Leveraging Multi-Stream Information Fusion for Trajectory Prediction in Low-Illumination Scenarios: A Multi-Channel Graph Convolutional Approach

Hailong Gong, Zirui Li, *Graduate Student Member, IEEE*, Chao Lu, *Member, IEEE*, Guodong Du, and Jianwei Gong, *Member, IEEE*

***Abstract*— Trajectory prediction is a fundamental problem and challenge for autonomous vehicles. Early works mainly focused on designing complicated architectures for deep-learning-based prediction models in normal-illumination environments, which fail in dealing with low-light conditions. The paper proposes a novel approach for trajectory prediction in low-illumination scenarios by leveraging multi-stream information fusion, which integrates image, optical flow, and object trajectory information. This is achieved by applying Convolutional Neural Network-based (CNN) Long Short-term Memory (LSTM) networks to extract temporal information from the image channel, Spatial-Temporal Graph Convolutional Network (ST-GCN) to model relative motion between adjacent camera frames through the optical flow channel, and recognizing high-level interactions between vehicles in the trajectory channel. Further, to investigate the reliability of the model in low-illumination scenarios, epistemic uncertainty estimation is conducted by applying Monte Carlo Dropout. The proposed approach is validated on HEV-I and newly generated Dark-HEV-I datasets focusing on graph-based interaction understanding and low illumination conditions. The experimental results show improved performance compared to baselines in both standard and low-illumination scenarios. Importantly, our approach is generic and applicable to scenarios with different types of perception data. The source code is available at https://github.com/TommyGong08/MSIF.**

***Index Terms*— Autonomous driving, trajectory prediction, low illumination scenarios, information fusion, graph convolutional network.**

I. Introduction

With the rapid development of autonomous driving, it has become apparent that ensuring the safety of autonomous systems in traffic scenarios is necessary for the widespread adoption of autonomous driving [1]. For self-driving cars to have driving capabilities comparable to those of human drivers, it is essential to understand the state of surrounding vehicles and predict their trajectories [2].

A. Motivation

Trajectory prediction in autonomous driving has been the focus of numerous studies in recent years, with researchers primarily investigating scenarios in normal-illumination environments using standard light conditions. However, these state-of-the-art approaches have proven inadequate for low-light conditions, rendering them unsuitable for nighttime use [3]. In low illumination scenarios, autonomous driving systems face a slew of challenges. First and foremost, limited perception capability is a significant concern, as the illumination level directly affects sensor performance. The ability to detect and recognize obstacles is also compromised in low-light conditions, as is the visibility of road conditions, which may be obscured by shadows or other factors. Furthermore, changes in illumination levels can have a significant impact on the accuracy of trajectory predictions. According to the National Highway Traffic Safety Administration's survey [4], fatal traffic accidents at night account for 51% of all such incidents in the United States, particularly in rural areas with extremely low illumination. As a result, accurate trajectory prediction in low-illumination scenarios is critical for traffic safety and reducing the number of fatalities and injuries resulting from nighttime accidents. To achieve this goal, advanced algorithms must be developed that can effectively operate in low-light conditions while accounting for the various challenges posed by such environments.

B. Related Works

Numerous types of research have proposed various methods for predicting trajectories. Physics-based methods employ the dynamics or kinematics models of vehicles [5]. In most cases, a simple physics model is preferred because complex physics models provide only marginal improvements in predictive accuracy. Kalman filtering is popularly applied in physics-based methods. Reference [6] models the noise of the current state of vehicles using Kalman filtering techniques. Based on

Manuscript received 2 November 2022; revised 11 August 2023; accepted 20 October 2023. Date of publication 7 November 2023; date of current version 13 May 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 52272411 and Grant 61703041 and in part by the China Scholarship Council (CSC). The Associate Editor for this article was J. W. Choi. *(Hailong Gong and Zirui Li contributed equally to this work.) (Corresponding author: Chao Lu.)*

Hailong Gong, Chao Lu, and Jianwei Gong are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: gonghailong2021@163.com; chaolu@bit.edu.cn; gongjianwei@bit.edu.cn).

Zirui Li is with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China, and also with the Chair of Traffic Process Automation, "Friedrich List" Faculty of Transport and Traffic Sciences, TU Dresden, 01069 Dresden, Germany (e-mail: z.li@bit.edu.cn).

Guodong Du is with the Institute of Dynamic System and Control, ETH Zürich, 8092 Zürich, Switzerland (e-mail: guoddu@ethz.ch).

Digital Object Identifier 10.1109/TITS.2023.3328294

the Gaussian Mixture Model(GMM), [7] predicts the lookahead distance for autonomous vehicles and [8] successfully recognizes the braking intensity levels of drivers. Combining vehicle-to-vehicle (V2V) communication and the Kalman filter, [9] predicts ego-vehicle trajectories to avoid obstacles. However, the accuracy of physics-based methods is heavily dependent on the description of physics models. If dynamic models of vehicles change, physics-based methods can only provide short-term predictions.

Recently, the most popular methods for predicting trajectories are based on deep learning, as they can effectively integrate physical constraints, interactions, and scene understanding [5]. To solve the problem of insufficient data and improve modeling efficiency, transfer learning is used in driver behavior modeling [10], especially in the lane-changing scenario [11], [12], [13], [14]. Reference [15] extracts features of spatial interactions with attention mechanism and employs the LSTM network to determine their temporal dependence. Reference [16] presents a social generative adversarial network (GAN) model that focuses on the normalization and rationality of trajectories. Reference [17] introduces Graph Parsing Neural Network (GPNN) to learn human-object interactions in videos and images. To effectively capture the social behaviors of relevant pedestrians, a graph neural network is implemented in social behavior modeling and trajectory prediction based on their timely location and speed direction [18], [19]. Graph neural network employs graphs to represent traffic scenarios in which nodes represent vehicles and edges represent the degree of interaction between vehicles [20], [21], [22], [23]. For capturing the spatial and temporal information during the interaction, [21] constructs the spatial-temporal graph model, in which the temporal graph extracts personal information and the spatial graph extracts pedestrian interaction information. To encode short-term data in autonomous driving scenarios, GMPNet [24] considers the grouped points as a node and uses a k-NN graph to make graph embedding. Reference [25] preserves the spatial information through fully connected GNNs and also effectively captures the relationship between two images via Attentive Graph Neural Networks (AGNN). The Social-STGCNN model proposed by [23] employs a graph convolutional neural network to embed the spatial-temporal graph and a time extrapolator to determine trajectories.

However, the physic model-based and deep learning-based trajectory prediction methods mainly focus on normal driving conditions. As illumination conditions change throughout the day, it is necessary for self-driving vehicles to make image enhancement and extract scene features in low-brightness environments [3]. Reference [26] develops a fusion-based enhancing method for weakly illuminated images. Reference [27] proposes a dataset with low-light images, finding that the effects of low-light reach far deeper into the features than can be solved by simple "illumination invariance". Reference [3] develops an image enhancement approach for autonomous driving at night. Optical flow information is introduced to ensure the consistency of transformed brightness or to realize optical flow tracking in response to the difficulty posed by low-illumination conditions. Reference [28] uses optical flow to improve image quality in low-light conditions. Utilizing dense optical flow, [29] encodes motion between consecutive frames and achieves visual emotion recognition in low resolution and poor illumination. Reference [30] integrates optical flow and LiDAR perception data for moving object detection in autonomous driving under low-light conditions. The above works have investigated image enhancement and object detection in low-light conditions of autonomous driving, but little research has explored trajectory prediction in the low-illumination environment.

A series of trajectory prediction approaches, such as physics-based, deep learning methods, have achieved state-of-the-art performance for normal driving conditions, but for complex autonomous driving traffic scenarios, environmental conditions are not constant. Autonomous vehicles can be exposed to extreme conditions such as low and strong lights. Moreover, some research work related to computer vision on low-light conditions seldom pays attention to the trajectory prediction problem [28], [29], [30], making it difficult to solve the trajectory prediction problem for extreme conditions.

C. Contributions

To overcome the detrimental effects of low-light conditions on autonomous driving, especially in the trajectory problem, this research proposes a multi-stream information (heterogeneous data) fusion-based method, MSIF, for trajectory prediction in low-illumination scenarios. The proposed method combines trajectories, optical flow, and image information, which ensures adaptability to various luminance levels and especially overcomes low-illumination conditions. To model the interaction in low-brightness conditions for trajectory prediction, the graph convolutional neural network (GCN) is applied to represent spatial-temporal features of trajectories [23] and the instantaneous speed of surrounding vehicles (from optical flow). Meanwhile, local spatial differences are identified using a novel recurrent-based image feature extraction technique [31]. To simulate realistic low-light driving conditions, the Dark-HEV-I dataset is derived from the HEV-I dataset by adjusting the image brightness. The main contributions of this paper are summarized as follows:

1) : A novel trajectory prediction method is proposed for low-illumination conditions by leveraging multi-stream information fusion, which flexibly integrates image, optical flow, and object trajectory information. The proposed method designs the ST-GCN-based method for temporal and spatial information representation and incorporates a novel multi-stream information fusion mechanism into its architecture.

2) : To simulate low-illumination driving conditions and evaluate the effectiveness of the proposed method, the Dark-HEV-I dataset is derived from the HEV-I dataset with the same scale. In Dark-HEV-I dataset, the low-illumination images are generated by adjusting the exposure of the original images, and optical flow is produced by using the low-illuminated images. Experimental results demonstrate that the proposed method could maintain high performance in the Dark-HEV-I dataset.

3) : To address the inherent challenges encountered in extreme conditions, such as low illumination scenarios, this

