

# FBLNet: FeedBack Loop Network for Driver Attention Prediction

Yilong Chen, Zhixiong Nan\*, Tao Xiang

Chongqing University, Chongqing, China

## Abstract

The problem of predicting driver attention from the driving perspective is gaining increasing research focus due to its remarkable significance for autonomous driving and assisted driving systems. The driving experience is extremely important for safe driving, a skilled driver is able to effortlessly predict oncoming danger (before it becomes salient) based on the driving experience and quickly pay attention to the corresponding zones. However, the nonobjective driving experience is difficult to model, so a mechanism simulating the driver experience accumulation procedure is absent in existing methods, and the current methods usually follow the technique line of saliency prediction methods to predict driver attention. In this paper, we propose a Feed-Back Loop Network (FBLNet), which attempts to model the driving experience accumulation procedure. By over-and-over iterations, FBLNet generates the incremental knowledge that carries rich historically-accumulative and long-term temporal information. The incremental knowledge in our model is like the driving experience of humans. Under the guidance of the incremental knowledge, our model fuses the CNN feature and Transformer feature that are extracted from the input image to predict driver attention. Our model exhibits a solid advantage over existing methods, achieving an outstanding performance improvement on two driver attention benchmark datasets.

## 1. Introduction

Accurately predicting where a driver should look is extremely important for assisted driving [14], accident avoidance [19], and driverless driving [37]. According to the road traffic injuries report <sup>1</sup> of WHO in 2022, nearly 1.3 million

Zhixiong Nan\* is the corresponding author.

This work was supported by the National Natural Science Foundation of China under Grant (62006180 and 62106026) and CCF-AFSG Research Fund (RF20220009, 02160026050171).

<sup>1</sup><https://www.who.int/zh/news-room/fact-sheets/detail/road-traffic-injuries>

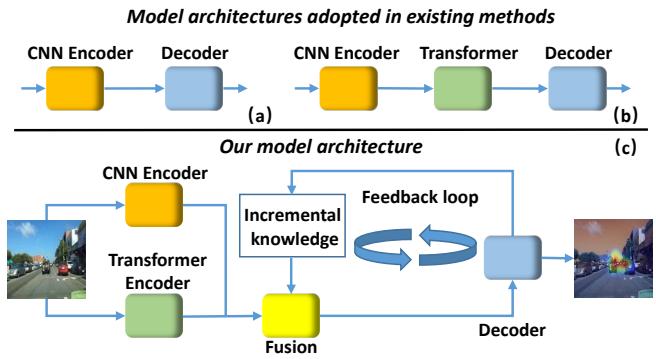


Figure 1: The comparison of the mainstream existing architectures (i.e., (a) CNN based encoder-decoder architecture and (b) CNN-Transformer based encoder-decoder architecture) and our architecture (i.e., (c) FBLNet).

people die in traffic accidents every year, and a high proportion of death result from driver attention distraction. Many accidents could be avoided if a system is able to accurately predict where a driver should look and timely send the attention distraction warning. Therefore, driver attention prediction is significant for developing safer autonomous driving or assisted driving systems. On account of the remarkable research significance of driver attention prediction, it is attracting increasingly wide research focuses, and many excellent models [36, 21, 39, 30] have been proposed.

The traffic scene is challenge and complex. It is highly dynamic, random and diverse. A driver may confront various scenarios, such as cutting in cars, crossing pedestrians and emergently-appearing motorbikes. To safely drive in such scenarios, **driving experience** is the key point. A skilled driver could effortlessly predict the most likely area where the danger may appear. For example, when a side front car is following another slowly driving car in front of it, the side front car may change to our lane, which is dangerous and we should pay more attention to it to avoid the collision. What's more, an experienced driver could predict the potentially dangerous zones according to the current traffic scene even if the danger is not visible at the current time, and pay attention to the dangerous zones to guarantee

safety. Therefore, it has great significance if we could design a driver attention prediction model that possesses the ability to accumulate the driving experience like a human.

Unfortunately, after reviewing the recent literature regarding attention prediction [28, 27], we have not found a work that is struggling in this direction. Existing models can be coarsely divided into two categories. One kind of model is CNN (Convolution Neural Network) based encoder-decoder, as shown in Fig. 1(a). The other kind is CNN-Transformer based encoder-decoder, as shown in Fig. 1(b), which first uses CNN to extract local features, then adopts Transformer to explore the global context of the local features, and finally adds a decoder at the end of Transformer. These models have remarkably driven the development of this field. However, their core idea is similar to that of models for saliency prediction, ignoring the importance of *long-term historically-accumulative information (similar to the driving experience)* for driver attention prediction. As a result, driver attention can be hardly predicted when saliency target is not consistent with driver attention.

