiLSTM layers.

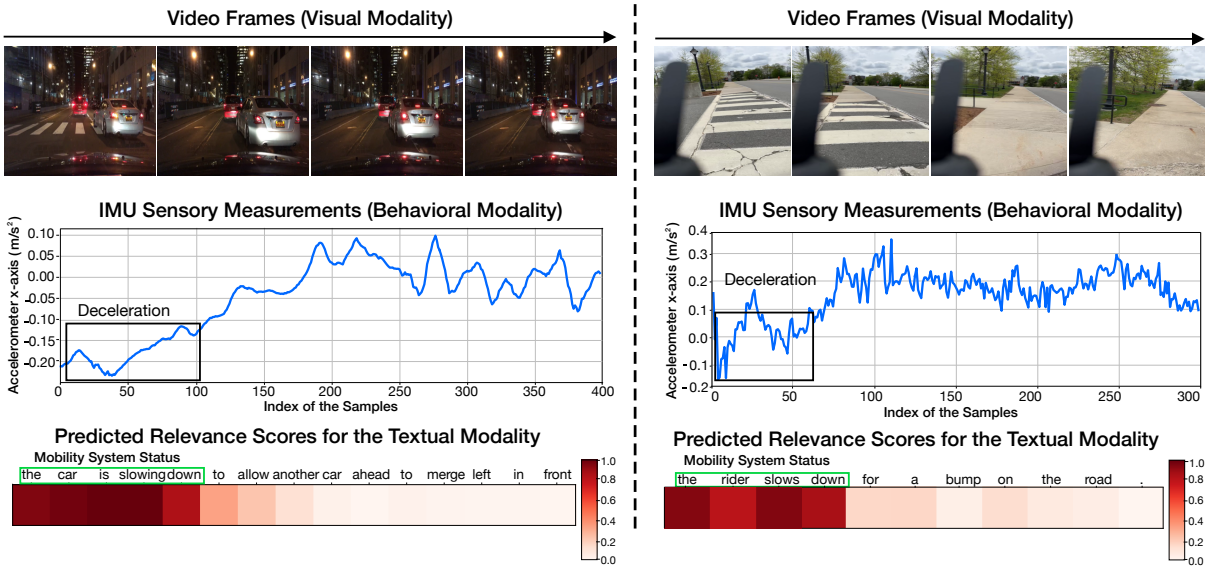


Fig. 16. Examples of the relevance scores estimated by CG-HMI for the HMI concepts from DS1 (left) and DS3 (right).

the pedestrian, extracting the HMI concepts of mobility system status and interactions with pedestrians. Fig. 17(b) demonstrates the strong correspondence of the textual nodes related to the mobility system status as well as the traffic lights and the acceleration behavior. Fig. 17(c) shows the e-scooter rider's interactions with the pedestrians, where the deceleration behavior is mapped towards the scene. We can observe from the three figures that the semantic correspondence has been captured by the HMIG representation, yielding the understandable results of the language and sensor data co-learning.

We further provide in Fig. 18 an illustrative example of determining the types and positions of a set of HMI concepts. We compare CG-HMI with the baseline model DAT [55] which was adapted to fuse multi-modal data. Note that (i), (ii), and (v) in Fig. 18 correspond to mobility system status, traffic environment status, and interactions with road infrastructures. We can observe that compared to DAT, CG-HMI further captures the intra-modality semantic correspondence across the textual annotations, and hence can identify the HMI concept regarding