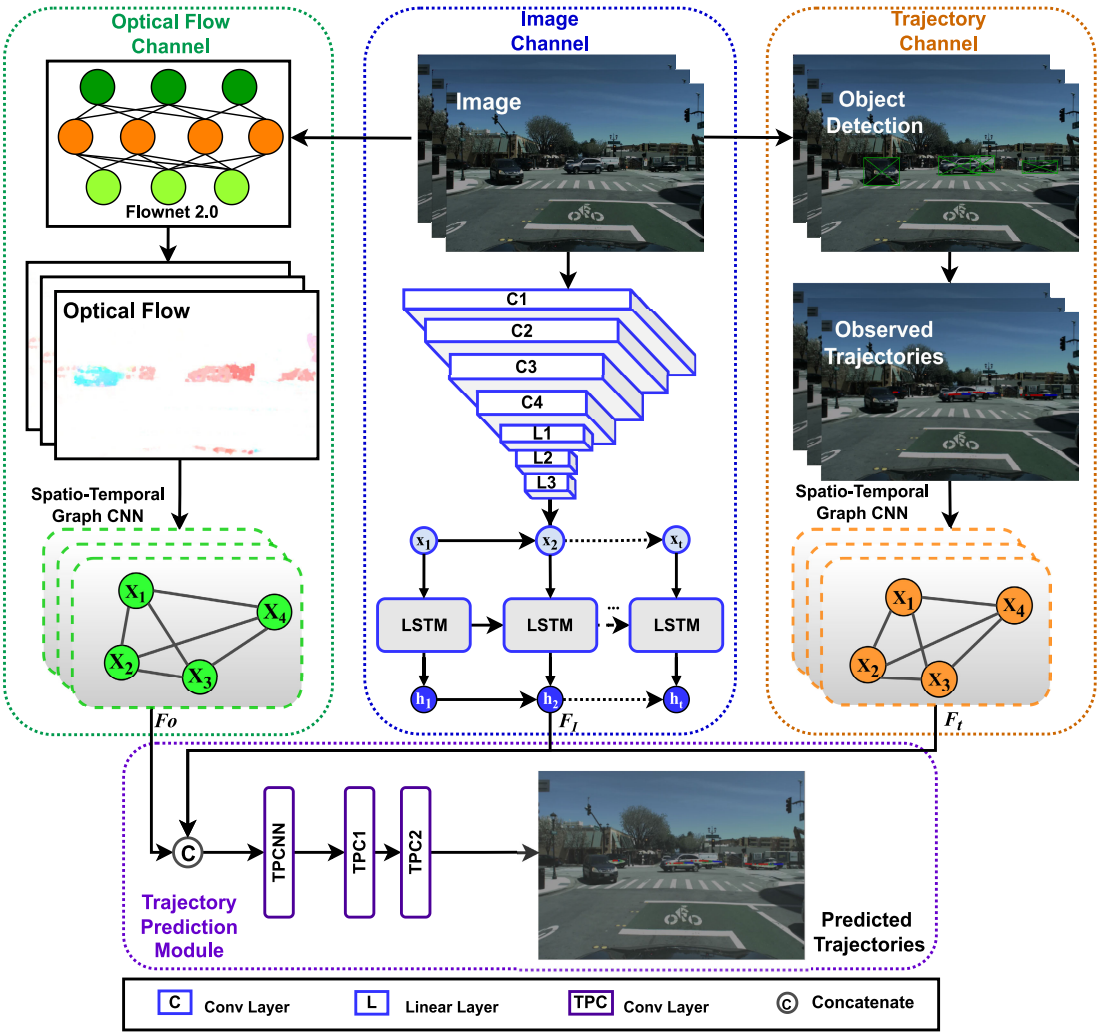

Fig. 1. The model consists of three channels: optical flow, image, and trajectory (from top left to top right). The output of this model is the distribution of predicted trajectories. The proposed approach implements CNN and LSTM layers for feature extraction and scene understanding for image information. A spatial-temporal graph convolutional neural network is used for feature extraction for optical flow and trajectory information. All the extracted features will be concatenated and transferred into the trajectory prediction module. Considering the weight and efficiency of the model, a convolution neural network is adopted in the future trajectories prediction module.

study also focuses on the estimation of epistemic uncertainty. In this pursuit, we employ the technique of sample-based Monte Carlo dropout to approximate the complex and elusive distribution. besides, an epistemic uncertainty score is defined, which serves to partially alleviate the issue of limited interpretability commonly associated with deep learning models.

D. Outline

This paper is organized as follows. In section II-A, the formulation of the multi-stream trajectory prediction is detailed. Section II begins with a description of the proposed method, followed by the formulation of graph representation, image feature extraction, and information fusion. Section III shows the HEV-I dataset, the newly generated Dark-HEV-I dataset, implementation details and experimental results. Finally, the conclusion is presented in Section IV.

II. Methodology

This section begins with the formulation of the trajectory prediction problem, including model inputs, outputs, data flow, and structural characteristics, as shown in Fig. 1. Then, the principle and functions of the three channels in the proposed model are described: the optical flow channel, the trajectory channel, and the image channel. Finally, the last subsection introduces the trajectory prediction module, specifically describing information fusion methods.

A. Problem Formulation

As shown in Fig. 1, a novel method involving an image channel, optical flow channel, trajectory channel, and the fusion module of trajectory prediction is proposed to predict trajectories of moving objects in the field of view. This study assumes that an autonomous driving vehicle can obtain heterogeneous data from its sensors. At each time step, at least the front-view image is provided. The speed and direction of objects in motion are not provided. As front-view camera images are commonly used in autonomous vehicles, this study assumes that the optical flow can be generated from the original images. Historical trajectories of objects in the scenario are obtained through object detection and trajectory

tracking. Specifically, the model's input at time t is defined as follows:

$$\boldsymbol{S}_t = \{\boldsymbol{I}_t, \boldsymbol{O}_t, \boldsymbol{X}_t\} \tag{1}$$

$$\boldsymbol{I}_t = [\boldsymbol{M}_{t-t_{obs}}, \ldots, \boldsymbol{M}_{t-1}, \boldsymbol{M}_t] \tag{2}$$

where $\boldsymbol{I}_t$ represents the image sequence captured by the front-view camera on the automatic vehicle. $\boldsymbol{O}_t$ and $\boldsymbol{X}_t$ denote the optical flow sequence and the sequence of the set formed by objects' trajectories in each frame, respectively, where

$$\boldsymbol{O}_t = [\boldsymbol{J}_{t-t_{obs}}, \ldots, \boldsymbol{J}_{t-1}, \boldsymbol{J}_t] \tag{3}$$

$$\boldsymbol{X}_t = [\boldsymbol{P}_{t-t_{obs}}, \ldots, \boldsymbol{P}_{t-1}, \boldsymbol{P}_t] \tag{4}$$

where $\boldsymbol{P}_t$ is the set of the objects' trajectories at time t, $\boldsymbol{P}_t = \{(x_t^i, y_t^i) \mid i = 0, 1, \ldots, n\}$, where n is a variable parameter due to the change of the number of the objects in the view at different times.

Furthermore, the trajectory and optical flow information will be represented by graphs, in which each node stands for a vehicle. At each time step t, the information of each $\boldsymbol{M}_t$ is abstracted into a graph $G_t = (V_t, A_t)$, in which $V = \{v^i \mid \forall i \in 1, 2, \ldots, N\}$. N is the total number of objects that appear in that sequence. V_t represents the node, and attributes of v_i are the coordinate of each object in pixel coordinates, denoted as $v_i = (x_i, y_i)$. A_t represents the adjacency matrix and $A_t = \{a_t^{i,j} \mid \forall i, j \in 1, 2, \ldots, N\}$. For the same sequence, A_t weights vertices' contributions to each in the convolution operation. Thus, a kernel function can be considered as prior knowledge about the interactive degree between vehicles as it maps attributes at v_i and v_j at time step t to the value $a_t^{i,j}$. As marked in Fig. 2, the spatial feature forwarded from node i to node j at time step t is denoted as $h_t^{i,j}$, used in graph convolution. During the graph convolution, spatial features will be taken part in the operation. Given the same definitions of the sampling function and weight function in [21], the graph spatial convolution is formulated as:

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{tj}) \cdot \boldsymbol{W}(l_{ti}(v_{tj})) \tag{5}$$

where $B(v_{ti})$ represents the neighbor set of v_{ti}, and $Z_{ti}(v_{tj})$ is the normalizing term.

Predicted trajectories are assumed to follow the bi-variate Gaussian distribution, which is estimated to $\Omega(\hat{\mu}, \hat{\sigma}$, and $\hat{\rho})$. The output of the model at time t is defined as $\boldsymbol{R}_t = [\boldsymbol{P}_{t+1}, \boldsymbol{P}_{t+2}, \ldots, \boldsymbol{P}_{t+t_{pred}}]$. Defining parameters of the i-th object's bi-variate Gaussian distribution at the moment t as μ_t^i,σ_t^i, ρ_t^i, the output can be formulated as:

$$\boldsymbol{\Psi}(\boldsymbol{R}_t^i \mid \boldsymbol{S}_t) \sim \Omega(\hat{\mu}_t^i, \hat{\sigma}_t^i, \hat{\rho}_t^i), i = 1, 2, \ldots, n \tag{6}$$

The ground truth of predicted trajectories is denoted as $Y_{t+1:t+t_{pred}}$. The goal of the proposed approach is to precisely map observations to predictions, which can be formulated as:

$$\arg\min_{f \in \boldsymbol{F}} \boldsymbol{L}(Y_{t+1:t+t_{\text{pred}}}, f(S_t)) \tag{7}$$

where $\boldsymbol{F}$ is the model set in the training process, f represents the model, and $\boldsymbol{L}$ is the measure of prediction error. The proposed method considers the Negatively Log-Likelihood

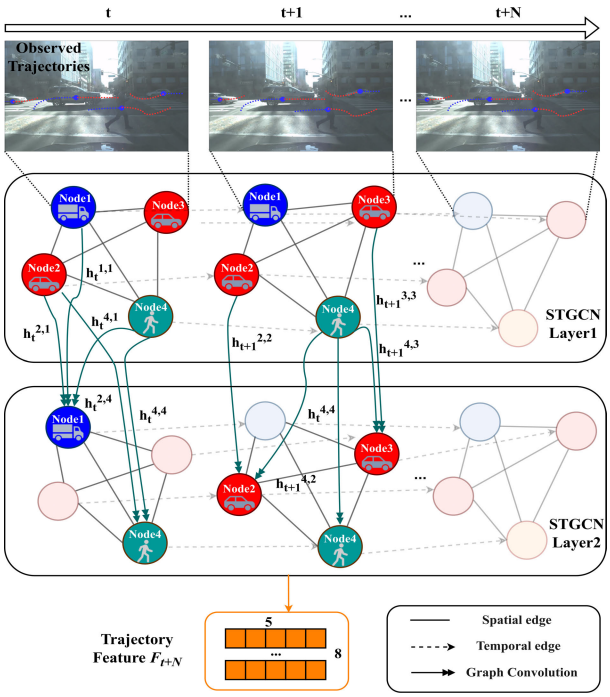

Fig. 2. This figure presents the detailed process of the Trajectory channel, which provides graph embedding. The spatial feature forward from node i to node j at time step t is denoted as $h_t^{i,j}$ used in graph convolution. During the graph convolution, the spatial features will be taken part in the operation. The trajectory change finally outputs the trajectory features which will be fused in TPM as shown in Fig.1.

Loss-based (NLL) function. Regarding the predicted bi-variate gaussian distribution trajectory, the loss function is computed by the following formula:

$$\boldsymbol{L} = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp(-\frac{1}{2(1-\rho^2)} [\frac{(x-\mu_X)^2}{\rho_X^2} + \frac{(y-\mu_Y)^2}{\rho_Y^2} - \frac{2\rho(x-\mu_X)(y-\mu_Y)}{\rho_X\rho_Y}]) \tag{8}$$

where $\mu_X, \mu_Y, \sigma_X, \sigma_Y, \rho$ are the components of the bi-variate gaussian distribution, and the x, y represent the ground truth of the trajectory points.

B. Multi-Stream Information Fusion Framework

The low illumination environment poses the challenge for trajectory prediction, which threatens the safety of autonomous vehicles. Due to the insufficient light and inadequate understanding of the scenario, it is inconsiderate to merely use trajectory information for prediction. For trajectory prediction in the low illumination scenarios, it is intuitive that the prediction method should make an image enhancement and capture the information about vehicles in motion. Therefore, our approach innovatively utilizes optical flow and image information for trajectory prediction besides trajectory information.