Aiming at designing a driver attention prediction model that is able to simulate human driving experience accumulation procedure, this paper proposes a **FeedBack Loop Network** (FBLNet). Essentially different from existing models, the novelty of FBLNet is the feedback loop mechanism that continuously fetches data from the decoder module in every round of training procedure to iteratively update the **incremental knowledge**. The feedback loop mechanism enables the incremental knowledge to carry historically-accumulative and long-term temporal information, making the whole model have a similar ability as a human accumulating the driving experience.

Fig. 1(c) illustrates the coarse structure of our model. Given an input image, CNN encoder and Transformer encoder extract the features of the input image, and the feedback loop is responsible for updating the incremental knowledge. The fusion module accepts image features and the incremental knowledge as the input, and outputs the fusion feature. The fusion feature is upsampled by the decoder module to construct the driver attention heatmap. To evaluate our model, a series of comparison and diagnostic experiments are conducted on two public datasets. The comparison experiment results show that our model has a solid advantage over the baselines, achieving significant improvement on six metrics. The diagnostic experiment results validate the effectiveness of our proposed feedback loop module, fusion module and encoder module.

The contributions of this paper are as follows. 1) As Fig 1 shown, our proposed FBLNet is essentially different from all existing methods for driver attention prediction. Our FBLNet can utilize the historically-accumulative and long-term temporal information by a feedback loop structure like a human driver accumulating his driving experi-

ence. Extensive experimental results on two driver attention benchmark datasets show that FBLNet outperforms existing models for attention prediction . 2) We also propose the incremental knowledge guided fusion module. This novel fusion module can explore the inter-relationship between different kinds of features and guides the CNN feature and the Transformer feature to update in accordance with incremental knowledge during the training process.

## 2. Related Works

Methods for this problem can be roughly divided into three types: 1) Conventional methods, 2) Convolutional neural network methods and 3) Transformer methods.

### 2.1. Conventional Methods

In the early days, the dominant method mainly adopts the probability model to predict attention. Itti *et al.* [17, 16] proposes a feature map combining image color, attributes and orientation features, which uses a dynamic neural network to select the concerned positions according to the significance. At that time, the mainstream is using a mathematical algorithm, to simulate human attention mechanisms, such as the Borji graph model based on probabilistic inference and top-down research methods [6], which uses the Bayesian network. Bayesian network is also used in [31], which uses a dynamic Bayesian model and combines visual salience with people’s cognitive state to predict. In the traffic significance detection model [8], a bottom-up and top-down combined model framework is constructed by using road vanishing points as guidance. Ban *et al.* [3] also uses the top-down and bottom-up fusion model, the core of his top-down approach is adaptive resonance theory (ART), which is also used in [9]. [13] is inspired by the biological concept of the ventral attention system, which focuses on incidents of low significance but high stimulation. These above methods do not use CNN, but their top-down and bottom-up ideas inspire later researches.

### 2.2. Convolutional Methods

With the continuous development of the convolutional neural network in the field of computer vision, CNN-based methods have become the first choice for researchers. These methods can be divided into two types, bottom-up type and a combination of bottom-up and top-down type. In the bottom-up approach, [35] uses Bayesian modeling, and adds a fully convolutional neural network to process the image. Palazzi *et al.* [29, 30] uses C3D network.[1] also uses C3D convolution operation. Lattef *et al.* [20] uses a generative model which has a simple structure. Deng *et al.* [36] proposes a Dual-Branch model that divides the input into a single frame and continuous frames. [33, 32, 34] are all based on U-Net. Meanwhile, many scholars integrate some top-down research methods into the convolu-

tional neural network [11, 39, 21, 25]. Fang *et al.* [10] proposes SCAFNet network is a method of simultaneous processing and a fusion of source images and semantic images. And a big step forward for the top-down method is the application of attention mechanism [7, 38, 26]. MEDRIL [2] is the method of maximum entropy depth reverse reinforcement learning. These methods solely rely on local features, thus failing to effectively capture the crucial global information necessary for accurate driver attention prediction.

