EXPLORING SITUATION AWARENESS FOR ADVANCED DRIVER-ASSISTANCE SYSTEMS

by

Chengxi Li

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Electrical and Computer Engineering

West Lafayette, Indiana

December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Stanley H. Chan, Chair

School of Electrical and Computer Engineering

Dr. Charles A. Bouman

School of Electrical and Computer Engineering

Dr. Mary Comer

School of Electrical and Computer Engineering

Dr. Fengqing Maggie Zhu

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

To my parents, Luqi and Yongmei.

ACKNOWLEDGMENTS

I would like to first and foremost thank my advisor, Prof. Stanley H. Chan, for his unwavering support and guidance during my PhD training. He offers me as much freedom to explore the research problems I am interested in, gives me inspirational advice in research, and helps me improve my writing and presentation skills. Stanley is not only my role model as a researcher, who works very hard and is always enthusiastic, but is also a man of kindness. I still remember the day that I almost gave up continuing my PhD and walked into his office for help. It is him who offered a helping hand to me, took me as his student, and saved me from my darkest moment. I feel so lucky and so proud to be one of his students.

I want to express my deepest appreciation to my PhD thesis committee members, Prof. Charles A. Bouman, Prof. Mary Comer, and Prof. Fengqing Maggie Zhu. I would like to thank them for joining my thesis committee without any hesitation, providing their critical feedback in my prelim exam, and valuable advice on my dissertation. I also want to thank my former advisor, Prof. Jeffrey M. Siskind. He motivated me to set a high standard for my research and I learned valuable research skills during my first two and half years in his lab.

I am also very lucky to work with a lot of hardworking and brilliant collaborators, Yue Meng, Xiangyu Qu, Abhiram Gnanasambandam, Omar A. Elgendy, Jiaju Ma, and Yi-Ting Chen. In particular, Yi-Ting and I have closely collaborated with weekly meetings for nearly two years. I am so grateful for his unending support and mentorship. Also, thank my lovely labmates from Intelligent Imaging Lab. Those gentlemen are always patient to answer my questions and spend their time in the TA sessions I host. A special thanks goes to my internship mentor, Enming Luo. He has taught me a lot and guided me on the career path in a technology company.

My sincerest thanks to my family and friends. I want to thank my parents, Luqi Li and Yongmei Chen, for their unconditional love and support over the past 27 years. Thank you to a special friend, Mingyu Sun, who accompanied and encouraged me through the toughest time of my PhD. I would also like to thank my friends, Jiayan Zhang and Xin Niu, who have always been supporting me academically and emotionally, without the company of which I would not be able to finish my PhD.

Contents

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ADAS	Advanced Driver-Assistance System
BDD	Berkeley DeepDrive
BEV	Bird’s-Eye-View
C3D	Convolutional 3D
CFA	Color Filter Array
CIS	CMOS Image Sensor
GCN	Graph Convolution Network
GNN	Graph Neural Networks
HDD	Honda Research Institute Driving Dataset
HMM	Hidden Markov Model
I3D	Inflated 3D Convnet
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IoU	Interation over Union
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
MNIST	Modified National Institute of Standards and Technology
MS COCO	Microsoft Common Objects in Context
NCC	Normalized Cross-Correlation
PASCAL	Pattern Analysis, Statistical modeling and Computational Learning
QIS	Quanta Image Sensor
RDN	Relation Distillation Network
ROI	Risk Object Identification
RoI	Region of Interest
SA	Situation Awareness
SID	See-in-the-Dark
SNR	Signal-to-Noise Ratio
SPAD	Single-Photon Avalanche Diode
STA	SpatioTemporal Accumulator

STAG	Spatio-Temporal Action Graph
TRN	Temporal Recurrent Network
VOC	Visual Object Classes
YOLO	You Only Look Once (object detector)

ABSTRACT

From prehistoric man who needs to be aware of the surrounding situations and hunt for food, to modern industry where machines and robots are programmed to explore the environment and accomplish assignments, situation awareness has always been an essential topic to everyone.

Advanced Driver-Assistance Systems (ADAS) is one of the modern technologies seeking effective solutions for driving safety. It also utilizes situation awareness model to interpret the driver's state in the environment and provide safe driving advice, with the potential to significantly reduce the traffic accident fatalities.

To enable situation awareness, an intelligent driving system needs to fulfill the following: (1) perceives the traffic elements in the environment, (2) comprehends the spatial-temporal interactions between a driver and other objects, and (3) projects the states of traffic elements to forecast future actions.

However, each level of situation awareness encounters its unique challenges in driving scenarios, for example, how to perceive vehicles in low-illuminated conditions? How to represent the complicated interactive relations in complicated driving situations? And how to anticipate the temporal dynamics of traffic elements and identify the where the potential risk comes from? To answer these questions, we explore situation awareness model for Advanced Driver-Assistance Systems at 3 levels: Perception, Comprehension and Projection. We discuss how to realize situation awareness based on three different computer vision tasks. We demonstrate that our proposed system is able to forecast the driver's operational intentions and identify risk objects to avoid hazards.

1. INTRODUCTION

More than 1.3 million people die in road accidents worldwide every year, or approximately 3,700 people per day [1]. Car accident deaths globally are the leading cause of death, excluding illness. A massive number of car accident fatalities are due to driver errors, such as lack of awareness [2]. To reduce the fatality rate through increased driving safety, Advanced Driver-Assistance Systems (ADAS) are in urgent need.

ADAS are groups of technological features that receive information from the environment, assess the situation, and provide assistance to drivers via the human-machine interface. Features such as Forward Collision Warning, Road Obstacle Detection, Traffic Signal Recognition, etc., capture drivers' attention or improve reaction time to potentially reduce road fatalities [3].

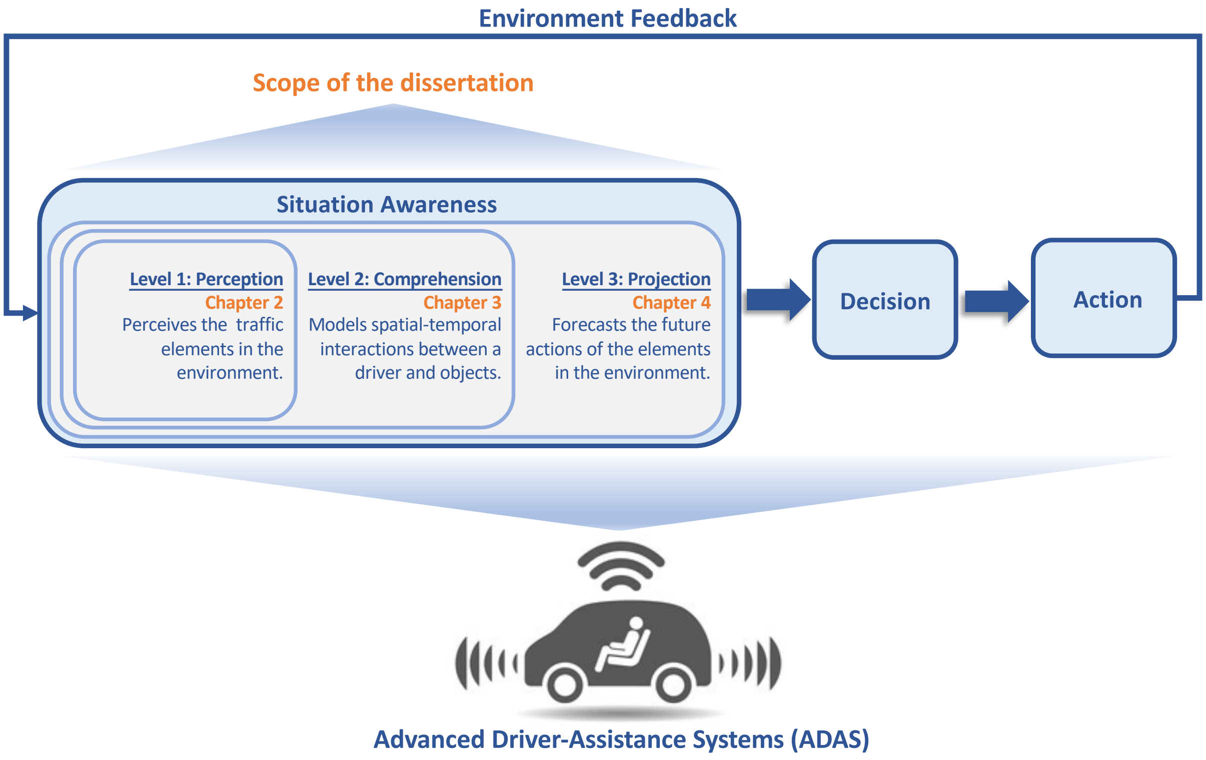

Environment Feedback
Scope of the dissertation
Situation Awareness
Level 1: Perception
Chapter 2
Perceives the traffic elements in the environment.
Level 2: Comprehension
Chapter 3
Models spatial-temporal interactions between a driver and objects.
Level 3: Projection
Chapter 4
Forecasts the future actions of the elements in the environment.
Decision
Action
Advanced Driver-Assistance Systems (ADAS)

Figure1.1. Situation awareness (SA) model for Advanced Driver-Assistance Systems (ADAS).

Since traffic scenes are extremely complicated, including a variety of elements (e.g., pedestrians, vehicles, and traffic signals), ongoing interactive relations (e.g., overtaking), and temporal evolution of elements (e.g., preparing to stop to avoid collision), ADAS need to process a torrent of information to disentangle the vehicle's state in the environment, make decisions based on that state, and operate the vehicle. "*Knowing what is going on around us*," also known as *situation awareness* (SA) [4], is the first step to enabling an intelligent driving assistance system. As long as the state of the vehicle in the environment is certain, the system can further determine what to do with the situation and perform actions accordingly. Therefore, SA serves as the main precursor to decision making and action [5], as shown in Figure 1.1.

The SA model is also widely applied in other domains, such as aircraft piloting, air traffic control [6], health care [7], etc. According to [5], SA is comprised of three different levels, defined as "*the perception of the elements in the environment within a volume time and space, the comprehension of their meaning, and the projection of their status in the near future*." In the language of ADAS applications, SA specifically means a system with the ability to perceive traffic elements on the road, understand the spatial-temporal interactions between a driver and objects, and forecast the future actions of the elements in the environment.

Although this three-level SA framework is general to most industrial domains, there are still unique challenges at each level for driving applications. We discuss these unique aspects in the following sections.

1.1 Level 1: Perception

Perception of traffic elements in the environment is a fundamental step of SA for ADAS. A basic perception of important information is a cornerstone of success in comprehension and projection. To perceive traffic elements, one can utilize object detection or semantic segmentation to recognize the existence of traffic objects (i.e., vehicles, pedestrians, traffic lanes, and traffic signs). State-of-the-art object detection methods such as Faster R-CNN [8] and YOLO [9] degrade under adverse weather (i.e., rain, snow, and fog) and low-illuminated photon-limited conditions (i.e., driving in the countryside with only moonlight). The latter – driving in a low light condition – is an especially common scenario in our daily lives,.

To address this problem, we present a photon-limited object detection framework in Chapter 3. We add two components to state-of-the-art object detectors: 1) a space-time non-local module that leverages the spatial-temporal information across an image sequence in the feature space, and 2) knowledge distillation in the form of student-teacher learning to improve the robustness of the detector's feature extractor against noise. Experiments are conducted on both object detection (PASCAL VOC 2007 [10]) and driving dataset (Berkeley DeepDrive 100K [11]) to demonstrate the improved performance of the proposed method in comparison with state-of-the-art baselines.

1.2 Level 2: Comprehension

SA is more than just perceiving information; it also includes how to interpret and understand information relevant to the goal. This phenomenon is defined in the literature as Level 2 SA – *Comprehension*. For an intelligent automated driving system, the goal is to guarantee human drivers' safety. A promising strategy to achieve this is to first understand how humans drive and interact with road users to avoid accidents in complicated driving situations. To be more specific, we need to develop a computational model which can capture the complicated spatial-temporal interactions between the ego vehicle and road users.

In Chapter 4, we propose a 3D-aware egocentric spatial-temporal interaction framework for automated driving applications. It is comprised of both Level 1 and Level 2 SA. Objects are first perceived by object detection and semantic segmentation [12] methods. Then Graph Convolution Networks (GCN) [13] are devised to model these interactions to facilitate the comprehension. By visualizing the learned affinity matrices, which encode object interactions, we showcase that the proposed framework can comprehend these interactions effectively.

1.3 Level 3: Projection

Projection is the highest level of SA. In terms of ADAS, it is interpreted as forecasting the future actions of the traffic elements. This ability will enable timely decision-making in the later stage. Given a driving scenario where a driver plans to pass through the intersection while a pedestrian is crossing the crosswalk, a Level 3 SA ADAS should be able to anticipate the future dynamics of the pedestrian and the vehicle. Moreover, a more intelligent driving system is expected to give driving advice (e.g., warn drivers of the collision risk) according to future projections.

Thus, one of the applications of the Level 3 SA ADAS is Driver-centric Risk Object Identification [14]; that is, to identify traffic elements that may cause hazards to the driver. In the previous example, the crossing pedestrian is a potential risk object to the vehicle if the driver does not stop. In Chapter 5, we formulate this task as a cause-effect problem and present two different novel two-stage risk object identification frameworks ([14], [15]), taking inspiration from models of situation awareness and causal inference.

1.4 Scope and Publications

In this dissertation, we study three levels of situation awareness for applications in Advanced Driver-Assistance Systems – (1) Perception; (2) Comprehension; and (3) Projection. Different levels of SA are represented as different computer vision tasks – (1) Photon-limited object detection; (2) Tactical driver behavior recognition; and (3) Driver-centric risk object identification.

Following is the relevant publication list for each chapter.

1. Chapter 3 – *Photon-Limited Object Detection using Non-local Feature Matching and Knowledge Distillation* (ICCV 2021 LCI Workshop)

2. Chapter 4 – *Learning 3D-aware Egocentric Spatial-Temporal Interaction via Graph Convolutional Networks* [16] (ICRA 2020)

3. Chapter 5 – *Who Make Drivers Stop? Towards Driver-centric Risk Assessment: Risk Object Identification via Causal Inference* [15] (IROS 2020) and *Driver-centric Risk Object Identification* [14] (T-PAMI Submission)

2. BACKGROUND

Since this dissertation focuses on the computer vision part of Advanced Driver-Assistance Systems, in this chapter, we introduce the background of computer vision research for driverless and self-driving vehicles.

2.1 Driving Models

The ultimate goal of ADAS is to realize fully autonomous navigation. Therefore, ADAS is an intermediate step to achieving Autonomous Driving (AD). To this end, various solutions have been proposed, and existing approaches can be categorized into two groups: modular pipelines and monolithic end-to-end learning methods[17].

Modular pipelines break down the entire driving system into functional modules from low-level perception, high-level scene understanding to path planning, and vehicle control. In 1995, Dickmanns [18] introduced a modular pipeline which can continuously estimate the state of the vehicle and output the control command. Following this approach, researchers put efforts into different related modules [19]–[21], which can be eventually applied to the self-driving system. Since the input, output, and function of each module are clear, this kind of method has two obvious advantages. First, it is easy to integrate prior knowledge into the system design. For example, we can apply distance constraints to the vehicle control module to avoid collisions. Second, it is easy to detect and fix the failure module of the whole system, as the intermediate results passing from module to module are straightforward to interpret. However, it also has drawbacks in that the intermediate representations designed by humans are not always optimal for the driving task. Similarly, since every module is designed and trained individually and independently, the optimization for each module might not be optimal for the final goal, after being integrated to the whole system. Taking the object detection task as an example, if we train the detector separately, it is highly possible that it would pay more attention to less relevant objects, such as birds, windows, etc., rather than traffic elements.

Another line of self-driving approaches falls into **end-to-end learning-based methods**, where the entire system can no longer be easily divided into components with explicit functions and the meaning of the intermediate information is more abstract to interpret. In 1989, ALVINN [22] was first proposed to achieve a mapping from images to navigation signals via a shallow neural network. More recently, [23] and [24] demonstrated that driving policy can be learned via an end-to-end supervised-learning manner from human demonstration ([11], [25], [26]). A notorious problem of the end-to-end driving model is lack of interpretability, as deep neural networks work as a "black box." To address the issue, Kim et al.([27], [28]) and Wang et al.([29]), propose attention-based mechanisms to provide better explanations for driving decisions. The driving models we discuss in Chapter 4 and Chapter 5 mainly belong to this end-to-end learning-based category, but they also utilize processed results from some low-level perception modules, such as object detection, semantic segmentation, depth estimation and tracking.

2.2 Computer Vision Tasks for Driving

To make reliable driving decisions, both aforementioned approaches first need to receive information from the environment via a group of sensors including camera, wheel odometry, and range sensors (e.g., RADAR and LiDAR). Among various signals, the visual data captured by the camera has the richest information, requiring computer vision techniques to process this information and utilize it either explicitly or implicitly.

Computer vision research in the autonomous driving field includes a wide spectrum of topics [17], including object detection, tracking, semantic segmentation, reconstruction, motion estimation, and scene understanding. Object detection ([8], [20], [30], [31]), tracking ([32]–[34]), and semantic segmentation ([12], [35]–[37]) are low-level perception tasks to recognize the traffic elements in the environment. Reconstruction maps 2D images into 3D geometry ([38]–[40]) and further provides the spatial configurations of the traffic scene. Motion estimation ([41]–[43]) determines spatial-temporal dynamics of the ego vehicle. Based on all of this processed information, scene understanding ([44]–[47]) aims to obtain a rich but compact representation of the scene.

All of these computer vision tasks are generally related to this dissertation. In particular, we focus on object detection and scene understanding in Chapter 3 and Chapter 4, respectively. Additionally, in Chapter 4 and Chapter 5, semantic segmentation and tracking are used to provide perception cues as the input to the driving model. Inspired by the basic 3D reconstruction techniques, we compute the 3D distance and use the distance constraint as prior knowledge to interaction modeling in Chapter 4 and Chapter 5.

3. PERCEPTION IN PHOTON-LIMITED CONDITIONS

3.1 Introduction

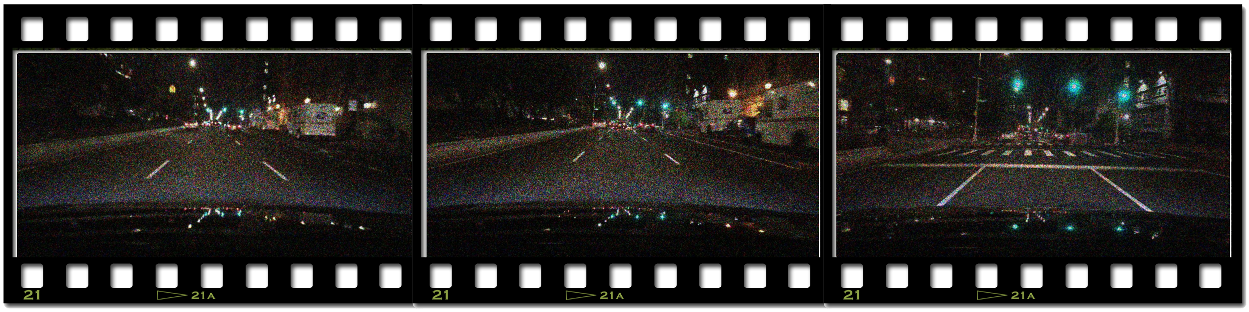


Figure3.1. Examples of driving in photon-limited conditions.

Robust perception is crucial and is the first level of SA. According to [48], 76% of SA errors in pilot are originated from failures of missing important information in the perception stage. Similarly, miss detection of traffic elements, e.g., oncoming vehicles and crossing pedestrians, can mislead ADAS to ignore the potential risks and cause danger. Thus, it is important to develop a reliable object detection system to perceive traffic elements.

State-of-the-art object detection methods such as Faster R-CNN [8] and YOLO [9] are widely used in driving applications to provide information of traffic objects, but their operating regimes have been limited to well-illuminated scenes with a sufficient amount of photons. As the number of photons decreases so that the signal-to-noise ratio becomes lower, the performance of these detectors will also degrade. For common situations – driving at night in suburban Figure 3.1, or even in the countryside with only moon light, developing a more robust object detection algorithm presents a pressing need. The goal of this chapter is to fill the gap by demonstrating object detection in real driving scenarios where existing methods fail to work.

Photon-limited imaging refers to image acquisition under a condition where the number of measured photons is very low. The fundamental limit is attributed to the Poisson process of the photon arrivals. This randomness is present even if the sensor is perfect – no read noise, no dark current, and has a uniform pixel response. Because the randomness is the nature of the problem, a photon-limited object detection algorithm must be able to extract the

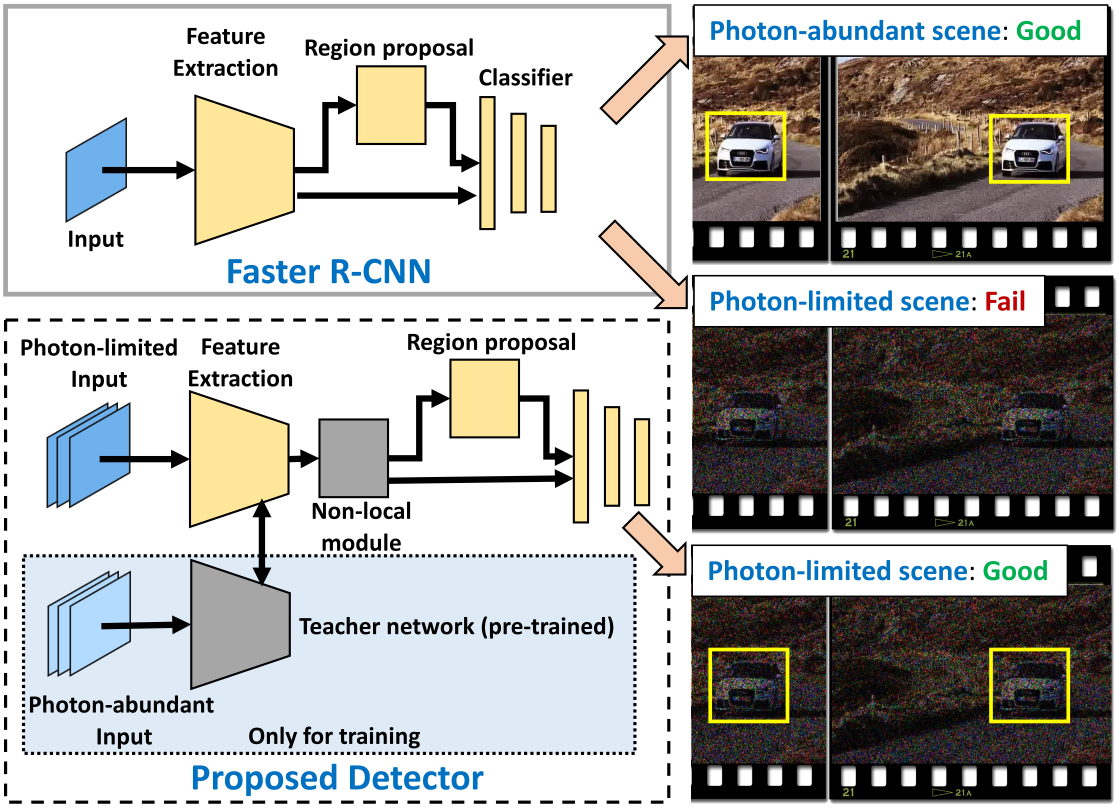


Figure3.2. We present a new object detection method for photon-limited conditions. While traditional detectors fail because the signal is too weak, our method addresses the problem by proposing two improvements: (1) Space-time non-local module, and (2) Student-teacher learning.

weak signal from the noise. Existing low-light enhancement algorithms have demonstrated promising results of improving the contrast of low-light images. In this chapter, we are interested in pushing the limit further by considering images that do not only have a low contrast but are also contaminated with shot noise.

The contributions of this chapter are summarized in Figure 3.2. While conventional methods such as Faster R-CNN fail to detect objects under photon-limited conditions, we propose two improvements to overcome the difficulty:

- Leverage spatial-temporal redundancy. We assume that the input data is a burst of photon-limited frames. Although motion exists across the burst of frames, the total signal-to-noise ratio (SNR) of a burst is higher than a single frame. By borrowing ideas from the non-local neural network [49], we build a space-time non-local feature aggregation module to assemble neighboring space-time features.

- Regularize features via student-teacher knowledge distillation. The construction of the non-local features is based on feature matching. The success of feature matching depends on the SNR of the features. To maximize the SNR of the features, we employ a knowledge distillation technique where the feature extraction module of a student network is trained to mimic the features produced by a pre-trained teacher.

By incorporating the two improvements into Faster R-CNN, we offer improved detection performance. We conduct extensive experiments on both synthetic and real data. For the synthetic experiment, we utilize a standard object detection benchmark (PASCAL VOC 2007 [10]) and a driving dataset (Berkeley DeepDrive 100K [11]). Our experimental results show that the new algorithm outperforms the baselines by more than 6% in mean accuracy precision (mAP). Specifically, for object classes related to driving scenes, our method surpasses the baseline by 10.2%, demonstrating the capability of perceiving traffic objects in the photon-limited condition. Given a desired mAP level, our system requires up to 50% fewer photons. When combined with the latest single-photon image sensors [50], we achieve object detection at 1 photon per pixels (PPP) or lower on real images.

3.2 Background

The taxonomy of the object detection methods is outlined in Figure 3.3, where we compare different detection tasks/methods against the photon-level (measured in lux) and the sensor gain (measured in ISO).

3.2.1 Baseline / Vanilla Methods

The mainstream object detection methods that are trained using large scale data set such as ILSVRC [69] and COCO [70] typically operate at the right most column of Figure 3.3 where the number of photons is sufficient. Depending on the input data format, the methods can be categorized into two group:

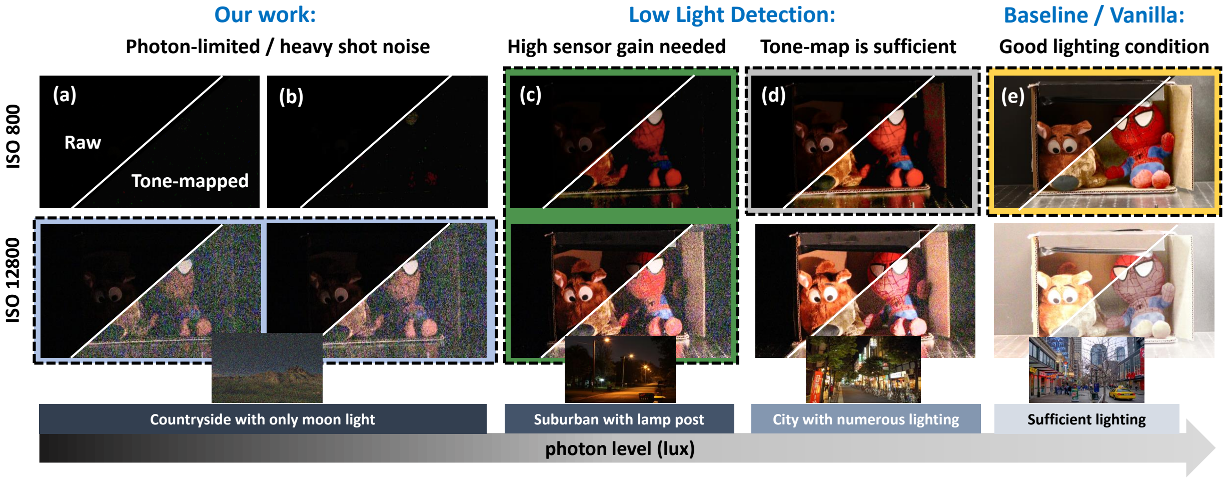


Figure3.3. While baseline/vanilla methods [8], [9], [12], [51]–[64] are designed to handle well-illuminated scenes, this chapter focuses on the photon-limited regime where signals are very weak. Existing "low-light" methods [65]–[68] typically do not operate in such an extreme condition where the signal is weak even after tone-map and/or adjusting the sensor's ISO.

- **Single-image** detection methods that detect objects from a single image. Some of these methods focus on speed and real time processing capability [9], [51]–[53], whereas other methods based on region proposal focus on detection performance [8], [12], [54], [55]. On top of these methods, various work are proposed by leveraging multi-scale information [71], making network fully convolutional [55], utilizing multi-task training [12], tackling foreground-background imbalance [52], and improving bounding box prediction quality [72], [73].
- **Video** detection methods that detect objects from multiple frames of a video. The premise of these methods is that the temporal information and the spatial-temporal redundancy provides valuable information for the detection. The aggregation of the temporal cues are typically done at two levels: (i) feature level aggregation [56]–[61], and (ii) box level aggregation [61]–[64].

Despite the abundance of baseline methods, the networks and training are not designed for photon-limited conditions. As a result, directly applying these methods to our problem is ineffective (performance is limited even if one augment training data) and inefficient (pre-processing could be computationally expensive but does not necessarily lead to unparalleled performance), as demonstrated in [66], [74] and in our experiment.