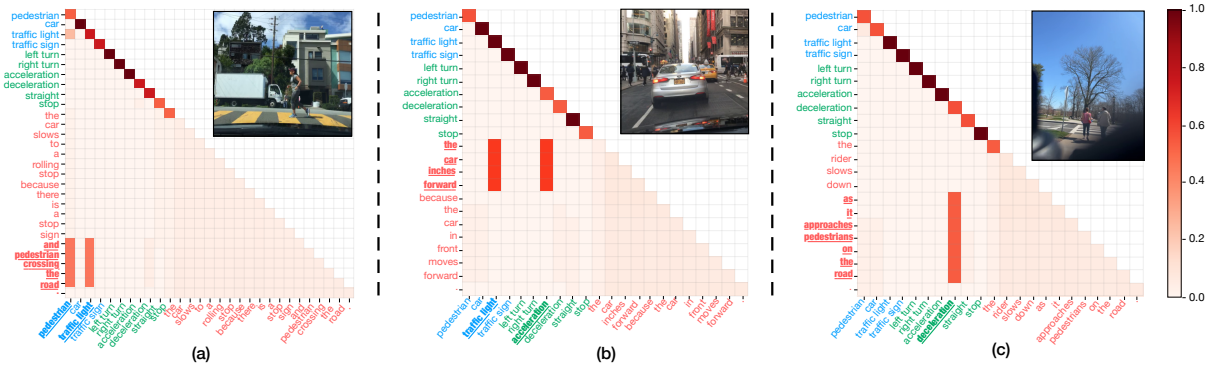


Fig. 17. Examples of the learned HMIGs by CG-HMI (words in blue, green, and red fonts represent visual, behavioral, and textual nodes, respectively) as well as the HMI scenes. (a) and (b) are from DS1. (c) is from DS3.

interactions with the road infrastructure. Furthermore, CG-HMI achieves more fine-grained recognition from the tokens in the textual annotations while discarding the irrelevant information.

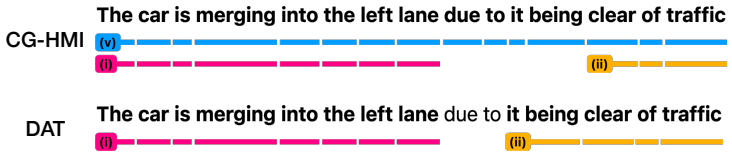


Fig. 18. An example (from DS1) of the identified HMI concepts by CG-HMI and a baseline model named DAT. We note that (i), (ii), and (v) correspond to mobility system status, traffic environment status, and interactions with road infrastructures.

## 6 DEPLOYMENT DISCUSSION

We would like to discuss further the deployment of CG-HMI in the following two aspects.

• **Integration with Other Modalities:** Our current prototype studies focused on 3 different modalities — i.e., visual, behavioral, and textual modalities — to co-learn the language and sensor data. CG-HMI could be further modified or extended to include more information modalities for understanding more complicated HMI scenes in AV or other emerging smart mobility systems. For instance, modalities, such as LiDAR measurements [18, 33, 35], speech commands or audio signals [20, 54], and magnetic field measurement can be further integrated for more comprehensive object detection and context recognition. This way, we can enable more fine-grained HMI learning.

• **Extension to Other HMI Concepts:** Our current studies focus on the designs of fusing three modalities (i.e., visual, behavioral, and textual) for identifying five different HMI concepts in the HMI scenes. Nevertheless, our formulation of CG-HMI, and the unified representations based on HMIGs, are general enough to be extended to other more fine-grained and complex HMI scenes and concepts [24, 40], such as mobility system user status under different lighting conditions, pedestrians' walking patterns when interacting with the vehicle, and vehicle trajectory when interacting with the other traffic participants. By fusing the unspoken and spoken language, our proposed designs can help analyze the causality or accountability, which is part of our future work.

## 7 CONCLUSION

We have proposed CG-HMI, a novel cross-modality graph-based language and sensor data co-learning approach to extract the HMI concepts from the HMI scenes. Specifically, we have designed a novel graph interaction fusion model with differentiable pooling-based graph attention to extract the inter- and intra-modality semantic correspondences and interactions from a unified graph representation of visual, behavioral, and textual modalities. The resulting graph embeddings are processed further to identify the existence and positions of the HMI concepts within the annotations, which benefits the further downstream HCI tasks and ubiquitous computing applications. Via extensive system prototype studies upon three real-world HMI datasets (two on car driving and one on e-scooter riding), we have corroborated the effectiveness and accuracy of our CG-HMI in recognizing the important HMI concepts from the complex HMI scenes.