### 2.3. Transformer Methods

At present, the successful applications of Transformer in the NLP field impels scholars to apply Transformer in the CV field and many excellent results are achieved. The applications of Transformer in the field of computer vision can be divided into: 1) pure use of Transformer, 2) integration of Transformer with CNN. [24] is a pure Transformer study. Also, many researchers notice the different characteristics between CNN and Transformer, meanwhile, plenty of excellent research has been done to explore it. [22, 40] uses U-Net and integrates the Swin-transformer. [15] fuses Transformer with a convolution network, then inputs the results into a ConvLSTM for time information learning. These studies indicate the potential of using Transformers in driver attention prediction. However, solely relying on the local and global features of the driving scene is insufficient for precise driver attention prediction. Neglecting the influence of historical driving experience, as previous researchers have done, can lead to incomplete results. Taking into account the driver’s accumulated experience is crucial for a more accurate and comprehensive understanding of where their focus lies during driving tasks.

## 3. Approach

### 3.1. Overview

The goal of this work is to predict driver attention  $\mathbf{A}$  given the image  $\mathbf{I}$  captured from the perspective of drivers, which is formulated as:

$$\mathbf{A} = \mathcal{N}(\mathbf{I}) \quad (1)$$

where  $\mathbf{I} \in \mathbb{R}^{3 \times W \times H}$ ,  $\mathbf{A} \in (0, 1)^{W \times H}$  and  $\mathcal{N}$  represents a driver attention prediction network.

Different from the prior conventional encoder-decoder network structure we propose the FeedBack Loop (FBL) encoder-decoder network, which is called FBLNet. The overview of FBLNet is illustrated in Fig. 2. FBLNet is composed of four modules, including CNN-Transformer Feature Extraction (§ 3.2), FeedBack Loop (§ 3.3), Incremental Knowledge Guided Fusion (§ 3.4) and Decoder (§ 3.5). The CNN-Transformer Feature Extraction module extracts the CNN feature  $\mathbf{C}$  and the Transformer feature  $\mathbf{T}$  from  $\mathbf{I}$ . FBL module continuously accepts the feedback information

$\mathbf{d}$  from the Decoder module, and iteratively generates the incremental knowledge  $\mathbf{K}$ . In the Incremental Knowledge Guided Fusion module,  $\mathbf{C}$ ,  $\mathbf{T}$ , and  $\mathbf{K}$  are fused to produce the fusion feature  $\mathbf{F}$ . The Decoder module takes  $\mathbf{F}$  as input, and finally outputs  $\mathbf{A}$ .

### 3.2. CNN-Transformer Feature Extraction

Traffic scenes are extremely dynamic, complex and random. In addition, traffic participants are numerous and different participants always have cooperation and competitive relationships. Considering the specific characteristics of traffic scenes, to accurately predict the driver’s attention, it is essential to simultaneously encode the local information (e.g., cutting in cars and persons) and analyze the global information (e.g., multiple participants at the crossroads). Therefore, different from most existing methods, we design the CNN-Transformer Encoder that consists of two pathways, one is the CNN pathway and the other is the Transformer pathway. The former enables the encoder to extract the local detailed feature, while the latter allows to extract the global long-range dependency feature.

The feature extraction of the Convolution Neural Network pathway is formulated as:

$$\{\mathbf{C}_i\}_{i=1}^5 = \mathcal{N}_{cnn}(\mathbf{I}) \quad (2)$$

where  $\mathbf{I}$  is the input image with the size of  $3 \times 224 \times 224$ , and  $\mathcal{N}_{cnn}$  denotes the backbone of CNN. Five CNN features  $\{\mathbf{C}_i\}_{i=1}^5$  are extracted, and the size of  $\mathbf{C}_i$  is  $(64 \times 2^{i-1}) \times \frac{224}{2^i} \times \frac{224}{2^i}$ . Among them,  $\{\mathbf{C}_i\}_{i=2}^4$  will be fused with decoder features to strengthen expressions (see § 3.5), and  $\mathbf{C}_5$  will be used as one input of fusion module (see § 3.4).

The feature extraction of the Transformer pathway is expressed as:

$$\{\mathbf{T}_i\}_{i=1}^4 = \mathcal{N}_{Trans}(\mathbf{I}) \quad (3)$$

where  $\mathcal{N}_{Trans}$  denotes the Transformer blocks, generating four Transformer features  $\{\mathbf{T}_i\}_{i=1}^4$ , and the size of  $\mathbf{T}_i$  is  $(64 \times 2^{i-1}) \times \frac{56}{2^{i-1}} \times \frac{56}{2^{i-1}}$ .  $\{\mathbf{T}_i\}_{i=1}^3$  will be fused with decoder features to strengthen expressions (see § 3.5), and  $\mathbf{T}_4$  will be used as one input of fusion module (see § 3.4).