3.2.2 Low-Light Detection Methods

Conventional low-light image processing methods can handle darker images than the baselines as shown in Figure 3.3(c) and (d). The easier case, as shown in Figure 3.3(d), happens when the lighting condition is not properly adjusted. However, information is mostly intact after tone-mapping and contrast enhancement. Image enhancement for this class of problem has been extensively studied [75]–[89]. For object detection, Loh et al. [65] and Yang et al. [66] created large-scale real low light detection data sets. The state-of-the-art detection systems in this scenario adopt Multi-Scale Retinex with Color Restoration (MSRCR) algorithm [75] for pre-processing and fine tune detectors on pre-processed data [66]. As will be shown in the experiment section, this strategy fails to work on photon-limited images; the strong photon shot noise will void the illumination smoothness assumption held by the Retinex model.

The harder case of the two, as shown in Figure 3.3(c), happens when the photon level is further reduced. In this operating regime, one needs to switch to a high sensor gain (higher ISO) so that the details can be observed. As far as object detection algorithms are concerned, to the best of our knowledge, no large scale detection dataset is available to date. Instead, Sasagawa et al. [67] treat detection in this scenario as a domain adaptation problem and use knowledge distillation to train a detector with normal lighting detection data and SID reconstruction data set [68]. In our study, we simulate the physical process of photon-limited image formation and demonstrate that our simulation enables our model to work on real photon-limited images.

3.2.3 Photon-Limited Imaging Methods

When the light level is extremely low or the exposure time is extremely short, each pixel only receives a handful of photons. Images captured under this condition are dominated by photon shot noise as shown in Figure 3.3(a)-(b), which are the cases of interest in this chapter.

For object detection at this photon level, the pioneer study by Chen et al. [90] shows the feasibility of performing classification under such condition on MNIST [91] data set. Various new types of image sensors have been developed over the past few years, including the single-photon avalanche diodes (SPAD) [92]–[99] and the quanta image sensors (QIS) [100]–[105]. A lot work has also been done in the signal processing side of both these sensors [106]–[116]. Specific to high-level computer vision tasks, Gyongy et al. demonstrated tracking and reconstruction of rigid planar object at this light level [117]. Gnanasambandam et al.[74] and Chi et al. [118] achieved image reconstruction and classification by combining student-teacher training scheme. The proposed idea is inspired by the student-teacher scheme. To further improve the performance, we introduce a spatial-temporal non-local module to leverage the information from neighbor frames. Our method generalizes the conventional detection methods by providing a more robust detection under photon-limited conditions.

3.3 Our Method

Given a sequence of photon-limited frames, our goal is to localize objects and identify their classes in *all* frames. Our proposed system is trained on data obtained from Sec 3.3.1 and consists of key components: the non-local module (Sec 3.3.2) and the student-teaching learning scheme (Sec 3.3.3).

3.3.1 Image Formation Model

Under a photon limited condition, the signal generated by the image sensor, $\boldsymbol{x}$, is modeled through a Poisson process [74], [90], [118]:

$$\boldsymbol{x} = \text{Poisson}(\alpha \cdot \text{CFA}(\boldsymbol{y}_{\text{RGB}}) + \boldsymbol{\eta}_{\text{dc}}) + \boldsymbol{\eta}_{\text{r}}, \tag{3.1}$$

where CFA stands for the color filter array. $\boldsymbol{y}_{\text{RGB}}$ is the clean RGB image in the range $[0, 1]$. α determines the average number of photons arriving at the sensor and therefore it depends on the exposure time and the average photon flux of the scene. $\boldsymbol{\eta}_{\mathrm{dc}}$ is the dark current, and $\boldsymbol{\eta}_r \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathrm{r}} I)$ is the readout noise with standard deviation σ_{r}.

The final output $\boldsymbol{x}$ is truncated at 3 standard deviation from mean pixel values and re-normalized to the range $[0, 1]$. All frames are assumed to be statistically independent, as the Poisson process and the noise are independent [119]. In our experiments, we used values listed in Table 3.1, following [74], [90], [120]. The dark current parameter is set to 0 as it is insignificant compared to other noise sources on modern sensors when the exposure time is short.

Table3.1. Data synthesis parameters used in our experiments.

α	$\boldsymbol{\eta}_{\mathrm{dc}}$	σ_{r}
0.25 — 5	$\mathbf{0}$	0.25 or 2

3.3.2 Space-Time Non-Local Module

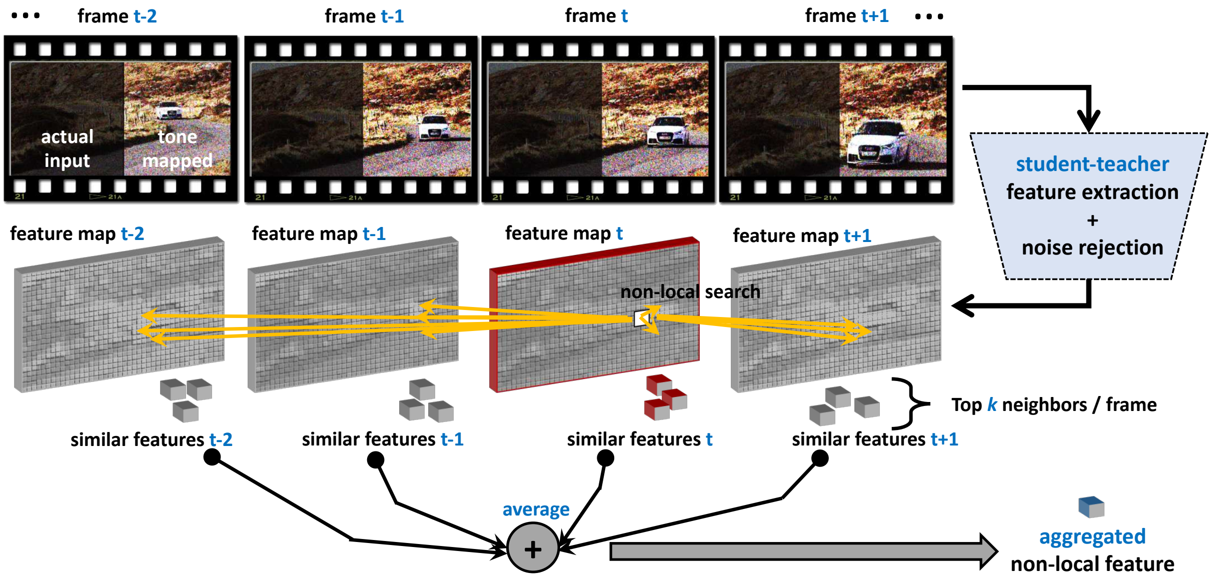


Figure3.4. Our proposed non-local module and student-teacher training scheme. The teacher network is first pre-trained on photon-abundant data and it enforces the student to extract noise-rejected features of each input frame. By applying the non-local search in the feature space, similar spatial-temporal features are aggregated to update the key frame features.

The biggest challenge of detecting objects under photon-limited conditions is the presence of intense shot noise. Our solution to extract signals from the noise is to utilize the spatial-temporal redundancy across a burst of frames. Our hypothesis is that if we are able to find similar patches in the space-time volume, we can take a non-local average to boost the signal. To achieve this goal, we design a non-local module as depicted in Figure 3.4.

Given an image sequence, each frame is fed into a feature extractor (the student-teacher module, which will be discussed in Sec 3.3.3) to obtain the feature maps. For each feature vector at location $(\mathrm{i}, \mathrm{j}, t)$, we conduct a non-local search for similar features by computing the inner-products of this feature and all the candidate features in the adjacent frames. This operation produces a set of scalars representing the similarities between the current feature and the features in the space-time neighborhood. Then for every time t, we select the top-k candidates with the highest inner product values. As shown in the experiments, we find that $k = 2$ is an appropriate number for most of the experiments. After picking the top-k features, we take the average to generate the aggregated non-local feature.

Our proposed space-time non-local module differs from the traditional non-local neural networks [49] in the following two aspects:

- Before computing the similarity, [49] uses convolutional layers to first project features onto another feature space. This additional feature space is designed to represent high-level semantic meanings of the scene, such as interactions. For photon-limited imaging where the SNR is low, such semantic-level features are generally more corrupted and hence they are less reliable than low-level features. In addition, feature projection could cause confusion to our spatial-temporal feature matching step because the noise is heavy.
- [49] aggregates *all* space-time information via a softmax weighted average. We only average partially the space-time information from the top-k features because irrelevant features in the time-space can distract our model. In the experiments, we demonstrate that the top-2 features per frame are sufficient for our purpose.

3.3.3 Knowledge Distillation

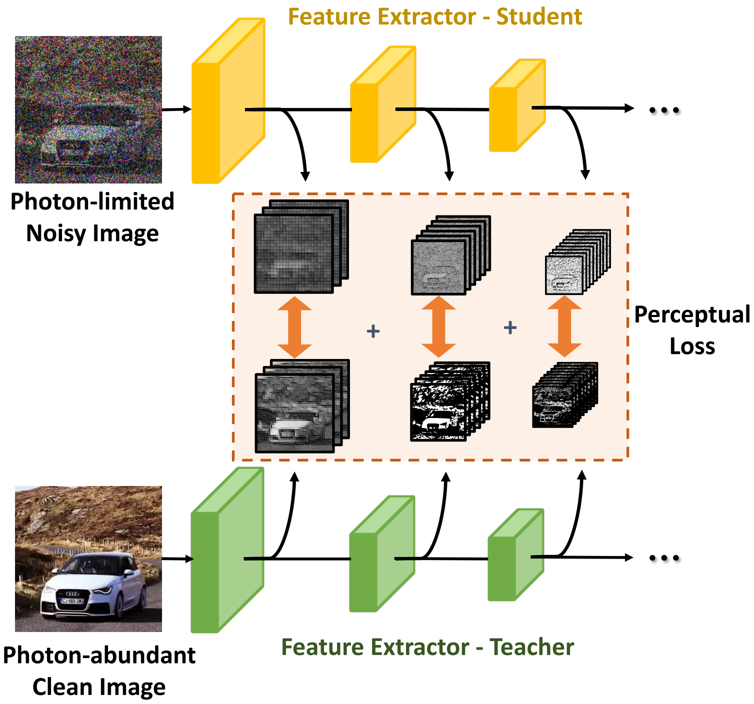

Feature Extractor - Student
Photon-limited Noisy Image
Perceptual Loss
+
+
Photon-abundant Clean Image
Feature Extractor - Teacher
...
...

Figure3.5. Knowledge distillation via student-teacher learning. The teacher network is pre-trained on clean images. We train the student network by minimizing the perceptual loss which measures the pixel-wise difference of the features.

The performance of the non-local feature matching depends heavily on the SNR of the features. If the features are contaminated by noise, finding correct feature correspondence would be difficult. Inspired by [74], [118], we address this issue by introducing a knowledge distillation step known as the student-teacher learning scheme to regularize the features. The idea is to train the student feature extractor by minimizing its L_2 distance with a teacher pre-trained on clean data so that the features extracted by the student are denoised.

Figure 3.5 depicts the idea of the proposed student-teaching learning scheme. In this figure, we have a teacher network and a student network. The teacher network is pre-trained using well-illuminated images. The student network has the same architecture but it is used to extract features from the photon-limited data (i.e., noisy). In the training stage, the parameters of the teacher network are fixed and those of the student network are trainable.

Because the teacher network is trained to handle clean images, it generates noise-free features when it is fed with clean images. We want features produced by the student network to be similar to those of the teacher. To this end, we introduce regularization to the student network by defining a **perceptual loss**:

$$\mathcal{L}_p = \sum_{\mathrm{i}=1}^{N} \|\widehat{\phi}_{\mathrm{i}}(x_{\text{clean}}) - \phi_{\mathrm{i}}(x_{\text{noisy}})\|^2, \tag{3.2}$$

where $\widehat{\phi}_{\mathrm{i}}(x_{\text{clean}})$ and $\phi_{\mathrm{i}}(x_{\text{noisy}})$ are the i-th layer's feature of the teacher and student network, respectively. The perceptual loss is the Euclidean distance measuring the difference between the student's and the teacher's features. Minimizing the perceptual loss forces them to be close in the feature space. This further enforces the network to denoise the image and generate good representations before non-local feature matching.

The overall training loss of our detector consists of the perceptual loss $\mathcal{L}_p$, the standard cross-entropy loss, and the regression loss [8].

3.3.4 Rationale of Our Design

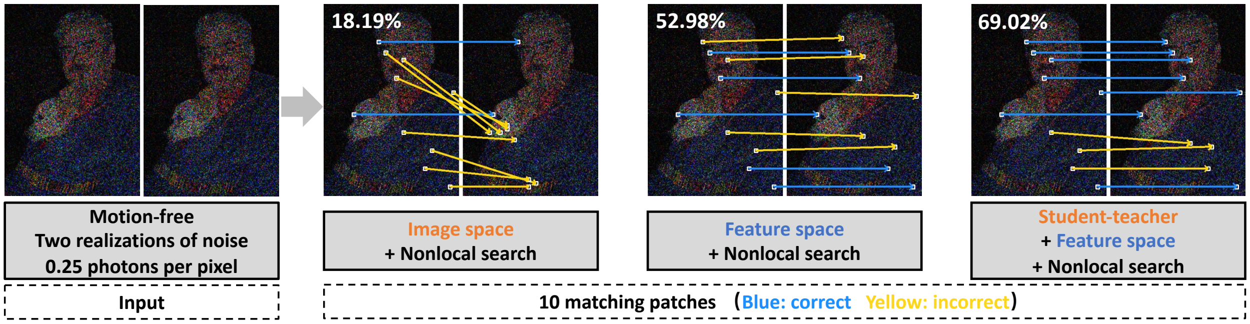


Figure3.6. Comparison of different non-local patch matching methods. We synthesize two i.i.d. copies of a photon-limited image. For each competing configuration, we visualize 10 matching patch examples. The blue and yellow arrows indicate correct and incorrect matching, respectively. As the image pair is motion-free, the correct matches should be indicated by horizontal arrows. The combination of non-local search and student-teacher learning demonstrates the best performance.

To illustrate the benefit of the proposed non-local module and the student-teacher learning scheme, we conduct an experiment in this section.

In Figure 3.6, we synthesize two independent and identically distributed (i.i.d.) copies of a photon-limited image at a photon level of 0.25 photons per pixel (ppp). We use this pair of images to check how the feature matching step performs. Three methods are compared: 1) non-local search in the image space (i.e., the original non-local search), 2) non-local search in the feature space, and 3) student-teacher + non-local search in the feature space. In the image space, for each $h \times w$ patch, we compute its normalized cross-correlation (NCC) with all $h \times w$ patches in the other image and choose the one with the highest NCC as its matching patch. In the feature space, we use features trained with or without student-teacher training and find correspondence for every feature vector. The correspondence is visualized by the center of the receptive field of feature vectors.

The benefit of the proposed method can be seen in two aspects: accuracy and speed. As illustrated in Figure 3.6, the non-local search in the feature space has a much higher success rate of finding correct correspondence than the same method applied to the image space. The student-teacher training further increases the performance by enhancing the robustness of the feature extractor against noise. We performed the experiment for 100 images and we observed that the trend was consistent.

For the speed, non-local search in image space is computationally more expensive than in the feature space. Given an $H \times W$ image with desired patch size $h \times w$, the feature matching process takes approximately $(HW)^2 hw$ floating-point operations (FLOP) in the image space and $(\frac{HW}{S})^2 C$ FLOP's in the feature space, where C is feature vector dimension and S is spatial resolution compression ratio by the feature extractor. Reducing the patch size reduces the computation cost, but the matching quality deteriorates significantly. In our implementation, we use 64×64 for the image space search and it takes ~ 256 times more computation than in the feature space.

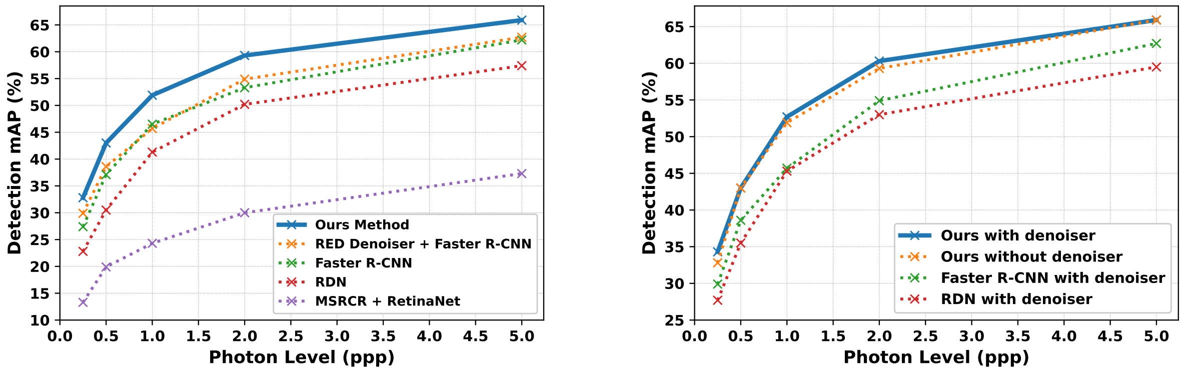
3.4 Experiment

3.4.1 Experimental Settings

Dataset. We use the procedure outlined in Sec 3.3.1 to synthesize training data of the photon-limited images from the Pascal VOC 2007 dataset [10]. To synthesize motion across the frames, we introduce a random translation of image patches. The total movement varies from 7 to 35 pixels across 8 frames similar to [118]. For testing, we created two synthetic testing datasets. One is based on standard object detection dataset – PASCAL VOC 2007 [10] and another is from a driving scene dataset – Berkeley DeepDrive (BDD) 100K [11]. We have also collected a dataset of real images for testing. The read noise of our model is assumed to be 0.25e^-, based on the sensor reported in [50]. The average photon level we tested ranges from 0.1 to 5.0 photons per pixel (ppp). With an f/1.4 camera, 1.1μm pixel pitch, and 30ms integration, this range of photons roughly translates to 0.02 lux to 5 lux (typical night vision scenarios). For real data, we use the GJ01611 16MP photon counting Quanta Image Sensor developed by GigaJot Technology [50].

Implementation Details. Our method is implemented in Pytorch based on [121]. The framework takes a T-frame image sequence as input, and T is set to be 1, 3, 5 and 8 in the following experiments. Following [8], we adopt ResNet-101[122] pretrained on ImageNet [123] as the backbone. We apply the perceptual loss to the features obtained from `block_1`, `block_2` and `block_3` of ResNet-101 and the non-local module is processed on the features from `block_3`. We utilize RoIAlign [12] to extract the features from object proposals and `block_4` is further applied to the extracted proposal features before the final classifier. The model is trained for 20 epochs and we use Adam [124] optimizer with default parameters, learning rate 0.001, and weight decay 0.1 every 5 epochs.

Competing Methods. We compare our method with four baselines. (a) A generic image object detector: Faster R-CNN [8], fine tuned using the photon-limited data we synthesized; (b) A video object detector: Relation Distillation Network (RDN)[62], also fine tuned using photon-limited data; (c) A low-light detection framework: color restoration algorithm (MSRCR) [75] plus a detection RetinaNet [52], which is one of the winning solutions of 2019



(a) Comparison with baselines

(b) Comparison with image denoisers.

Figure3.7. Experiments on synthetic data. (a) Compare different object detection methods: Faster R-CNN[8], RED[125] + Faster R-CNN[8], RDN[62], and MSRCR[75] + RetinaNet[52]. (b) Compare methods that use image denoising as a pre-processing step.

UG^2+ low-light face detection challenge; (d) A two-stage pre-denoised detection framework: RED-Net [125] plus Faster R-CNN [8].

3.4.2 Main Results

Our first experiment is conducted on synthetic data of PASCAL VOC 2007 [10] test set. We use 8-frame inputs with the number of features for non-local aggregation set to 2 per frame in the following experiments.

Comparison with the baselines. Figure 3.7a shows the detection rate, measured in mean average precision (mAP), as a function of the photon level, measured in photons per pixel (ppp). The proposed method consistently outperforms the competing methods across the tested photon levels from 0.25 ppp to 5.0 ppp. The difference between our method and the second-best method is as large as 6% in terms of mAP when the photon level is 2.0 ppp.

Comparison with image denoisers. When handling noisy images, a natural solution is to first run a denoiser and feed the denoised images into a standard object detector. Figure 3.7b depicts the comparisons with such baseline methods. The denoiser we use is the RED-Net [125] previously used in other photon-limited imaging papers such as [118]

Table3.2. Comparison of different network designs. Relative mAP increase are reported with respect to Faster R-CNN baseline. The unit is %. ST: student-teacher learning; NL: non-local module; ST+NL:student-teacher learning + non-local module.

Photon Level (ppp)	0.25	0.5	1.0	2.0	5.0
ST	9.12	6.20	4.52	5.44	2.57
NL	16.06	14.56	9.89	10.13	5.14
ST+NL	**20.07**	**15.90**	**11.61**	**11.26**	**5.95**

and [74]. As the figure indicates, the proposed method outperforms the baselines by a big margin. In addition, adding a denoiser to the proposed method offers almost no additional benefit. Therefore, the proposed method has effectively executed the denoising task without requiring another network for denoising.

Different network designs. Table 3.2 demonstrates the importance of the space-time non-local module and the student-teacher learning module. In this table, we present the relative performance gain compared with Faster R-CNN baseline [8]. The addition of the non-local module and the student-teacher training shows improvement upon the baseline. We observe that the performance gain shrinks when the photon level increases, as detection becomes easier. The combination of both designs shows the best performance across all photon levels, especially in extremely low light, where the relative gain is 20.07%.

Required Photon Levels for Detection. In Figure 3.8, we discuss how many photons are needed for each pixel in order to achieve the target detection performance. The x-axis represents the detection accuracy we want to achieve and the y-axis is the minimal numbers of photons per pixel needed in the images. We compare four settings by switching the inputs from synthetic CIS to QIS images and changing the baseline method to our method. When the target mAP is 50%, QIS data only needs half photons of CIS data to reach the same accuracy by just using Faster R-CNN. By introducing our method, we can further decrease the required photon level by half on average.

Choice of Frame Numbers and K. Non-local module is applied to multi-frame input and searches for K similar features in each frame. Thus, we study the best and practical

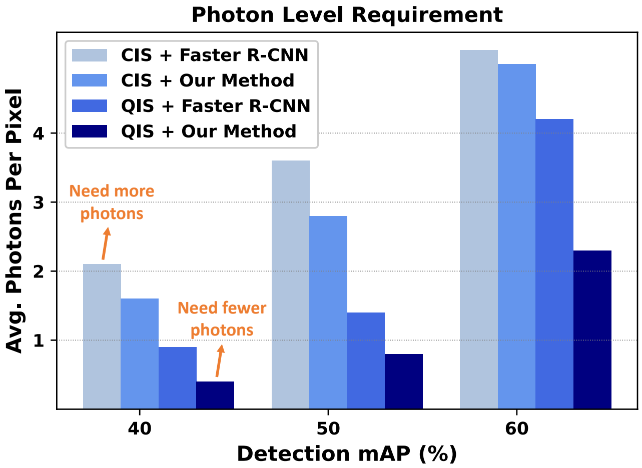


Figure3.8. Photon level requirement vs. detection performance.

Table3.3. A study of frame numbers and searched similar feature numbers. T is the number of frames input to our model and K is the number of searched features per frame for feature aggregation. We test our model under different photon levels from 0.25 to 5.0. For each column, the best mAP is shown in bold.

mAP (%)	ppp = 0.25		ppp = 0.5		ppp = 1.0		ppp = 2.0		ppp = 5.0	
	$T = 3$	$T = 8$	$T = 3$	$T = 8$	$T = 3$	$T = 8$	$T = 3$	$T = 8$	$T = 3$	$T = 8$
$K = 1$	32.3	**33.3**	41.5	42.8	49.6	51.9	58.4	59.0	65.1	**66.0**
$K = 2$	**32.7**	33.2	**41.6**	**43.0**	**50.0**	51.9	**58.7**	**59.3**	**65.6**	**66.0**
$K = 3$	32.4	33.2	41.5	42.8	49.9	**52.1**	58.6	59.2	65.4	65.9
$K = 4$	32.5	33.0	41.5	**43.0**	**50.0**	**52.1**	58.6	59.1	65.4	65.9

settings for our designed network. In Table 3.3, we find that using 8 frames is always better than 3 frames no matter which photon levels. It is easy to interpret this result because more frames provide more information and the proposed non-local module is able to associate similar patches across multiple frames. However, more input frames require more computations and processing time. When we set the frame number larger than 8, it will exceed the GPU memory. Thus we use 8-frame sequences as input for practical usage. Moreover, we discover that K=2 is the best choice for the number of searched similar features per frame. Too many selected features could be a distractor for the denoising purpose.

Table3.4. Detection results of real data. Each class column shows the number of correct detections versus ground truth. The last column is the overall mAP.

	person	car	sheep	mAP (%)
Faster R-CNN	54/105	58/60	60/60	66.9
Ours	73/105	60/60	60/60	**87.9**

Real data. We collected 225 real images in low light and annotate objects from 3 categories: `person`, `sheep`, and `car`. We train our model using the synthetic data and verify the results using the real data. The results of these 225 testing images are shown in Table 3.4. On average, our proposed method achieves an mAP of 87.9% while the baseline method achieves 66.9%.

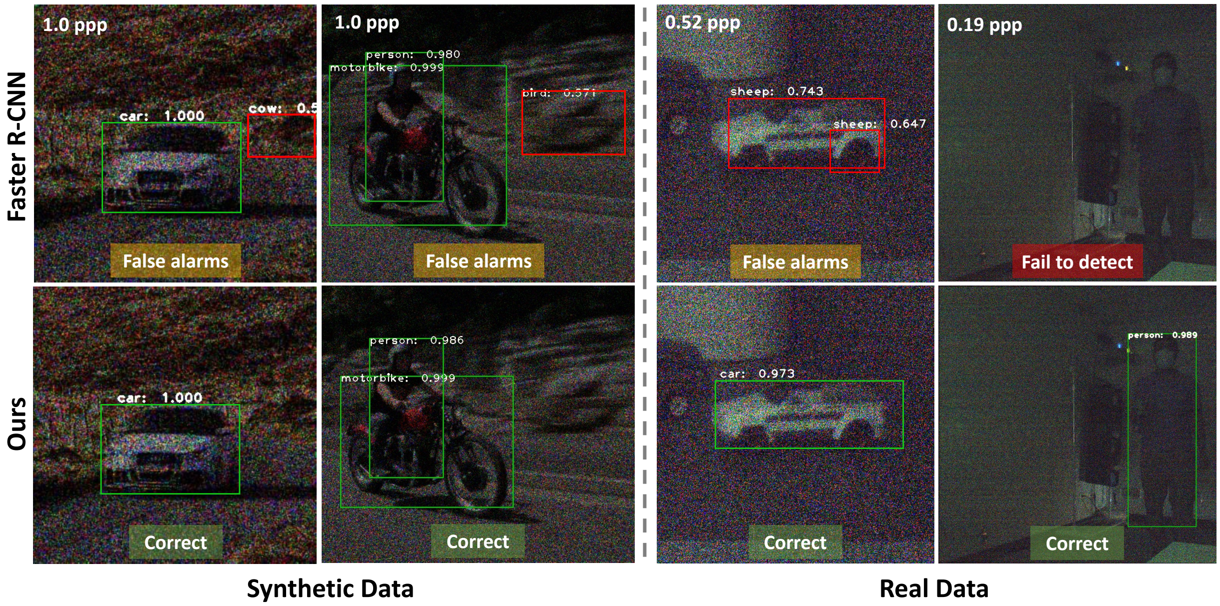


Figure3.9. Detection results on synthetic and real data. The top row is the Faster R-CNN baseline. The bottom row is our method. The photon level is shown in the top-left corner. Correct/Incorrect results are in green/red, respectively. The real data is captured by Gigajot Technology 16 MP Photon Counting Quanta Image Sensor (GJ01611).

Figure 3.9 shows a qualitative comparison of traffic object examples between our method and the baseline Faster R-CNN. The result shows that the baseline suffers from either false alarms or missed detection. In contrast, the proposed method is able to detect the static toy car and moving person on the real data when the photon level is 0.52 ppp and 0.19 ppp, respectively. In Figure 3.10, we show more qualitative examples of comparisons. All of the four scenarios are dynamic scenes. The first two are synthetic data and the photon levels are set to 2.0 ppp and 1.0 ppp. The last two scenes are real data captured at photon levels of 0.28 ppp and 0.19 ppp. We observe that the presence of heavy shot noise results in false alarms detected in the background, such as the sheep and the bird in scene 2. Also, the baseline method fails to detect the moving person for most of the time in scene 3.

Table3.5. Detection results of 6 traffic objects. Each class column shows the average precision (AP in %) of each class. We compare our method with a baseline method (Faster R-CNN [8]) at five different photon levels.