## ACKNOWLEDGMENT

We thank the editors and reviewers for the constructive feedback. This project is supported, in part, by the National Science Foundation (NSF) under Grant 2239897, Google Research Scholar Program Award (2021–2022), and NVIDIA Applied Research Accelerator Program Award (2021–2022).

## REFERENCES

[1] 2022. How Mercedes-Benz Is Using AI & NLP To Give Driving A Tech Makeover. https://analyticsindiamag.com/how-mercedes-benz-is-using-ai-nlp-to-give-driving-a-tech-makeover/.
[2] 2022. Natural language processing enhances autonomous vehicles experience. https://www.autonomousvehicleinternational.com/features/natural-language-processing-enhances-autonomous-vehicles-experience.html.
[3] Utku Günay Acer, Marc van den Broeck, Chulhong Min, Mallesham Dasari, and Fahim Kawsar. 2022. The City as a Personal Assistant: Turning Urban Landmarks into Conversational Agents for Serving Hyper Local Information. *Proc. ACM IMWUT* 6, 2 (2022), 1–31.
[4] Sonia Baee, Erfan Pakdamanian, Inki Kim, Lu Feng, Vicente Ordonez, and Laura Barnes. 2021. Medirl: Predicting the visual attention of drivers via maximum entropy deep inverse reinforcement learning. In *Proc. IEEE/CVF ICCV*. 13178–13188.
[5] Hédi Ben-Younes, Éloi Zablocki, Patrick Pérez, and Matthieu Cord. 2022. Driving behavior explanation with multi-level fusion. *Pattern Recognition* 123 (2022), 108421.
[6] Qingqing Cao, Prerna Khanna, Nicholas D Lane, and Aruna Balasubramanian. 2022. MobiVQA: Efficient On-Device Visual Question Answering. *Proc. ACM IMWUT* 6, 2 (2022), 1–23.
[7] Hou Pong Chan, Mingxi Guo, and Cheng-Zhong Xu. 2022. Grounding Commands for Autonomous Vehicles via Layer Fusion with Region-specific Dynamic Layer Attention. In *Proc. IEEE/RSJ IROS*. IEEE, 12464–12470.
[8] Dongyao Chen, Kyong-Tak Cho, Sihui Han, Zhizhuo Jin, and Kang G Shin. 2015. Invisible sensing of vehicle steering with smartphones. In *Proc. ACM MobiSys*. 1–13.
[9] Dawei Chen, Zhixu Li, Binbin Gu, and Zhigang Chen. 2021. Multimodal named entity recognition with image attributes and image knowledge. In *Proc. Springer DASFAA*. Springer, 186–201.
[10] Dongyao Chen and Kang G Shin. 2019. TurnsMap: Enhancing driving safety at intersections with mobile crowdsensing and deep learning. *Proc. ACM IMWUT* 3, 3 (2019), 1–22.
[11] Xiang Chen, Ningyu Zhang, Lei Li, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Good Visual Guidance Makes A Better Extractor: Hierarchical Visual Prefix for Multimodal Entity and Relation Extraction. *arXiv* (2022).
[12] Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the association for computational linguistics* 4 (2016), 357–370.
[13] Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2020. An effective transition-based model for discontinuous NER. *arXiv* (2020).
[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* (2018).
[15] Chaoyue Ding, Shiliang Sun, and Jing Zhao. 2023. MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection. *Information Fusion* 89 (2023), 527–536.
[16] Drover. 2022. AI-Powered Computer Vision for Micromobility. Retrieved October 11, 2022 from https://drover.ai/
[17] Jianwu Fang, Dingxin Yan, Jiahuan Qiao, Jianru Xue, and Hongkai Yu. 2021. DADA: Driver attention prediction in driving accident scenarios. *IEEE T-ITS* (2021).