### 3.3. Feedback Loop

The major novelty of our FBLNet is the FeedBack Loop (FBL) structure, which is inspired by the core theory (i.e., feedback) of Wiener’s cybernetics claiming that the feedback mechanism makes a system stable by conveying information from the back end to the front end. As shown in Fig. 2, FBL continuously fetches the decoder feature from a feedback node and iteratively carries the feedback feature forward to accumulate it to the incremental knowledge. It is important to note that FBL is exclusively employed during the training phase. Once the training phase is completed,  $\mathbf{K}$  remains constant. This process is analogous to a driver

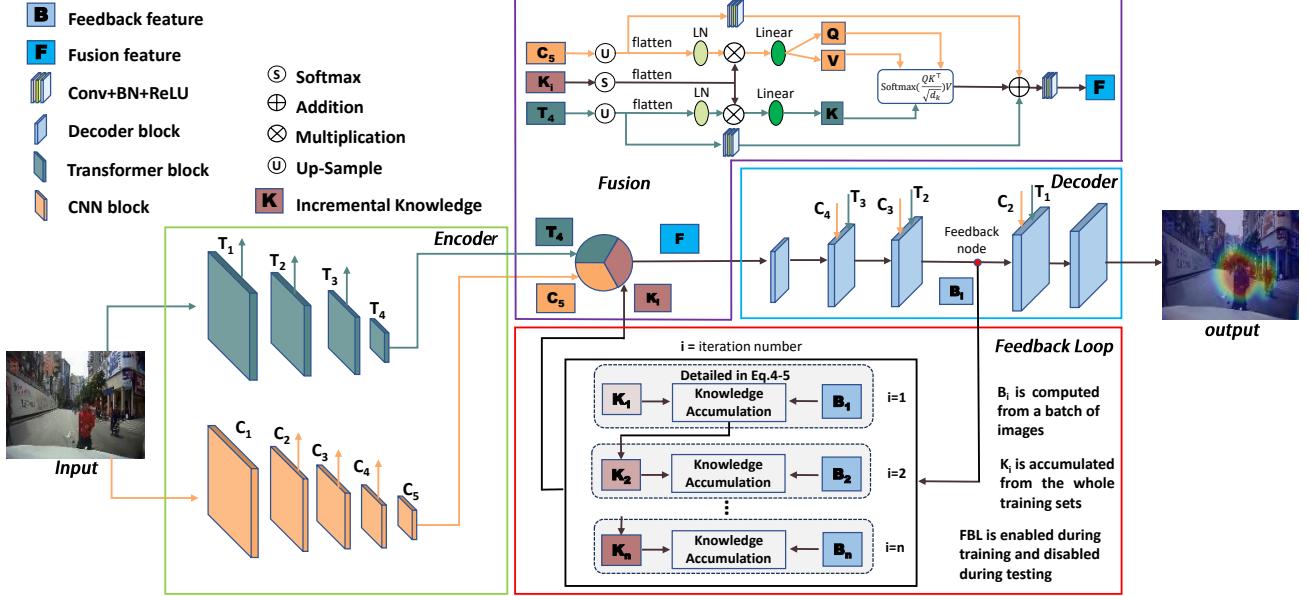


Figure 2: The overview of our method. **Encoder** module extracts CNN feature  $C_5$  and Transformer feature  $T_4$  from input image. **Feedback Loop** module continuously accepts the feedback feature  $\mathbf{d}_i$  from the Decoder module, and iteratively generates the incremental knowledge  $\mathbf{K}_i$ . **Fusion** module fuses the  $C_5$ ,  $T_4$  and  $\mathbf{K}_i$  to generate the fusion feature  $\mathbf{F}$ . **Decoder** module takes  $\mathbf{F}$  as input, and finally outputs the driver attention prediction.

accumulating driving experience from various driving scenarios (training set), and the accumulated experience (**K**) can be effectively utilized when the driver encounters new road conditions (testing set).