As shown in Fig. 1, the MSIF method combines front-view image streams, optical flow, and trajectories from object detection using three input channels: 1) Optical channel, 2) Image channel, and 3) Trajectory channel. The model's inputs consist

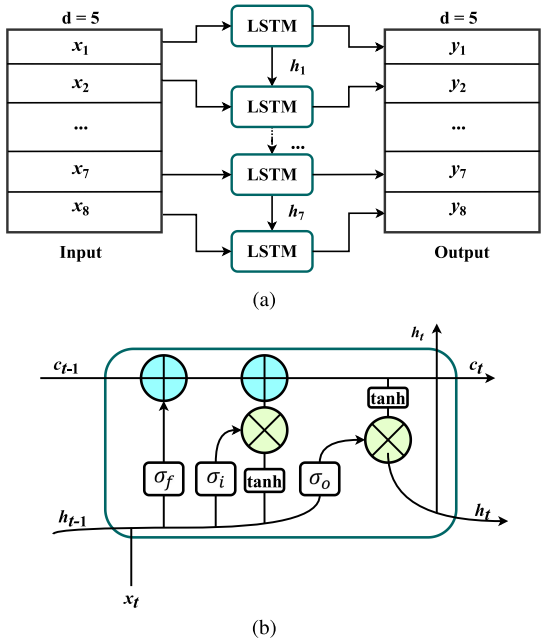

Fig. 3. The subfigure (a) shows the working principle of LSTM layer, the subfigure (b) shows the working the principle of LSTM cell. LSTM is a network with a long-term memory function consisting of forgetting gate, input data, and output date, as shown in Fig.3(b).

of the pixel matrices of the image stream, the optical flow, and the trajectories, where the image stream is the original data, and the optical flow and trajectories are generated from the image stream.

For the image channel, after obtaining the images of the front-view camera, the approach first resizes the images. It implements the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers to extract the features of the resized image. Several optical flow generation techniques, such as [32] and [33], have been developed for the optical flow channel. Flownet 2.0 generates optical flow in this framework [32]. Then, the optical flow channel distinguishes moving objects from background and enhances the image channel by leveraging a spatial-temporal graph convolutional neural network. For the trajectory channel, the target detection algorithm takes original images as inputs and generates trajectories of the geometric center of the bounding boxes. Trajectories are served as inputs of a spatial-temporal graph convolutional neural network for interactive behavior modeling in the trajectory channel. Finally, the trajectory prediction module accepts extracted features from the three channels mentioned above, where the features are fused and used to generate predicted trajectories. Considering the weight and efficiency of the model, a convolution neural network is utilized in the future trajectory prediction module.

The output of this model is the bivariate Gaussian distribution of the predicted trajectories. It must be emphasized that all trajectory coordinates in this paper are in the pixel coordinate system.

C. Graph Representation

This subsection describes the reason and process for the graph representation of the optical flow and trajectories. As stated in the previous section, the model for predicting trajectories based on a single trajectory cannot solve the randomness and mutability of trajectories. To accurately predict multivariate trajectories, it is necessary to consider incorporating additional heterogeneous data into models of interaction behavior. Considering that no matter what scenario the vehicle is in, it is more concerned with moving objects. Thus the proposed method introduces data that can reflect the velocity of object motion - optical flow. The approach combines two types of heterogeneous data, optical flow, and trajectory, which respond to partial information, to model the interactive behavior accurately. In addition, previous research has demonstrated that spatial-temporal graph convolutional networks (ST-GCN) can effectively model social behaviors; therefore, this approach employs ST-GCN for graph embedding of optical flow and trajectories. The upper left part of Fig. 1 depicts the optical flow generation process. Previous researchers have invented various algorithms for optical flow generation [33], [34]. The paper [35] presents a novel two-frame motion estimation algorithm based on polynomial expansion transform. Reference [36] uses optical flow to represent motion and predicts dynamic visual salience by combining spatial and temporal features. To obtain precise motion estimation, the optical flow channel implements the architecture of Flownet 2.0, which provides pixel-level motion between consecutive frames, accurately reflecting the instantaneous speed of objects in the view [32]. Flownet 2.0 consists of a feature extraction network and a flow regression network. The feature network takes two consecutive frames as input and extracts a set of feature maps that capture the spatial and temporal information of the images. The flow regression network infers the corresponding optical flow vectors using these feature maps. Under the assumption of small movement, spatial coherence [37], and brightness constancy, the optical flow could be computed as follows:

$$\frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial y}\Delta y + \frac{\partial G}{\partial t}\Delta t = 0 \tag{9}$$

$$\frac{\partial G}{\partial x}\Delta V_x + \frac{\partial G}{\partial y}\Delta V_y + \frac{\partial G}{\partial t} = 0 \tag{10}$$

where G is the grayscale image, x and y are pixel coordinates, and t represents the time index. V_x and V_y are the velocities of the pixel (x,y) in the X and Y direction, respectively. The brighter the image color is, the faster the object moves. This method emphasizes the relationship between the local features of optical flow, i.e., the relationship behind the pixels with brighter colors. To learn this relationship, the proposed method employs the ST-GCN layers for embedding optical flow graphs, efficiently reducing computational complexity.

Another type of perception data utilized in this method is the trajectory. As shown in the upper right part of Fig. 1, the target detection module takes the original images as inputs, identifies the position of vehicles in the image, and outputs the coordinates of the top left and bottom right vertexes of the detection result bounding boxes. For i-th object in the image, the top left vertex is denoted as $p_i^{tl} = (x_i^{tl}, y_i^{tl})$, and the bottom right vertex is denoted as $p_i^{br} = (x_i^{br}, y_i^{br})$, tl stands for top left while br stands for bottom right. The sequence of points P_i constitutes a complete trajectory, where $P_i = (x_i, y_i)$. Thus,

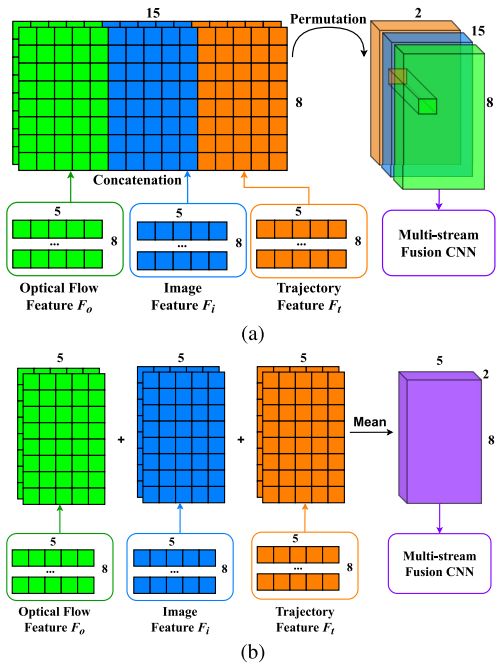

Fig. 4. The figures present two methods of feature fusion. (a) shows the stitching fusion operation (concatenation) of image feature F_i, optical flow feature F_o, and trajectory feature F_t. (b) shows the isotopic fusion operation (calculating arithmetic mean).

the geometric center coordinates of the bounding box can be calculated as:

$$\begin{cases} x_i = \dfrac{x_i^{br} + x_i^{tr}}{2} \\ y_i = \dfrac{y_i^{br} + y_i^{tr}}{2} \end{cases} \tag{11}$$

The optical flow and trajectory information reflect the speed and direction of vehicles, respectively. To characterize their interactions in low illumination conditions, the proposed approach implements ST-GCN for graph lrepresentation. The method defines a new graph G with attributes corresponding to the set of attributes G_t. In this study, the novel kernel function [23] is adopted within the graph representing, and formulated as:

$$a_t^{i,j} = \frac{1}{\|v_t^i - v_t^j\|_2 + \epsilon} \tag{12}$$

where $V = \{v^i \mid \forall i \in 1, 2, \ldots, N\}$ is the set of vertices of the graph G_t as mentioned in Section II, ϵ is an infinitesimal.

After representing the interaction by using the graph, ST-GCN implements graph convolutional layers and temporal convolutional layers to extract features according to the definition given in [23]:

$$f(V^l, A) = \sigma(\Lambda^{-\frac{1}{2}} \hat{A}_t \Lambda^{-\frac{1}{2}} V^l \boldsymbol{W}^l) \tag{13}$$

where A_t is symmetrically normalized by

$$A_t = \Lambda^{-\frac{1}{2}} \hat{A}_t \Lambda^{-\frac{1}{2}} V^l \tag{14}$$

$$\hat{A}_t = A_t + I \tag{15}$$

and Λ_t is the diagonal node degree matrix of $\hat{A}_t$, and I is the identity matrix.

D. Image Feature Extraction

It is necessary to integrate image data to comprehend the features of the environment. CNN is used in the image channel to extract image features. LSTM layers are used to learn this partial and temporal information when small spatial differences between feature maps are considered. The sequence of images labeled $\boldsymbol{I}_t$ in Section II serves as input for the image feature extraction module. Firstly, images are resized to 600 × 480 before consecutively input to the convolutional neural network and LSTM layers. This process of CNN could be formulated as:

$$\boldsymbol{L}_t = CNN(\boldsymbol{I}_t) \tag{16}$$

Details of the CNN network structure are shown in Fig.15. The size of the feature map output by CNN is 8×15 as shown on the left side of Fig.3 (a). The feature map is serialized to get a sequence with a stride of 8, and the data 1×5 from each row is used as the input of the LSTM cell.

LSTM is a network with a long-term memory function consisting of a forgetting gate, input data, and output date, as shown in Fig.3(b). The forward process of LSTM could be represented as Eqs. (17a)-(17e). The forget gate decides what information is left in the cell state and updates the cell state Eqs. (17a)-(17b). The input gate decides what information to discard from the cell state Eq.(17c). The output gate controls the output of the cell state Eqs. (17d)-(17e). At time step t, the input and output of the LSTM hidden layer are x_t and h_t, and the memory unit is c_t.

$$f_t = \sigma(\boldsymbol{W}_{xf} x_t + \boldsymbol{W}_{hf} h_{t-1} + b_f) \tag{17a}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(\boldsymbol{W}_{xc} x_t + \boldsymbol{W}_{hc} h_{t-1} + b_c) \tag{17b}$$

$$i_t = \sigma(\boldsymbol{W}_{xi} x_t + \boldsymbol{W}_{hi} h_{t-1} + b_i) \tag{17c}$$

$$o_t = \sigma(\boldsymbol{W}_{xo} x_t + \boldsymbol{W}_{ho} h_{t-1} + b_o) \tag{17d}$$

$$h_t = o_t \odot tanh(c_t) \tag{17e}$$

where $\boldsymbol{W}_{xf}$, $\boldsymbol{W}_{xi}$, $\boldsymbol{W}_{xo}$ are the weight matrices in LSTM cell, $\boldsymbol{b_f}$, $\boldsymbol{b_c}$, $\boldsymbol{b_i}$, $\boldsymbol{b_o}$ are bias, and σ represents the Sigmoid activation function.

E. Information Fusion

This subsection first introduces components of the trajectory prediction module and the investigation of multi-stream heterogeneous data fusion methods.

The main objective of the proposed method is to predict the future trajectory of vehicles in interactive scenarios. Thus, the *Trajectory Prediction Module* (TPM) is designed at the end of the framework. TPM consists of two parts: A multi-stream fusion convolutional neural network (MFC) and a trajectory prediction convolutional neural network (TPC).

The TPC takes the optical flow graph embedding feature $\boldsymbol{F}_o$, trajectory graph embedding feature $\boldsymbol{F}_t$, and image feature $\boldsymbol{F}_i$ with the same dimensions as inputs. Considering the structural differences between heterogeneous data from multi-stream sources, this subsection introduces two types of data fusion: stitching and isotopic fusions.