[18] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. 2021. Fog simulation on real LiDAR point clouds for 3D object detection in adverse weather. In *Proc. IEEE/CVF CVPR*. 15283–15292.
[19] Suining He and Kang G. Shin. 2022. Socially-Equitable Interactive Graph Information Fusion-Based Prediction for Urban Dockless E-Scooter Sharing. In *Proc. WWW*. Association for Computing Machinery, 3269–3279.
[20] Jingwen Hu, Yuchen Liu, Jinming Zhao, and Qin Jin. 2021. MMGCN: Multimodal Fusion via Deep Graph Convolution Network for Emotion Recognition in Conversation. In *Proc. ACL Anthology*. 5666–5675.
[21] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* (2015).
[22] Jinkyu Kim, Teruhisa Misu, Yi-Ting Chen, Ashish Tawari, and John Canny. 2019. Grounding human-to-vehicle advice for self-driving vehicles. In *Proc. IEEE/CVF CVPR*. 10591–10599.
[23] Jinkyu Kim, Suhong Moon, Anna Rohrbach, Trevor Darrell, and John Canny. 2020. Advisable learning for self-driving vehicles by internalizing observation-to-action rules. In *Proc. IEEE/CVF CVPR*. 9661–9670.
[24] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. 2018. Textual explanations for self-driving vehicles. In *Proc. ECCV*. 563–578.
[25] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv* (2016).
[26] Luc Le Mero, Dewei Yi, Mehrdad Dianati, and Alexandros Mouzakitis. 2022. A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE T-ITS* (2022).
[27] Fei Li, ZhiChao Lin, Meishan Zhang, and Donghong Ji. 2021. A span-based model for joint overlapped and discontinuous named entity recognition. *arXiv* (2021).
[28] Fei Li, Zheng Wang, Siu Cheung Hui, Lejian Liao, Dandan Song, and Jing Xu. 2021. Effective named entity recognition with boundary-aware bidirectional neural networks. In *Proc. WWW*. 1695–1703.
[29] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE TKDE* 34, 1 (2020), 50–70.
[30] Max Guangyu Li, Bo Jiang, Zhengping Che, Xuefeng Shi, Mengyao Liu, Yiping Meng, Jieping Ye, and Yan Liu. 2019. DBUS: Human Driving Behavior Understanding System.. In *Proc. IEEE/CVF ICCV*. 2436–2444.
[31] Pei Li, Mohamed Abdel-Aty, Qing Cai, and Zubayer Islam. 2020. A deep learning approach to detect real-time vehicle maneuvers based on smartphone sensors. *IEEE T-ITS* (2020).
[32] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Alireza Dirafzoon, Aparajita Saraf, Amy Bearman, and Babak Damavandi. 2022. IMU2CLIP: Multimodal Contrastive Learning for IMU Motion Sensors from Egocentric Videos and Text. *arXiv preprint arXiv:2210.14395* (2022).
[33] Ramin Nabati and Hairong Qi. 2019. RRPN: Radar region proposal network for object detection in autonomous vehicles. In *Proc. IEEE ICIP*. IEEE.
[34] Hongliang Pan, Zheng Lin, Peng Fu, Yatao Qi, and Weiping Wang. 2020. Modeling intra and inter-modality incongruity for multi-modal sarcasm detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 1383–1392.
[35] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. 2021. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *Proc. IEEE/CVF CVPR*. 444–453.
[36] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D Lane, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. 2018. Multimodal deep learning for activity and context recognition. *Proc. ACM IMWUT* 1, 4 (2018), 1–27.
[37] Dave Raggett. 2015. The Web of Things: Challenges and Opportunities. *Computer* 48, 5 (2015), 26–32.
[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proc. IEEE CVPR*. 779–788.
[39] Matteo Simoncini, Douglas Coimbra de Andrade, Leonardo Taccari, Samuele Salti, Luca Kubin, Fabio Schoen, and Francesco Sambo. 2022. Unsafe Maneuver Classification From Dashcam Video and GPS/IMU Sensors Using Spatio-Temporal Attention Selector. *IEEE T-ITS* (2022).
[40] Jithesh Gugan Sreeram, Xiao Luo, and Renran Tian. 2021. Contextual and Behavior Factors Extraction from Pedestrian Encounter Scenes Using Deep Language Models. In *International Conference on Big Data Analytics and Knowledge Discovery*. Springer, 131–136.
[41] Lin Sun, Jiquan Wang, Kai Zhang, Yindu Su, and Fangsheng Weng. 2021. RpBERT: a text-image relation propagation-based BERT model for multimodal NER. In *Proc. AAAI*, Vol. 35. 13860–13868.
[42] Mahan Tabatabaie and Suining He. 2023. Naturalistic E-Scooter Maneuver Recognition with Federated Contrastive Rider Interaction Learning. *Proc. ACM IMWUT* 6, 4, Article 205 (Jan 2023), 27 pages.
[43] Mahan Tabatabaie, Suining He, and Xi Yang. 2021. Reinforced Feature Extraction and Multi-Resolution Learning for Driver Mobility Fingerprint Identification. In *Proc. ACM SIGSPATIAL*. 69–80.
[44] Mahan Tabatabaie, Suining He, and Xi Yang. 2022. Driver Maneuver Identification with Multi-Representation Learning and Meta Model Update Designs. *Proc. ACM IMWUT* 6, 2 (2022), 1–23.
[45] Chuanqi Tan, Wei Qiu, Mosha Chen, Rui Wang, and Fei Huang. 2020. Boundary enhanced neural span classification for nested named entity recognition. In *Proc. AAAI*, Vol. 34. 9016–9023.