Photon Level (ppp)	Method	bike	bus	car	motor-bike	person	train
0.25	Faster R-CNN	34.1	38.3	45.5	41.3	38.2	40.5
	Ours	37.5	45.1	54.4	48.0	48.4	45.1
0.5	Faster R-CNN	44.6	49.6	56.6	49.8	51.2	51.7
	Ours	54.8	56.9	62.6	56.1	57.7	58.4
1.0	Faster R-CNN	58.1	59.5	66.0	57.8	58.2	60.3
	Ours	62.8	65.7	69.5	65.5	64.9	64.7
2.0	Faster R-CNN	66.3	68.2	70.9	65.6	64.1	63.3
	Ours	74.1	72.5	75.5	70.3	69.5	68.2
5.0	Faster R-CNN	74.4	72.6	75.8	71.6	72.6	72.4
	Ours	78.5	76.4	78.5	72.3	74.2	74.3

Perception of Traffic Elements. Among the 20 classes of objects from PASCAL VOC 2007 [10], there are 6 classes highly related to traffic scenes – `bike`, `bus`, `car`,`motorbike`, `person` and `train`. In Table 3.5, we compare our method with Faster R-CNN [8] by showing

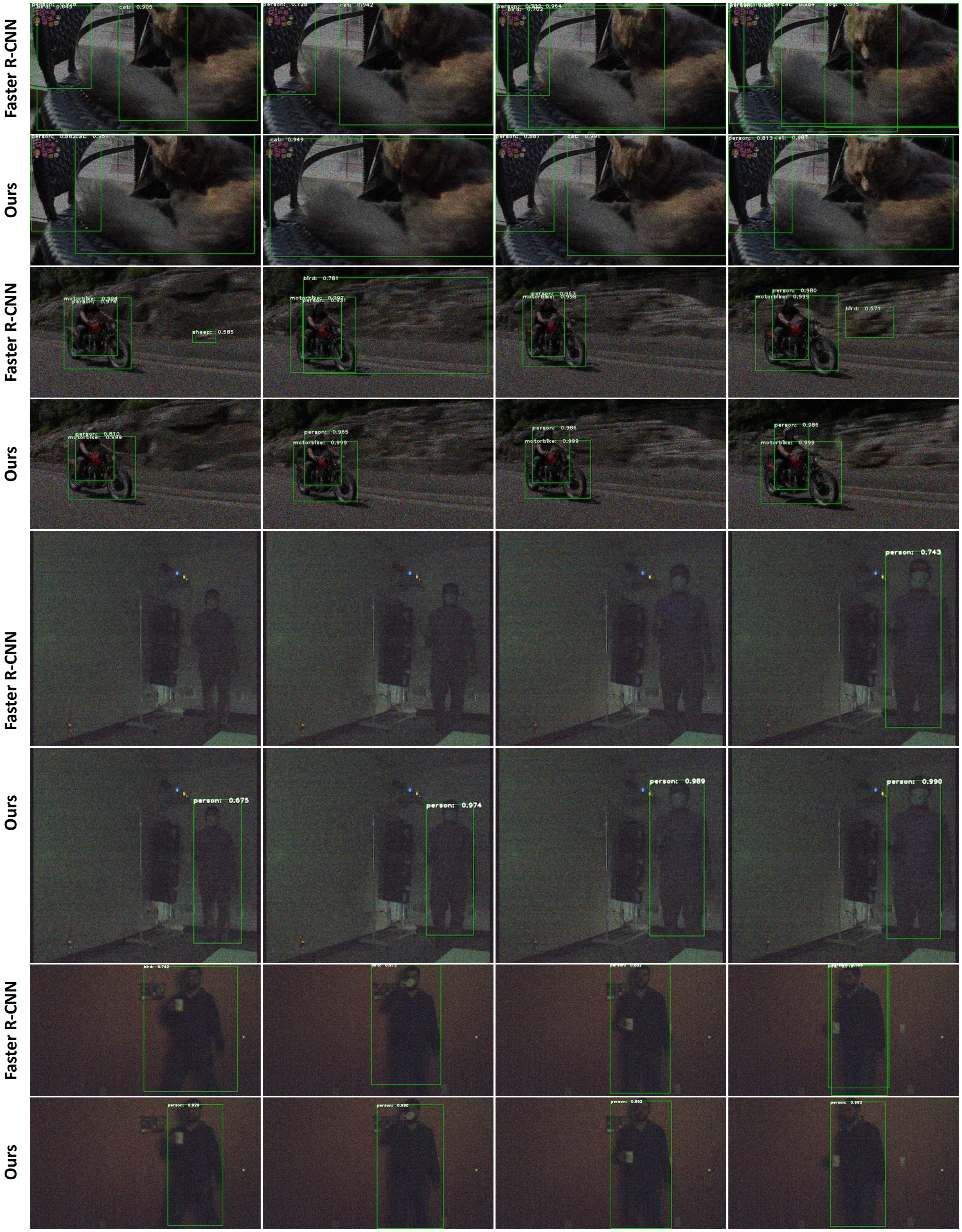


Figure3.10. More detection results on synthetic and real data.

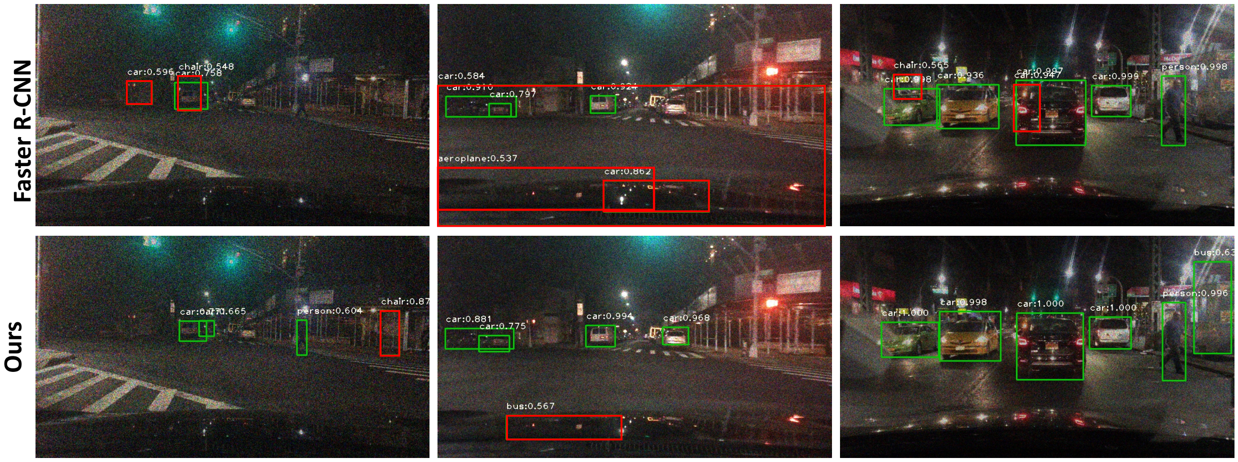


Figure3.11. Examples of qualitative results on BDD 100K [11] The top row is the Faster R-CNN baseline. The bottom row is our method. Correct/Incorrect results are in green/red, respectively.

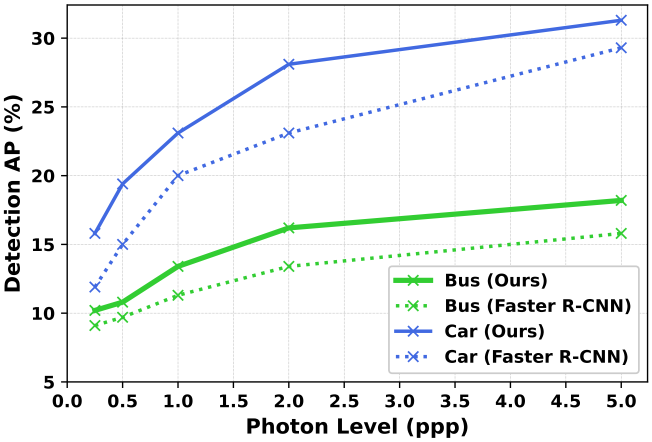


Figure3.12. Experiments on BDD 100K [11]. The x-axis represents the photon level and y-axis is the average precision (AP) of each class.

detection results of these traffic objects at five different photon levels. Under the same condition, our method is always better than the baseline for all traffic classes. The performance difference can be as large as 10.2% when the photon level is less than 1.0. To obtain a deeper understanding of perception in driving scenes, we also test our algorithm on a dataset called Berkeley DeepDrive (BDD) 100K [11], which provides diverse driving videos with rich annotations. We focus on two main object categories, i.e., bus and car, which dominates the dataset. Results are shown in Figure 3.12. We discover that there is always a significant performance gap between our method and Faster R-CNN [8]. The difference of car category between our method and the competing method is as large as 5 % in terms of AP when the photon level is 2.0 ppp. We also present some qualitative results in Figure 3.11 and observe that the baseline method tends to produce more false alarms. In the first example (the most left one), the baseline method misses the crossing pedestrian which is a fatal mistake and may cause serious serious accidents.

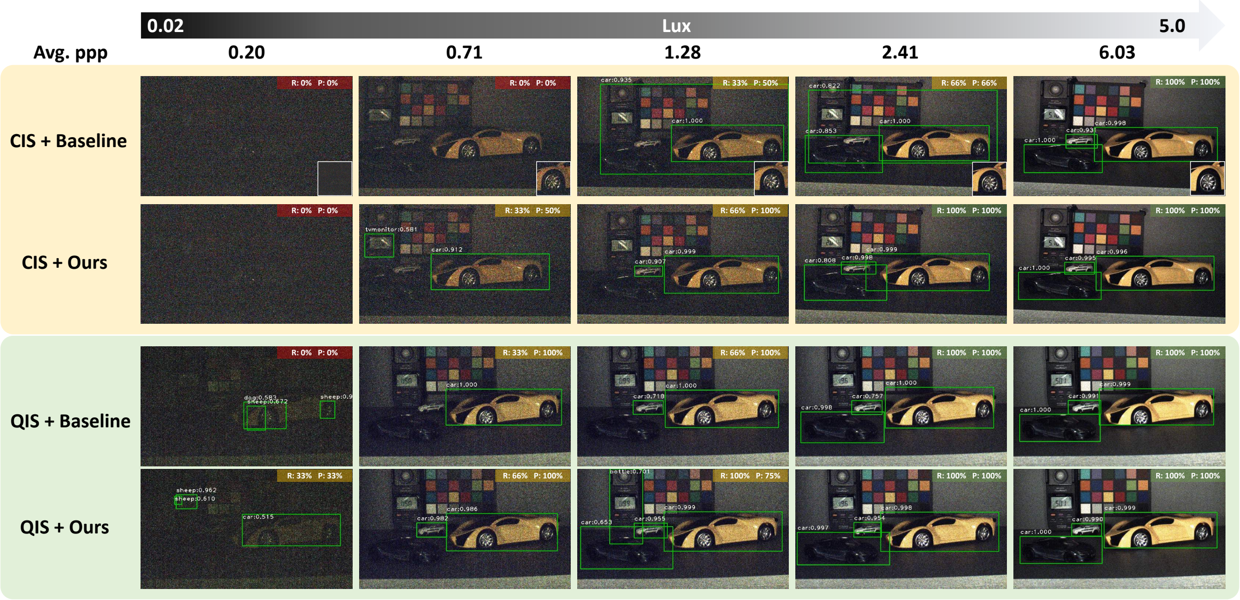


Figure3.13. Comparison of different sensors and different methods on real data. The visualized figures are tone mapped and the baseline method is Faster R-CNN. We choose 5 different lux levels ranging from 0.02 to 5.0, equivalent to Avg. ppp ranging from 0.20 to 6.03. In the right-top corner of images, the recall (R) and precision (P) are computed, enclosed in frames with different colors. Red/Yellow/Green indicates totally failed/partially correct/totally correct, respectively. In the first row, we zoom into the left-front side of the yellow car and show details in the right-bottom box. We can see that in the extremely low light condition, the images suffer from the high-noise problem.

3.4.3 Performance comparison with CIS and QIS

We evaluate the proposed method with a conventional CMOS image sensor (CIS) from Google Pixel 3XL and a GJ01611 Quanta Image Sensor (QIS) from Gigajot Technology [103] under different illumination levels. By combining the proposed algorithm with the QIS device, we demonstrate the performance of the proposed detection method under extremely photon-limited conditions (0.02 lux and only 0.20 ppp).

To ensure a fair comparison, we note that the CIS has a pixel pitch of 1.4μm and read noise of 2.14e$^-$, while the QIS has 1.1μm pixels and read noise of 0.22e$^-$. In the experiments, the f-number of the lens is adjusted to balance the difference of pixel sizes (f/1.8 for CIS and f/1.4 for QIS) in the two sensors and 30msec exposure time is used for both sensors.

The comparison results are shown in Figure 3.13. The images were taken under illumination levels from 0.02 lux to 5.0 lux. Under strong illumination conditions such as 5.0 lux, all the compared methods show high detection accuracy without any false alarms. However, as the illumination level decreases, the proposed algorithm shows significant advantages over the baseline methods. This performance improvement is further enhanced with the QIS compared to the CIS because of its ultra-low read noise. For example, under 0.02 lux and an average photon level of 0.20 ppp, only the combination of the proposed algorithm and the QIS device can successfully detect the yellow car in the scene.

3.5 Discussion

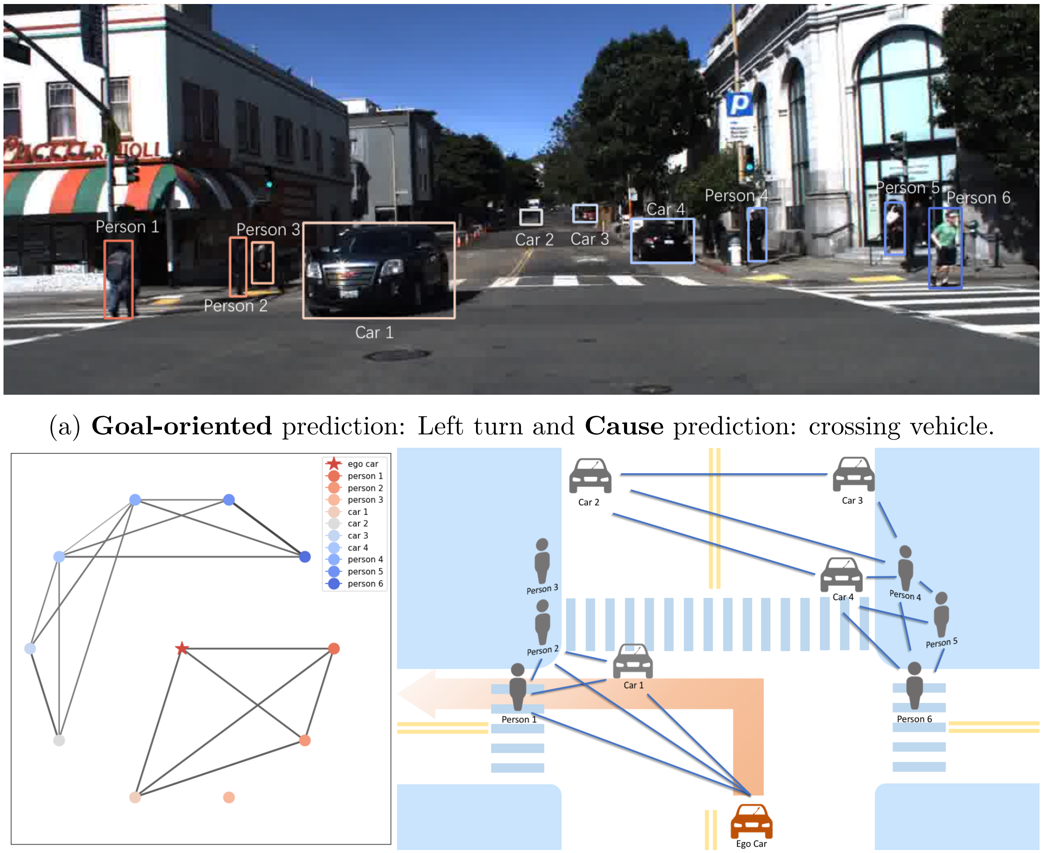
We proposed a photon-limited object detection framework. Our solution integrates a new non-local feature aggregation method and a knowledge distillation technique with the state-of-the-art detector networks. The two new modules offer better feature representations for photon-limited images. In comparison with the baselines, the proposed detector demonstrated superior performance in synthetic and real experiments.

Particularly, for driving scenarios, our proposed method also showcases the capability of detecting and perceiving traffic objects in the photon-limited condition, e.g. driving in the suburban at night. It is a fundamental step to enable Perception (Level 1 SA) and provides important cues for the safe driving as the goal of ADAS.

Also, when applied to the latest photon counting devices, we demonstrated object detection at a photon level of 1 photon per pixel or lower, significantly surpassing the existing CMOS image sensors and algorithms. It is envisioned that the new detection framework will enable a variety of applications, such as security, defense, life science, and consumer, as well as the emerging medical applications.

4. COMPREHENSION OF SPATIAL-TEMPORAL INTERACTIONS

4.1 Introduction



(a) **Goal-oriented** prediction: Left turn and **Cause** prediction: crossing vehicle.

(b) Learned affinity matrix

(c) BEV visualization of the learned affinity matrix.

Figure4.1. In a complicated traffic situation at intersections, the ego-vehicle intends to take a left turn while yielding to a upcoming vehicle. Our model learns a graph structure in (b) using edge connections to represent the interactions among road users and the ego-vehicle. The top-view scene representation in (c) is derived from (a) and (b) by overlaying learned relations on a scene layout for better illustration.

The comprehension of SA is about how to *combine*, *interpret*, *store* and *retain* information [5] given tons of perceived data. This includes processing multiple pieces of information and determining their relevance to the goal. As we know, the goal of ADAS is to increase drivers' safety in the complicated traffic environment. Then understanding how human drives and interacts with road users is an essential part of Level 2 SA for an intelligent automated driving system. The first step to achieve this is to develop a computational model which can capture the complicated spatial-temporal interactions between the ego-vehicle and road users.

Over the past decade there has been a significant advance in modeling spatial-temporal interactions [126]–[132]. However, most of the existing work still cannot effectively model complex interactions since many of them are leveraging "hand-crafted interaction models" [130]. Data-driven approaches are better options as they can learn subtle and complex interactions [130], [132]–[134]. However, existing approaches are still insufficient for three reasons.

First, the input used by several existing methods [130], [133], [134] is the human's 2D-location on bird's-eye-view (BEV) images. However, it is more desirable to use ego-perspective sensing devices, e.g, cameras, as humans use two eyes to sense. This calls for a specific design for egocentric interaction models. Second, using 2D pixel coordinates to model the 3D interactions (such as [132]) is insufficient because of perspective projection. BEV images can resolve this problem since the depth and spatial positions are both embedded in the BEV images. Third, the existing approaches only consider human-human or human-robot interactions, ignoring the environment factors, such as lane markings, crosswalks, and traffic lights. However, modeling these objects is nontrivial because they have irregular shapes.

In this chapter, we propose a 3D-aware egocentric spatial-temporal interaction framework for automated driving applications. Our method is the first method based on egocentric images and can address the aforementioned problems. The specific approach we take is to design two graph convolutional networks (GCN) [13] to model the egocentric interactions. We define two graphs, *Ego-Thing Graph* and *Ego-Stuff Graph* to encode how the ego-vehicle interacts with the *Thing* objects (e.g., cars and pedestrians) and the *Stuff* objects (e.g., lane markings and traffic lights). The *Ego-Thing Graph* is an improvement of Wu et al. [135]. We introduce two new concepts. We add an *Ego* node (i.e., the ego-vehicle) for egocentric interaction modeling, and we incorporate the objects' 3D locations (recovered from image-based depth estimation). The *Ego-Stuff Graph* is designed similarly. However, in order to extract features from irregular *Stuff* objects, we introduce a new method known as the MaskAlign operation.

We validate the proposed framework on tactical driver behavior recognition using Honda Research Institute Driving Dataset (HDD) [136]. The HDD is the largest dataset in the field. It provides 104-hour egocentric videos with frame-level annotations of tactical driver behavior. We validate our method based on two types of settings: 1) the ego-vehicle has interactions with *Stuff* objects (e.g., lane change, lane branch, and merge) and 2) the ego-vehicle has interactions with *Thing* objects (e.g., stop for crossing pedestrian and deviate for parked car). Our approach offers substantial performance boost (in terms of mAP, See Experiment section for definitions) over baselines on the two settings by 3.9% and 6.0%, respectively.

4.2 Background

4.2.1 Tactical Driver Behavior Recognition

Significant efforts have been made in tactical driver behavior recognition [24], [29], [126]–[128], [136]–[139]. Hidden Markov networks (HMM) were leveraged to recognize driver behaviors [126]–[128], [137], [138]. A single node in HMM encodes the states from the ego-vehicle, roads and traffic participants [126] into a state vector. In the proposed framework, we explicitly model the above three states using different nodes, each of which encodes its own representation according to the semantic context. Recently, convolutional and recurrent neural network based algorithms [24], [136], [139] are proposed. They implicitly encode the states of the ego-vehicle and road users using 2D convolution, and the state transition is via recurrent units. Our method explicitly models the states using graph convolutional networks (GCN) and uses temporal convolution networks for the state transition.

Wang et al., [29] designed an object-level attention layer to capture the impacts of objects on driving policies. Instead of simply weighting and concatenating objects' features, our framework preserves more complicated forms of interactions benefiting from GCN. Additionally, interactions between the ego-vehicle and road infrastructure are included in our system.

4.2.2 Graph Neural Networks for Driving Scenes

Recently, graph neural networks (GNN) [13], [140] has made significant progress in situation recognition [141], action recognition [142], [143], group activity recognition [135], and scene graph generation [144]. However, considerably less attention has been paid to driving scene applications

Herzig et al. [145] proposed a Spatio-Temporal Action Graph (STAG) network to detect driving collision. While STAG is similar to the proposed *Ego-Thing Graph*, our model explicitly exploits 3D locations of objects and the ego-vehicle into the design of nodes and edges. The 3D cue is essential in understanding scenes from egocentric perspective. This design is motivated by [135]. Note that 2D locations are used in [135] while we use 3D locations extracted from [146]. Moreover, we consider interactions between the ego-vehicle and road infrastructure that enable the proposed framework to be applied for diverse driving scene applications, e.g., learning driving model from images [147]. The details of our graph design can be found in Sec 4.3.1.

4.3 Our Method

An overview of the proposed framework is depicted in Figure 4.2. Given video frames, we apply instance segmentation and semantic segmentation in [12] to obtain *Thing* objects and *Stuff* objects, respectively. Object features are extracted from intermediate I3D [148] features via RoIAlign [12] and MaskAlign (Sec 4.3.2). Afterwards, we construct *Ego-Thing Graphs* and *Ego-Stuff Graphs* in a timely manner and apply graph convolutional networks (GCN) [13] for message passing. The updated *Ego* features from two graphs are fused and processed via a temporal fusion module. Additionally, the temporally fused *Ego* features are concatenated with the I3D head feature, which serves as a global video embedding, to form the egocentric representation. At last, this egocentric feature is passed through a fully connected layer to obtain the final classification.

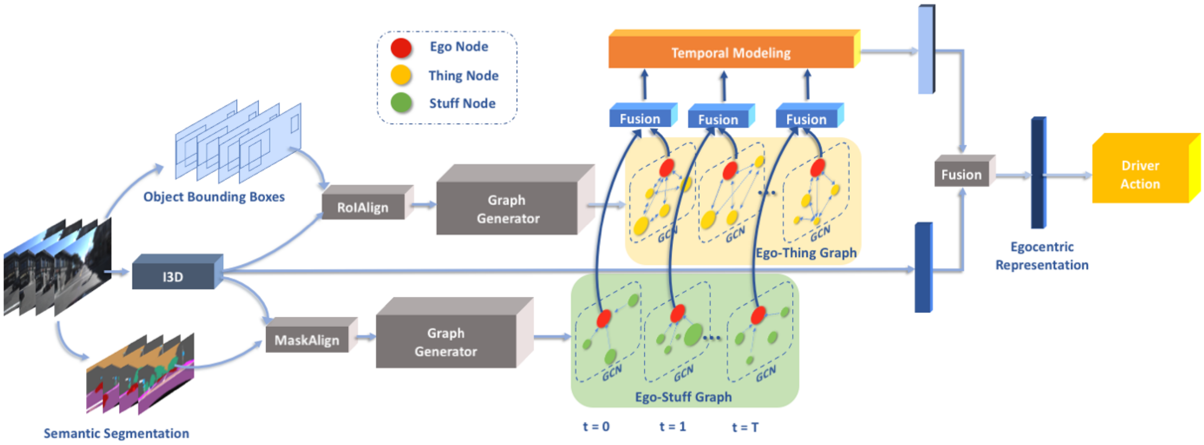


Figure4.2. An overview of our framework. Given a video segment, our model applies 3D convolutions to extract visual features followed by two branches: RoIAlign is employed to extract object features from object bounding boxes and MaskAlign is designed to extract features of irregular shape objects from semantic masks. Then, frame-wise *Ego-Thing Graph* and *Ego-Stuff Graph* are constructed to propagate interactive information among objects via graph convolution networks. The outputs of the two graphs are fused and fed into a temporal fusion module to form interactive representation. Finally, global video representation from I3D head and interactive features are aggregated as an input to tactical driver behavior recognizer.

4.3.1 Ego-Thing Graph

The *Ego-Thing Graph* is designed to model interactions among ego-vehicle and movable traffic participants, such as ⟨*car, ego-vehicle*⟩, ⟨*car, person*⟩ and so on.

Node feature extraction. In our design, *Thing* objects are *car*, *person*, *bicycle*, *motorcycle*, *bus*, *train*, and *truck*. Given bounding boxes generated from Mask R-CNN [12], we keep the top-K detections on each frame from all the classes above and set K to 20. Then RoIAlign [12] and a max pooling layer are applied to obtain $1 \times D$ dimensional appearance features as *Thing* node features in a *Ego-Thing Graph*. The *Ego* node feature is obtained by the same procedure from a frame-size bounding box.

Graph definition. We denote the sequence of frame-wise *Ego-Thing Graphs* as $\mathbf{G}^{ET} = \{\mathbf{G}^{ET}_t | t = 1, \cdots, T\}$, where T is the number of frames, and $\mathbf{G}^{ET}_t \in R^{(K+1)\times(K+1)}$ is the *Ego-Thing* affinity matrix at frame t representing the pair-wise interactions among *Thing* objects and ego. Specifically, $G_t^{ET}(\mathrm{i},\mathrm{j})$ denotes the influence of object j on object i. Nodes in graph correspond to a set of objects $\{(\mathbf{x}_\mathrm{i}^t, \mathbf{p}_\mathrm{i}^t)|\mathrm{i} = 1, \cdots, K+1\}$, where $\mathbf{x}_\mathrm{i}^t \in R^D$ is i-th object's appearance feature, and $\mathbf{p}_\mathrm{i}^t \in R^3$ is the 3D location of the object in world frame. Note that index $K+1$ corresponds to *Ego* object and $\mathrm{i} = 1, \cdots, K$ correspond to *Thing* objects.

Interaction modeling. *Ego-Thing* interactions are defined as second-order interactions, where not only the original state but also the changing state of the *Thing* object caused by other objects will altogether influence the *Ego* state. To sufficiently model these interactions, we consider both appearance features and distance constraints inspired by [135]. We compute the edge value $G_t^{ET}(\mathrm{i},\mathrm{j})$ as:

$$G_t^{ET}(\mathrm{i},\mathrm{j}) = \frac{f_s(\mathbf{p}_\mathrm{i}^t, \mathbf{p}_\mathrm{j}^t)\exp(f_a(\mathbf{x}_\mathrm{i}^t, \mathbf{x}_\mathrm{j}^t))}{\sum_{\mathrm{j}=1}^{K+1} f_s(\mathbf{p}_\mathrm{i}^t, \mathbf{p}_\mathrm{j}^t)\exp(f_a(\mathbf{x}_\mathrm{i}^t, \mathbf{x}_\mathrm{j}^t))} \tag{4.1}$$

where $f_a(\mathbf{x}_\mathrm{i}^t, \mathbf{x}_\mathrm{j}^t)$ indicates the appearance relation between two objects, and we set up a distance constraint via a spatial relation $f_s(\mathbf{p}_\mathrm{i}^t, \mathbf{p}_\mathrm{j}^t)$. Softmax function is used to normalize the influence on object i from other objects.

The appearance relation is calculated as below:

$$f_a(\mathbf{x}_\mathrm{i}^t, \mathbf{x}_\mathrm{j}^t) = \frac{\phi(\mathbf{x}_\mathrm{i}^t)^\mathrm{T}\phi(\mathbf{x}_\mathrm{j}^t)}{\sqrt{D}} \tag{4.2}$$

where $\phi(\mathbf{x}_\mathrm{i}^t) = \mathbf{w}\mathbf{x}_\mathrm{i}^t$ and $\phi(\mathbf{x}_\mathrm{j}^t) = \mathbf{w}\mathbf{x}_\mathrm{j}^t$. Both $\mathbf{w} \in R^{D\times D}$ and $\mathbf{w} \in R^{D\times D}$ are learnable parameters which map appearance features to a subspace and enable learning the correlation of two objects. $\sqrt{D}$ is a normalization factor.