The incremental knowledge is physically represented as a feature map, where all elements are initialized to “1”. The feedback loop is repeated in each forward propagation step, thus the Incremental Knowledge is constantly updated. This procedure is formulated as:

$$\mathbf{K}_i = \begin{cases} 1, & i = 1 \\ \mathcal{F}_{iter}(\mathbf{K}_{i-1}, \mathbf{B}_{i-1}), & i \geq 2 \end{cases} \quad (4)$$

where  $i$  represents the iteration step number,  $\mathbf{K}_1$  is the initialization matrix which has the same size as the feedback feature,  $\mathbf{K}_i$  is the incremental knowledge after  $i$  iterations, and we note that each iteration equals that a batch of data goes through a complete forward propagation procedure.  $\mathcal{F}_{iter}$  in Eq. 4 is the iteration rule defined as:

$$\mathcal{F}_{iter}(\mathbf{K}_{i-1}, \mathbf{B}_{i-1}) = \text{mean}(\text{ReLU}(\text{BN}(\text{Conv}(\mathbf{K}_{i-1} \oplus \mathbf{B}_{i-1}))) + \mathbf{K}_{i-1}) \quad (5)$$

$\mathbf{B}_{i-1}$  in Eq. 4 is the feedback feature  $\mathbf{B}$  at  $(i-1)_{th}$  iteration, and  $\oplus$  is represented for concatenation.  $\mathbf{B}$  is fetched from a certain feedback node in decoder:

$$\mathbf{B} = \mathbf{d}_j, \mathbf{d}_j \in \{\mathbf{d}_j\}_{j=0}^4 \quad (6)$$

where  $\mathbf{d}_j$  is the decoder feature after  $j$  decoder layers.

We can find that  $\mathbf{B}$  could be fetched from any decoder

layers, thus “where to start feedback” is a crucial section. According to the theory in classic Wiener’s cybernetics, the feedback node is near to the output, which impels to select  $\mathbf{B} = \mathbf{d}_4$ . However, considering the characteristics of the deep neural network, we selected  $\mathbf{B} = \mathbf{d}_2$ , and the analysis will be detailed in the diagnostic experiment (§ 4.3).

As a short summary, the core idea of our proposed Feedback Loop is to feedback the decoder feature to iteratively update the incremental knowledge. By continuous feedback, the information in the temporal dimension is increasingly accumulated to the incremental knowledge, making it carry rich historically-accumulative and long-temporal information, thus the incremental knowledge could perform a crucial guiding role for driver attention prediction.

### 3.4. Incremental Knowledge Guided Fusion

The input of our fusion module is CNN feature  $C_5$  in Eq. 2, Transformer feature  $T_4$  in Eq. 3, and the incremental knowledge  $\mathbf{K}_i$  in Eq. 4. The output is the fusion feature denoted as  $\mathbf{F}$ . Different from the classic fusion mechanisms (e.g., addition and concatenation), we propose an incremental knowledge guided and Transformer based fusion mechanism.  $\mathbf{K}_i$  is used to guide the transition of  $C_5$  and  $T_4$ , and Transformer is adapted to allow the network to learn the potential inter-relationship between different kinds of features.

To utilize  $\mathbf{K}_i$  for guiding the transition of  $C_5$  and  $T_4$ , we start by applying a softmax operation on  $\mathbf{K}_i$  to obtain an attention feature  $\mathbf{K}_i^a$ .

$$\mathbf{K}_i^a = \text{Softmax}(\mathbf{K}_i), \quad (7)$$

and then impose  $\mathbf{K}_i^a$  on  $\mathbf{C}_5$  and  $\mathbf{T}_4$  by the multiplication:

$$\mathbf{T}_4^g = \mathbf{K}_i^a \times \text{LN}(\text{Flatten}(\mathcal{F}_{up}(\mathbf{T}_4))) \quad (8)$$

$$\mathbf{C}_5^g = \mathbf{K}_i^a \times \text{LN}(\text{Flatten}(\mathcal{F}_{up}(\mathbf{C}_5))) \quad (9)$$

where  $\mathcal{F}_{up}$  denotes up-sample operation to make  $\mathbf{T}_4$  and  $\mathbf{C}_5$  have the same size with  $\mathbf{K}_i$ , LN denotes the layer normalization. By Eqs. 8-9,  $\mathbf{T}_4$  and  $\mathbf{C}_5$  are updated under the guidance of  $\mathbf{K}_i$ .  $\mathbf{T}_4^g$  and  $\mathbf{C}_5^g$  are updated Transformer feature and updated CNN feature, respectively.