[46] Ultralytics. 2022. Yolov5. Retrieved October 3, 2022 from https://github.com/ultralytics/yolov5
[47] Yonatan Vaizman, Nadir Weibel, and Gert Lanckriet. 2018. Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification. *Proc. ACM IMWUT* 1, 4 (2018), 1–22.
[48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[49] Xuwu Wang, Junfeng Tian, Min Gui, Zhixu Li, Jiabo Ye, Ming Yan, and Yanghua Xiao. 2022. PromptMNER:Prompt-Based Entity-Related Visual Clue Extraction and Integration for Multimodal Named Entity Recognition. In *International Conference on Database Systems for Advanced Applications*. Springer, 297–305.
[50] Youze Wang, Shengsheng Qian, Jun Hu, Quan Fang, and Changsheng Xu. 2020. Fake news detection via knowledge-driven multimodal graph convolutional networks. In *Proc. ACM ICMR*. 540–547.
[51] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *Proc. ICML*. PMLR, 6861–6871.
[52] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
[53] Ye Xia, Jinkyu Kim, John Canny, Karl Zipser, Teresa Canas-Bajo, and David Whitney. 2020. Periphery-fovea multi-resolution driving model guided by human attention. In *Proc. IEEE/CVF CVPR*. 1767–1775.
[54] Shuo Xu, Yuxiang Jia, Changyong Niu, and Hongying Zan. 2022. MMDAG: Multimodal Directed Acyclic Graph Network for Emotion Recognition in Conversation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 6802–6807.
[55] Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474* (2019).
[56] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proc. ACL Anthology*. 1480–1489.
[57] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *Proc. NeurIPS* 31 (2018).
[58] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *Proc. IEEE/CVF CVPR*. 2636–2645.
[59] Jianfei Yu, Jing Jiang, Li Yang, and Rui Xia. 2020. Improving multimodal named entity recognition via entity span detection with unified multimodal transformer. *Proc. ACL*.
[60] Yuting Zhan and Hamed Haddadi. 2019. Towards automating smart homes: Contextual and temporal dynamics of activity prediction. In *UbiComp/ISWC*. 413–417.
[61] Dong Zhang, Suzhong Wei, Shoushan Li, Hanqian Wu, Qiaoming Zhu, and Guodong Zhou. 2021. Multi-modal graph fusion for named entity recognition with targeted visual guidance. In *Proc. AAAI*, Vol. 35. 14347–14355.
[62] Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. Adaptive Co-attention Network for Named Entity Recognition in Tweets. *Proc. AAAI* 32, 1 (Apr. 2018).