The necessity of defining spatial relation arises from that the interactions of two distant objects are usually scarce. To calculate this relation, we first unproject objects from 2D image plane to the 3D space in the world frame [146]:

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T = \delta_{u,v} \cdot \mathbf{P}^{-1} \begin{bmatrix} u & v & 1 \end{bmatrix}^T \tag{4.3}$$

where $\begin{bmatrix} u & v & 1 \end{bmatrix}^T$ and $\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ are homogeneous representations in 2D and 3D coordinate systems, $\mathbf{P}$ is the camera intrinsic matrix, and $\delta_{u,v}$ is the relative depth at (u, v) obtained by depth estimation [146]. In the 2D plane, we choose the centers of bounding boxes to locate *Thing* objects. The location of the ego-vehicle is fixed at the middle-bottom pixel of the frame. Then the spatial relation function f_s is formulated as:

$$f_s(\mathbf{p}_{\mathrm{i}}^t, \mathbf{p}_{\mathrm{j}}^t) = I(d(\mathbf{p}_{\mathrm{i}}^t, \mathbf{p}_{\mathrm{j}}^t) \leq \mu) \tag{4.4}$$

where $I(\cdot)$ is the indicator function, $d(\mathbf{p}_{\mathrm{i}}^t, \mathbf{p}_{\mathrm{j}}^t)$ computes the Euclidean distance between object i and object j in the 3D space, and μ is the distance threshold which regulates the spatial relation value to be zero if the distance is beyond this upper bound. In our implementation, the value of μ is set to be 3.0.

4.3.2 Ego-Stuff Graph

The *Ego-Stuff Graph* $\mathbf{G}^{ES}$ is constructed in a similar manner as the *Ego-Thing Graph* $\mathbf{G}^{ET}$ in Equation 4.1 except for the following aspects:

Node feature extraction. We include the following classes as *Stuff* objects: *Crosswalk*, *Lane Markings*, *Lane Separator*, *Road*, *Service Lane*, *Traffic Island*, *Traffic Light* and *Traffic Sign*. The criterion we use to distinguish *Stuff* objects from *Thing* objects is based on whether the change of states can be caused by other objects. For example, cars stop and yield to person, but a traffic light turns red to green by itself. Another distinction lies in that the contour of most *Stuff* objects cannot be well depicted as rectangular bounding boxes. Thus, it is difficult either to detect it by algorithms like Faster R-CNN [8], YOLO [9] or to extract features by RoIAlign [12] without enclosing irrelevant information. For this, we propose a feature extraction approach named MaskAlign to extract features for a binary mask $\mathbf{M}_{\mathrm{i}}^t$, which is the i-th *Stuff* object at time t. $\mathbf{M}_{\mathrm{i}}^t$ is downsampled to $\mathbf{M}_{\mathrm{i}}^t$ $(W \times H)$ with the same spatial dimension as the intermediate I3D feature map $\mathbf{X}$ $(T \times W \times H \times D)$. We compute the *Stuff* object feature by MaskAlign as following:

$$\mathbf{x}_{\mathrm{i}}^{t} = \frac{\sum_{w=1}^{W} \sum_{h=1}^{H} \mathbf{X}_{(w,h)}^{t} \cdot \mathbf{M}_{\mathrm{i}\,(w,h)}^{t}}{\sum_{w=1}^{W} \sum_{h=1}^{H} \mathbf{M}_{\mathrm{i}\,(w,h)}^{t}} \tag{4.5}$$

where $\mathbf{X}_{(w,h)}^{t} \in R^{1\times D}$ is the D-dimension feature at pixel (w, h) for time t, and $\mathbf{M}_{\mathrm{i}\,(w,h)}^{t}$ is a binary scalar indicating whether object i exists at pixel (w, h).

Interaction Modeling. In *Ego-Stuff Graph*, we ignore interactions among *Stuff* objects since they are insusceptible to other objects. Hence, we set f_s to zeros for every pair of *Stuff* objects and only pay attention to the influence that *Stuff* objects act on ego-vehicle. We call it as the first-order interaction. To better model the spatial relations, instead of unprojecting bounding box centers, we map every pixel inside the downsampled binary mask $\mathbf{M}_{\mathrm{i}}^{t}$ to 3D space and calculate the Euclidean distance between every pixel with the ego-vehicle. The distance is the minimum distance of the all. The distance threshold in *Ego-Stuff Graph* is designed as 0.8.

4.3.3 Reasoning on Graphs

To perform reasoning on graphs, we introduce graph convolutional networks (GCN) proposed in [13]. GCN takes a graph as input, passes information through the learned edges, and refreshes nodes' features as output. Specifically, graph convolution can be expressed as:

$$\mathbf{Z}^{l+1} = \mathbf{G}\mathbf{Z}^{l}\mathbf{W}^{l} + \mathbf{Z}^{l} \tag{4.6}$$

where $\mathbf{G}$ is the affinity matrix from graphs. Taking the *Ego-Thing Graph* as an example, $\mathbf{Z}^{l} \in R^{(K+1)\times D}$ is the appearance feature matrix of nodes in the l-th layer. $\mathbf{W}^{l} \in R^{D\times D}$ is the learnable weight matrix. We also build a residual connection by adding $\mathbf{Z}^{l}$. In the end of each layer, we adopt Layer Normalization [149] and ReLU before $\mathbf{Z}^{l+1}$ is fed to the next layer. As second-order interaction is not considered in *Ego-Stuff Graph* but in *Ego-Thing Graph*, we use one layer GCN in *Ego-Stuff Graph* and two layers in *Ego-Thing Graph*.

4.3.4 Temporal Modeling

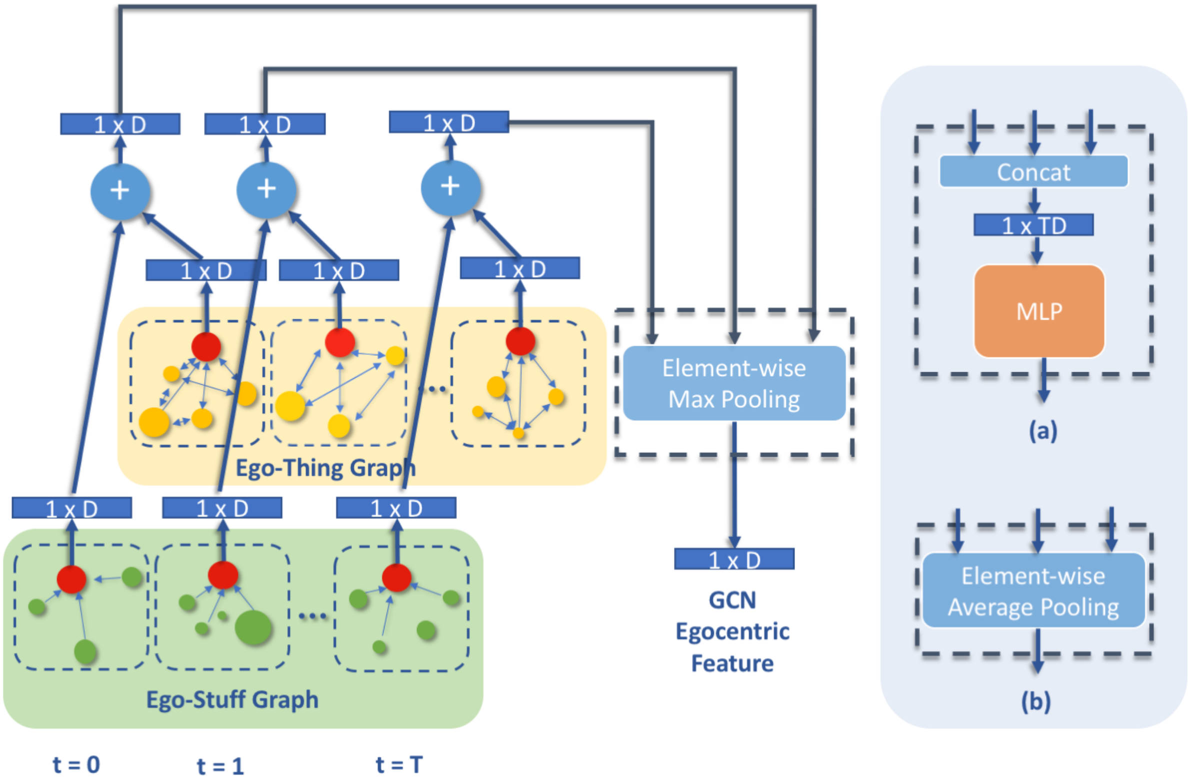


Figure4.3. Architecture of our temporal modeling module.

GCN interactive features in each frame are processed independently without considering temporal context information. Therefore, we append a temporal fusion module to the late stage in our framework as illustrated in Figure 4.3. Unlike prior works [135], [142], [143], which fuse features of every node in different graphs, we only focus on *Ego* node. *Ego* features are aggregated by a element-wise summation from two types of graphs. Then these time-specific *Ego* features are fed into a temporal fusion module, which applies element-wise max pooling to obtain a $1 \times D$ feature vector, namely GCN egocentric feature. We also propose another two designs for temporal fusion: (a) Inspired by Temporal Relation Network [150], which utilizes multi-layer perceptrons (MLP) as temporal modeling, we follow the similar approach in order to capture the temporal ordering of patterns. (b) The temporal fusion can also be replaced by element-wise average pooling. In Sec 4.4.5, we conduct different experiments to investigate all three temporal modeling approaches.

4.4 Experiment

4.4.1 Dataset

We evaluate the proposed framework on the HDD dataset [136], the largest dataset that provides 104-hour egocentric videos with frame-level annotations of tactical driver behavior. It has a diverse set of scenarios where complicated interactions happen between the ego-vehicle and road users. The data was collected within San Francisco Bay Area including urban, suburban and highways. We follow the same Train/Test data split as [136].

The videos are labeled by a 4-layer representation to describe tactical driver behaviors. Among these 4 layers, **Goal-oriented action** layer (e.g., left turn and right lane lane change) and **Cause** layer (e.g., stop for crossing vehicle) consist of the actions with interactions. We leverage those labels and analyze the effectiveness of the proposed interaction modeling framework in Sec 4.4.3.

4.4.2 Implementation Details

We implemented our framework in TensorFlow. All experiments are performed on a server with 4 NVIDIA TITAN-XP. The input to the framework is a 20-frame clip with a resolution of 224 × 224 at 3 fps, approximately 6.67s. We adopt Inception-v3 [151] pre-trained on ImageNet [123] as the backbone, following [148] to inflate 2D convolution into a 3D ConvNet, and fine-tune it on the Kinetics action recognition dataset [152]. The intermediate feature map used in RoIAlign and MaskAlign is extracted from the `Mixed_3c` layer, where $D = 512$ is the number of feature channels. The global I3D feature is generated from a $1 \times 1 \times 1$ convolution on `Mixed_5c` layer feature, which reduces the output channel number from 1024 to 512. The downsampled binary mask $\mathbf{M}_{\mathrm{i}}^{t}$ is 28×28. The model is trained in a two-stage training scheme with batch size set to 32: (1) we fine-tune the Kinetics pre-trained model on the HDD dataset for 50K iterations without using GCN. We refer to this model the baseline for our experiment. (2) We load the weights trained in Stage 1, and further train the network together with GCN for 20K iterations. We use Adam [124] optimizer with default parameters. We set learning rate as 0.001 and 0.0002 for the first and second stage for training, respectively.

4.4.3 Analysis on Interactions

To understand the benefits of modeling interactions, we perform analysis on the following two aspects.

Goal-oriented Action Layer. Table 4.2 presents **Goal-oriented** action recognition results. We use the per-frame mean average precision (mAP) as evaluation metric in all experiments. We pay attention to the 5 'lane-related' classes in frames: *Left Lane Change*, *Right Lane Change*, *Left Lane Branch*, *Right Lane Branch* and *Merge*. Our model obtains 49.9% mAP over these 5 classes, which surpasses the I3D baseline 46.0% mAP by a gain of 3.9%. This improvement showcases the effectiveness of modeling interactions between ego-vehicle and traffic lanes, which also can be validated by visualization in Sec 4.4.6.

Cause Layer. 6 classes from **Cause** layer are designed to explain the reason for *stop* and *deviate* actions, such as *Deviate for Parked Vehicle*, which is an example of *Ego-Thing* interaction. We extend our framework to multi-head classifiers to simultaneously predict **Goal-oriented** actions and **Causes**. Note that we train a multi-head I3D as the baseline for this experiment. Our design achieves a steady increase in recognizing **Goal-oriented** actions by improving the baseline of 48.5% to 50.2%. Meanwhile, the result of **Cause** layer in Table 4.1 shows a significant gain of 6.0% in overall mAP. We further demonstrate the strength of the proposed interaction modeling by using a *Deviate for Parked Vehicle* scenario in Figure 4.5 in Sec 4.4.6.

4.4.4 Comparison with the State of the Art

We compare our approach with the state-of-the-art in Table 4.2. We categorize the existing methods tested on HDD into *online* and *offline*. The online approaches aim to detect driver actions as soon as a frame arrives. Future context is not considered. The offline approaches take future frames into consideration. Since future information is processed, the

Table4.1. Results of driver behavior recognition in **Cause** layer on HDD. The unit is %.

	Individual actions						
Method	Stop for Congestion	Stop for Sign	Stop for Red Light	Stop for Crossing Vehicle	Deviate for Parked Vehicle	Stop for Crossing Pedestrian	Overall mAP
I3D [148]	64.8	71.7	63.6	21.5	15.8	26.2	43.9
Ours	**74.1**	**72.4**	**76.3**	**26.9**	**20.4**	**29.0**	**49.9**

Table4.2. Results of **Goal-oriented** driver behavior recognition on HDD. The unit is %.

		Individual actions											
Method	Online/Offline	intersection passing	L turn	R turn	L lane change	R lane change	L lane branch	R lane branch	crosswalk passing	railroad passing	merge	u-turn	Overall mAP
CNN [136]	Online	53.4	47.3	39.4	23.8	17.9	25.2	2.9	4.8	1.6	4.3	7.2	20.7
CNN-LSTM [136]		65.7	57.7	54.4	27.8	26.1	25.7	1.7	16.0	2.5	4.8	13.6	26.9
ED [139]		63.1	54.2	55.1	28.3	35.9	27.6	8.5	7.1	0.3	4.2	14.6	27.2
TRN [139]		63.5	57.0	57.3	28.4	37.8	31.8	10.5	11.0	0.5	3.5	25.4	29.7
DEPSEG-LSTM [153]		70.9	63.4	63.6	48.0	40.9	39.7	4.4	16.1	0.5	6.3	16.7	33.7
C3D [154]		72.8	64.8	71.7	53.4	44.7	52.2	3.1	14.6	2.9	10.6	15.8	37.0
C3D [154]	Offline	82.4	77.4	**80.7**	67.9	56.9	59.7	5.2	17.4	**3.9**	20.1	29.5	45.5
I3D [148]		**85.6**	**79.1**	78.9	74.0	**62.4**	59.0	14.3	**29.8**	0.1	20.1	**41.4**	49.5
Ours		85.5	77.9	79.1	**76.0**	62.0	**64.0**	**19.8**	29.6	1.0	**27.7**	39.0	**51.1**

offline approaches exhibit an overwhelming advantage over the online approaches. Among the offline methods, our model significantly outperforms the C3D [154] and I3D [148] by 5.6% and 1.6% in terms of mAP, respectively.

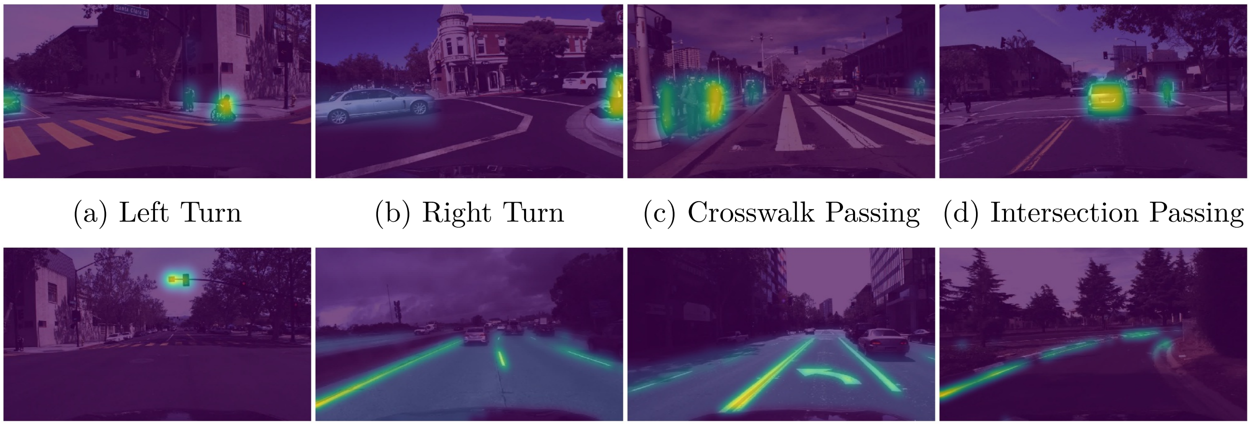
4.4.5 Ablation Studies

To provide a comprehensive understanding of the contributions from each module, we decompose our model into three components and conduct ablation studies using the **Goal-oriented** action recognition shown in Table 4.3.

Comparison of Different Graphs. The first section of Table 4.3 analyzes the influence of each graph to the tactical driver behavior recognition. The baseline is the I3D. When *Ego-Stuff Graph* or *Ego-Thing Graph* is included, the results are boosted from 49.5% to 50.6% and 50.7%, respectively. If both graphs are trained jointly with the baseline model, we achieve the best performance 51.1% on the **Goal-oriented** action recognition.

Table4.3. Ablation Studies

	Method	Overall mAP
Different Graphs	I3D [148]	49.5
	Ego-Stuff Graph	50.6
	Ego-Thing Graph	50.8
	Ego-Thing Graph + Ego-Stuff Graph	**51.1**
Spatial Modeling	Appearance Relation	50.9
	Appearance + Spatial Relation	**51.1**
Temporal Modeling	Average	50.0
	MLP	50.9
	Max	**51.1**


(a) Left Turn (b) Right Turn (c) Crosswalk Passing (d) Intersection Passing

(e) Left Turn (f) Left Lane Change (g) Left Lane Branch (h) Merge

Figure4.4. Attention visualization from egocentric view. The first and second row show examples from *Ego-Thing Graph* and *Ego-Stuff Graph*, respectively. In (a)-(c), pedestrians intending to cross the street have significant influence on *Ego* behavior when turning left, turning right and passing the crosswalk. The ego-vehicle passes an intersection in (d) while paying attention to the moving car and bicycle in front of it. The figure (e) illustrates a left turn case when the heat map shows a high attention around the traffic light, which is green. In (f)-(h), lane markings show strong influences to ego's lane-related behaviors.

Importance of Spatial Relation. To investigate the effectiveness of spatial relation function in Equation 4.4, we conduct two experimental settings: using only the appearance relations, and embedding 3D spatial relation as an additional constraint. Without using the proposed 3D spatial relation, the performance decreases by 0.2%, indicating the advantage of encoding spatial context.

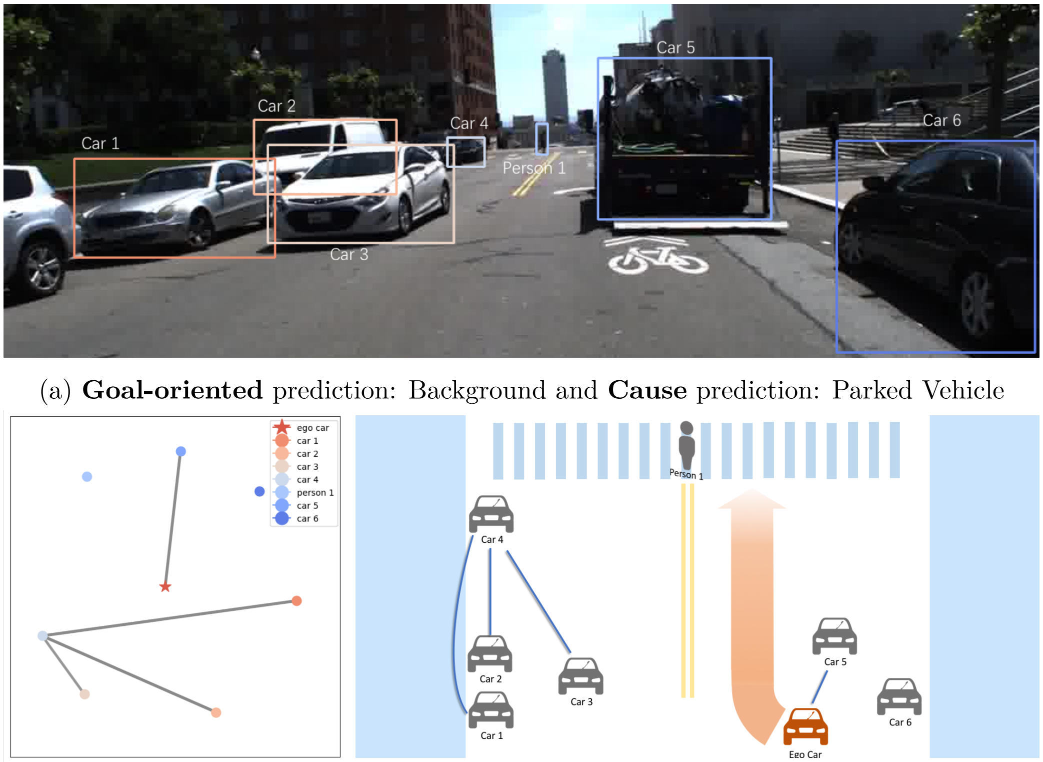
Variations of Temporal Modeling. We analyze the impact of temporal modeling approaches. The best mAP – 51.1% is obtained by element-wise max pooling. If we use element-wise averaging for the features from each time step, the model has a mAP of 50.0%. Our conjecture is that, for a 20-frame video clip, the key change takes place within a short duration. For example, in a *Left Lane Change* behavior, the most noticeable moment is when the ego-vehicle intersects the traffic lanes within a few frames. Temporal modeling using averaging features potentially degrades the distinguishable features, which will unavoidably result in information loss. A multi-layer perceptron (MLP), which takes temporal ordering patterns into account, exceeds averaging pooling by 0.9% but is 0.2% lower than the best performance. Our hypothesis is that significant change of interactive relations plays an more important role in recognizing tactical driver behavior than the ordering in time.

4.4.6 Visualization

Apart from quantitative evaluation, we demonstrate interpretability of our method by the following two visualization strategies.

Attention Visualization from Egocentric View. Given the learned affinity matrices in *Ego-Thing Graph* and *Ego-Stuff Graph*, we highlight those objects with strong connection to the *Ego* node in Figure 4.4. The visualization results provide a strong proof that the proposed model captures the underlying interactions, which is essential for tactical driving behavior understanding. Note that in the example shown in Figure 4.4 (e), the model captures the relation between the ego-vehicle (turning left) and the traffic light (green light).

Attention Visualization from BEV. In addition to the interactions with ego, we can represent the complicated traffic scene in a graph structure as well. Figure 4.5(b) shows the visualized *Ego-Thing Graph* from the multi-head model for a scenario where the ego-vehicle deviates for a parked truck. Each circle in the graph corresponds to a *Thing* object in the frame and the ego-vehicle is represented by a star. The edge linking two nodes represents



(a) **Goal-oriented** prediction: Background and **Cause** prediction: Parked Vehicle

(b) Learned affinity matrix

(c) BEV visualization of the learned affinity matrix.

Figure4.5. Attention visualization from top-view.

the interactive relation among them. We manually draw a BEV map Figure 4.5(c) to better represent the interactions based on spatial context.

4.5 Discussion

In this chapter, we propose a framework to realize Level 2 SA for ADAS which comprehends the spatial-temporal interactions between driver and road users. The proposed framework utilizes graph convolution networks to combine, interpret, store and retain information from the complicated driving scenes. It demonstrates favorable quantitative performance on the HDD dataset. Qualitatively, we show the model can captures interactions between the ego-vehicle and *Stuff* objects, and the ego-vehicle and *Thing* objects. The ability of comprehension provides the premise of embedding the projection to ADAS in order to forecast future states and avoid hazards.

5. PROJECTION AND RISK OBJECT IDENTIFICATION

5.1 Introduction

Projection, as the highest level of SA, is the ability to forecast the future states and dynamics of traffic elements in the environment. It includes anticipating where the driver and other vehicles intend to go, predicting if pedestrians prepare to cross the road, inferring the probability of crushing if obstacles appear and etc.

Among these future events, one of the most important information is the potential risk (i.e., collision) in the near future. If an intelligent driving system can assist drivers to identify the upcoming potential risks, it will significantly reduce traffic accident fatalities caused by driver errors. This task of identifying risks has been studied extensively in the risk assessment literature [155]. In the context of intelligent vehicles, the risk is generally defined based on collision prediction. While this definition is widely applied, road collision is only one source of potential hazards in driving [155]. We believe a more generic definition is needed.

We propose a novel driver-centric definition of risk, i.e., *risky objects influence driver behavior*. Figure 5.1 depicts the idea of the proposed definition. While driving toward an intersection, we react to the crossing pedestrian (i.e., slow down). After passing the intersection, we react to the construction cone (i.e., deviate to a clear path). From these examples, we observe that we constantly attend to those traffic participants potentially influencing driver behavior, because we humans are equipped with risk perception. In other words, a dangerous situation would occur if we do not react to them immediately. The proposed definition captures the observation. We believe the definition gives a new perspective to the definition of risk assessment.

A natural question arises: *Who changes drivers' behavior?* We propose a new task called *risk object identification*, which aims to identify the object(s) influencing drivers' behavior. The proposed task can be approached via three existing tasks: (1) salient object identification learned from human gaze behavior [156]; (2) object importance estimation or risky region localization learned from human annotations [157], [158]; and (3) salient regions/objects identification learned from end-to-end driving models with self-attention mechanisms [27], [29].

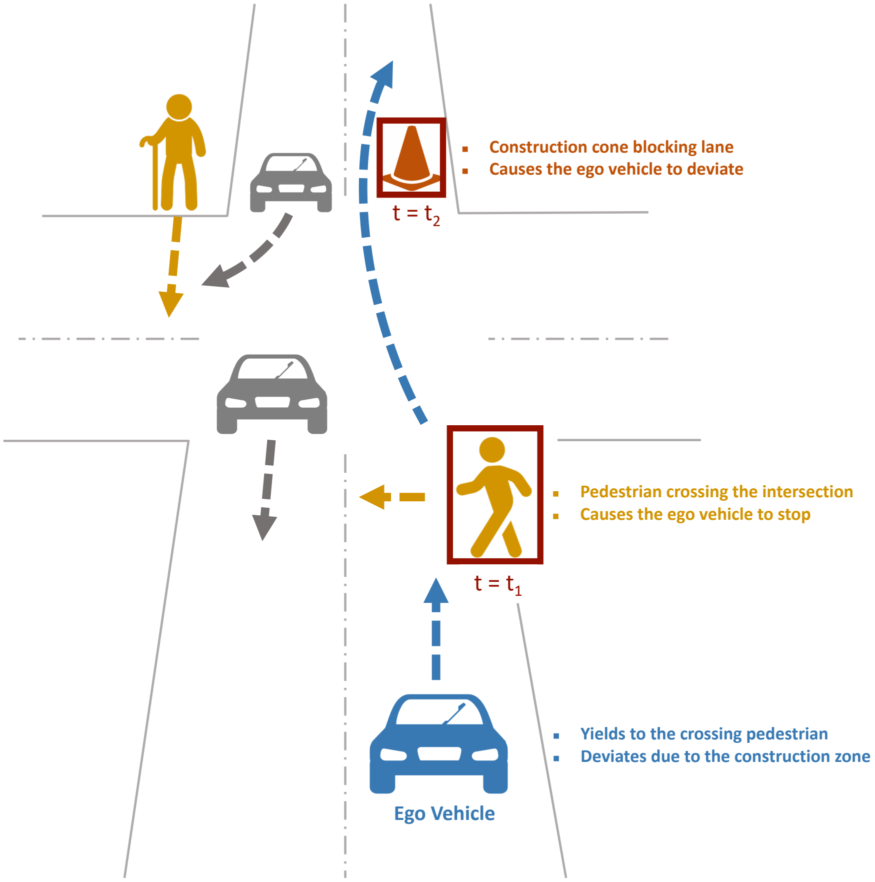

Construction cone blocking lane
Causes the ego vehicle to deviate
t = t2
Pedestrian crossing the intersection
Causes the ego vehicle to stop
t = t1
Yields to the crossing pedestrian
Deviates due to the construction zone
Ego Vehicle

Figure5.1. Human drivers perceive scenes, assess risks, make a plan, and take actions while driving. Risk assessment, identifying hazards and risk factors that have the potential to cause harm, is indispensable for driving safety. In this chapter, we cast the identification of potential hazards as a cause-effect problem. A new task called risk object identification is introduced. We propose a novel computational framework that learns to reason how humans react (effect) to these objects (cause).