Given the updated  $\mathbf{T}_4^g$  and  $\mathbf{C}_5^g$ , Transformer is utilized to explore the relationship between them. First, the attention weight is computed by:

$$\mathbf{W} = \text{Softmax} \left( \frac{(\mathbf{W}_q \mathbf{C}_5^g) (\mathbf{W}_k \mathbf{T}_4^g)^T}{\sqrt{D}} \right) \quad (10)$$

where  $\mathbf{C}_5^g$  is *query*,  $\mathbf{T}_4^g$  is *key*,  $\mathbf{W}_q$  and  $\mathbf{W}_k$  represent learnable parameters of updating *query* and *key*, and  $D$  is the dimension of *key*.

Then, the fusion feature  $\mathbf{F}$  is computed by aggregating *Value* ( $\mathbf{V}$ ) weighted with attention weight  $\mathbf{W}$  in Eq. 10:

$$\mathbf{F} = \mathbf{WV} \quad (11)$$

where  $\mathbf{V} = \mathbf{C}_5^g$ .

Ultimately, to further enrich the fusion feature, the original CNN feature and the Transformer feature are utilized once again:

$$\mathbf{F} = \mathcal{F}_{cnn}(\mathbf{F} + \mathcal{F}_{cnn}(\mathbf{C}_5) + \mathcal{F}_{cnn}(\mathbf{T}_4)) \quad (12)$$

where  $\mathcal{F}_{cnn}(\cdot)$  represents a small CNN block,  $\mathcal{F}_{cnn}(\cdot) = \text{ReLU}(\text{BN}(\text{Conv}(\cdot)))$ . By Eq. 12, the fusion feature is updated, which has considered not only the individual information of  $\mathbf{T}_4$  and  $\mathbf{C}_5$  but also the relationship between them.

### 3.5. Decoder

The goal of the decoder module is to upsample the fusion feature  $\mathbf{F}$  to finally predict the driver attention heatmap  $\mathbf{A}$ . The decoder module consists of five blocks denoted as  $\{\mathcal{D}_i\}_{i=0}^4$ , and their corresponding outputs are represented by  $\{\mathbf{d}_i\}_{i=0}^4$ .

To make clear each decoder block, let's take  $\mathcal{D}_1$  as an example. The direct input of  $\mathcal{D}_1$  is the output  $\mathbf{d}_0$  of the previous decoder block  $\mathcal{D}_0$ . Besides  $\mathbf{d}_0$ , to further strengthen the expression of the decoder feature, a skip connection mechanism is adapted. Specifically, the CNN feature  $\mathbf{C}_4$  and Transformer feature  $\mathbf{T}_3$  skip from the encoder module to the decoder module to serve as the inputs of  $\mathcal{D}_1$ . Therefore,  $\mathbf{d}_0$ ,  $\mathbf{C}_4$  and  $\mathbf{T}_3$  are the inputs of  $\mathcal{D}_1$ , which outputs  $\mathbf{d}_1$ . This procedure is simplistically formulated as follows:

$$\mathbf{d}_1 = \mathcal{F}_{up}(\mathcal{F}_{cnn}(\text{Concat}(\mathbf{C}_4, \mathbf{T}_3))) + \mathcal{F}_{up}(\mathcal{F}_{cnn}(\mathbf{d}_0)) \quad (13)$$

where  $\mathcal{F}_{up}$  denotes up-sample operation, *Concat* represents the concatenation operation, and  $\mathcal{F}_{cnn}(\cdot) = \text{ReLU}(\text{BN}(\text{Conv}(\cdot)))$  represents a small CNN block.

The output of last decoder block is  $\mathbf{d}_4$ , based on which the driver attention heatmap  $\mathbf{A}$  is computed:

$$\mathbf{A} = \text{sigmoid}(\text{Conv}(\mathbf{d}_4)) \quad (14)$$

## 4. Experiments

### 4.1. Experimental Settings

**Datasets.** BDDA [39] (Berkeley DeepDrive Attention) is a driver attention dataset that focuses on critical situations such as occlusion, truncation, and emergency braking. The dataset is composed of 1,232 videos from different weather and lighting conditions. To obtain the annotations, 45 drivers are asked to observe videos, and their eye movements are recorded by the infrared eye tracker to generate attention points. The ground truth attention map of each frame is obtained by smoothing the attention point aggregation of several observers. DADA [10] is a driver attention dataset involving 54 kinds of accident scenarios, including 658,476 video frames of 2,000 videos collected in various scenes (highways, urban, rural, and tunnel), weather conditions and light conditions. With the help of the infrared eye tracker, 20 volunteers annotated multiple attention points for each frame, and the ground truth attention map is obtained by applying Gaussian filters on attention points.