Stitching fusion can retain the feature information of multi-stream data to the maximum extent. As shown in

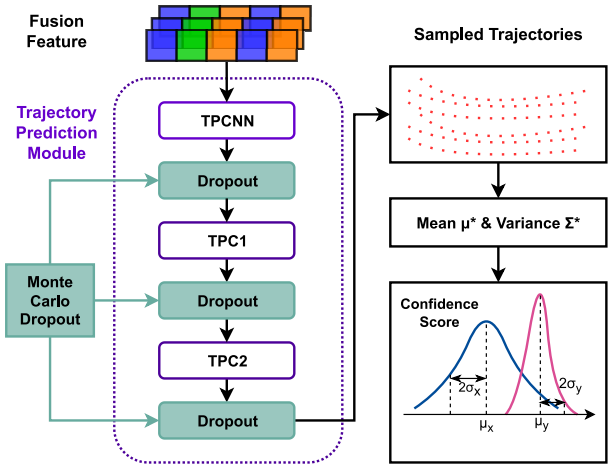

Fig. 5. The proposed framework implements Monte Carlo Dropout for epistemic uncertainty estimation. The dropout layers are inserted in the TPM module and stochastically dropout with a probability p during the inference process.

Fig.4(a), stitching fusion concatenates features from three channels of the same size together and transfers them to the Multi-stream Fusion CNN, which can be formulated as Eq.(18).

Fig.4(b) shows the isotopic fusion process, which performs an equal sum operation on three features with the same dimension and then takes the average. Eq.(19) formulates the isotopic fusion process:

$$\boldsymbol{F}_{\text{fusion}} = concat(\boldsymbol{F}_t|\boldsymbol{F}_i|\boldsymbol{F}_o) \tag{18}$$

$$\boldsymbol{F}_{\text{fusion}} = mean(\boldsymbol{F}_t + \boldsymbol{F}_i + \boldsymbol{F}_o) \tag{19}$$

$$\boldsymbol{F}_{\text{fusion}} = MFC(\boldsymbol{F}_{\text{fusion}}; \boldsymbol{W}_{\text{fusion}}) \tag{20}$$

The trajectory prediction convolutional neural network (TPC), expanding the temporal dimension by using convolution, is adapted to generate final bi-variate Gaussian distributed trajectories $\boldsymbol{Y}$:

$$\boldsymbol{Y} = TPC(\boldsymbol{F}_{\text{fusion}}; \boldsymbol{W}_{\text{TP}}) \tag{21}$$

where $\boldsymbol{W}_{\text{TP}}$ represents the weight matrix of TPC. Details of MFC and TPC structure are shown in Fig.15.

Algorithm 1 summarizes the training process of the MSIF Framework. f, g, τ, H stand for the structure of the optical flow channel, image channel, trajectory channel, and TPM module respectively.

Algorithm 1 Trajectory Prediction Training via Multi-Stream Information Fusion

Require: epoch number T, batch size N, batch of Trajectory x, batch of images y, pre-calculated optical flow z, structure of g, f, τ, H

```
# generate pre-trained optical flow feature;
ẑ_k = f(z_k)
for t = 0 to T do
    for k = 1 to N do
        # generate trajectory feature;
        x̂_k = τ(x_k)
        # generate image feature;
        ŷ_k = g(y_k)
        if no MC Dropout: then
            # fused by TPM and predict the trajectory
            ŝ_k = H(x̂_k, ŷ_k, ẑ_k)
        else
            # fused by TPM with MC-Dropout and predict the trajectory
            ŝ_k = H_MC(x̂_k, ŷ_k, ẑ_k)
        end if
        l ← equation(8)
        L ← l(s_k, ŝ_k)
        update networks f, g, τ to minimize the loss L
    end for
end for
return the best framework, which consists of f, g, τ, H
```

F. Epistemic Uncertainty Estimation

Autonomous driving often encounters complex scenarios, such as tricky traffic conditions, complex road structures, extreme weather, etc. These disadvantages are usually not covered by the datasets used to train deep learning models, which means that the input data is outside the distribution of the training set. Epistemic uncertainty estimation aims to detect inputs that lie outside the model's training distribution, and it measures how reliable the model is as well. When the predictive model faces inputs outside the training set's distribution, the system can calculate the epistemic uncertainty score according to a specific formula. Furthermore, the epistemic uncertainty score solves the dilemma of the insufficient interpretability of the deep learning model to some extent, and the overall autonomous driving system can make further decisions according to the epistemic uncertainty score. On this point, Bayesian Neural Network (BNN) incorporates probabilistic principles to model uncertainty. Unlike traditional neural networks with fixed weight and bias, BNN offers distribution over parameters, allowing for better uncertainty quantification. We denote by y^*, x^* the predicted trajectories and the test input, while $\mathcal{X}, \mathcal{Y}$ represent the training input and output. The model uncertainty could be defined by:

$$p(y^*|\, x^*, \mathcal{X}, \mathcal{Y}) = \int p(y^*|\, x^*, w)\ p(w|\mathcal{X}, \mathcal{Y}) dw \tag{22}$$

As shown in Fig.5, to approximate the intractable distribution $p(y^*|\, x^*, w)$, we learn variational distribution $p(w)$ by the sample-based Monte Carlo dropout (MC dropout) [38], which is a typical method for model uncertainty estimation without changing the architecture of the network. The proposed framework implements dropout layers in the TPM module and stochastically dropout with a probability p during the inference process. After inference N times, a set of trajectory $\{y_1^*, y_2^*, y_3^*, \ldots, y_N^*\}$ are stored and used to evaluate the epistemic uncertainty. The confidence score can be used to measure the uncertainty of the trajectory prediction model [39]. After MC dropout, the mean $\bar{p}_{t_i}^*$ and variance $\bar{\Sigma}_{p_{t_i}^*}$ of N collected trajectories at the same timestep t_i could

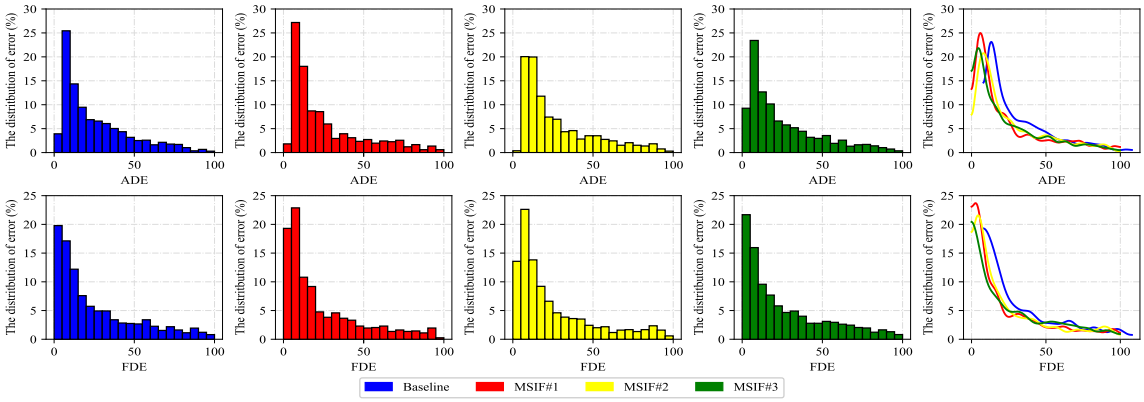

Fig. 6. The histograms show comparative results of error distributions for ADE and FDE in HEV-I dataset. Error distributions of baseline, MSIF#1, MSIF#2, and MSIF#3 are presented in blue, red, yellow, and green. The X-axis is the range of predicted error for each test sample. The Y-axis is the percentage of samples in a different range of errors.

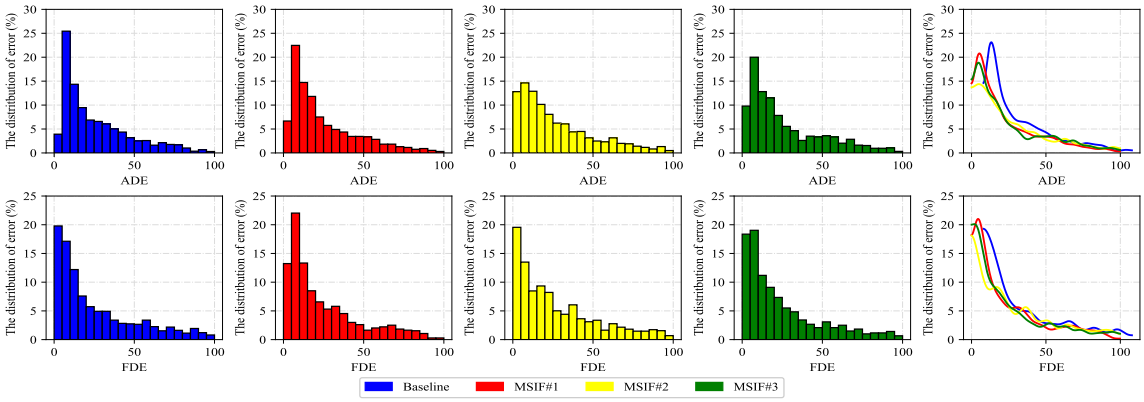

Fig. 7. The histograms show comparative results of error distributions for ADE and FDE in Dark-HEV-I dataset. Error distributions of baseline, MSIF#1, MSIF#2, and MSIF#3 are presented in blue, red, yellow, and green. The X-axis is the range of predicted error for each test sample. The Y-axis is the percentage of samples in a different range of errors.

be calculated by Eq.23 and Eq.24.

$$\bar{p}^*_{t_i} = \frac{1}{N}\sum_{n\in N} p^{*(n)}_{t_i} \tag{23}$$

$$\bar{\Sigma}_{p^*_{t_i}} = \frac{1}{N}\sum_{n\in N} (p^{*(n)}_{t_i} - \bar{p}^*_{t_i})^2 \tag{24}$$

$$CS_x = \frac{|x_{true} - \mu_x|_{i=1,2,3,\ldots,T_P} < 2\sigma_x}{T_P} \times 100\% \tag{25}$$

$$CS_y = \frac{|y_{true} - \mu_y|_{i=1,2,3,\ldots,T_P} < 2\sigma_y}{T_P} \times 100\% \tag{26}$$

Hence, it is understandable that the variance Σ_{xx} and Σ_{yy} measure the dispersion of the predicted data in both the x and y directions. Then, the standard deviation $\sigma = \sqrt{\Sigma}$ is taken into account to gauge the confidence score. The confidence score is calculated by judging whether the prediction lies in the confidence interval within two standard deviations (2σ), thereby quantifying the perceived uncertainty of the model through the Eq.25 and Eq.26.

III. Experiments

This section introduces the Honda Egocentric view-Intersection dataset, which focuses on interactive scenarios, to validate the method mentioned above. Experiments demonstrate that the proposed method performs well on both the HEV-I and the generated Dark-HEV-I datasets. Besides, details of experiments, including evaluation metrics, baselines, implementation details, and experimental results, are presented in the following subsection. Finally, this section conducts quantitative and qualitative analyses to demonstrate the feasibility and superiority of the proposed approach.