First, learning to predict pixel-level driver attention by imitating human gaze behavior has been explored by [156], [159], [160]. This area of research is motivated by psychological studies suggesting that there is a connection between driving, attention, and gaze [161]. Alletto et al., [156] collect a large-scale dataset including drivers' gaze fixations acquired during actual driving. While the direction of this study is promising, human gaze behavior is intrinsically noisy, and fixations may not directly associate with objects influencing drivers' behavior. Second, risky region localization [158] or object importance estimation [157] formulate the task as a two-class object detection problem. Human annotators are asked to

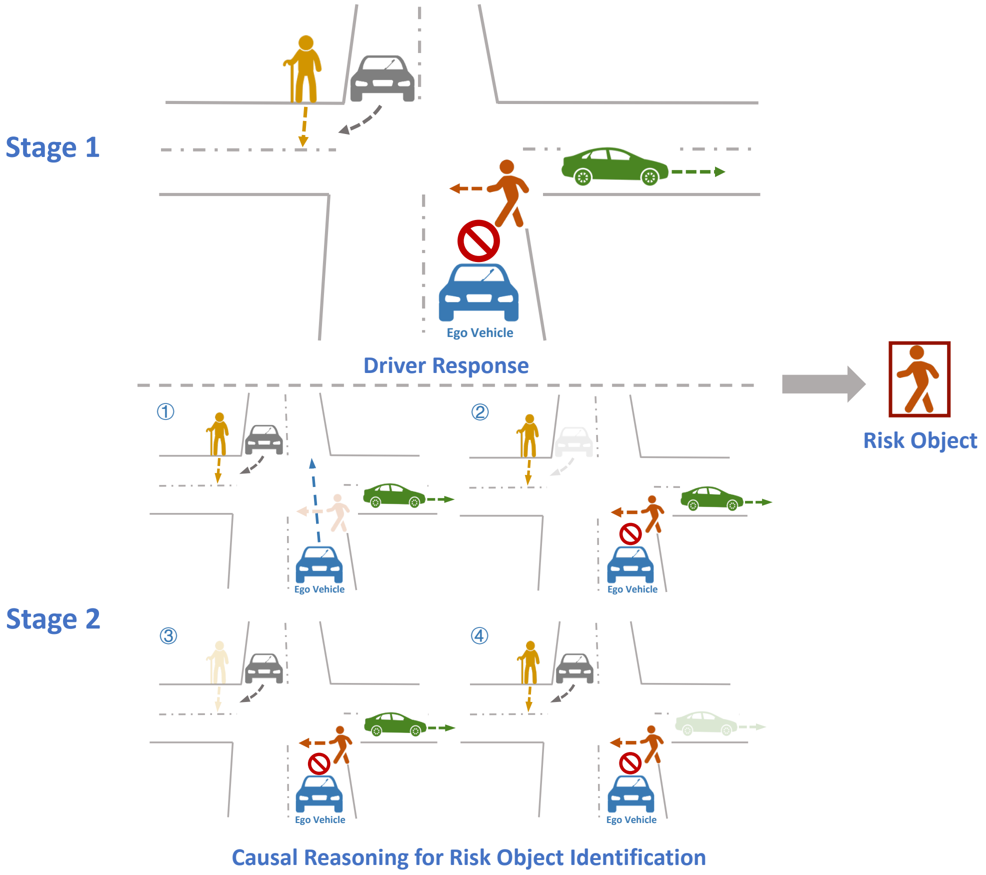


Figure5.2. An conceptual diagram of the proposed two-stage risk object identification framework. We first predict driver response in a given situation. To identify object(s) influencing driver behavior, we intervene the input observation by removing a traffic participant at a time (i.e., simulating a situation with the traffic participant), and predict the corresponding driver response. For instance, removing the crossing pedestrian changes driver response (effect) from *Stop* to *Go*. The effects of removing other traffic participants remain the same. We conclude that the crossing pedestrian is the risk object (cause).

label risky regions or important objects. While favorable results are obtained, the supervised learning-based formulation requires a significant amount of human-labeled annotations, and the performance in unseen situations cannot be guaranteed. Third, the task can be formulated as selecting regions/objects with high activations in visual attention heat maps learned from end-to-end driving models [27], [29]. Specifically, pixel- and object-level attention maps are obtained via optimizing task-driven objective functions and self-attention mechanisms. However, highly activated objects/regions do not necessarily associate with models' decisions. The issue outlines the confusion between causation and correlation. In [162], the authors also identify "causal misidentification" as an under-explored problem in training end-to-end driving models.

To address the aforementioned issues, we propose a novel two-stage risk object identification framework based on the proposed definition of risk. Specifically, we formulate the risk object identification as a cause-effect problem [163]. The core concept is depicted in Figure 5.2.

In the first stage, we integrate the concept of projection and develop a Level 3 SA [5] driving model to anticipate driver response in a given situation. We simplify the response of drivers to be *Stop* and *Go*. The proposed model encapsulates the goal (i.e., driver intention), perception (i.e., elements of the environment), comprehension (i.e., interactions between driver and *Thing* objects and interactions between driver and *Stuff* objects in 3D), and projection (i.e., intention-aware interaction forecasting) for driver response prediction. *Thing* and *Stuff* objects are defined in Sec 5.4.1.

In the second stage, given a *Stop* response (i.e., driver behavior is influenced by certain objects), we intervene input video by removing a tracklet at a time and inpainting the removed area in each frame to simulate a scenario without the presence of the tracklet. The same driving model is applied to predict the effect of the removal. The process iterates through all tracklets and records the corresponding effects. Note that we assume that the cause of driver response change is either vehicles or pedestrians. The tracklet that causes a maximum response change is the risk object. Our preliminary exploration of risk object identification [15] also follows the cause-effect concept but uses a simpler driving model. The details are included in Sec 5.7.

Our work differs from existing methods [27], [29], [156]–[158] in the following three aspects:

1. A novel driver-centric notion of risk, whereby risky objects are defined as those that influence driver behavior, is proposed;

2. An unsupervised framework is introduced as an initial step toward generalization;

3. A causal inference-based framework is proposed to address the issue of "causal misidentification" in end-to-end driving models for risk object identification.

In this chapter, substantial extensions are made to our early results (driving model previously mentioned in Chapter 4. Specifically, we pay more attention to the projection (Level 3 SA) by adding two ideas to the existing driving model [16]: (1) an encoder-decoder architecture based on Temporal Recurrent Network (TRN) that uses both historical evidence and predicted future information to better predict current action, and (2) an intention-aware design as cues to aid better prediction of the driver response. Also, we systematically benchmark three different tasks, i.e., driver response prediction, driver intention prediction, and risk object identification on the proposed driver-centric ROI dataset. We conduct thorough ablative studies to justify the architectural designs.

5.2 Background

5.2.1 Risk Assessment

Living agents can assess risk for decision-making. Earlier attempts have been made to study this problem from different angles, and can be categorized into four categories. First, the works [164], [165] design a set of rules based on the current state of vehicles and contextual states for detecting dangerous situations. These rule-based approaches ignore uncertainties of dynamic driving environments, leading to instabilities in their decisions. Second, risky situations can be determined by the similarity of a pattern between a pair of traffic participants with accident patterns obtained from accident databases [166], [167]. However, real-world accident data are hard to obtain. It is also challenging to realistically simulate accident with a simulator. Moreover, it is insufficient to consider pairwise relations in complicated driving scenarios, where multiple traffic participants interact with each other.

Third, a popular risk assessment methodology is to predict all possible colliding future trajectories [168]–[170]. Please refer to [155] for a detailed survey of motion prediction and risk assessment in the context of intelligent vehicles. While predicting all possible colliding future trajectories is well-received by this research field, the approach involves a large number of computations since it requires pairwise comparisons. Fourth, Lefèvre et al. [171] define the risk of a situation by detecting conflicts between driver intention and expectation via a probabilistic framework. While this paradigm is very close to our proposed definition of risk, the underlying mechanism for risk object identification is different. Specifically, in [171], a risk object is identified by computing the probability of intention-expectation mismatch for each vehicle based on vehicle states. If the probability exceeds a threshold, the corresponding vehicle is considered to be a “hazard”. In contrast, we discover the risk object based on causal inference, reasoning the effect of an object removal (i.e., intervention).

5.2.2 Vision-based End-to-end Driving Models

The history of vision-based end-to-end driving models can be traced back to 1989 when ALVINN [22], the framework that learns a mapping from images to navigation signals via a shallow neural network, is introduced. Recently, Bojarski et al. [23] demonstrate a similar idea by extending it to modern convolutional neural networks for extracting better visual representations from images. In [24], visual representations are learned with an auxiliary semantic segmentation task to better represent driving scenes. While significant progress has been demonstrated, neural network-based frameworks lack interpretability, crucial for safety-critical applications. To address the issue, Kim et al. [27], [28] and Wang et al., [29] propose pixel-and object-level attention mechanisms, respectively. Particularly, Wang et al., [29] propose an object-level attention scoring mechanism as a means to model how certain traffic participants impact actions of driving models.

Interactions modeling between traffic participants is commonly studied in trajectory prediction literature [130], [172]–[174]. However, interaction modeling for learning driving policies is under-explored. To address this problem, our method explicitly models the interactions using Graph Convolutional Networks (GCNs). Instead of simply weighting and concatenating objects’ visual representations as interaction modeling [29], we model interaction as message passing that incorporates relative distances between traffic participants and ego-vehicle. Moreover, interactions between the ego-vehicle and road infrastructure (e.g., traffic light) are considered in the proposed framework. We show that the two interaction modelings are essential for driver response prediction. Additionally, the proposed driving model exploits the inductive biases motivated by situation awareness [5]. We empirically demonstrate the effectiveness of these inductive biases for both driver response prediction and risk object identification.

While the aforementioned driving models have shown remarkable advances in following roads and avoiding obstacles, they cannot be guaranteed to achieve a goal (e.g., left turn). Codevilla et al. [25], [175] incorporate navigational commands as an extra input for learning driving policies. Instead of inputting a navigational command, the proposed driving model infers drivers' intention from egocentric videos for driver response prediction.

5.2.3 Causality in Computer Vision

Computer vision research has proliferated over the past decades due to the advance of deep learning algorithms. However, current deep learning models suffer from spurious correlation problems [176] because of ignoring causality in data. Humans perceive causality of the physical world. To address the issue, recent studies[177]–[181] explicitly consider the concept of causality into deep learning architectural designs.

Particularly, the authors of [179], [180] propose a novel training objective as a practical approximation for imaginative intervention (i.e., *do* operator proposed in [163]) to eliminate noncausal relations and unobserved confounders for image captioning and visual Q&A. In this chapter, we also leverage causal intervention but in a different way. Specifically, instead of using an imaginative causal intervention, we explicitly conduct *do* operator via image inpainting.

To our best knowledge, we are among the first to utilize causal inference for driving scene applications. Kim et al. [27] propose a causality test to verify the effectiveness of inferred attention maps obtained from the proposed driving model. We also employ causal inference similar to the causality test. However, the purpose of causal inference in this chapter is to identify risk objects. Moreover, we design a simple but effective data augmentation strategy using causal intervention. This leads to a more robust driving model.

Haan et al. [162] propose to incorporate functional causal models [163] into imitation learning to address the issue of "causal misidentification". In [182], they overcome the causal

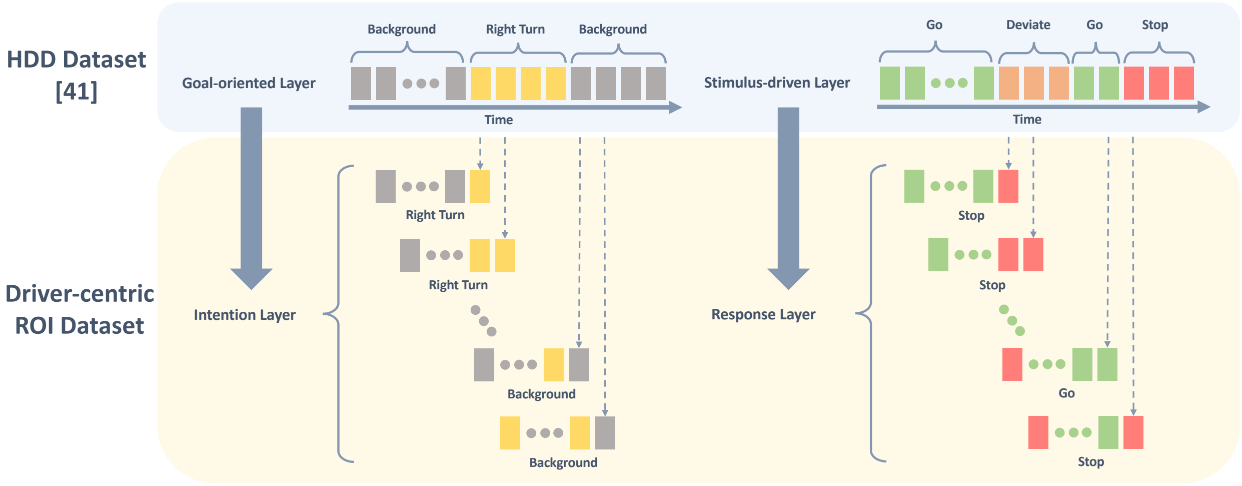


Figure5.3. Driver-centric Risk Object Identification (ROI) dataset. To study risk object identification, a dataset with a diverse of reactive scenarios is essential. We build the driver-centric ROI dataset on top of the Honda Research Institute Driving Dataset (HDD). In particular, we introduce two layers, i.e., **Driver Intention** and **Driver Response** in the proposed dataset. Further detail of the two layers can be found in Sec 5.3. To obtain **Intention** labels, we form n-frame clips, and the corresponding **Intention** label of each clip is the last frame's label defined in the **Goal-oriented** layer. A similar procedure is applied to construct the **Response** layer (as shown on the right-hand side of the figure). Notice that both *Stop* and *Deviate* annotated in HDD are merged into *Stop* in our dataset.

misidentification issue by adding noises to inputs. Our work is complementary to [162], [182]. Specifically, the focus of [162], [182] is to improve the robustness of driving models, whereas the proposed framework leverages driving models to determine the response of drivers in a counterfactual situation for risk object identification. We believe the two lines of work should be studied jointly and will leave for future work.

5.3 Dataset

To study driver-centric risk object identification, a dataset with diverse reactive scenarios (i.e., drivers react to potential hazards while navigating to their goals) is indispensable. For instance, when human drivers intend to turn left at an unprotected intersection, they react (e.g., slowing down or stopping) to certain traffic participants to avoid dangerous situations.

We curate a driver-centric Risk Object Identification (ROI) dataset from the Honda Research Institute Driving Dataset (HDD) [136].

5.3.1 Dataset Annotation

The driver-centric ROI dataset utilizes a two-layer representation — **Intention** and **Response**. Figure 5.3 illustrates how we construct the proposed driver-centric ROI dataset from the HDD dataset.

The **Goal-oriented** layer defined in the HDD dataset denotes tactical driver behavior such as *right turn*, *left turn*, or *lane change*. As shown in Figure 5.3, each frame is labeled with either a goal-oriented or background action. To obtain the **Intention** of a n-frame clip (the parameter n is 20 in our implementation), we use the last frame's label of the **Goal-oriented** layer as the **Intention** label. While performing a tactical behavior, drivers might have to *stop* or *deviate* due to traffic participants or obstacles. We extend the **Stimulus-driven** actions, i.e., *Stop* and *Deviate*, defined in the HDD dataset as the **Response** label. Note that both *Stop* and *Deviate* are merged into *Stop* as depicted in Figure 5.3. The rest of the frames are labeled as *Go*. The HDD dataset also annotates a **Cause** layer to explain the reason for *Stop* and *Deviate* actions. We create our **Test2** set by selecting frames from the four **Cause** scenarios, i.e., *Congestion*, *Crossing Pedestrian*, *Crossing Vehicle* and *Parked Vehicle*. Moreover, in the **Test2** set, we provide bounding boxes of risk objects (i.e., object[s] influencing driver's behavior) for risk object identification benchmarks. We focus on scenarios in which drivers react to vehicles or pedestrians.

5.3.2 Dataset Statistics

The driver-centric ROI dataset has 184 890 frames for training driver response and intention predictors. Two test sets are constructed for driver response prediction and risk object identification, respectively. The **Test1** split has 63 314 frames for both driver response and intention benchmarks. The **Test2** has 630 frames (i.e., 630 different risk objects) covering four different reactive scenarios, i.e., *Congestion*, *Crossing Pedestrian*, *Crossing Vehicle*, and

Table5.1. Statistics (annotated frames) of the proposed driver-centric ROI dataset.

		Intention												Response	
Split		BG	IP	LT	RT	LLC	RLC	LLB	RLB	CP	RP	MG	UT	STP	G
Train		737 949	48 933	21 819	19 824	4 815	4 386	1 833	717	2 364	588	1 182	2 001	184 890	661 521
Test1		236 622	17 772	7 017	6 195	1 098	1 212	435	324	432	123	327	432	63 314	208 675
	Cause														
Test2	Congestion	98	/	/	/	1	/	/	/	/	/	/	/	99	/
	Crossing Pedestrian	62	15	5	/	/	/	/	/	2	/	/	/	84	/
	Crossing Vehicle	263	2	7	35	/	/	/	4	/	/	/	/	311	/
	Parked Vehicle	120	3	/	/	9	4	/	/	/	/	/	/	136	/
	All	543	20	12	35	10	4	/	4	2	/	/	/	630	/

Intention: (BG) background, (IP) intersection passing, (LT) left turn, (RT) right turn,(LLC) left lane change, (RLC) right lane change, (LLB) left lane branch, (RLB) right lane branch,(CP) crosswalk passing, (RP) railroad passing, (MG) merge, (UT) u-turn.
Response: (STP) stop, (G) go.

Parked Vehicle for risk object identification benchmarks. Detailed statistics are shown in Table 5.1.

5.4 Level 3 SA Driving Model

Given a reactive scenario with T RGB images $I \coloneqq \{I_1, I_2, \cdots, I_T\}$, the goal is to identify the object influencing driver response in the last frame. The task is called risk object identification. We formulate the task as a cause-effect problem [163]. Specifically, a two-stage framework is proposed to identify the cause (i.e., the object) of an effect (i.e., driver response) via the proposed *Intention-aware Driving Model* and *Causal Reasoning for Risk Object Identification*.

An overview of the proposed intention-aware driving model is depicted in Figure 5.4. To predict the response of a driver, a driving model should capture complicated spatio-temporal interactions between a driver and traffic participants. We propose a novel driving model motivated by the model of situation awareness (SA) [5]. Specifically, the proposed model encapsulates the four essential components defined in SA for driver response prediction: goal/objective (i.e., driver intention), perception (i.e, elements of a traffic environment), comprehension (i.e., interactions between driver and *Thing* objects, and interactions between driver and *Stuff* objects in 3D), and projection (i.e., intention-aware interaction forecasting). The detail of each component is discussed in the following.

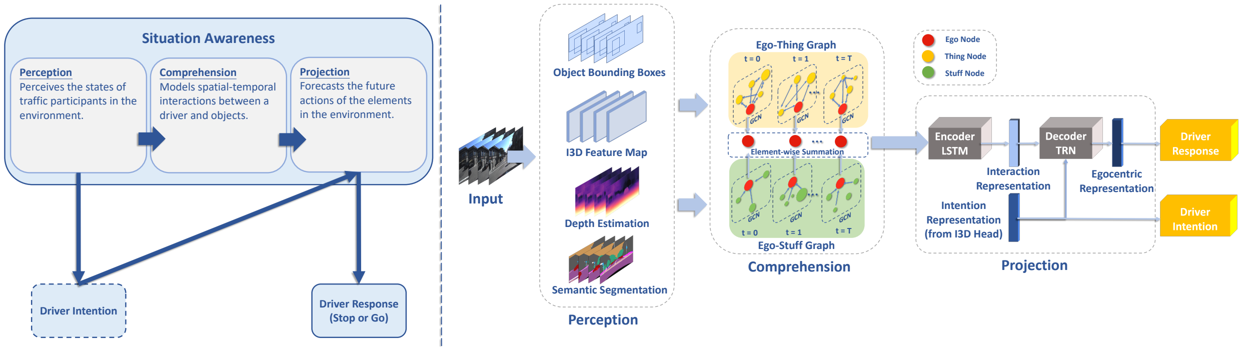


Figure5.4. An overview of the proposed intention-aware driving model for driver response prediction (right). The proposed architecture is motivated by the model of situation awareness [5] (left). Given a video clip, 3D convolutions (I3D), object detection, semantic segmentation, and depth estimation are applied to obtain states of traffic participants in a traffic environment at the *Perception* stage (Sec 5.4.1). At the *Comprehension* stage, an *Ego-Thing Graph* and an *Ego-Stuff Graph* are constructed to model spatial-temporal interactions between a driver and traffic participants (Sec 5.4.2). In this chapter, we categorize traffic participants into two types, i.e., *Thing* and *Stuff*. The details are discussed in Sec 5.4.2. The final stage, *Projection* (Sec 5.4.3), forecast future interactions between driver and traffic participants for driver response prediction. Frame-wise interactions obtained from *Ego-Thing Graph* and *Ego-Stuff Graph* are fused and fed into an encoder LSTM to form interaction representation. Intention representation obtained from the I3D head and interaction representation are sent a decoder TRN (the architecture is shown in Figure 5.5) to predict driver response.

5.4.1 Perception

Perception plays an essential role in the SA model [5]. This component perceives the status, attributes, and dynamics of relevant traffic participants of a traffic environment. Similar to Chapter 4, given T RGB images, we apply object detection and semantic segmentation [12] to obtain *Thing* and *Stuff* objects, respectively. We use the same criterion to distinguish *Stuff* objects from *Thing* objects: whether states of an object can be influenced by other objects. If yes, we categorize the object as a *Thing* object. For instance, cars stop or yield to pedestrians, but a traffic light turns red or green by itself. In addition to detection and segmentation, we perform object tracking using Deep SORT [183] and depth estimation [146].

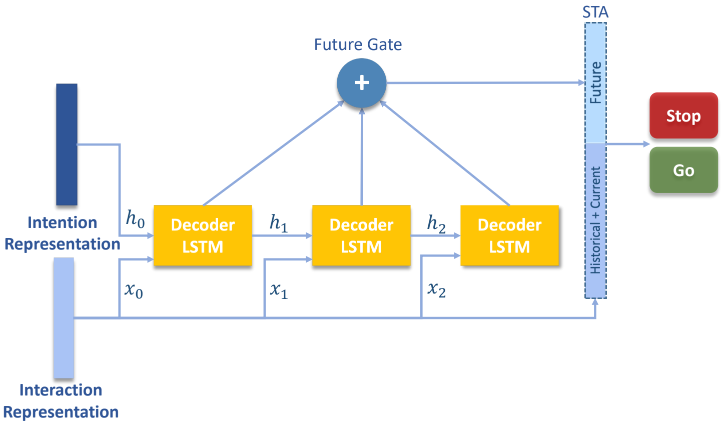


Figure5.5. Decoder Temporal Recurrent Network (TRN) [139]. The inputs to this module are intention and interaction representations. Note that intention representation is used to initialize the hidden state of the first decoder LSTM cell. The future gate and spatiotemporal accumulator (STA) aggregate features from historical, current, and predicted future information to predict driver response.

5.4.2 Comprehension

We interpret *Comprehension* as the spatio-temporal interactions between the driver and *Thing* objects, and interactions between the driver and *Stuff* objects in the 3D world. We follow the same procedures in Chapter 4 to realize the modeling of *Comprehension.* Specifically, we construct two graphs, i.e., *Ego-Thing Graphs* and *Ego-Stuff Graphs.* They are modeled with Graph Convolutional Networks (GCNs) [13]. The details can be found in Chapter 4

5.4.3 Projection

The role of *Projection* is to forecast future actions of elements in the environment. The updated appearance feature $\mathbf{Z}^{l+1}$, discussed in Chapter 4, is processed independently at every frame without considering temporal changes. An encoder-decoder architecture is proposed to capture temporal interactions for forecasting future interactions.

Encoder-decoder Architecture. We implement the proposed encoder-decoder architecture based on the Temporal Recurrent Network (TRN) [139], which makes use of both accumulated historical evidence and predicted future information to better predict current action. Following [139], we use long short-term memory (LSTM) [184] as the backbone for both encoder and decoder.

We aggregate updated *Ego* features from *Ego-Stuff Graphs* and *Ego-Thing Graphs* by an element-wise summation. Time-specific updated *Ego* features are fed into the encoder LSTM to obtain a $1 \times D$ feature vector called interaction representation. Note that prior works [135], [142], [143] fuse all nodes' features in a graph, and the fused features are sent to the encoder LSTM. In contrast, we only send updated *Ego* features in $\mathbf{Z}^{l+1}$ to the encoder-decoder architecture, because updated *Ego* features are expected to capture interactions among traffic participants that are key to robust driver response prediction. Unlike typical decoder architectures implemented as other LSTMs, TRN includes an LSTM decoder, a future gate, and a spatiotemporal accumulator (STA). We extend TRN for the predicting driver response, and the corresponding architecture is depicted in Figure 5.5. The LSTM decoder learns a feature representation of the evolving interactions. The future gate receives a vector of hidden states from the decoder LSTM and embeds features via the element-wise summation as the future context. The STA concatenates historical, current, and predicted future spatiotemporal features, and estimates driver response occurring in the very next frame.

Intention-aware Design. Driver intention is indispensable for planning the next action [25], estimating the importance of road users[185], and assessing risk [186]. Similarly, in our task, driver response (i.e., *Go* and *Stop*) is determined not only by interactions among traffic participants but also driver intention (e.g., *Left Turn* or *Right Turn*). For instance, a vehicle turning right at an intersection will not stop for pedestrians walking on the left sidewalk. Hence, we treat features extracted from the I3D head as the intention representation. The representation is used to initialize the hidden state of the first decoder LSTM cell. Note that the design differs from [139], which initializes the hidden state h_0 with zeros. To acquire a good intention representation, the representation is trained to predict driver intention in a supervised learning manner.

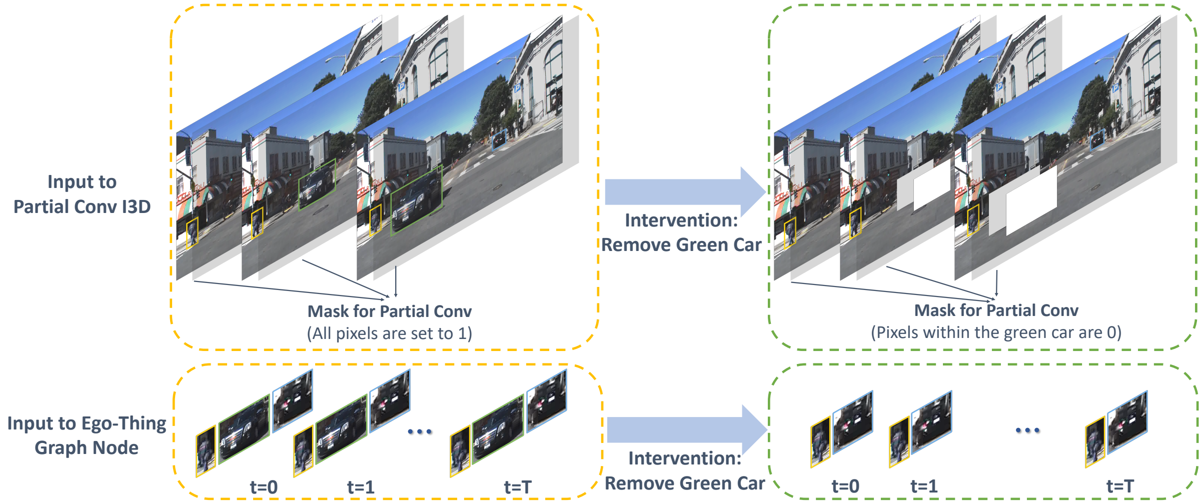


Figure5.6. We simulate a situation using partial convolutional layers [187]. Note that a partial convolutional layer is initially introduced for image inpainting. We utilize partial convolutions to simulate a scenario without the presence of an object. The left-hand side of the figure depicts when an intervention is disabled. To simulate a situation without an object (e.g., the car in the green box), we set the pixels of the binary mask within the car's box to 0. In addition, the *Ego-Thing Graph* is constructed without considering the car in the green box as a node.

5.5 Causal Reasoning

The previous section introduces the proposed intention-aware driving model. In this section, we discuss how we utilize *intervention*, a powerful tool for causal inference, as a means for data augmentation to improve the performance of the driving model (Sec 5.5.1) and apply causal inference to identify the risk object (Sec 5.5.2).

5.5.1 Driving Model Training with Data Augmentation via Intervention

The performance of driving models depends on the amount of training data under different traffic configurations [24]. Due to limited real-world human driver demonstrations, we propose a novel data augmentation strategy via *intervention* [163]. Specifically, we generate a new data point based on a simple yet effective notion, i.e., removing non-causal objects does not influence driver behavior. For instance, in a *Go* scenario, a driver enters an intersection while pedestrians walk on the sidewalk in an opposite direction. It is reasonable to

Algorithm 1 : Driving Model Training

```
$T$: Number of frames
$N$: Number of *Thing* objects in a given tracklet list
$\mathbf{A}_r$: Ground truth driver response (either *Go* or *Stop*)
**Input**: A sequence of RGB frames $I \coloneqq \{I_1, I_2, \cdots, I_T\}$
**Output**: Predicted driver response $\mathbf{a}_r$ and intention $\mathbf{a}_\mathrm{i}$. Notice that $\mathbf{a}_r$ consists of confidence scores of *Go* or *Stop*. $\mathbf{a}_r \coloneqq \{r^{go}, r^{stop}\}$.

$O \coloneqq$ DetectionAndTracking($I$)
    $\coloneqq \{O_1, O_2, \cdots, O_N\}$ // List of *Thing* object tracklets
$S \coloneqq$ SemanticSegmentation($I$)
    $\coloneqq \{S_1, S_2, \cdots, S_T\}$ // List of *Stuff* objects
// **Data Augmentation via Intervention (Sec 5.5.1)**
**if** $\mathbf{A}_r$ is *Go* and $N > 1$ **then**
    // Randomly remove a tracklet
    $k \coloneqq$ RandomSelect($N$)
**else**
    $k$ is empty
**end if**
// Mask out *Thing* object $k$ on each mask frame
$M \coloneqq$ MaskGenerator($I, O_k$)
// Remove a *Thing* object $k$ from the tracklet list
$O = O - \{O_k\}$
$\mathbf{a}_r, \mathbf{a}_\mathrm{i} \coloneqq$ DrivingModelTraining($I, M, O, S$) //Discussed in Sec 5.5.1
**return** $\mathbf{a}_r, \mathbf{a}_\mathrm{i}$
```

assume that driver behavior is the same if a pedestrian is not present. Note that, in this work, we only use labeled driver response and intention as the supervision signals. Therefore, the proposed augmentation strategy is only applicable to *Go* scenarios.