**Evaluation Metrics.** Six classical metrics, including three distribution-based metrics: Pearson's Correlation Coefficient (CC), Kullback-Leibler divergence (Kldiv), and Similarity (SIM). Three location-based metrics: Normalized Scanpath Saliency (NSS) and two types of Area Under the ROC curve (AUC\_J [18], AUC\_B [5]) are selected for the performance evaluation. AUC is to calculate the Area Under the ROC which is created by plotting the true positive rate (TPR) against the false positive rate (FPR).

$$FPR = \frac{FP}{FP + TN}, TPR = \frac{TP}{TP + FN} \quad (15)$$

CC calculates the linear relationship of the prediction ( $\mathbf{P}$ ) and ground truth ( $\mathbf{Q}$ ).

$$CC(\mathbf{P}, \mathbf{Q}) = \frac{Cov(\mathbf{P}, \mathbf{Q})}{\sigma(\mathbf{P}) \times \sigma(\mathbf{Q})} \quad (16)$$

Kldiv is used to measure the difference between two probability distributions.

$$Kldiv(\mathbf{P}, \mathbf{Q}) = \sum_i \mathbf{Q}_i \log \left( \varepsilon + \frac{\mathbf{Q}_i}{\varepsilon + \mathbf{P}_i} \right) \quad (17)$$

SIM indicates the similarity between  $P$  and  $Q$ .

$$SIM(\mathbf{P}, \mathbf{Q}) = \sum_i \min(\mathbf{P}_i, \mathbf{Q}_i) \quad (18)$$

NSS is used to evaluate the difference between  $P$  and  $Q$ .

$$NSS(\mathbf{P}, \mathbf{Q}) = \frac{1}{N} \sum_i (\overline{\mathbf{P}_i} \times \overline{\mathbf{Q}_i}) \quad (19)$$

Table 1: Quantitative comparison results on DADA [10] and BDDA [39] datasets.

Method	BDDA						DADA					
	AUC_J↑	AUC_B↑	SIM↑	CC↑	Kldiv↓	NSS↑	AUC_J↑	AUC_B↑	SIM↑	CC↑	Kldiv↓	NSS↑
BDDA [39] <sub>ACCV'2018</sub>	0.9383	0.8847	0.3513	0.4805	2.0723	3.4515	0.8960	0.8297	0.2232	0.3261	2.7671	2.5229
U2NET [33] <sub>PR'2020</sub>	<b>0.9594</b>	<b>0.9403</b>	0.3612	0.5586	<b>1.4737</b>	3.9504	0.9481	<b>0.9088</b>	0.3008	0.4737	<b>1.8568</b>	3.7740
MINET [32] <sub>CVPR'2020</sub>	0.8645	0.7546	0.3571	0.4882	10.5031	4.2808	0.8651	0.7152	0.3000	0.3855	9.9914	3.6239
DRIVE [4] <sub>ICCV'2021</sub>	0.7637	0.6890	0.2589	0.3158	13.8349	2.5615	0.9050	0.8533	0.2468	0.3743	4.0270	3.0614
DADA [10] <sub>T-ITS'2021</sub>	0.9546	0.9056	0.3986	0.5566	1.4824	4.2205	<b>0.9500</b>	0.8737	<b>0.3412</b>	<b>0.4797</b>	2.1689	<b>4.0549</b>
DBNET [36] <sub>IEEE/CAA'2022</sub>	0.9554	0.9270	0.3825	0.5589	1.8533	3.9321	0.9190	0.8792	0.2582	0.3957	2.7716	2.9301
PGNET [40] <sub>CVPR'2022</sub>	0.9237	0.8184	<b>0.4396</b>	<b>0.5698</b>	6.0989	<b>4.9290</b>	0.9250	0.8124	<b>0.3728</b>	0.4571	5.2839	4.0455
<b>Ours</b>	<b>0.9587</b>	<b>0.9290</b>	<b>0.4666</b>	<b>0.6447</b>	<b>1.4010</b>	<b>5.0273</b>	<b>0.9540</b>	<b>0.9098</b>	0.3307	<b>0.5020</b>	<b>1.9205</b>	<b>4.1316</b>

For AUC, CC, SIM and NSS metrics, higher values are expected. For the Kldiv metric, a lower value is expected.