A. Dataset and Evaluation Metrics

1) HEV-I Dataset: *Honda Egocentric View-Intersection* (HEV-I) is a vision dataset that focuses primarily on urban intersection scenarios where vehicles move in uncertain directions due to the complexity of road layouts and traffic conditions. The reasons for selecting HEV-I in this work include: 1) Unlike other standard datasets in autonomous driving scenarios, the HEV-I dataset contains more videos and vehicles and pays close attention to the vehicle states in interactive scenarios. In contrast, a dataset such as KITTI focuses on the ego vehicle instead of interaction scenarios. For example, vehicles are parked on roadsides or driving in one direction in most of their videos. Therefore, the HEV-I dataset is preferable for trajectory prediction problems in interactive scenarios. 2) The HEV-I dataset includes driving scenes captured at various times and under varying light intensities (including backlighting and low light), enabling

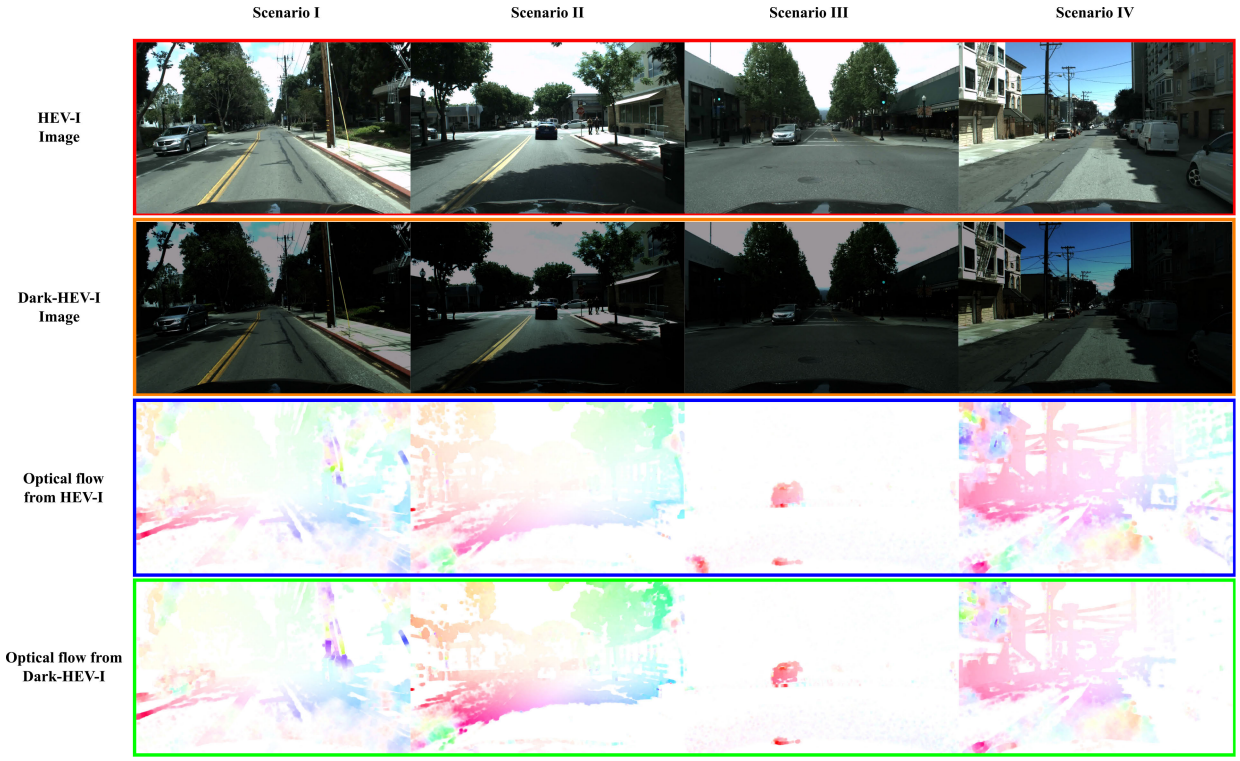

Fig. 8. This figure presents the sample of HEV-I dataset and generated Dark-HEV-I dataset. **Top to bottom:** HEV-I image; Dark-HEV-I image; optical flow from HEV-I image; optical flow from Dark-HEV-I image. Each column corresponds to a scenario.

the model to examine the prediction of trajectories under low illumination conditions. 3) HEV-I is an egocentric view dataset as opposed to a BEV view dataset. This vision-based dataset is more suitable for feature extracting and understanding the interaction scene of vehicles. The HEV-I dataset contains 230 videos as 1920 × 1200 images in 10Hz and ground truth trajectories belonging to eight object classes. To obtain dense optical flow, this approach uses Flownet 2.0 [32] with a 5×5 Region Of Interest (ROI) Pooling operator to generate a final flattened feature vector $\boldsymbol{F}_o \in \mathbb{R}^{50}$.

2) Dark HEV-I Dataset: The Dark-HEV-I has exactly the same scale as the HEV-I. Based on the HEV-I dataset, the new Dark-HEV-I dataset is generated to simulate a low-illumination autonomous driving environment. To simulate the low-illumination conditions, the exposure of each image is adjusted by utilizing the *scikit-image* library and implement gamma correction. The principle of gamma correction could be formulated as follows [40]:

$$V_{out} = V_{in}^{g} \tag{27}$$

where V_{out} is the output luminance value, and V_{in} is the input luminance value. g denotes the *gamma* value. If $gamma > 1$, the new image will be darker than the original image. If $gamma < 1$, the new image will be brighter than the original image. This function transforms the input image pixelwise after scaling each pixel to the range 0 to 1. The optical flow is regenerated by the new images with low exposure. HEV-I image, Dark-HEV-I image, HEV-I optical flow, and Dark-HEV-I optical flow are presented from top to bottom in Fig.8. The new images and the regenerated optical flow together form the Dark-HEV-I dataset. Each column in Fig. 8 represents one scenario.

3) Metrics: Average Displacement Error (ADE) [41], and Final Displacement Error (FDE) [42] are two metrics commonly used in trajectory prediction problems to evaluate the model performance accurately. ADE measures the average deviation from the ground truth, while FDE measures the absolute deviation at the endpoints of predicted trajectories. The lower the ADE and FDE, the better the model performance. Given that the HEV-I dataset is an image-based ego-centric dataset, the experimental results and evaluation metrics hereinafter are calculated in pixel coordinates.

Similar to Social-STGCNN [23] and Social-LSTM [42], the experiments generate 20 samples based on the predicted distribution and use the following formula to compute ADE and FDE with respect to the ground truth:

$$ADE = \frac{\sum_{n \in N} \sum_{t \in T_e} \left\| \hat{p}_t^n - p_t^n \right\|_2}{N \times T_p} \tag{28}$$

$$FDE = \frac{\sum_{n \in N} \left\| \hat{p}_{t_e}^n - p_{t_e}^n \right\|_2}{N} \tag{29}$$

where N represents samples of the test set, T_p is the prediction time, the ground truth, and the n^{th} sample coordinates at time step t are denoted as $\hat{p}_{t_e}^n$ and $p_{t_e}^n$.

4) Baselines: Comparing the proposed approach with the most classical and state-of-the-art models: Structural-RNN, Social-LSTM, and Social-STGCNN, the main differences between our models and baseline are shown in the middle column of the Table II:

- **Structural-RNN** Structural-RNN [43] takes the trajectories as input and successfully combines the high-level representation of the spatial-temporal graphs with the sequence learning success of recurrent neural networks.
- **Social-LSTM** Social-LSTM [42] takes the trajectories as input, which models the potentially conflicting social interactions among pedestrians by adopting long short-term memory cells.
- **TrafficPredict** TrafficPredict [44] is a long short-term memory-based (LSTM-based) real-time traffic prediction algorithm, that consists of an instance layer and a category layer.
- **Social-STGCNN** Social-STGCNN [23] is a spatial-temporal graph convolutional network that combines CNN and GCN. It extracts spatial and temporal information from the graph to generate suitable embeddings, which are then utilized by the time convolutional network to predict pedestrian trajectories.
- **MSIF#1** The base model fuses optical and trajectory information but without image information.
- **MSIF#2** The base model fuses image and trajectory information but without optical flow information.
- **MSIF#3** The base model fuses image information, optical flow, and trajectory information.

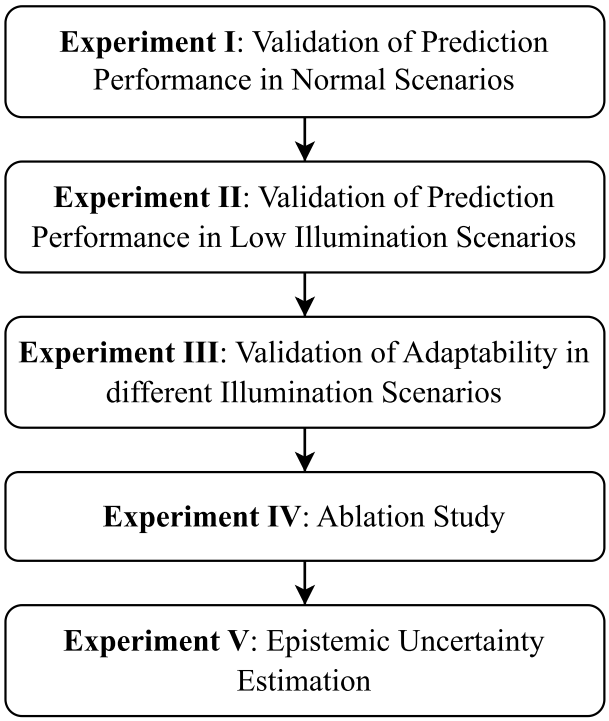

Fig. 9. This figure illustrates the experiment design of this work.

B. Implementation Details and Experimental Settings

All experiments utilize the HEV-I dataset, which is divided into training, validation, and testing sets in a ratio of 7:2:1, with the corresponding numbers being 4631:1529:665. All models are implemented using Pytorch, and experiments are run on an RTX3090 GPU. The optimizer is Adaptive Moment Estimation (Adam). The initial training rate is set as $1.0e-6$, and a learning rate scheduler is used to adjust the learning rate to its 10% every 50 epochs. Table I contains a summary of the training parameters. Given the modular design of each stream, a plug-and-play method is developed to fuse heterogeneous data, satisfying the requirement for flexibility and generality for the trajectory prediction approach, so that in the future, additional types of potential perception data can be readily utilized.

TABLE I
PARAMETERS OF IMPLEMENTATION DETAILS

Parameters	Value
Learning Rate	$1.0e-6$
Optimizer	Adam
Batch Size	1024
Number of training episode	250
Dimension of input	2
Dimension of output	5
Number of ST-GCN layers	1
Number of TPC layers	5
Dimension of output	5
Length of observed trajectory	8
Length of predicted trajectory	12

Fig. 9 illustrates the experiment design of this work. The first experiment validates the performance of the trajectory predictor in standard-illumination scenarios. The second experiment is designed to validate the predictor performance in low-illumination conditions. Next, we design case studies under varying light conditions to verify the adaptability of the model under different intensities of light. Then, as multi-stream information exists in the model, the ablation study is conducted to investigate various feature fusion techniques to identify a suitable method for producing the most accurate predictions. Finally, we design an experiment to evaluate the epistemic uncertainty of the proposed framework.