In *Stop* scenarios, we need to know causal objects' locations to remove non-causal objects. However, exhaustive risk object labeling is costly, and that is not the focus of this chapter.

Moreover, even if causal objects are given, we cannot remove causal objects and assume the corresponding driver response to be *Go*, because traffic situations are inherently complicated, so the corresponding driver response is unclear. For instance, a driver is in a congestion situation (i.e., driver stops for the frontal vehicle), and the traffic light of the driver's lane is red. In this situation, the frontal vehicle is labeled as the risk object (cause).

Algorithm 2 : Causal Inference for Risk Object Identification

```
$T$: Number of frames
$N$: Number of objects
Input: A sequence of RGB frames $I \coloneqq \{I_1, I_2, \cdots, I_T\}$ where the ego car stops
Output: Risk object ID

$O \coloneqq$ DetectionAndTracking($I$)
    $\coloneqq \{O_1, O_2, \cdots, O_N\}$// List of Thing object tracklets
$S \coloneqq$ SemanticSegmentation($I$)
    $\coloneqq \{S_1, S_2, \cdots, S_T\}$ // List of Stuff objects
for $O_k \in O$ do
    // Mask out Thing object $k$ on each frame
    $M \coloneqq$ MaskGenerator($I, O_k$)
    // Remove the Thing object $k$ from the tracklet list
    $O = O - \{O_k\}$
    // Predict driver response and intention
    without the object $k$, where $\mathbf{a}_r \coloneqq \{r_k^{go}, r_k^{stop}\}$
    $\mathbf{a}_r, \mathbf{a}_i \coloneqq$ DrivingModel($I, M, O, S$)
end for
return $\arg\max_k(r_k^{go})$
```

However, driver response remains the same if the frontal vehicle were not present because of the red light. Generating *Stop* scenarios is non-trivial, and we leave it for future works.

To train the intention-aware driving mode with the proposed data augmentation strategy, the model should be able to "intervene," i.e., remove a non-causal object from images. We realize the strategy by replacing standard convolutional layers in I3D with *partial convolutional layers* [187], [188]. Note that a partial convolutional layer is initially introduced for image inpainting. We utilize partial convolutions to simulate a scenario without the presence of an object. A 3D partial convolutional layer takes two inputs, i.e., a sequence of RGB frames and a one-channel binary mask for each frame. The pixel values of a mask are 1 by default. While training the driving model with data augmentation, we set the pixels within the selected object to be 0. In addition, the node of the selected object in a graph is disconnected from the rest of the objects.

The proposed training process is outlined in 1. Given training samples in a *Go* scenario, we randomly select an object k to intervene, i.e., simulating a situation without the presence of the object. Specifically, given a tracklet o_k, a one-channel binary mask M_t at time t is defined as

$$M_t(\mathrm{i,j}) = \begin{cases} 0, & \text{if } (\mathrm{i,j}) \text{ in region } o_k^t \\ 1, & \text{otherwise} \end{cases}, \tag{5.1}$$

where o_k^t is the bounding box of the k-th object at time t, and $(\mathrm{i,j})$ is a pixel coordinate within the box. Note that k-th object is discarded from the tracklet list while training the driving model.

5.5.2 Causal Inference for Risk Object Identification

Given a "Stop" scenario, we aim to identify the corresponding risk object. We deploy the same intervention process discussed in Sec 5.5.1 to identify the risk object. Specifically, the masks of a tracklet and the corresponding video frames are processed by the same driving model. The model outputs the confidence score of *Go* and *Stop* without the presence of the tracklet. After iterating through all tracklets, we select the object with the highest *Go* confidence score to be the risk object. This is because the object causes the most driver behavior change. 2 describes the overall causal inference process.

5.6 Experiment

5.6.1 Implementation Details

We implement our framework in TensorFlow. All experiments are performed on a server with 4 NVIDIA TITAN-XP cards. The input to the framework is a 20-frame clip with a resolution of 224×224 at 3 fps, approximately 6.67s. The framework outputs the predictions of driver intention and response of the very next frame. We adopt Inception-v3 [151] pre-trained on ImageNet [123] as the backbone, following [148] to inflate 2D convolution into a 3D ConvNet, and fine-tune it on the Kinetics action recognition dataset [152]. The intermediate feature used in RoIAlign and MaskAlign is the `Mixed_3c` layer, where the number of feature channels is 512. The intention feature is generated from a 1 × 1 × 1 convolution on the `Mixed_5c` layer's feature, and the channel number of the feature is 512. The downsampled binary mask $\mathbf{M}_{\mathrm{i}}^{t}$ is 28 × 28. The decoder length is set to be 3. The model is trained in a two-stage training scheme with a batch size of 32. First, we finetune the Kinetics pre-trained model on the driver-centric ROI dataset for 50 000 iterations without using GCN. Second, we load the weights trained in the first stage and finetune the network with GCN for another 20 000 iterations. Note that we employ the augmentation strategy mentioned in Sec 5.5.1 in the second stage. We use the Adam optimizer [124] with the default parameters. The learning rate is set to be 0.001 and 0.0002 for the first and second stage, respectively.

5.6.2 Driving Model Performance

Evaluation Setup

The performance of the driving model is evaluated as a discrete feasible action prediction, in accordance with [24], [26], [29]. The two discrete actions, *Go* and *Stop* are evaluated. We follow the train/test split defined in [136], where 846 411 and 271 989 samples are used for training and testing, respectively. Four evaluation metrics are utilized. First, we report perplexity as in [24], [26], [29]. Perplexity calculates the negative log-likelihood of predicted probability of **Response** given ground truth (lower is better). Second, the macro-averaged accuracy is reported. Note that, in a multi-class classification setup, the micro-averaged accuracy is preferable if the label distribution is imbalanced. In our task, the *Go* to *Stop* ratio is approximately 4:1. Therefore, we also report the micro-averaged accuracy as the third metric. **Response** prediction can be treated as an online action detection task [139], [189]. We use per-frame mean average precision (mAP) as the fourth evaluation metric.

Evaluation

Table 5.2 summarizes the results of the driving models. We compare the following baselines. To compare different models, we keep their backbone network (i.e., Inception-v3) the same.

Table5.2. Results of driver response prediction compared with baselines. Perplexity (lower is better), macro- and micro-average accuracies, and overall mAP are used as metrics for driver response prediction. The unit is % for all metrics except perplexity. The best and second best performances are shown in bold and underlined, respectively. We also report the performance of driver intention prediction using the overall mAP as the metric.

Model	Response				Intention
	Perplexity	Macro Accuracy	Micro Accuracy	Overall mAP	Overall mAP
1. CNN + LSTM	1.00	64.37	77.95	71.07	/
2. CNN + LSTM + Multi-head	0.93	68.27	79.04	70.12	36.41
3. Pixel-level Attention[27]	0.89	76.15	80.21	78.57	/
4. Object-level Attention [29]	0.84	78.81	83.19	79.02	/
5. GCN (ours)	0.83	77.57	82.64	80.33	/
6. GCN + Multi-head (ours)	0.72	76.30	85.68	84.46	36.31
7. GCN + TRN Head (ours)	0.69	79.32	86.17	83.44	**36.80**
8. GCN + TRN Head + Data Augmentation (ours)	**0.37**	**87.63**	**92.56**	**95.44**	36.75

- **CNN+LSTM.** We extract visual features from the `Mixed_5c` layer of I3D and sequentially input the features at each time step to a two-layer LSTM [139] for temporal modeling.

- **Pixel-level attention.** The pixel-level attention module is proposed by [27] to improve model's intepretability and the performance of driving models.

- **Object-level attention.** In [29], the authors propose an object-centric attention mechanism to augment end-to-end policy learning. Both pixel- and object-level attention modules are incorporated into **CNN+LSTM**.

The following summarizes our proposals.

- **GCN.** The key difference between GCN and three baselines is the input feature to the LSTM module. Specifically, the feature is processed via Graph Convolution Networks and contains interaction among traffic participants and driver.

- **Multi-head.** We add an additional head for driver intention prediction to **CNN+LSTM** and **GCN**. A standard cross-entropy loss is used for driver in-

Table5.3. Ablative study of our design choices.

	Model	Perplexity
Intention Modeling	Without intention modeling	0.83
	Multi-head	0.72
	TRN Head	**0.69**
Different Graphs	Ego-Stuff Graph	0.74
	Ego-Thing Graph	0.80
	Ego-Thing Graph + Ego-Stuff Graph	**0.69**
Spatial Modeling	Appearance Relation	0.73
	Appearance + Spatial Relation	**0.69**
Data Augmentation	Without Augmentation	0.69
	With Augmentation	**0.37**

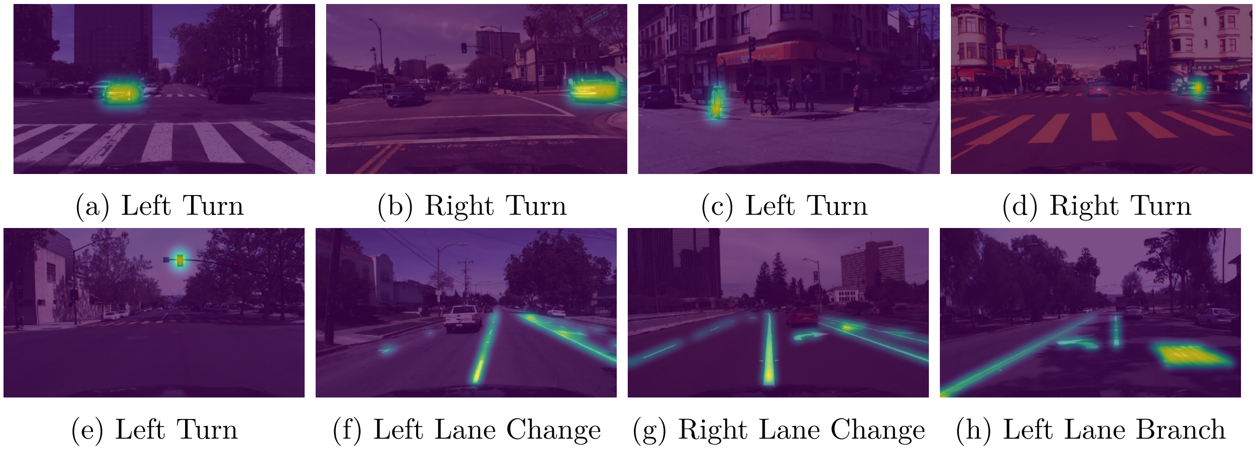
tention prediction. Note that both the interaction and intention features share the same features from the `Mixed_5c` layer of I3D.

- **TRN Head.** To forecast future interactions, we incorporate TRN [139]. We initialize TRN with intention representation (as shown in Figure 5.5) .
- **Intervention.** The concept of intervention is utilized to augment training data to improve the performance of driver response prediction discussed in Sec 5.5.1.

We show that **GCN** outperforms baselines, demonstrating the importance of interaction modeling. By incorporating **Multi-head**, i.e., intention modeling, both extensions reduce the perplexity by 0.07 and 0.11, respectively. With **TRN Head**, we observe that perplexity is reduced by 0.03. Finally, we demonstrate that **Intervention** significantly improves the performance of the driver response prediction (0.32 decrease in perplexity).

While promising improvements are observed for driver response prediction, the trend does not hold for driver intention prediction, as shown in Table 5.2. This is because the intention representations used in the four models (Models 1, 6, 7, and 8 listed in Table 5.2) are features obtained from the `Mixed_5c` layer of I3D, which has negligible gradients in back propagation. A better architectural design for intention prediction is needed, and we leave it for future work.

Ablation Study


(a) Left Turn (b) Right Turn (c) Left Turn (d) Right Turn

(e) Left Turn (f) Left Lane Change (g) Right Lane Change (h) Left Lane Branch

Figure5.7. Visualization of Learned *Ego-Thing Graph* and *Ego-Stuff Graph* on egocentric images. The first and second rows show examples from an *Ego-Thing Graph* and an *Ego-Stuff Graph*, respectively. Comparing (a) and (b), which have similar traffic configurations, our model attends to objects at different locations based on distinct intentions. In (c) and (d), pedestrians intending to cross the street have a significant influence on ego behavior when turning left or turning right. Fig. (e) illustrates a left turn case when the heat map shows high attention around the traffic light, which is green. In (f)-(h), lane markings show strong influences on the ego's lane-related behaviors.

We conduct ablation studies to understand the contributions of the proposed architecture designs. The studies are summarized in Table 5.3.

Analysis of Intention Modeling. The first section of Table 5.3 analyzes the influence of intention modeling. The baseline does not consider intention. When intention representation is incorporated into **Multi-head** and **TRN Head**, the results are improved by 0.11 and 0.14, respectively.

Variations of Different Graphs. When both *Ego-Stuff* and *Ego-Thing Graphs* are considered, the model achieves the best perplexity performance. The results indicate the importance of the proposed interaction modeling of drivers, traffic participants, and road infrastructure.

Importance of Spatial Relation. We study the importance of the spatial relation function (Equation 4.4) to the **Response** prediction. We conduct two experiments, i.e., 1) using only the appearance relations, and 2) appending 3D spatial relation as an additional constraint. Without using the proposed 3D spatial relation, the perplexity increases by 0.04, indicating the need for a spatial constraint.

Data Augmentation via Intervention. We study the impact of data augmentation by comparing the performance of two models trained with and without the data augmentation strategy. The last section in Table 5.3 showcases the advantage of using augmented data, cutting the perplexity by nearly half. The data augmentation strategy adds variations to the training set that improve the robustness of the proposed driving model.

Visualization

We visualize learned affinity matrices in *Ego-Thing Graph* and *Ego-Stuff Graph* to determine if our approach can highlight those objects influencing driver behavior. The visualization results as shown in Figure 5.7 provide a strong evidence that the proposed model captures the underlying interactions between traffic participants and driver.

Figure 5.7a and Figure 5.7b showcase similar traffic configurations where the driver approaches a four-way intersection with the presence of other vehicles. Given different intentions, i.e., *Left Turn* in Figure 5.7a and *Right Turn* in Figure 5.7b, our model attends to objects that impact the ego-vehicle navigation. A similar phenomenon is observed in Figure 5.7c and Figure 5.7d. Different attention map characteristics are observed. While similar driving model architectures are leveraged, three major differences, i.e., different supervision signals, training strategy, and intention-aware design, are introduced in the proposed architectures. Particularly, the additional supervision signal—the driver response, encourages the model attending to object(s) that influence driver behavior.

The bottom row of Figure 5.7 represents attention maps obtained from the *Ego-Stuff Graph*. In Figure 5.7e, the model captures the relation between driver intention (turning left) and traffic light (green light). Note that we observe that the *Ego-Stuff Graph* modeling captures lane-related driver intention, i.e., *Lane Change*, *Lane Branch* and *Merge*, as shown in Figure 5.7f, Figure 5.7g, and Figure 5.7h.

Table5.4. Comparison with baselines. The methods with * are re-implemented by us to ensure the same backbone is used for fair comparisons. *mAcc* stands for mean accuracy, and the unit is %. The best and second best performances are shown in bold and underlined, respectively.

Model	*mAcc*			
	Crossing Vehicle	Crossing Pedestrian	Parked Vehicle	Congestion
Random Selection	15.1	7.1	6.4	5.5
Driver's Attention Prediction * [159]	16.8	8.9	10.0	21.3
Object-level Attention * [29]	22.6	9.5	22.6	40.7
Pixel-level Attention * [27]	28.0	8.1	15.6	35.7
GCN (ours)	27.5	**13.6**	26.0	51.3
GCN + TRN Head (ours)	29.0	13.2	27.3	52.2
GCN + TRN Head + Data Augmentation (ours)	**32.5**	12.9	**28.4**	**57.5**

5.6.3 Risk Object Identification

Evaluation Setup

We evaluate risk object identification in the four reactive scenarios: *Congestion*; *Crossing Pedestrian*; *Crossing Vehicle*; and *Parked Vehicle*. We use accuracy (number of correct predictions over the number of samples) as the metric. A correct prediction is one that has an Intersection over Union (IoU) score between a selected box and a ground truth box that is larger than a predefined threshold. Similar to [70], [190], accuracies at IoU thresholds of 0.5 and 0.75 are reported. In addition, mean accuracy (*mACC*) is calculated by using IoU thresholds ranging from 0.5 to 0.95 (in increments of 0.05).

Evaluation

We compare the performance of Risk Object Identification with the following baselines. The results are shown in Table 5.4.

Random Selection. Random selection randomly picks an object as the risk object from all the detections for a given frame randomly. Note that the method does not process any visual information except object detection. The method is used to contextualize the challenge of this task.

Driver Attention Prediction uses a pre-trained model [159] trained on the BDD-A dataset to predict the driver's gaze attention maps at each frame. We compute an average attention weight of every detected object region based on a predicted attention map. The risk object is the object with the highest attention weight, indicating the driver's gaze attends to this region. The model is trained with human gaze signals that are unavailable in the proposed dataset. The performance of this method is slightly better than **Random Selection** as reported in the second row of Table 5.4. We observe that predicted attention maps tend to focus at a vanishing point. Note that this issue has been raised in [160], highlighting the problem as one of the challenges of imitating human gaze behavior.

Object-level Attention Selector. The object-level attention driving model [29] is reformulated for risk object identification. The risk object is the object with the highest object-attention score.

Pixel-level Attention. Kim et al. [27] propose a causality test to search for regions that influence the network's output behavior. Note that region proposals are formed based on sampling predicted pixel-level attention maps. To identify a risk object, we replace the region proposal strategy used in [27] with object detection, and utilize the inferred pixel-level attention map to filter out detections with low attention values. In the experiments, we set the threshold at 0.002. The modification ensures a fair comparison as region proposals obtained from [27] are not guaranteed to be an object entity. Note that the code of region proposal generation detailed in [27] is not publicly available.

We report favorable risk object identification performance over existing baselines [27], [29] in Table 5.4. The results indicate the effectiveness of the proposed intention-aware driving model and causal inference for the task. In the next section, we perform ablation studies to examine the contributions of each part of our model. Notice that our evaluation protocol differs from [15]. In [15], we train four different driving models and test four scenarios independently, whereas a single intention-aware driving model is trained in this chapter.

Table5.5. Ablation study of the proposed risk object identification framework. The unit is %. The best and second best performances are shown in bold and underlined, respectively.

Driving Model	Data Augmentation	Causal Inference	Crossing Vehicle			Crossing Pedestrian			Parked Vehicle			Congestion		
			$Acc_{0.5}$	$Acc_{0.75}$	$mAcc$	$Acc_{0.5}$	$Acc_{0.75}$	$mAcc$	$Acc_{0.5}$	$Acc_{0.75}$	$mAcc$	$Acc_{0.5}$	$Acc_{0.75}$	$mAcc$
CNN + LSTM	✗	✓	29.9	29.9	26.3	15.5	14.3	12.4	33.1	28.7	25.4	39.4	35.4	32.9
GCN (ours)	✗	✓	31.8	31.5	27.5	<u>16.7</u>	15.5	<u>13.6</u>	32.4	29.4	26.0	56.6	56.6	51.3
GCN + Multi-head (ours)	✗	✓	31.8	31.8	28.0	**17.9**	**17.9**	**14.6**	32.4	29.4	26.3	61.6	57.6	53.8
GCN + TRN Head (ours)	✗	✓	<u>33.1</u>	<u>33.1</u>	<u>29.0</u>	<u>16.7</u>	<u>16.7</u>	13.2	<u>33.8</u>	<u>30.2</u>	<u>27.3</u>	60.6	56.6	52.2
GCN + TRN Head (ours)	✓	✗	28.3	28.0	25.0	13.1	11.9	9.6	22.1	21.3	18.7	<u>65.7</u>	<u>61.6</u>	<u>57.4</u>
GCN + TRN Head (ours)	✓	✓	**37.0**	**37.0**	**32.5**	15.5	15.5	12.9	**35.3**	**31.6**	**28.4**	**66.7**	**62.6**	**57.5**

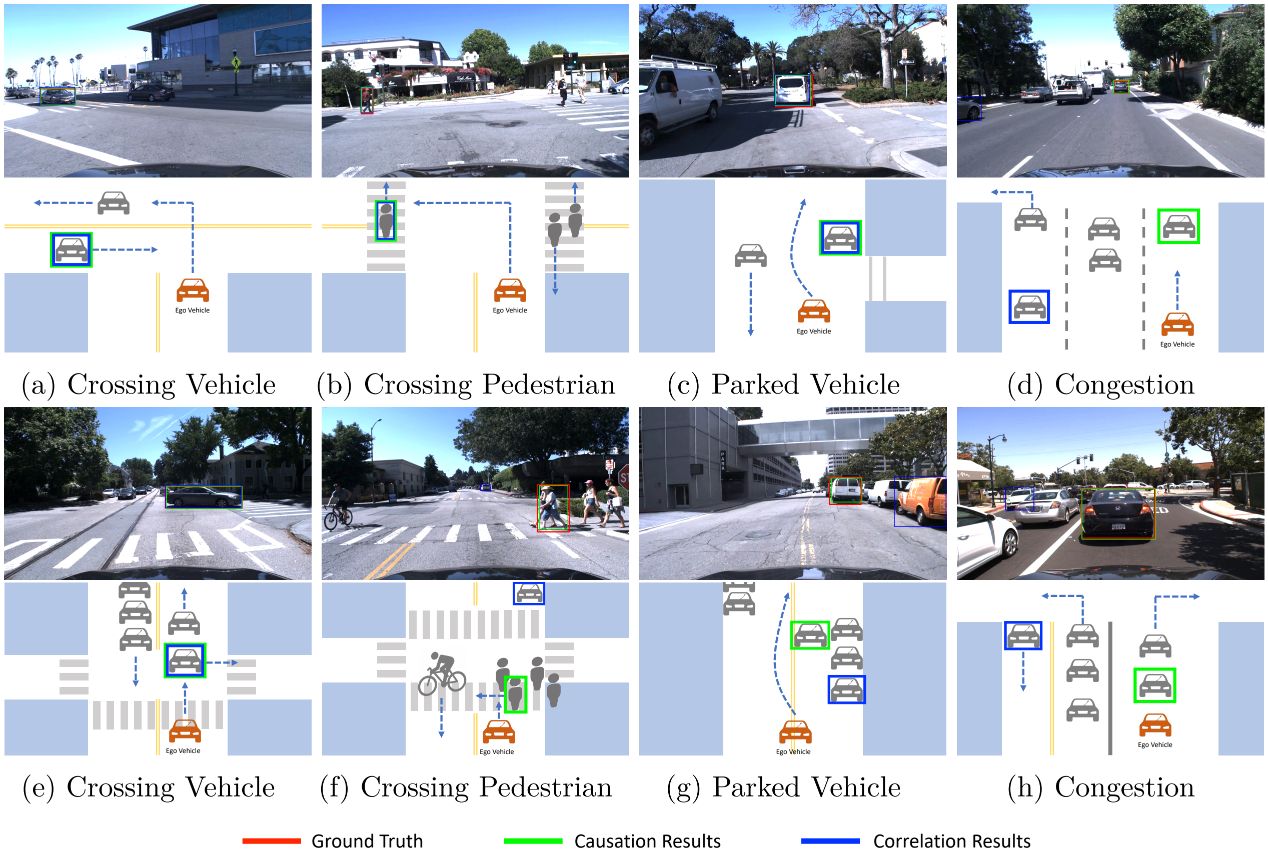
Ablation Study

Three variations are studied to analyze their impacts on the performance of risk object identification: (1) architecture of the driving model, (2) intention modeling and (3) training strategy. The results are summarized in Table 5.5.

Architecture. The completed framework (GCN + TRN Head, reported in the last row of Table 5.5) boosts the *mACC*s of GCN by 6.2%, 0.5%, 3.0% and 24.6% in four different scenarios, respectively. The architecture ranks first in three senarios (*Crossing Vehicle*, *Parked Vehicle*, and *Congestion*). We found interaction modeling is crucial, as it improves performance over a pure CNN+LSTM model.

Intention Modeling. Both multi-head and TRN head based intention modelings improve overall performance. While the two modelings have similar risk object identification results, we choose TRN Head because it achieves better performance of the driver response prediction task.

Training with Data Augmentation. We observe significant improvement in all scenarios with the proposed data augmentation strategy except *Crossing Pedestrian.* The results indicate the effectiveness of the proposed training strategy. For *Crossing Pedestrian*, our conjecture is that vehicles are likely to be chosen because of the natural imbalanced distribution in the training data. Note that the ratio of detected vehicles to pedestrians is approximately 17:1. Our model learns how to identify risk objects under traffic configurations (especially different vehicle configurations) so that the model performs favorably for scenarios that involve interacting with vehicles. In contrast, scenarios that involve interact-



(a) Crossing Vehicle (b) Crossing Pedestrian (c) Parked Vehicle (d) Congestion

(e) Crossing Vehicle (f) Crossing Pedestrian (g) Parked Vehicle (h) Congestion

Figure5.8. Risk object identification results obtained by **Causation** and **Correlation**. Note that **Causation** is the causal inference based approach proposed in the paper. Instead of using causal inference, **Correlation** determines the risk object by selecting the object with the highest attention weight to *Ego* in the *Ego-Thing Graph*. The top row shows an egocentric view where green boxes indicate our **Causation** results, blue boxes are **Correlation** results, and ground truth boxes are in red. A bird's-eye-view representation is presented in the bottom row, providing information including scene layout and intentions of traffic participants.

ing with pedestrians are less emphasized. To solve this problem, a possible solution is to perform a category-aware intervention so that a balanced distribution can be obtained.

In summary, with the proposed components, i.e., TRN Head, intention modeling, and training with data augmentation, we demonstrate state-of-the-art risk object identification performance. Note that this observation is also found in driver response prediction, discussed in Sec 5.6.2.

Correlation vs. Causation

We study the importance of causal modeling for this task. Instead of using causal inference (called **Causation**) to identify the risk object, the risk object is the object with the highest attention weight between *Ego* in *Ego-Thing Graph*. We call this method **Correlation**. In Table 5.5, the second to the last row shows the results of **Correlation**. Our **Causation** approach significantly outperforms **Correlation** in all reactive scenarios. We empirically demonstrate the need of casual modeling for this task.

In Figure 5.8, ground truth risk objects are enclosed in red bounding boxes, our **Causation** results are shown in green, and the **Correlation** predictions are shown in blue boxes. In addition, we provide a bird's-eye-view (BEV) pictorial illustration of scenes in the second row. Note that BEVs depict scene layouts, driver intention, and traffic participants' intentions, with identified risk objects in green boxes. In Figure 5.8 (b), three crossing pedestrians with different intentions are depicted. Our **Causation** approach correctly identifies the left-hand side pedestrian as the risk object while the driver intends to turn left. While **Correlation** predicts the same result, our method is more explainable because the decision is made by considering driver intention. Figure 5.8 (d), (f),(g) and (h) showcase examples where **Correlation** fails but the proposed framework identifies risk objects successfully.

5.7 Compare with Our Preliminary Work

As mentioned previously, we also conducted an preliminary exploration of risk object identification in [15]. The preliminary work [15] also formulates the task as a cause-effect problem and utilizes idea of causal inference. In terms of method, the only distinction is the driving model architecture. In this section, we provide the details of our preliminary driving model, implementation details and experimental results.

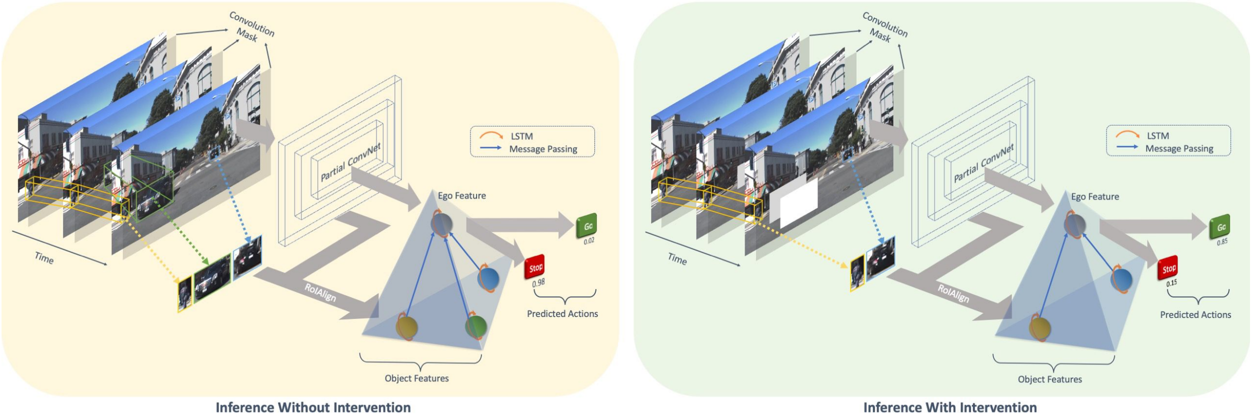


Figure5.9. An overview of the preliminary framework. The right and left figures show the inference process with and without intervention, respectively. Both employ the same driving model to output the predicted driver response. The inputs to the driving model include a sequence of RGB frames, a sequence of binary masks and object tracklets. Partial convolution and average pooling are employed to obtain the ego features while object features are extracted by RoIAlign. Each feature is modeled temporally and then propagates information to form a visual representation of the scene for final prediction. On the right, the input is intervened at an object level by masking out the selected object on the convolution mask and also removing it from the tracklets. For example, we remove the car in the green box and the driving model returns a high confidence score of 'go'.

5.7.1 Preliminary Driving Model

An overview of the preliminary driving model architecture is visualized in Figure 5.9. Given video frames, we utilize Partial Convolution Networks [187], [188] and average pooling to represent pixel-level features of the ego vehicle.

To obtain object-level representation, we apply Mask R-CNN [12] and Deep SORT [183] to detect and track every object throughout time. RoIAlign [12] is employed to extract object representations. At time t, the ego vehicle features and object features are updated via long short-term memory (LSTM) module [184]. This temporal modeling process captures the dynamics of ego vehicle and objects.

Motivated by [29] and [16], both pixel-level and object-level features are essential for driving scene tasks. Hence, we aggregate the two sources of features via message passing,

$$g = h_{\mathrm{e}} \oplus \frac{1}{N}(\sum_{\mathrm{i}=1}^{N} h_{\mathrm{i}}) \tag{5.2}$$

where g is defined as the aggregated features, h_{e} represents the ego's features obtained after temporal modeling and $h_o = \{h_{\mathrm{i}}, h_2, \cdots, h_N\}$ are the N object features. $\oplus$ indicates a concatenation operation. To manipulate the representation at an object level, we set the pixel value of the binary mask to be 0 at the location of the selected object. The mask influences the features extracted from partial convolution and disconnects the message of the selected object from the rest. In the end, this representation g is passed through fully connected layers to obtain the final classification of the driver response (i.e., ‘go’ or ‘stop’).

The main differences between this preliminary driving model and the aforementioned completed Level 3 SA driving model are *Comprehension* and *Projection*:

- **Comprehension.** The completed Level 3 SA driving model uses Graph Convolution Networks to represent and learn the spatial-temporal interactions among objects. Whereas, this preliminary driving model uses a simple averaging operation to aggregate information of other objects. There is no specific design to model the spatial-temporal relations.

- **Projection.** More attention is paid to the projection module in the completed Level 3 SA driving model. It applies two ideas: (1) an encoder-decoder architecture based on Temporal Recur-rent Network (TRN) that uses both historical evidence and predicted future information to better predict current action, and (2) an intention-aware design as cues to aid better prediction of the driver response. However, in the preliminary work, object states is processed and projected to the future by a simple LSTM temporal modeling.

Table5.6. Ablation studies. Results of risk object identification in four scenarios on the HDD. The unit is %. The best and second performances are shown in bold and underlined, respectively.

Driving Model	Mask	Training with Intervention	Crossing Vehicle			Crossing Pedestrian			Parked Vehicle			Congestion		
			$Acc_{0.5}$	$Acc_{0.75}$	$mAcc$	$Acc_{0.5}$	$Acc_{0.75}$	$mAcc$	$Acc_{0.5}$	$Acc_{0.75}$	$mAcc$	$Acc_{0.5}$	$Acc_{0.75}$	$mAcc$
Vanilla CNN	RGB	✗	36.0	35.7	31.4	20.2	16.7	14.9	36.8	32.4	29.7	87.9	83.9	76.8
Partial CNN	RGB	✗	38.6	37.6	33.5	22.6	19.0	16.2	36.0	32.4	29.0	81.8	81.8	73.7
	RGB	✓	41.2	40.5	36.2	19.0	16.7	13.5	$\underline{39.0}$	$\underline{36.8}$	$\underline{32.4}$	**94.9**	**91.0**	**82.6**
	Convolution	✗	38.6	37.6	33.6	22.6	17.9	16.2	36.8	33.1	29.5	88.9	84.8	78.0
	Convolution	✓	$\underline{44.4}$	$\underline{43.1}$	$\underline{38.5}$	25.0	$\underline{22.6}$	$\underline{19.3}$	34.6	33.1	28.8	88.0	84.8	77.3
Partial CNN + Object	Convolution	✗	39.9	38.9	34.4	$\underline{27.4}$	$\underline{22.6}$	18.9	31.6	27.9	24.7	91.9	87.9	79.7
	Convolution	✓	**49.2**	**48.6**	**43.0**	**35.7**	**32.1**	**27.0**	**47.1**	**44.9**	**39.8**	$\underline{92.9}$	$\underline{88.9}$	$\underline{81.0}$

5.7.2 Implementation Details

We implemented this framework in PyTorch, and performed all experiments on a system with Nvidia Quadro RTX 6000 graphics cards. The input to the framework is a sequence of frames with a resolution of 299 × 299 at 3 fps, and T is set to 3 in all the experiments, approximately 1s. The corresponding input mask maintains the same size as the input image. The convolutional backbone is a InceptionResnet-V2 [191], pre-trained on ImageNet [123] and modified with partial convolution operation [187], [188]. A Detectron model [192] trained on MSCOCO [70] is used to generate bounding boxes for objects. RoIAlign extracts object features with size $20 \times 8 \times 8$ from the `Conv2d_7b` layer, which is then padded into a 1-D vector of size 1280.

We follow the same way as [139] to initialize the hidden states with channel number set to 512 and also use dropout [193] of 0.5 at hidden state connections in the LSTM module. The aggregated feature g concatenated from ego features and object features is a 1-D vector with 1024 channels. Similar to [27], the output sizes of 3 fully-connected layers before the final binary classifier are 100, 50 and 10, respectively.

The network is trained end-to-end for 10 epochs with batch size set to 16. We use Adam [124] optimizer with default parameters, learning rate 0.0005, and weight decay 0.0005.

5.7.3 Ablation Studies

We conduct ablation studies in Table 5.6 to provide a comprehensive understanding of the contributions for each component.

Architecture of the Driving Model. Our proposed driving model uses features from CNN features and object features. For CNN features, we test two backbone features, i.e., vanilla convolution and partial convolution.

Intervention Mask. Different from vanilla CNN, the input to partial CNN includes an extra mask, offering two options to intervene an image. We either input a RGB image with selected region masked out or feed in a binary mask with selected region set to 0 and the rest to 1. We denote the two ways of intervention as “RGB mask” and “Convolution mask” in Table 5.6.

Training with Intervention. To discover how the framework performs, especially when using the model trained with more traffic configuration variations, we explore two experimental settings — training with and without intervention. Notice that for Partial CNN model, we always use convolution mask to remove selected objects when training with intervention. In Partial CNN + Object model, we additionally remove the selected object features during message passing.

By analyzing the results, our completed framework (last row in Table 5.6) boosts the *mACC* by 11.6%, 13.5%, 11% and 7.3%, respectively, compared with the lowest accuracies. It ranks first in three senarios (Crossing Vehicle, Crossing Pedestrian and Parked Vehicle) and second in Congestion case.

Training with intervention always leads to an increase in accuracy when the driving model is modeled with object-level information. However, it does not necessarily help the performance when the driving model is downgraded to Partial CNN only.

In terms of intervened mask type, an interesting phenomenon is observed that in Crossing Vehicle and Crossing Pedestrian scenarios, intervening with convolution mask achieves higher accuracy than RGB mask in general. However, in the other two scenarios, this trend is no more noticeable. Our conjecture is that, when the ego vehicle deviates for parked vehicle or stops for congestion, the target risk object is salient pixel-wise, taking up the majority area of the frame. Under such circumstance, inputting a masked RGB frame could be enough for changing the driving model output significantly. Thus, the increased performance resulting from hallucination effect of convolution mask is relatively unremarkable .

5.7.4 Quantitative Evaluation

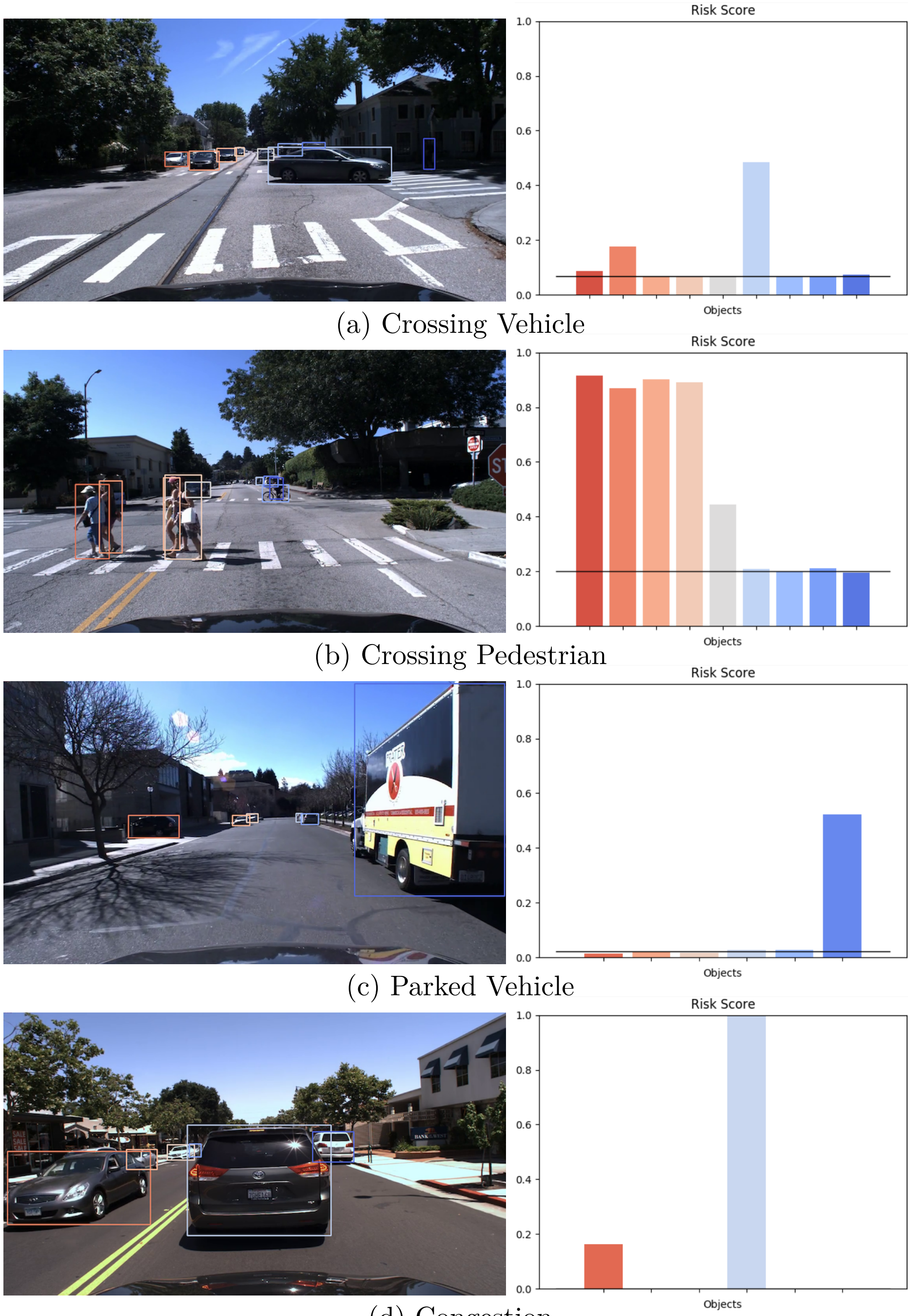
Table5.7. Comparison with other risk object identification methods. The methods with * are re-implemented by us. The unit is %. The best and second performances are shown in bold and underlined, respectively.

Method	*mAcc*			
	Crossing Vehicle	Crossing Pedestrian	Parked Vehicle	Congestion
Random Selection	15.1	7.1	6.4	5.5
Driver's Attention Prediction * [159]	16.8	8.9	10.0	21.3
Object-level Attention Selector * [29]	36.5	21.2	20.1	8.9
Pixel-level Attention + Causality Test * [27]	41.9	21.5	34.6	62.7
Ours	**43.0**	**27.0**	**39.8**	**81.0**

Since by the time we finished this work there was no existing work which could be directly applied to the risk object identification task, we re-implemented three approaches mentioned in Sec 5.6.3 and followed their spirits to select important/risk objects in the driving scenario. The comparison with our method is shown in Table 5.7. Note that the results of the last row are not directly comparable to the results in Table 5.4 because we train four different driving models and test four scenarios independently, whereas a single intention-aware driving model is trained in Table 5.4. As shown in the table, our preliminary result still achieves the best performance among the five methods which demonstrates the effeteness of our design.

5.7.5 Qualitative Evaluation

In addition to select only one risk object, our framework can also be used to assess the risk of every object in the scene. We visualize the results in Figure 5.10 and the ego vehicles in the samples are supposed to take a “stop” action. All detected objects are encased in bounding boxes with different colors, and their risk scores are in a bar chart with corresponding color.



(a) Crossing Vehicle

(b) Crossing Pedestrian

(c) Parked Vehicle

(d) Congestion

Figure5.10. Sample scenes from the HDD dataset with object risk score visualized. On the left, all detected objects are shown in bounding boxes with different colors. The risk score of each object is depicted in a bar chart on the right. The color of each bar is one-to-one matched to the bounding box. We use a black horizontal line to indicate the predicted 'go' score of the ego vehicle without applying any intervention.

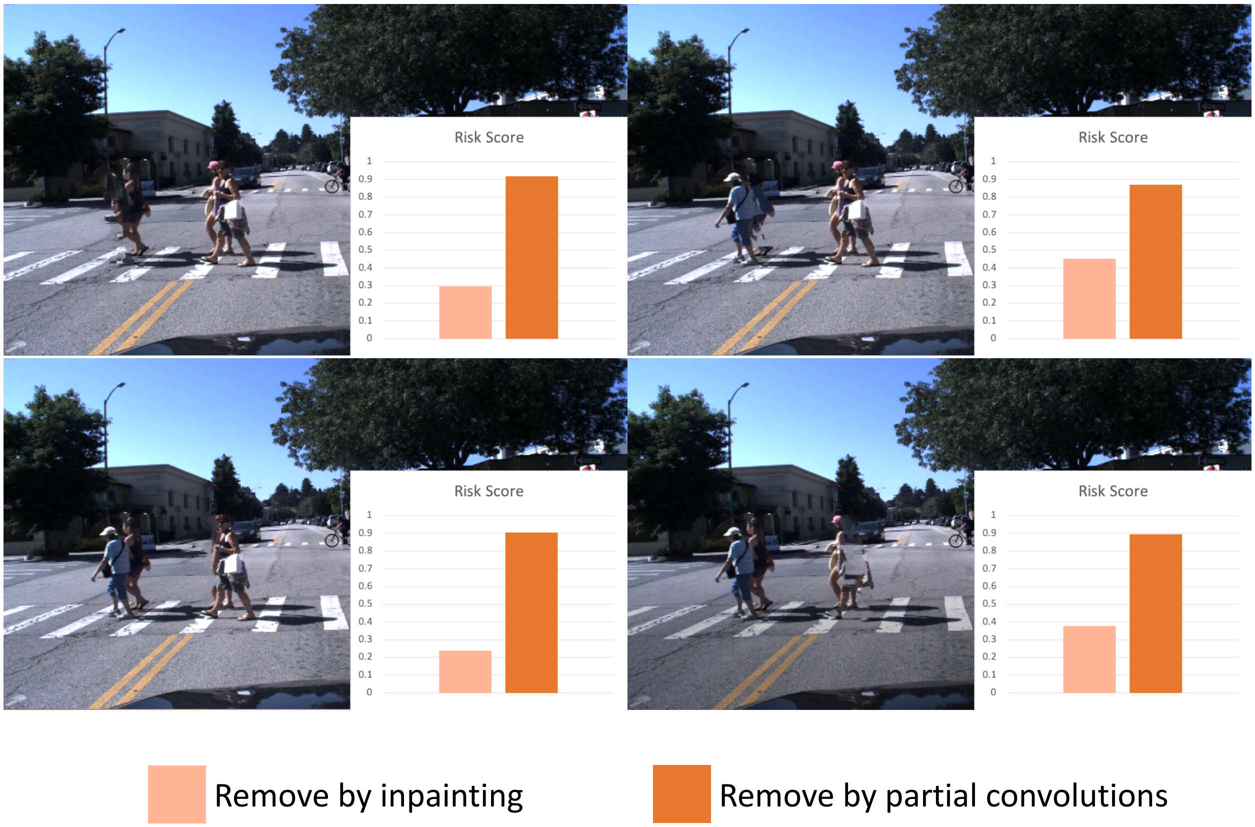


Figure5.11. An example of computed risk scores by using inpainting images compared with our method.

The risk score of an object is equivalent to the predicted confidence score of 'go' action after removing it. A higher score of "go" action means a higher possibility that it is the object that stops the ego vehicle. We use a black horizontal line to indicate the predicted confidence score of 'go' action when the input is not intervened. If the score is less than 0.5, then the sample is classified as "stop". As we see in the figures, our framework generates a reasonable risk assessment result.

In Figure 5.10 (b), when multiple risk objects (a group of people) exist, our framework assigns high risk scores to every potential risk object. It seems correct at first glance. However, re-thinking the results leads to some questions since removing any of the four pedestrians will not make the ego car move. Our conjecture is that the partial convolutional operation not only hallucinates the removed the area but also the surrounding regions are affected due to the growing receptive field as networks go deeper. As pedestrians are adjacent in this example, removing one single person by partial convolutions may dilute the surrounding ones and return high risk scores. To verify, we manually inpainting the image by removing every person iteratively and feed the inpainting image to the same driving model without applying partial convolutions. Results are shown in Figure 5.11 with lower risk scores, indicating the correctness of the driving model. And it may also prove our guess about partial convolutional operations. On the other hand, we see the potential of identifying a group of adjacent risk objects via our framework.

5.7.6 Failure Cases

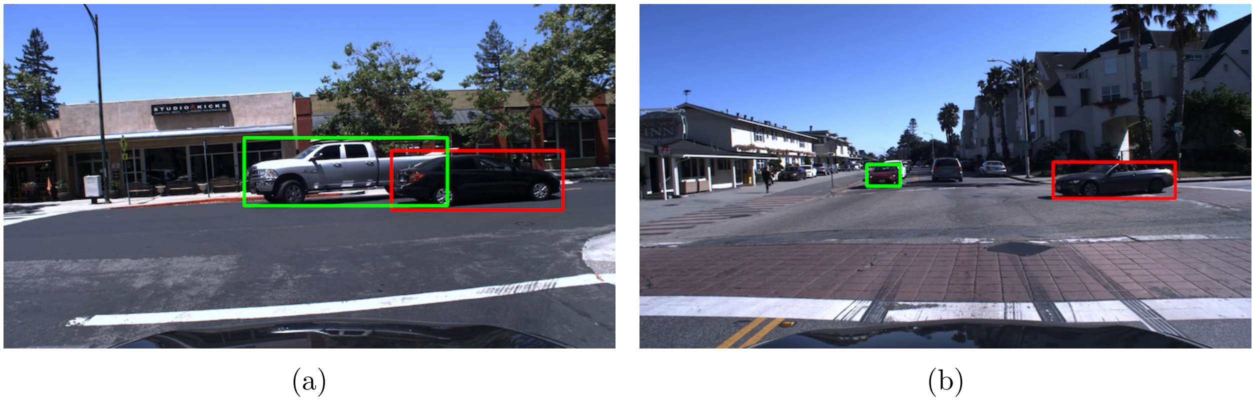


Figure5.12. Examples of failure cases. Our prediction is in green and ground truth is in red.

While our model shows the possibility to identify the intention of the ego vehicle based on the past motion (Figure 5.8 (b)), there are situations that our driving model is confused and chooses an incorrect risk object when the changes of historical motion are not obvious. In Figure 5.12 (a), the ego vehicle plans to take a right turn and stops for the vehicle in the red box. However, our framework selects the white pickup truck over the black vehicle as the risk object. The reason could be the intention of the ego vehicle is ambiguous and historical cues are not informative. Additionally, in Figure 5.12 (b), our driving model is not able to distinguish which vehicle will move first at a 4-way stop intersection and where it is going, resulting in a wrong selection. Hence, we believe explicitly modeling the ego's intention, as well as other participants', in the driving model will render better inference results. Inspired by this observation, this is also the initial motivation that we embed intention-aware design into our completed Level 3 SA driving model.

5.8 Discussion

In this chapter, we explore the projection part of SA for the driving model. One of the realizations of projection is to identify the risk in the near future. Thus, we propose a novel driver-centric definition of risk, i.e., risky objects influence driver behavior. A new task called risk object identification is introduced and is formulated as a cause-effect problem. We present a novel two-stage risk object identification framework inspired by the model of Level 3 situation awareness and causal inference. Especially for projection, we utilize an encoder-decoder Temporal Recur-rent Network (TRN) to aggregate information from both accumulated past and predicted future. Besides, an intention-aware design is introduced to make driver response predictions based on the driver intention.

We also create a driver-centric Risk Object Identification (ROI) dataset to evaluate the proposed system. Extensive quantitative and qualitative evaluations are conducted. Favorable performance compared with strong baselines is demonstrated. Future work can leverage road topology explicitly to improve driver intention prediction. Additionally, a single shot risk object identification framework would be interesting to explore for practical applications.

6. CONCLUSION

6.1 Summary

We studied three levels of situation awareness for advanced driver-assistance systems based on different computer vision tasks. For Level 1 – *Perception*, we focused on a specific and common scenario – driving in photon-limited conditions where traffic elements are difficult to perceive. To tackle this problem, we integrated a new non-local feature aggregation method and a knowledge distillation technique with the state-of-the-art detector networks to produce better feature representations for photon-limited images. For Level 2 – *Comprehension*, we studied how to understand the spatial-temporal relations between a driver and objects. We proposed a 3D-aware egocentric spatial-temporal model by using Graph Convolution Networks and demonstrated the effectiveness of our design via a task called tactical driver behavior recognition. To achieve the highest level of situation awareness, *Projection*, we presented a completed Level 3 SA driving model which is an integration of perception, comprehension, and projection. Combined with causal inference, we are able to forecast the driver's operation intentions and anticipate potential risks to aid driving safety.

6.2 Future Work

There are also many interesting problems which can be done in the future work:

First, in Chapter 5, before non-local feature aggregation, we need the feature extractor to obtain features individually for each frame. Although the overall design already reduces computations compared with non-local search in the image space, it is still relatively memory- and time-consuming if we want to apply it to real-time applications. It is a promising direction to study how to compress the detection network size, or how to speed up the feature extraction and aggregation, so we can realize the photon-limited detection in real-time.

Second, in Chapter 4 and Chapter 5, we make the assumption that the state of *Thing* object can be influenced by other *Thing* objects and *Ego* but do not consider the influence from *Stuff* objects. We are interested in how to merge two graphs into one, and how to represent a more comprehensive and precise representation of interactions among all the traffic elements.

Third, for projection in Chapter 5, future work can leverage road topology explicitly to improve driver intention prediction. Additionally, a single shot risk object identification framework would be interesting to explore for practical applications.

The evaluations of Chapter 5 does not apply the photon-limited object detection at the perception level. It is due to the dataset limitations. HDD [136] dataset is captured during the daytime and BDD 100K [11] does not provide the annotations of risk objects. Currently, there is no existing dataset that fulfills our evaluation needs of risk object identification in photon-limited conditions. We will leave it as future work, and we believe that it will be very feasible to combine photon-limited object detection into the completed Level 3 SA driving model.

REFERENCES

[1] World Health Organization, *Global status report on road safety 2018: Summary*, 2018.

[2] *National Highway Traffic Safety Administration*, https://www.nhtsa.gov/research/.

[3] Wikipedia contributors, *Advanced driver-assistance systems — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Advanced_driver-assistance_systems&oldid=1034822989, [Online; accessed 3-August-2021], 2021.

[4] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems”, *Human factors*, vol. 37, no. 1, pp. 32–64, 1995.

[5] M. R. Endsley, “Theoretical Underpinnings of Situation Awareness: A Critical Review”, in *Situation Awareness Analysis*, 2000.

[6] M. R. Endsley and M. W. Smolensky, “Situation awareness in air traffic control: The picture.”, 1998.

[7] C. M. Schulz, M. R. Endsley, E. F. Kochs, A. W. Gelb, and K. J. Wagner, “Situation awareness in anesthesia: Concept and research”, *The Journal of the American Society of Anesthesiologists*, vol. 118, no. 3, pp. 729–742, 2013.

[8] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks”, *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge”, *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[11] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, “Bdd100k: A diverse driving video database with scalable annotation tooling”, *arXiv preprint arXiv:1805.04687*, vol. 2, no. 5, p. 6, 2018.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn”, in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[13] T. N. Kipf and M. Welling, “Semi-supervised Classification with Graph Convolutional Networks”, in *ICLR*, 2017.

[14] C. Li, S. H. Chan, and Y.-T. Chen, “Driver-centric risk object identification”, *arXiv preprint arXiv:2106.13201*, 2021.

[15] C. Li, S. H. Chan, and Y.-T. Chen, “Who make drivers stop? towards driver-centric risk assessment: Risk object identification via causal inference”, in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 10 711–10 718.

[16] C. Li, Y. Meng, S. H. Chan, and Y.-T. Chen, “Learning 3d-aware egocentric spatial-temporal interaction via graph convolutional networks”, in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 8418–8424.

[17] J. Janai, F. Güney, A. Behl, A. Geiger, *et al.*, “Computer vision for autonomous vehicles: Problems, datasets and state of the art”, *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, no. 1–3, pp. 1–308, 2020.

[18] rnst D. Dickmanns, *Dynamic Machine Vision*, http://dyna-vision.de/, 1995.

[19] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. v. Hundelshausen, *et al.*, “Team annieway’s autonomous system for the 2007 darpa urban challenge”, *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.

[20] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3d object detection network for autonomous driving”, in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[21] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[22] D. A. Pomerleau, “AlVINN: An Autonomous Land Vehicle in a Neural Network”, in *Technical report, Carnegie Mellon University, Computer Science Department*, 1989.

[23] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, “End to End Learning for Self-Driving Cars”, in *arXiv preprint arXiv:1604.07316*, 2016.

[24] H. Xu, Y. Gao, F. Yu, and T. Darrell, “End-To-End Learning of Driving Models From Large-Scale Video Datasets”, in *CVPR*, 2016.

[25] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end Driving via Conditional Imitation Learning", in *IEEE International Conference on Robotics and Automation*, 2018.

[26] D. Wang, C. Devin, Q.-Z. Cai, P. Krähenbühl, and T. Darrell, "Monocular Plan View Networks for Autonomous Driving", in *International Conference on Intelligent Robots and Systems*, 2019.

[27] J. Kim and J. Canny, "Interpretable Learning for Self-driving Cars by Visualizing Causal Attention", in *IEEE International Conference on Computer Vision*, 2017.

[28] J. Kim, T. Misu, Y.-T. Chen, A. Tawari, and J. Canny, "Grounding Human-to-vehicle Advice for Self-driving Vehicles", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[29] D. Wang, C. Devin, Q.-Z. Cai, F. Yu, and T. Darrell, "Deep Object Centric Policies for Autonomous Driving", in *ICRA*, 2019.

[30] A. Arcos-Garcia, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems", *Neurocomputing*, vol. 316, pp. 332–344, 2018.

[31] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second", in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2903–2910.

[32] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks", in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[33] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking", in *2008 IEEE Conference on computer vision and pattern recognition*, IEEE, 2008, pp. 1–8.

[34] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3d pose estimation and tracking by detection", in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 623–630.

[35] J. M. Alvarez, T. Gevers, and A. M. Lopez, "3d scene priors for road detection", in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 57–64.

[36] R. Mohan, "Deep deconvolutional networks for scene parsing", *arXiv preprint arXiv:1411.4101*, 2014.

[37] G. L. Oliveira, W. Burgard, and T. Brox, “Efficient deep models for monocular road segmentation”, in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 4885–4891.

[38] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, “Mvsnet: Depth inference for unstructured multi-view stereo”, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 767–783.

[39] Q. Xu and W. Tao, “Multi-scale geometric consistency guided multi-view stereo”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5483–5492.

[40] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, “Recurrent mvsnet for high-resolution multi-view stereo depth inference”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5525–5534.

[41] N. Yang, R. Wang, J. Stuckler, and D. Cremers, “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry”, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 817–833.

[42] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, “Soft-slam: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles”, *Journal of field robotics*, vol. 35, no. 4, pp. 578–595, 2018.