C. Experiment I: Trajectory Prediction in Normal Scenarios

This study conducts experiments on heterogeneous multi-stream sensing data in HEV-I dataset. Using Structural-RNN, Social-LSTM, and Social-STGCNN, TrafficPredict as the baseline, this experiment investigates the impact of scenario images and optical information on the accuracy of intersection trajectory prediction in standard illumination scenarios. As shown in Table II, baselines only utilize trajectories of ground truth bounding boxes, based on which the proposed approach incorporates optical flow and image information. In this comparative study, the difference between MSIF#1 and MSIF#2 is that the input of MSIF#1 is trajectories and optical, whereas the input of MSIF#2 is trajectories and images, while the MSIF#3 combines the information of the three. Considering the significance of fast prediction in autonomous driving, we investigate the time cost of MSIF#1, MSIF#2, and MSIF#3. In the configuration environment outlined in this article, it is observed that MSIF#1 solely relies on the trajectory and optical flow modules for its operations. Consequently, the inference time for MSIF#1 is approximately 49 ms. MSIF#2 incorporates the VGG-MINI architecture to facilitate image feature extraction, which involves a convolution operation that

TABLE II
QUANTITATIVE RESULTS OF PROPOSED APPROACH AND BASELINES ON HEV-I AND DARK-HEV-I WITH METRICS ADE/FDE

Model	Trajectory	Multi-stream Optical flow	Image	HEV-I ADE / FDE	Dark-HEV-I ADE / FDE
Structural-RNN [43]	✓	×	×	35.40 / 53.99	- / -
Social LSTM [42]	✓	×	×	**32.57** / 51.23	- / -
Social-STGCNN [23]	✓	×	×	58.65 / 56.95	- / -
TrafficPredict [44]	✓	×	×	31.28 / 50.04	- / -
MSIF**#1**	✓	✓	×	51.10 / **50.76**	50.32 / **50.15**
MSIF**#2**	✓	×	✓	60.95 / 52.93	220.60 / 140.52
MSIF**#3**	✓	✓	✓	**33.18** / **45.77**	**44.94** / 64.76

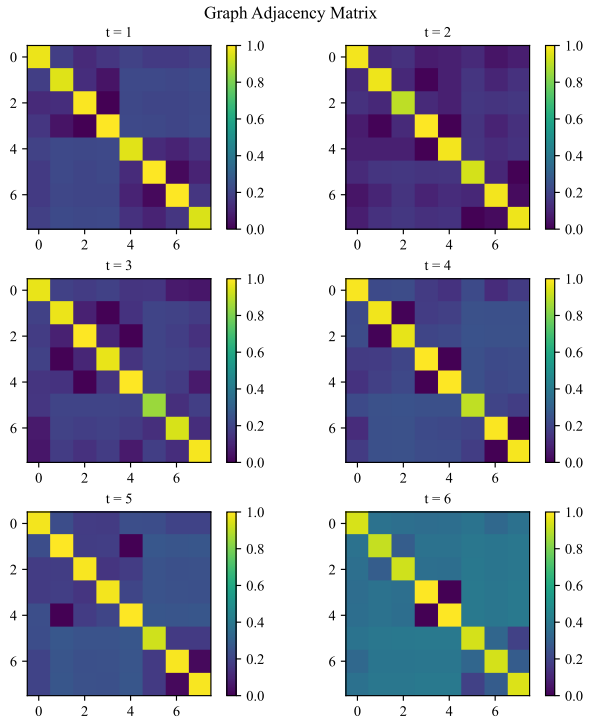

Fig. 10. The visualization of the adjacency matrix reflects the effectiveness of spatial-temporal GCN in modeling the agents' interaction.

is relatively computationally intensive, resulting in an inference time of around 1,090 ms. Lastly, MSIF#3 amalgamates diverse multi-stream information, making the inference time the lengthiest among the three configurations, amounting to approximately 1,510 ms. The experimental results presented in Table II demonstrate that the proposed method outperforms the baseline Social-STGCNN in HEV-I dataset. Comparing the baseline to MSIF#1, ADE decreases from 58.65 px to 51.10 px, and FDE decreases from 56.95 px to 50.76 px, indicating that optical flow improves the accuracy of trajectory prediction results. MSIF#2 achieves ADE of 60.95 px and FDE of 52.93 px, which indicates that MSIF#2 (with images) is mediocre on the metrics of ADE and FDE but does not indicate its misunderstanding of interactive scenarios adequately. The best-performing model MSIF#3 achieves ADE of 33.18 px and FDE of 45.77 px, representing respective increases of 43.43% and 19.63% over the baseline.

Fig.6 presents comparative results of the error distribution. Baseline, MSIF#1, MSIF#2, and MSIF#3 error distributions are depicted in blue, red, yellow, and green, respectively. The X-axis depicts the predicted error range for each test sample. The Y-axis represents the percentage of samples within each error range. The closer the histogram is to the Y-axis, the more accurate the prediction. In Fig.6, the error percentages of ADE and FDE in the range of 0 px to 5 px for MSIF#3 (green) are more than 20%, far exceeding those of MSIF#1. For MSIF#1, the proportion of ADE between 8 and 12 is close to 35%, and the percentage of FDE between 5 px and 10 px is close to 30%. The closer the histogram of MSIF#3 is to the Y-axis, the higher the percentage, so MSIF#3 has the best prediction performance.

Understanding the implicit interactions among agents within a given scene presents a formidable challenge. In the proposed framework, we leverage the spatial-temporal graph neural network to effectively capture and represent trajectory information. Following the provided formula 12, the constructed graph's adjacency matrix, derived from the trajectory points during the preceding moment, highlights the extent of influence between agents and delineates their interaction correlation. Fig. 10 illustrates the trend of the adjacency matrix normalized to [0, 1] spanning time intervals $t = 1$ to $t = 6$ for a scenario involving seven agents. Notably, the intensity of the color directly corresponds to the strength of interaction, with brighter shades indicating heightened interaction and darker hues signifying reduced correlation between the two agents.

Fig.12 reflects the loss during the training and validation process. The loss value during training is stable after 20 epochs, whereas the loss value during validation decreases gradually in the early stages and stops decreasing after ten epochs. The loss curve demonstrates that the model can effectively fit the trajectory.

D. Experiment II: Trajectory Prediction in Low-Illumination Scenarios

This experiment is conducted to validate the performance of the proposed method in low-illumination scenarios. The experimental results of the generated Dark-HEV-I dataset presented in Table II demonstrate that the proposed MSIF method achieves accurate prediction results under low-illumination conditions. MSIF#1 has an ADE of 50.32 px and an FDE of 50.15 px. MSIF#2 achieves ADE of 220.60 px and FDE of 140.52 px, indicating that only integrating images with low illumination significantly affects the comprehension of scenarios. MSIF#3 achieves ADE of 44.94 px and FDE of 64.76 px. MSIF#1 achieves a lower FDE, while MSIF#3 has a lower ADE in the Dark-HEV-I dataset. Fig.7 presents

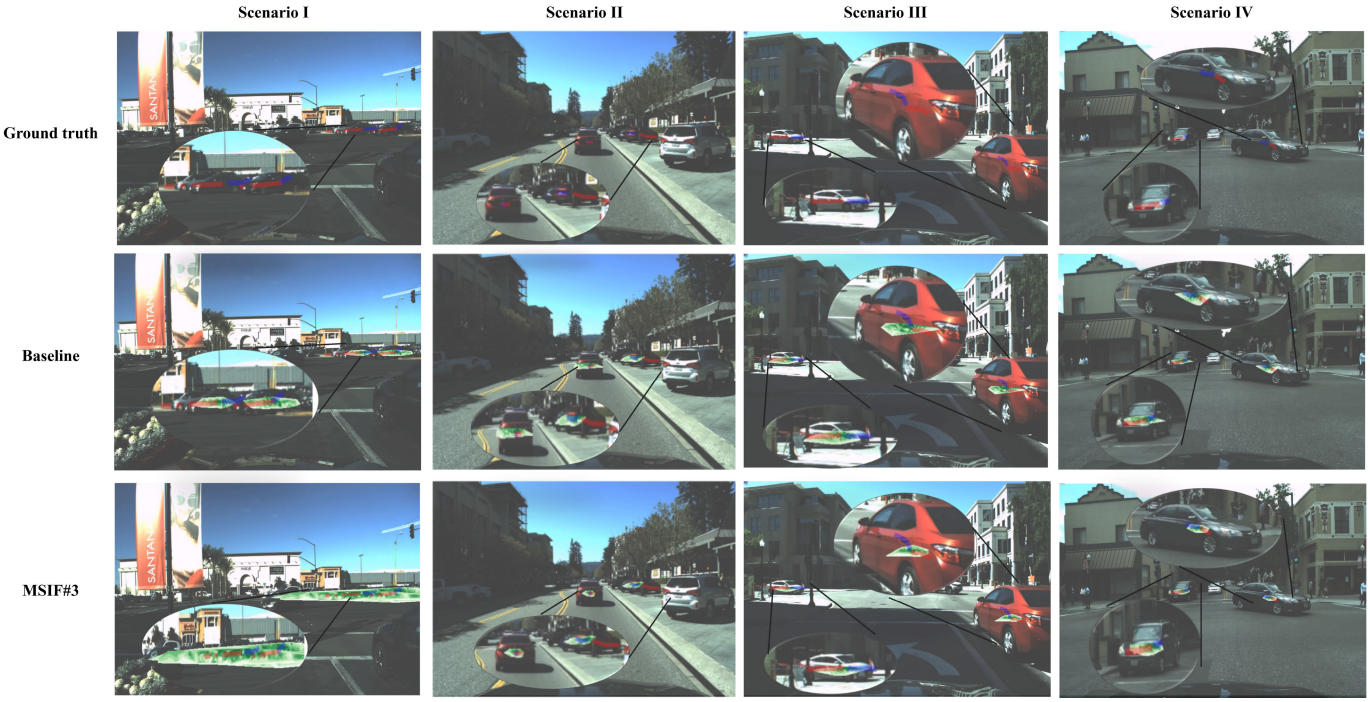

Fig. 11. This figure shows the performance of the proposed approach in the low-illumination scenario. We choose four illumination scenarios in the test set and visualize the predicted trajectories distribution. **Top to Bottom**: Ground truth, Baseline, MSIF#3. The result shows that the trajectory distribution of MSIF#3 is closer to the ground truth.

comparative results of the error distribution in the Dark-HEV-I dataset. MSIF#2 (yellow) is underperforming, as the ADE is lower than 15% in each interval in the range of 0 px to 100 px. In terms of ADE, the performance of MSIF#3 (green) is similar to MSIF#1 (red) with 20% the percentage of the ADE error distribution. While in terms of FDE, the percentage of the error distribution of FDE is greater than 15% in both the range of 0 px to 5px and 5px to 10 px. Furthermore, the distribution curve (the 5th column) shows that the histograms of MSIF#1 and MSIF#3 are closer to the Y-axis which indicates that MSIF#1 and MSIF#3 outperform the baseline in low illumination conditions.

The performance of the proposed method in the low-illumination scenario[1] is depicted in Fig. 11. We select four poor illumination scenarios from the Dark-HEV-I dataset and visualize the distribution of predicted trajectories. The ground truth, the result of baseline, and the result of MSIF#3 are presented from top to bottom. Although scenario I is a complex scenario where many vehicles pass through the intersection at a fast speed, the predicted trajectory of MSIF#3 effectively covers the ground truth, demonstrating the accuracy of MSIF#3 for long trajectory prediction. Scenario II is a car-following case, in which the predicted result of MSIF#3 is more reasonable than the baseline. In scenario III and scenario IV, MSIF#3 can forecast the direction of the opposite vehicles. In summary, the predicted distribution of MSIF#3 trajectories is closer to the ground truth, indicating that MSIF#3 achieves accurate trajectory prediction in a low-brightness environment.

[1]The video demo is available at https://www.youtube.com/watch?v=jnGJwwthkFE.

E. Experiment III: Validation of Adaptability in Different Illumination Scenarios

The first and the second experiment validate the performance of the proposed method in standard and low-illumination conditions, respectively. This experiment is conducted to verify the adaptability of the proposed MSIF to different illumination conditions. The illumination of the scenario is changed by using the gamma correction as mentioned in Eq.27, which adjusts the exposure degree (gamma from 1.0 to 2.5) of the images to simulate the different light levels throughout the day. Fig. 13 shows the gradual change in illumination for two scenes. From top to bottom, the *gamma* value equals 1, 1.4, 1.8, 2.0, and 2.5, respectively. Then, these newly generated images are used to produce corresponding optical flow information.

This experiment uses the model weights obtained by training at the Dark-HEV-I dataset (gamma equal to 2) to test the trajectory prediction performance under other light intensities. As shown in Table III, when gamma varies from 1.0 to 2.5, the ADE and FDE metrics of MSIF#1 fluctuate in a small range, and metrics of MSIF#3 show an upward trend. As MSIF#3 integrates the image information with low light level, but MSIF#1 does not, it is reasonable to speculate that the low-illumination images do have a serious impact on the trajectory prediction results.

F. Experiment IV: Ablation Study

This subsection investigates the influence of the parameters in the neural network on the model performance. Firstly, we adjusted the number of ST-GCN layers to explore the

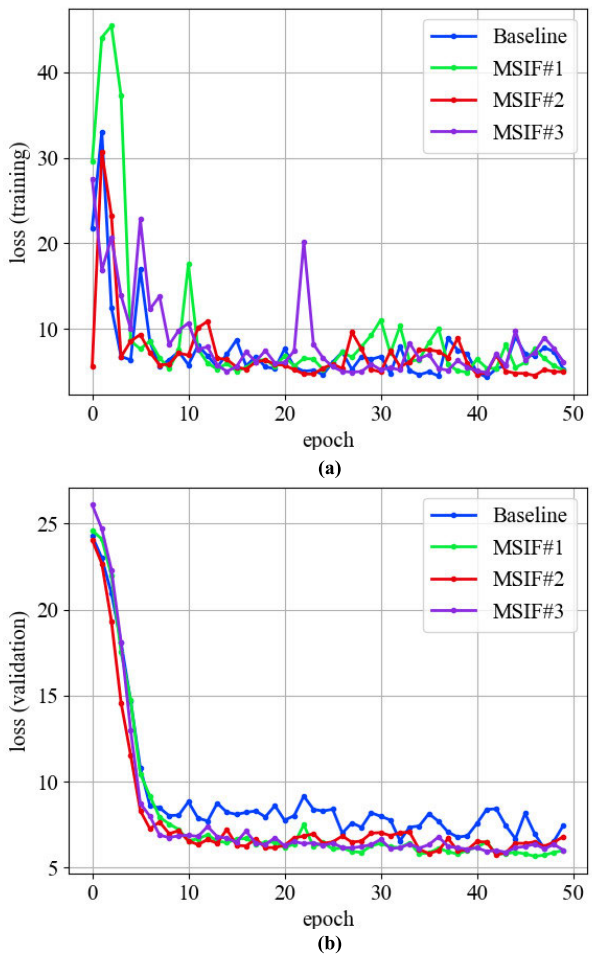

Fig. 12. Figure (a) reflects the loss during training and figure (b) reflects the validation process. The loss value during training is basically stable after 20 epochs, and the loss value during validation decreases smoothly in the early stage and stops decreasing after 10 epochs. The loss curve demonstrates that the model can fit the trajectory effectively.

TABLE III
COMPARATIVE RESULTS WITH DIFFERENT ILLUMINATION CONDITIONS ON MSIF#1 AND MSIF#3

Gamma	MSIF#1 ADE / FDE	MSIF#3 ADE / FDE
1.0	51.10 / 50.76	33.18 / 45.47
1.4	51.41 / 49.28	38.55 / 52.68
1.8	50.84 / 51.26	45.27 / 49.32
2.0	50.32 / 50.15	44.94 / 64.76
2.5	79.24 / 88.07	67.95 / 80.41

Fig. 13. This figure shows the gradual change in illumination for two scenes. **Top to bottom:** gamma is equal to 1, 1.4, 1.8, 2.0, and 2.5, respectively.

TABLE IV
QUANTITATIVE RESULTS OF MSIF#3 WITH DIFFERENT NUMBER OF ST-GCN LAYERS ON HEV-I AND DARK-HEV-I

Number of ST-GCN	HEV-I ADE / FDE	Dark-HEV-I ADE / FDE
1	**33.18** / **45.77**	**44.94** / 64.76
2	37.26 / 54.92	45.01 / **63.89**
3	38.99 / 56.20	46.97 / 65.86
4	40.51 / 58.07	47.76 / 67.03

TABLE V
QUANTITATIVE RESULTS OF MSIF#3 WITH DIFFERENT FUSION METHODS ON HEV-I AND DARK-HEV-I

Fusion Method	HEV-I ADE / FDE	Dark-HEV-I ADE / FDE
Mean	60.28 / 54.30	915.42 / 972.13
Weighted Mean	57.23 / 52.40	168.33 / 77.02
FCNN (×1)	88.69 / 64.50	107.27 / 82.21
FCNN (×2)	**33.18** / **45.77**	**44.94** / 64.76
FCNN (×3)	65.09 / 60.19	79.44 / 68.18

relationship between the number of network layers and trajectory prediction performance. Then, we design experiments on various feature fusion techniques to identify a suitable method for producing the most accurate predictions. As shown in Fig.15, the proposed approach adapts three different fusion methods (Mean, Weighted Mean, and Concatenation). Referring to Fig.15, the output features of all three channels have the same dimension, such as [1, 5, 8, 2]. The fusion module calculates the numerical mean or concatenates these features in the second dimension. To ensure that the output of the fusion operation meets the TPC, the fused feature will be input MFC to change the number of channels if concatenation fusion is selected.

As shown in Table. IV, the result of two ST-GCN layers and three ST-GCN layers is close. One layer of ST-GCN is the best setting that achieves the lowest ADE in both the HEV-I dataset and the Dark-HEV-I dataset. Thus, one layer of ST-GCN is chosen as the best parameter in the following experiment. Table. V reflects that the feature concatenation methods outperform other fusion methods. The TPC with two convolutional neural network layers (FCNN x2) achieves the lowest ADE and FDE in both HEV-I and Dark-HEV-I datasets. The Mean feature fusion method gets ADE of 60.28 and FDE

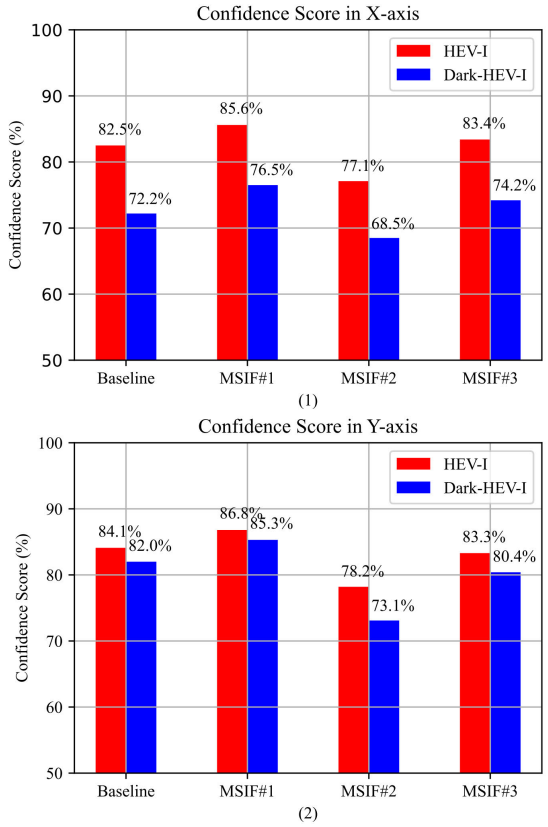

Fig. 14. A histogram of confidence scores in the X and Y directions. Using confidence scores to measure epistemic uncertainty based on the MC Dropout method.

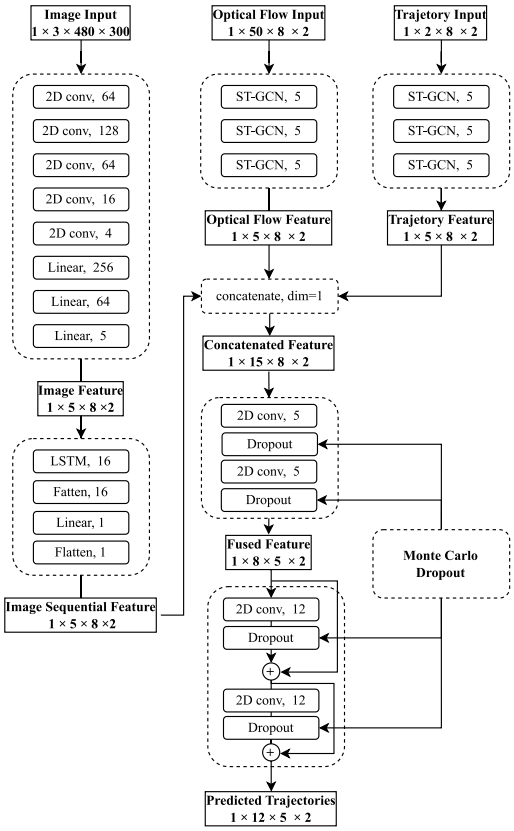

Fig. 15. This figure presents the detailed architecture of MSIF, including the image channel, the optical channel and the trajectory channel. The upper left side shows the architectural details of image feature extraction. The upper middle and upper right parts show layer details of the optical flow channel and trajectory channel, which consist of spatial-temporal GCN (ST-GCN). The lower middle part shows how features from three channels fuse and the architecture details of multi-stream Fusion CNN.

of 54.30, however, due to the low illumination conditions in Dark-HEV-I, the Mean fusion method does not perform well, with ADE of 915.43 and FDE of 972.13. FCNN (x2) is finally implemented in the TPC module due to its better performance.

G. Experiment V: Epistemic Uncertainty Estimation

Fig.14 reflects the model uncertainty of the proposed framework on HEV-I and Dark-HEV-I datasets. To assess the model's epistemic uncertainty, we followed the methodology outlined in Section II-F and calculated separate confidence scores for the X direction and the Y direction. Analysis of Fig.14 (a) and Fig.14 (b) reveals that, across all models and datasets, the confidence scores in the Y direction tend to be higher compared to those in the X direction. Notably, Baseline, MSIF#1, and MSIF#3 consistently achieved confidence scores above 80%, with MSIF#1 reaching an impressive score of 85.3% on the Dark-HEV-I dataset. Conversely, beyond a certain probability of dropout, MSIF#2 exhibited lower confidence scores below 80% in both the X and Y directions. Specifically, in the X direction, MSIF#2 achieved confidence scores of 77.1% and 68.5% on the two datasets, while the corresponding scores in the Y direction were slightly higher at 78.2% (HEV) and 73.1% (Dark-HEV-I), respectively.