[43] K. Lenac, J. Ćesić, I. Marković, and I. Petrović, “Exactly sparse delayed state filter on lie groups for long-term pose graph slam”, *The International Journal of Robotics Research*, vol. 37, no. 6, pp. 585–610, 2018.

[44] C. Wojek and B. Schiele, “A dynamic conditional random field model for joint labeling of object and scene classes”, in *European Conference on Computer Vision*, Springer, 2008, pp. 733–747.

[45] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele, “Monocular visual scene understanding: Understanding multi-object traffic scenes”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 4, pp. 882–897, 2012.

[46] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, “3d traffic scene understanding from movable platforms”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2013.

[47] A. Seff and J. Xiao, “Learning from maps: Visual common sense for autonomous driving”, *arXiv preprint arXiv:1611.08583*, 2016.

[48] D. G. Jones and M. R. Endsley, "Sources of situation awareness errors in aviation.", *Aviation, space, and environmental medicine*, 1996.

[49] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local Neural Networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[50] J. Ma, S. Masoodian, D. A. Starkey, and E. R. Fossum, "Photon-number-resolving Megapixel Image Sensor at Room Temperature without Avalanche Gain", *Optica*, 2017.

[51] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018, pp. 734–750.

[52] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection", in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007. DOI: 10.1109/ICCV.2017.324.

[53] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, "Ssd: Single shot multibox detector", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015. DOI: 10.1109/TPAMI.2015.2389824.

[55] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks", in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16, Barcelona, Spain: Curran Associates Inc., 2016, pp. 379–387, ISBN: 9781510838819.

[56] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 408–417.

[57] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards high performance video object detection", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 7210–7218.

[58] F. Xiao and Y. J. Lee, "Video object detection with an aligned spatial-temporal memory", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018, pp. 485–501.

[59] M. Liu and M. Zhu, "Mobile video object detection with temporally-aware feature maps", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 5686–5695.

[60] G. Bertasius, L. Torresani, and J. Shi, "Object detection in video with spatiotemporal sampling networks", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018, pp. 331–346.

[61] S. Wang, Y. Zhou, J. Yan, and Z. Deng, "Fully motion-aware network for video object detection", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018, pp. 542–557.

[62] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Relation distillation networks for video object detection", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 7023–7032.

[63] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 10 337–10 346.

[64] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 3038–3046.

[65] Y. P. Loh and C. S. Chan, "Getting to know low-light images with the exclusively dark dataset", *Computer Vision and Image Understanding*, vol. 178, pp. 30–42, 2019, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2018.10.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314218304296.

[66] W. Yang, Y. Yuan, W. Ren, J. Liu, W. J. Scheirer, Z. Wang, T. Zhang, Q. Zhong, D. Xie, S. Pu, Y. Zheng, Y. Qu, Y. Xie, L. Chen, Z. Li, C. Hong, H. Jiang, S. Yang, Y. Liu, X. Qu, P. Wan, S. Zheng, M. Zhong, T. Su, L. He, Y. Guo, Y. Zhao, Z. Zhu, J. Liang, J. Wang, T. Chen, Y. Quan, Y. Xu, B. Liu, X. Liu, Q. Sun, T. Lin, X. Li, F. Lu, L. Gu, S. Zhou, C. Cao, S. Zhang, C. Chi, C. Zhuang, Z. Lei, S. Z. Li, S. Wang, R. Liu, D. Yi, Z. Zuo, J. Chi, H. Wang, K. Wang, Y. Liu, X. Gao, Z. Chen, C. Guo, Y. Li, H. Zhong, J. Huang, H. Guo, J. Yang, W. Liao, J. Yang, L. Zhou, M. Feng, and L. Qin, "Advancing image understanding in poor visibility environments: A collective benchmark study", *IEEE Transactions on Image Processing*, vol. 29, pp. 5737–5752, 2020. DOI: 10.1109/TIP.2020.2981922.

[67] Y. Sasagawa and H. Nagahara, "Yolo in the dark - domain adaptation method for merging multiple models", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 345–359.

[68] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 3291–3300.

[69] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[70] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context", in *IEEE European Conference on Computer Vision*, 2014.

[71] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

[72] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network", in *Proceedings of the 24th ACM International Conference on Multimedia*, ser. MM '16, Amsterdam, The Netherlands: Association for Computing Machinery, 2016, pp. 516–520, ISBN: 9781450336031. DOI: 10.1145/2964284.2967274. [Online]. Available: https://doi.org/10.1145/2964284.2967274.

[73] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Barcelona, Spain, 2019, pp. 2888–2897, ISBN: 9781510838819.

[74] A. Gnanasambandam and S. H. Chan, "Image classification in the dark using quanta image sensors", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 502–519.

[75] D. Jobson, Z. Rahman, and G. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes", *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 965–976, 1997. DOI: 10.1109/83.597272.

[76] D. Coltuc, P. Bolon, and J.-M. Chassery, "Exact histogram specification", *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1143–1152, 2006. DOI: 10.1109/TIP.2005.864170.

[77] H. Ibrahim and N. S. Pik Kong, "Brightness preserving dynamic histogram equalization for image contrast enhancement", *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1752–1758, 2007. DOI: 10.1109/TCE.2007.4429280.

[78] Y.-T. Kim, “Contrast enhancement using brightness preserving bi-histogram equalization”, *IEEE Transactions on Consumer Electronics*, vol. 43, no. 1, pp. 1–8, 1997. DOI: 10.1109/30.580378.

[79] X. Fu, D. Zeng, Y. Huang, Y. Liao, X. Ding, and J. Paisley, “A fusion-based enhancing method for weakly illuminated images”, *Signal Processing*, vol. 129, pp. 82–96, 2016, ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2016.05.031. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168416300949.

[80] C. Guo, C. Li, J. Guo, C. C. Loy, J. Hou, S. Kwong, and R. Cong, “Zero-reference deep curve estimation for low-light image enhancement”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 1780–1789.

[81] W. Ren, S. Liu, L. Ma, Q. Xu, X. Xu, X. Cao, J. Du, and M.-H. Yang, “Low-light image enhancement via a deep hybrid network”, *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4364–4375, 2019. DOI: 10.1109/TIP.2019.2910412.

[82] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, “Underexposed photo enhancement using deep illumination estimation”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 6849–6857.

[83] W. Yang, S. Wang, Y. Fang, Y. Wang, and J. Liu, “From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 3063–3072.

[84] Y. Atoum, M. Ye, L. Ren, Y. Tai, and X. Liu, “Color-wise attention network for low-light image enhancement”, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 2130–2139. DOI: 10.1109/CVPRW50498.2020.00261.

[85] S. Gu, Y. Li, L. V. Gool, and R. Timofte, “Self-guided network for fast image denoising”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 2511–2520.

[86] K. Xu, X. Yang, B. Yin, and R. W. Lau, “Learning to restore low-light images via decomposition-and-enhancement”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 2281–2290.

[87] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, "Burst photography for high dynamic range and low-light imaging on mobile cameras", *ACM Trans. Graph.*, vol. 35, no. 6, Nov. 2016, ISSN: 0730-0301. DOI: 10.1145/2980179.2980254. [Online]. Available: https://doi.org/10.1145/2980179.2980254.

[88] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll, "Burst denoising with kernel prediction networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 2502–2510.

[89] F. Lv, Y. Li, and F. Lu, "Attention guided low-light image enhancement with a large scale low-light simulation dataset", *International Journal of Computer Vision*, vol. 129, pp. 2175–2193, Jul. 2021.

[90] B. Chen and P. Perona, "Vision without the image", *Sensors*, vol. 16, no. 4, 2016, ISSN: 1424-8220. DOI: 10.3390/s16040484. [Online]. Available: https://www.mdpi.com/1424-8220/16/4/484.

[91] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[92] N. A. Dutton, I. Gyongy, L. Parmesan, and R. K. Henderson, "Single photon counting performance and noise analysis of CMOS SPAD-based image sensors", *Sensors*, vol. 16, no. 7, p. 1122, 2016.

[93] N. Dutton, T. Al Abbas, I. Gyongy, F. Mattioli Della Rocca, and R. Henderson, "High dynamic range imaging at the quantum limit with Single Photon Avalanche Diode based image sensors", *MDPI Sensors*, vol. 18, no. 4, p. 1166, 2018.

[94] C. Bruschini, S. Burri, S. Lindner, A. C. Ulku, C. Zhang, I. M. Antolovic, M. Wolf, and E. Charbon, "Monolithic SPAD arrays for high-performance, time-resolved single-photon imaging", in *IEEE International Conference on Optical MEMS and Nanophotonics*, IEEE, 2018, pp. 1–5.

[95] K. Morimoto, A. Ardelean, M.-L. Wu, A. C. Ulku, I. M. Antolovic, C. Bruschini, and E. Charbon, "Megapixel time-gated SPAD image sensor for 2D and 3D imaging applications", *OSA Optica*, vol. 7, no. 4, pp. 346–354, 2020.

[96] N. A. Dutton, I. Gyongy, L. Parmesan, S. Gnecchi, N. Calder, B. R. Rae, S. Pellegrini, L. A. Grant, and R. K. Henderson, "A SPAD-based QVGA image sensor for single-photon counting and quanta imaging", *IEEE Transactions on Electron Devices*, vol. 63, no. 1, pp. 189–196, 2015.

[97] G. Mora-Martín, A. Turpin, A. Ruget, A. Halimi, R. Henderson, J. Leach, and I. Gyongy, "High-speed Object Detection using SPAD Sensors", in *Photonic Instrumentation Engineering VIII*, Y. Soskind and L. E. Busse, Eds., International Society for Optics and Photonics, vol. 11693, SPIE, 2021, pp. 73–82.

[98] I. Gyongy, G. Mora-Martín, A. Turpin, A. Ruget, A. Halimi, R. Henderson, and J. Leach, "High-speed Vision with a 3D-stacked SPAD Image Sensor", in *Advanced Photon Counting Techniques XV*, M. A. Itzler, J. C. Bienfang, and K. A. McIntosh, Eds., International Society for Optics and Photonics, vol. 11721, SPIE, 2021, pp. 1–7.

[99] P. Chandramouli, S. Burri, C. Bruschini, E. Charbon, and A. Kolb, "A bit too much? high speed imaging from sparse photon counts", in *2019 IEEE International Conference on Computational Photography (ICCP)*, 2019, pp. 1–9. DOI: 10.1109/ICCPHOT.2019.8747325.

[100] E. R. Fossum, "Some thoughts on future digital still cameras", *Image sensors and signal processing for digital still cameras*, p. 305, 2006.

[101] E. R. Fossum, "Gigapixel digital film sensor (DFS) proposal", *Nanospace Manipulation of Photons and Electrons for Nanovision Systems*, 2005.

[102] E. R. Fossum, "Modeling the performance of single-bit and multi-bit quanta image sensors", *IEEE Journal of the Electron Devices Society*, vol. 1, no. 9, pp. 166–174, 2013.

[103] J. Ma, D. Zhang, O. A. Elgendy, and S. Masoodian, "A 0.19 e-rms Read Noise 16.7 Mpixel Stacked Quanta Image Sensor With 1.1 μm-Pitch Backside Illuminated Pixels", *IEEE Electron Device Letters*, vol. 42, no. 6, pp. 891–894, 2021.

[104] J. Ma, D. Starkey, A. Rao, K. Odame, and E. R. Fossum, "Characterization of quanta image sensor pump-gate jots with deep sub-electron read noise", *IEEE Journal of the Electron Devices Society*, vol. 3, no. 6, pp. 472–480, 2015. DOI: 10.1109/JEDS.2015.2480767.

[105] E. R. Fossum, J. Ma, and S. Masoodian, "Quanta Image Sensor: Concepts and Progress", in *Advanced Photon Counting Techniques X*, M. A. Itzler and J. C. Campbell, Eds., International Society for Optics and Photonics, vol. 9858, SPIE, 2016, pp. 1–14.

[106] A. Gupta, A. Ingle, A. Velten, and M. Gupta, "Photon-flooded single-photon 3d cameras", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 6770–6779.

[107] A. Ingle, A. Velten, and M. Gupta, “High flux passive imaging with single-photon sensors”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 6760–6769.

[108] A. Gupta, A. Ingle, and M. Gupta, “Asynchronous single-photon 3d imaging”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 7909–7918.

[109] S. Ma, S. Gupta, A. C. Ulku, C. Bruschini, E. Charbon, and M. Gupta, “Quanta burst photography”, *ACM Trans. Graph.*, vol. 39, no. 4, Jul. 2020, ISSN: 0730-0301. DOI: 10.1145/3386569.3392470. [Online]. Available: https://doi.org/10.1145/3386569.3392470.

[110] A. Ingle, T. Seets, M. Buttafava, S. Gupta, A. Tosi, M. Gupta, and A. Velten, “Passive inter-photon imaging”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 8585–8595.

[111] F. Yang, Y. M. Lu, L. Sbaiz, and M. Vetterli, “Bits from photons: Oversampled image acquisition using binary poisson statistics”, *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1421–1436, 2012. DOI: 10.1109/TIP.2011.2179306.

[112] S. H. Chan, O. A. Elgendy, and X. Wang, “Images from bits: Non-iterative image reconstruction for quanta image sensors”, *Sensors*, vol. 16, no. 11, 2016, ISSN: 1424-8220. DOI: 10.3390/s16111961. [Online]. Available: https://www.mdpi.com/1424-8220/16/11/1961.

[113] A. Gnanasambandam, O. Elgendy, J. Ma, and S. H. Chan, “Megapixel photon-counting color imaging using quanta image sensor”, *Opt. Express*, vol. 27, no. 12, pp. 17 298–17 310, Jun. 2019. DOI: 10.1364/OE.27.017298. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-27-12-17298.

[114] O. A. Elgendy, A. Gnanasambandam, S. H. Chan, and J. Ma, “Low-light Demosaicking and Denoising for Small Pixels using Learned Frequency Selection”, *IEEE Transactions on Computational Imaging*, vol. 7, pp. 137–150, 2021.

[115] A. Gnanasambandam and S. H. Chan, “HDR Imaging with Quanta Image Sensors: Theoretical Limits and Optimal Reconstruction”, *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1571–1585, 2020.

[116] O. A. Elgendy and S. H. Chan, “Color Filter Arrays for Quanta Image Sensors”, *IEEE Transactions on Computational Imaging*, vol. 6, pp. 652–665, 2020.

[117] I. Gyongy, N. A. Dutton, and R. K. Henderson, “Single-photon tracking for high-speed vision”, *Sensors*, vol. 18, no. 2, 2018, ISSN: 1424-8220. DOI: 10.3390/s18020323. [Online]. Available: https://www.mdpi.com/1424-8220/18/2/323.

[118] Y. Chi, A. Gnanasambandam, V. Koltun, and S. H. Chan, “Dynamic low-light imaging with quanta image sensors”, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 122–138.

[119] D. L. Snyder, C. W. Helstrom, A. D. Lanterman, M. Faisal, and R. L. White, “Compensation for readout noise in ccd images”, *J. Opt. Soc. Am. A*, vol. 12, no. 2, pp. 272–283, Feb. 1995. DOI: 10.1364/JOSAA.12.000272. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-12-2-272.

[120] K. Wei, Y. Fu, J. Yang, and H. Huang, “A physics-based noise formation model for extreme low-light raw denoising”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 2758–2767.

[121] J. Yang, J. Lu, D. Batra, and D. Parikh, “A faster pytorch implementation of faster r-cnn”, *https://github.com/jwyang/faster-rcnn.pytorch*, 2017.

[122] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[123] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A Large-scale Hierarchical Image Database”, in *Proceedings of the IEEE International Conference on Computer Vision and pattern Recognition (CVPR)*, 2009.

[124] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization”, in *International Conference on Learning Representations (ICLR)*, 2014.

[125] X. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections”, in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/0ed9422357395a0d4879191c66f4faa2-Paper.pdf.

[126] N. Oliver and A. Pentland, “Graphical Models for Driver Behavior Recognition in a SmartCar”, in *IV*, 2000.

[127] D. Mitrovic, “Reliable Method for Driving Events Recognition”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 198–205, 2005.

[128] A. Jain, H. Koppula, B. Raghavan, S. Soh, and A. Saxena, “Car that Knows before You Do: Anticipating Maneuvers via Learning Temporal Driving Models”, in *ICCV*, 2015.

[129] T. Gindele, S. Brechtel, and R. Dillmann, “Learning Driver Behavior Models from Traffic Observations for Decision Making and Planning”, *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, 2015.

[130] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human Trajectory Prediction in Crowded Spaces”, in *CVPR*, 2016.

[131] J. Schulz, C. Hubmann, N. Morin, J. Löchner, and D. Burschka, “Learning Interaction-Aware Probabilistic Driver Behavior Models from Urban Scenarios”, in *IV*, 2019.

[132] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, “TraPHic: Trajectory Prediction in Dense and Heterogeneous Traffic using Weighted Interactions”, in *CVPR*, 2019.

[133] A. Vemula, K. Muelling, and J. Oh, “Social Attention: Modeling Attention in Human Crowds”, in *ICRA*, 2018.

[134] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, “Crowd-Robot Interaction: Crowd-aware Robot Navigation with Attention-based Deep Reinforcement Learning”, in *ICRA*, 2019.

[135] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu, “Learning Actor Relation Graphs for Group Activity Recognition”, in *CVPR*, 2019.

[136] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, “Toward Driving Scene Understanding: A Dataset for Learning Driver Behavior and Causal Reasoning”, in *CVPR*, 2018.

[137] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, “A Driver Behavior Recognition Method Based on a Driver Model Framework”, in *SAE 2000 World Congress*, 2000.

[138] B.-F. Wu, Y.-H. Chen, C. Yeh, and Y.-F. Li, “Reasoning Based Framework for Driving Safety Monitoring using Driving Event Eecognition”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1231–1241, 2013.

[139] M. Xu, M. Gao, Y.-T. Chen, L. Davis, and D. Crandall, “Temporal Recurrent Networks for Online Action Detection”, in *ICCV*, 2019.

[140] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated Graph Sequence Neural Networks”, in *ICLR*, 2016.

[141] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler, “Situation Recognition with Graph Neural Networks”, in *ICCV*, 2017.

[142] S. Yan, Y. Xiong, and D. Lin, “Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition”, in *AAAI*, 2018.

[143] X. Wang and A. Gupta, “Videos as Space-Time Region Graphs”, in *ECCV*, 2018.

[144] J. Yang, J. Lu, S. Lee, D. Batra1, and D. Parikh, “Graph R-CNN for Scene Graph Generation”, in *ECCV*, 2018.

[145] R. Herzig, E. Levi, H. Xu, H. Gao, E. Brosh, X. Wang, A. Globerson, and T. Darrell, “Spatio-Temporal Action Graph Networks”, in *ICCVW*, 2019.

[146] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer”, 2020.

[147] M. Müller, A. Dosovitskiy, B. Ghanem, and V. Koltun, “Driving Policy Transfer via Modularity and Abstraction”, in *CoRL*, 2018.

[148] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”, in *CVPR*, 2017.

[149] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer N ormalization”, in *arXiv preprint arXiv:1607.06450*, 2016.

[150] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, “Temporal relational reasoning in videos”, in *ECCV*, 2018.

[151] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision”, in *CVPR*, 2016.

[152] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, “The Kinetics Human Action Video Dataset”, in *arXiv preprint arXiv:1705.06950*, 2017.

[153] A. Narayanan, Y.-T. Chen, and S. Malla, “Semi-supervised Learning: Fusion of Self-supervised, Supervised Learning, and Multimodal Cues for Tactical Driver Behavior Detection”, in *CVPRW*, 2018.

[154] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri1, “Learning Spatiotemporal Features with 3D Convolutional Networks”, in *ICCV*, 2015.

[155] S. Lefèvre, D. Vasquez, and C. Laugier, “A Survey on Motion Prediction and Risk Assessment for Intelligent Vehicles”, *ROBOMECH Journal*, vol. 1, p. 1, 2014.

[156] S. Alletto, A. Palazzi, F. Solera, S. Calderara, and R. Cucchiara, “DR(eye)VE: A Dataset for Attention-based Tasks with Applications to Autonomous and Assisted Driving”, in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2016.

[157] M. Gao, A. Tawari, and S. Martin, “Goal-oriented Object Importance Estimation in On-road Driving Videos”, in *IEEE International Conference on Robotics and Automation*, 2019.

[158] K.-H. Zeng, S.-H. Chou, F.-H. Chan, J. C. Niebles, and M. Sun, “Agent-Centric Risk Assessment: Accident Anticipation and Risky Region Localization”, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[159] Y. Xia, D. Zhang, J. Kim, and D. W. Ken Nakayama Karl Zipser, “Predicting Driver Attention in Critical Situations”, in *Asian Conference on Computer Vision*, 2018.

[160] A. Tawari, P. Mallela, and S. Martin, “Learning to Attend to Salient Targets in Driving Videos using Fully Convolutional RNN”, in *IEEE Intelligent Transportation Systems Conference*, 2018.

[161] B. Tatler, M. Hayhoe, M. Land, and D. B. lard, “Eye Guidance in Natural Vision: Reinterpreting Salience”, *Journal of Vision*, vol. 11, p. 5, 2011.

[162] P. de Haan, D. Jayaraman, and S. Levine, “Causal Confusion in Imitation Learning”, in *IEEE Conference on Neural Information Processing Systems*, 2019.

[163] J. Pearl, “Causality”, *Cambridge University Press*, 2009.

[164] J. Ibanez-Guzman, S. Lefevre, A. Mokkadem, and S. Rodhaim, “Vehicle to vehicle communications applied to road intersection safety, field results”, in *13th International IEEE Conference on Intelligent Transportation Systems*, IEEE, 2010, pp. 192–197.

[165] S. Worrall, D. Orchansky, F. Masson, and E. Nebot, “Improving Vehicle Safety using Context based Detection of Risk”, in *IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2010, pp. 379–385.

[166] A. Chinea and M. Parent, “Risk Assessment Algorithms based on Recursive Neural Networks”, in *2007 International Joint Conference on Neural Networks*, IEEE, 2007, pp. 1434–1440.

[167] F. D. Salim, S. W. Loke, A. Rakotonirainy, B. Srinivasan, and S. Krishnaswamy, "Collision Pattern Modeling and Real-time Collision Detection at Road Intersections", in *2007 IEEE Intelligent Transportation Systems Conference*, IEEE, 2007, pp. 161–166.

[168] M. Althoff and A. Mergel, "Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars", *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1237–1247, 2011.

[169] D. Greene, J. Liu, J. Reich, Y. Hirokawa, A. Shinagawa, H. Ito, and T. Mikami, "An Efficient Computational Architecture for a Collision Early-warning System for Vehicles, Pedestrians, and Bicyclists", *IEEE Transactions on intelligent transportation systems*, vol. 12, no. 4, pp. 942–953, 2011.

[170] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss, "Interactive Scene Prediction for Automotive Applications", in *2013 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2013, pp. 1028–1033.

[171] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmàn, "Evaluating Risk at Road Intersections by Detecting Conflicting Intentions", in *International Conference on Intelligent Robots and Systems*, 2012.

[172] A. Alahi, V. Ramanathan, and L. Fei-Fei, "Socially-aware Large-scale Crowd Forecasting", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[173] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[174] Y. Xu, Z. Piao, and S. Gao, "Encoding Crowd Interaction with Deep Neural Network for Pedestrian Trajectory Prediction", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[175] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the Limitations of Behavior Cloning for Autonomous Driving", 2019.

[176] J. Pearl, M. Glymour, and N. P. Jewell, *Causal Inference in Statistics: A Primer.* John Wiley & Sons, 2016.

[177] S. Nair, Y. Zhu, S. Savarese, and L. Fei-Fei, "Causal Induction from Visual Observations for Goal Directed Tasks", in *NeurIPS 2019 Workshop on Causal Machine Learning*, 2019.

[178] Y. Li, A. Torralba, D. F. Animashree Anandkumar, and A. Garg, “Causal Discovery in Physical Systems from Videos”, in *Conference on Neural Information Processing Systems*, 2020.

[179] J. Qi, Y. Niu, J. Huang, and H. Zhang, “Two Causal Principles for Improving Visual Dialog”, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[180] T. Wang, J. Huang, H. Zhang, and Q. Sun, “Visual Commonsense R-CNN”, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[181] X. Yang, H. Zhang, and J. Cai, “Deconfounded Image Captioning: A Causal Retrospect”, in *arXiv preprint arXiv:2003.03923*, 2020.

[182] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah, and A. Kendall, “Urban Driving with Conditional Imitation Learning”, in *arXiv preprint arXiv:1912.00177*, 2019.

[183] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric”, in *2017 IEEE International Conference on Image Processing*, 2017.

[184] S. Hochreiter and J. Schmidhuber, “Long Short-term Memory”, *Neural computation*, 1997.

[185] A. Rahimpour, S. Martin, A. Tawari, and H. Qi, “Context Aware Road-user Importance Estimation (iCARE)”, in *IEEE Intelligent Vehicles Symposium*, 2019.

[186] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmàn, “Intention-aware Risk Estimation for General Traffic Situations, and Application to Intersection Safety”, 2013.

[187] G. Liu, a. K. J. S. Fitsum A. Reda, T.-C. Wang, A. Tao, and B. Catanzaro, “Image Inpainting for Irregular Holes using Partial Convolutions”, in *IEEE European Conference on Computer Vision*, 2018.

[188] G. Liu, K. J. Shih, T.-C. Wang, F. A. Reda, K. Sapra, Z. Yu, A. Tao, and B. Catanzaro, “Partial Convolution based Padding”, *arXiv preprint arXiv:1811.11718*, 2018.

[189] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, “Online Action Detection”, in *IEEE European Conference on Computer Vision*, 2016.

[190] Z. Zhang, C. Yu, and D. Crandall, “A Self Validation Network for Object-Level Human Attention Estimation”, in *IEEE Conference on Neural Information Processing Systems*, 2019.

[191] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alem, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”, in *AAAI*, 2017.

[192] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, *Detectron*, https://github.com/facebookresearch/detectron, 2018.

[193] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *JMLR*, 2014.

VITA

CHENGXI LI

EDUCATION

Purdue University, West Lafayette, IN, USA *Aug. 2016 - Present*

PhD Student in Department of Electrical and Computer Engineering **GPA:4.0/4.0**

Research Interests: Computer Vision, Image Processing and Machine Learning

Fudan University, Shanghai, China *Sep. 2012 - Jun. 2016*

B.S. in Electrical and Electronics Engineering **GPA:3.76/4.0, rank: 1/104**

National University of Singapore, Singapore *Jan. 2015 - May. 2015*

Exchange student in Electrical and Computer Engineering **GPA:5.0/5.0**

PUBLICATIONS

C. Li, Y. Meng, S. Chan and Y. Chen,"Learning 3D-aware Egocentric Spatial-Temporal Interaction via Graph Convolutional Networks," in *IEEE International Conference on Robotics and Automation (ICRA), 2020.*

C. Li, S. Chan and Y. Chen,"Who Make Drivers Stop? Towards Driver-centric Risk Assessment: Risk Object Identification via Causal Inference," in *IEEE International Conference on Intelligent Robots and Systems (IROS), 2020.*

C. Li, X. Qu, A. Gnanasambandam, O. Elgendy, J. Ma and S. Chan,"Photon-Limited Object Detection using Non-local Feature Matching and Knowledge Distillation," in *IEEE International Conference on Computer Vision (ICCV) Workshop, 2021.*

C. Li, S. Chan and Y. Chen,"Driver-centric Risk Object Identification," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI), 2021.*

PROFESSIONAL EXPERIENCE

Machine Learning Engineer Intern, Facebook, Menlo Park, CA, USA *Aug. 2020 - Nov. 2020*

Mentor: **Dr. Enming Luo**

Worked on imbalanced data problem and enhance machine learning model performance through data curation.

Research Intern, Honda Research Institute, San Jose, CA, USA *May. 2019 - Aug.2020*

Mentor: **Dr. Yi-Ting Chen**

Project 1: Risk Perception Modeling in Driving Scene

- Proposed a two-stage framework based on causal inference for risk object identification.
- Evaluated the proposed framework on the Honda Research Institute Driving Dataset (HDD) and demonstrated a substantial average performance boost over a strong baseline by 7.5%.

Project 2: Video Representation for Egocentric Activity Recognition

- Proposed a 3D-aware egocentric spatial-temporal interaction framework for automated driving.
- Validated the proposed framework on tactical driver behavior recognition using Honda Research Institute Driving Dataset (HDD), boosting over baselines on two experimental settings by 3.9% and 6.0%, respectively and published in ICRA 2020 as first author.

TEACHING EXPERIENCE

Teaching Assistant, Purdue University, West Lafayette, IN, USA *Jan. 2019 - May. 2019*

Instructor: Prof. Mary Comer

Course: Probabilistic Methods in Electrical and Computer Engineering

Teaching Assistant, Purdue University, West Lafayette, IN, USA *Jan. 2021 - May. 2021*

Instructor: Prof. Stanley H. Chan

Course: Machine Learning

TECHNICAL SKILLS

Programming Languages	Python, Matlab, C++, C
Deep Learning Tools	PyTorch, Tensorflow, Torch, Caffe
Softwares	OpenCV, LaTeX, OrCAD, Xilinx ISE, Altium Designer