Evidently, the model epistemic uncertainty of MSIF#1, which exclusively integrates trajectory and optical flow information, manifests as the most minimal. In contrast, the MSIF#2, which incorporates image information, proves to be susceptible to the influence of multi-stream fusion data. This susceptibility is reflected in its notably low confidence score and heightened epistemic uncertainty.

IV. Conclusion

To address the trajectory prediction issue in low-light conditions, this article proposes MSIF, a multi-stream information fusion-based approach, considering the interaction of the vehicle in the low-illumination environment. The image channel uses a convolutional neural network and LSTM layers for feature extraction and scene understanding. ST-GCN describes the interactivity of vehicles in both the optical flow channel and the trajectory channel. The proposed approach uses the Trajectory Prediction Module (TPM) to achieve feature fusion and trajectory prediction. To simulate the low-illumination conditions, the Dark-HEV-I dataset is created. The model is validated using both the HEV-I dataset and the generated Dark-HEV-I dataset. The multi-stream comparative experiment demonstrates that the proposed method outperforms baseline methods regarding trajectory prediction metrics. The study on feature fusion demonstrates that our method effectively combines multi-stream heterogeneous data. The qualitative analysis demonstrates that the trajectories predicted by our model in complex interaction scenarios are more reasonable

and realistic, demonstrating the adaptation to a low-light environment and achieving scene understanding. This method applies to intelligent networked vehicle driving scenarios and can be implemented on the roadside for various applications.

In the future, we will investigate the application of richer perception data for trajectory predictions in extreme conditions, as well as more effective data fusion techniques. Besides, novel architecture will be investigated for interaction and scene understanding, especially graph-based neural networks [45], [46], [47]. Furthermore, considering the high-efficiency and real-time requirements of autonomous driving [48], [49], we will explore the light-weighted trajectory prediction networks in low-illumination scenarios.

APPENDIX

Fig.15 presents the detailed architecture of MSIF, including the image channel, optical channel, and trajectory channel. The upper left side of Fig.15 shows architecture details of image feature extraction. The upper middle and upper right parts show layer details of the optical flow channel and trajectory channel, which consist of spatial-temporal GCN (ST-GCN). The lower middle part shows how features from three channels fuse and the architecture details of multi-stream Fusion CNN.

REFERENCES

[1] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 90–96, Spring 2017.

[2] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, no. 1, pp. 187–210, May 2018.

[3] G. Li, Y. Yang, X. Qu, D. Cao, and K. Li, "A deep learning based image enhancement approach for autonomous driving at night," *Knowl.-Based Syst.*, vol. 213, Feb. 2021, Art. no. 106617.

[4] *Traffic Safety Facts 2017 (DOT HS 812 806)*, National Highway Traffic Safety Administration, U.S. Department of Transportation, Washington, DC, USA, 2017.

[5] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 652–674, Sep. 2022.

[6] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. J. Dietmayer, "IMM object tracking for high dynamic driving maneuvers," in *Proc. IEEE Intell. Vehicles Symp.*, 2004, pp. 825–830.

[7] Z. Li, B. Wang, J. Gong, T. Gao, C. Lu, and G. Wang, "Development and evaluation of two learning-based personalized driver models for pure pursuit path-tracking behaviors," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 79–84.

[8] Z. Li, J. Gong, C. Lu, and J. Li, "Personalized driver braking behavior modeling in the car-following scenario: An importance-weight-based transfer learning approach," *IEEE Trans. Ind. Electron.*, vol. 69, no. 10, pp. 10704–10714, Oct. 2022.

[9] R. Zhang, L. Cao, S. Bao, and J. Tan, "A method for connected vehicle trajectory prediction and collision warning algorithm based on V2V communication," *Int. J. Crashworthiness*, vol. 22, no. 1, pp. 15–25, Jan. 2017.

[10] C. Lu, F. Hu, D. Cao, J. Gong, Y. Xing, and Z. Li, "Virtual-to-real knowledge transfer for driving behavior recognition: Framework and a case study," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6391–6402, Jul. 2019.

[11] C. Lu, F. Hu, D. Cao, J. Gong, Y. Xing, and Z. Li, "Transfer learning for driver model adaptation in lane-changing scenarios using manifold alignment," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3281–3293, Aug. 2020.

[12] Z. Li et al., "Transferable driver behavior learning via distribution adaption in the lane change scenario," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 193–200.

[13] Z. Li, J. Gong, C. Lu, and J. Xi, "Importance weighted Gaussian process regression for transferable driver behaviour learning in the lane change scenario," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12497–12509, Nov. 2020.

[14] C. Gong, Z. Li, C. Lu, J. Gong, and F. Hu, "A comparative study on transferable driver behavior learning methods in the lane-changing scenario," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3999–4005.

[15] Y. Xu, D. Ren, M. Li, Y. Chen, M. Fan, and H. Xia, "Tra2Tra: Trajectory-to-trajectory prediction with a global social spatial–temporal attentive neural network," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1574–1581, Apr. 2021.

[16] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.

[17] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 401–417.

[18] Z. Li, C. Lu, Y. Yi, and J. Gong, "A hierarchical framework for interactive behaviour prediction of heterogeneous traffic participants based on graph neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9102–9114, Jul. 2022.

[19] Z. Li, J. Gong, C. Lu, and Y. Yi, "Interactive behavior prediction for heterogeneous traffic participants in the urban road: A graph-neural-network-based multitask learning framework," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 3, pp. 1339–1349, Jun. 2021.

[20] L. Zhang, Q. She, and P. Guo, "Stochastic trajectory prediction with social graph network," 2019, *arXiv:1907.10233*.

[21] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–12.

[22] S. Haddad, M. Wu, H. Wei, and S. K. Lam, "Situation-aware pedestrian trajectory prediction with spatio-temporal attention model," 2019, *arXiv:1902.05437*.

[23] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14412–14420.

[24] J. Yin, J. Shen, X. Gao, D. Crandall, and R. Yang, "Graph neural network and spatiotemporal transformer attention for 3D video object detection from point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9822–9835, Aug. 2023.

[25] W. Wang, X. Lu, J. Shen, D. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9235–9244.

[26] X. Fu, D. Zeng, Y. Huang, Y. Liao, X. Ding, and J. Paisley, "A fusion-based enhancing method for weakly illuminated images," *Signal Process.*, vol. 129, pp. 82–96, Dec. 2016.

[27] Y. P. Loh and C. S. Chan, "Getting to know low-light images with the exclusively dark dataset," *Comput. Vis. Image Understand.*, vol. 178, pp. 30–42, Jan. 2019.

[28] E. Jung, N. Yang, and D. Cremers, "Multi-frame gan: Image enhancement for stereo visual odometry in low light," in *Proc. Conf. Robot Learn.*, 2020, pp. 651–660.

[29] I. Pikoulis, P. P. Filntisis, and P. Maragos, "Leveraging semantic scene characteristics and multi-stream convolutional architectures in a contextual approach for video-based visual emotion recognition in the wild," in *Proc. 16th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Dec. 2021, pp. 1–8.

[30] H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu, and S. Yogamani, "FuseMODNet: Real-time camera and LiDAR based moving object detection for robust low-light autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2393–2402.

[31] G. D'Angelo and F. Palmieri, "Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal features extraction," *J. Netw. Comput. Appl.*, vol. 173, Jan. 2021, Art. no. 102890.

[32] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1647–1655.

[33] A. Dosovitskiy et al., "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.
[34] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.
[35] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proc. Scandin. Conf. Image Anal.* Cham, Switzerland: Springer, 2003, pp. 363–370.
[36] Z. Wu, L. Su, Q. Huang, B. Wu, J. Li, and G. Li, "Video saliency prediction with optimized optical flow and gravity center bias," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 1–6.
[37] D. Fortun, P. Bouthemy, and C. Kervrann, "Optical flow modeling and computation: A survey," *Comput. Vis. Image Understand.*, vol. 134, pp. 1–21, May 2015.
[38] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
[39] A. Nayak, A. Eskandarian, and Z. Doerzaph, "Uncertainty estimation of pedestrian future trajectory using Bayesian approximation," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 617–630, 2022.
[40] C. Poynton, *Digital Video and HD: Algorithms and Interfaces*. Amsterdam, The Netherlands: Elsevier, 2012.
[41] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 261–268.
[42] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.
[43] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5308–5317.
[44] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "TrafficPredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6120–6127.
[45] X. Lu, W. Wang, J. Shen, D. Crandall, and J. Luo, "Zero-shot video object segmentation with co-attention Siamese networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2228–2242, Apr. 2022.
[46] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, "Video object segmentation with episodic graph memory networks," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 661–679.
[47] Z. Li et al., "Continual driver behaviour learning for connected vehicles and intelligent transportation systems: Framework, survey and challenges," *Green Energy Intell. Transp.*, vol. 2, no. 4, Aug. 2023, Art. no. 100103.
[48] J. Shen, Y. Liu, X. Dong, X. Lu, F. S. Khan, and S. Hoi, "Distilled Siamese networks for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8896–8909, Dec. 2022.
[49] Z. Zhao, S. Zhao, and J. Shen, "Real-time and light-weighted unsupervised video object segmentation network," *Pattern Recognit.*, vol. 120, Dec. 2021, Art. no. 108120.

Hailong Gong received the B.S. degree in computer science and technology from the Beijing Institute of Technology (BIT), Beijing, China, in 2022. His research interests include computer vision, image processing, and intelligent systems.

Zirui Li (Graduate Student Member, IEEE) received the B.S. degree from the Beijing Institute of Technology (BIT), Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree in mechanical engineering. From June 2021 to July 2022, he was a Visiting Researcher with the Delft University of Technology (TU Delft). Since August 2022, he has been a Visiting Researcher with the Chair of Traffic Process Automation, Faculty of Transportation and Traffic Sciences Friedrich List, TU Dresden. His research interests include interactive behavior modeling, risk assessment, and motion planning of automated vehicles.

Chao Lu (Member, IEEE) received the B.S. degree in transport engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 2009, and the Ph.D. degree in transport studies from the University of Leeds, Leeds, U.K., in 2015. In 2017, he was a Visiting Researcher with the Advanced Vehicle Engineering Centre, Cranfield University, Cranfield, U.K. He is currently an Associate Professor with the School of Mechanical Engineering, BIT. His research interests include intelligent transportation and vehicular systems, driver behavior modeling, reinforcement learning, and transfer learning and its applications.

Guodong Du received the B.S. degree in mechanical engineering from the Beijing Institute of Technology, Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree in automobile engineering. He is an Academic Guest with ETH Zürich. His research interests include motion planning and control, reinforcement learning algorithms, vehicle dynamics control, and energy management of hybrid electric vehicles.

Jianwei Gong (Member, IEEE) received the B.S. degree from the National University of Defense Technology, Changsha, China, in 1992, and the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2002. From 2011 to 2012, he was a Visiting Scientist with the Robotic Mobility Group, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor with the School of Mechanical Engineering, Beijing Institute of Technology. His research interests include intelligent vehicle environment perception and understanding, decision making, path/motion planning, and control.