**Implementation Details.** **1)** The backbone of our CNN branch in Eq. 2 is ResNet-18 [12]. **2)** Swin-Transformer [23] is used as the backbone of our Transformer branch in Eq. 3. **3)** The size of input image  $\mathbf{I}$  is  $3 \times 224 \times 224$  to conform the requirement of Swin-Transformer. **4)** We choose  $\mathbf{d}_2$  as our feedback feature Eq. 6. **5)** The size of incremental knowledge is initially defined as  $64 \times 56 \times 56$ . And it is resized to  $256 \times 14 \times 14$  before entering into fusion module. **6)** The number of channels for  $\mathbf{T}_4$  and  $\mathbf{C}_5$  are both 512, a squeeze channel operation is applied to reduce the channel number to 256 so that it meets the requirement of the size of  $\mathbf{B}$ . **7)** The Adam optimizer is used with the initial learning rate set to  $10^{-4}$ , the momentum and the weight decay are set as 0.9 and  $10^{-4}$ , respectively. **8)** The batch size is set to 8, and experiments are conducted in an NVIDIA RTX3090 GPU. **9)** Our model implementation is based on PyTorch.

**Loss Function.** The loss function  $\mathcal{L}(\mathbf{P}, \mathbf{Q})$  is defined between the predicted saliency map  $\mathbf{P}$  and the corresponding ground-truth saliency map  $\mathbf{Q}$ .

$$\mathcal{L}(\mathbf{P}, \mathbf{Q}) = \mu Kldiv(\mathbf{P}, \mathbf{Q}) - \eta NSS(\mathbf{P}, \mathbf{Q}) - \xi CC(\mathbf{P}, \mathbf{Q}) \quad (20)$$

where  $\mu = 1$ ,  $\eta = 0.1$ , and  $\xi = 0.1$  are the scalar factors for Kldiv, NSS, and CC, respectively.

## 4.2. Comparison Experiment

**Baselines.** In order to illustrate the performance of our model, we compared it with other seven models, among which DBNet [36], DRIVE [4], U2Net [33], MINet [32], BDDA [39] and DADA [10] are completely based on CNN, and PGNet [40] is based on CNN and Transformer.

**Quantitative Comparison.** Tab. 1 shows the comparison results on BDDA [39] and DADA [10] datasets. We can observe that our model overall outperforms all seven baselines. Compared with the second-best result on SIM, CC and Kldiv metrics, our model achieves 6.14%, 13.4%, and 7.27% performance improvement, respectively, on the BDDA dataset. On the DADA dataset, our model achieves 4.65% and 3.43% performance improvement, on CC and Kldiv metrics, respectively. Especially, solid comparison

margins are obtained on the BDDA dataset across CC and Kldiv metrics. The BDDA dataset focuses on accident scenarios where driver attention prediction is extremely important for danger warnings.

In addition, the comparison results indicate that our proposed model structure presents the advantage over the commonly-used Encoder-Decoder structure (i.e., DBNet [36], DRIVE [4], U2Net [33] and MINet [32]) and recently-proposed CNN+Transformer structure (i.e., PGNet [40]). The superiority of our model structure is three-fold: **1)** A novel FeedBack Loop (FBL) network is proposed. FBL continuously fetches decoder features backward to the incremental knowledge, which is able to guide attention prediction using historically-accumulative long-range temporal information. **2)** A incremental knowledge guided and Transformer based fusion network is proposed. This fusion mechanism not only strengthens the CNN feature and Transformer feature by applying the incremental Knowledge, but also digs out the inter-relationship among all features via the Transformer idea. **3)** CNN-Transformer encoder is designed to extract features. CNN presents the advantage of extracting detailed local features, and Transformer is powerful at modeling long-range global contexts.

**Qualitative Comparison.** Fig. 3 illustrates some examples of qualitative comparison results. We list six complex scenarios and they are represented by (a)-(f). On the whole, our model outperforms all baselines. Especially in scenes (a), (b) and (c), our model shows great superiority. 1) In scene (a), two pedestrians crossing the road in the left shaded area demand attention, posing a challenge for models without FBL. However, FBLNet accurately focuses on this zone. 2) In scene (b), no baseline method matches the performance of FBLNet, as they all wrongly predict the white vehicle with the most salient color as the driver's attention, while the actual focus should be on the pedestrian crossing on the left. 3) In scene (c), FBLNet correctly notices the pedestrian on the left, while other methods neglect this important object. 4) In scene (d), with a large bicyclist and a small traffic light, only FBLNet effectively detects both objects of different sizes. 5) In scene (e), three baseline methods fail to accurately identify the attention area, while FBLNet

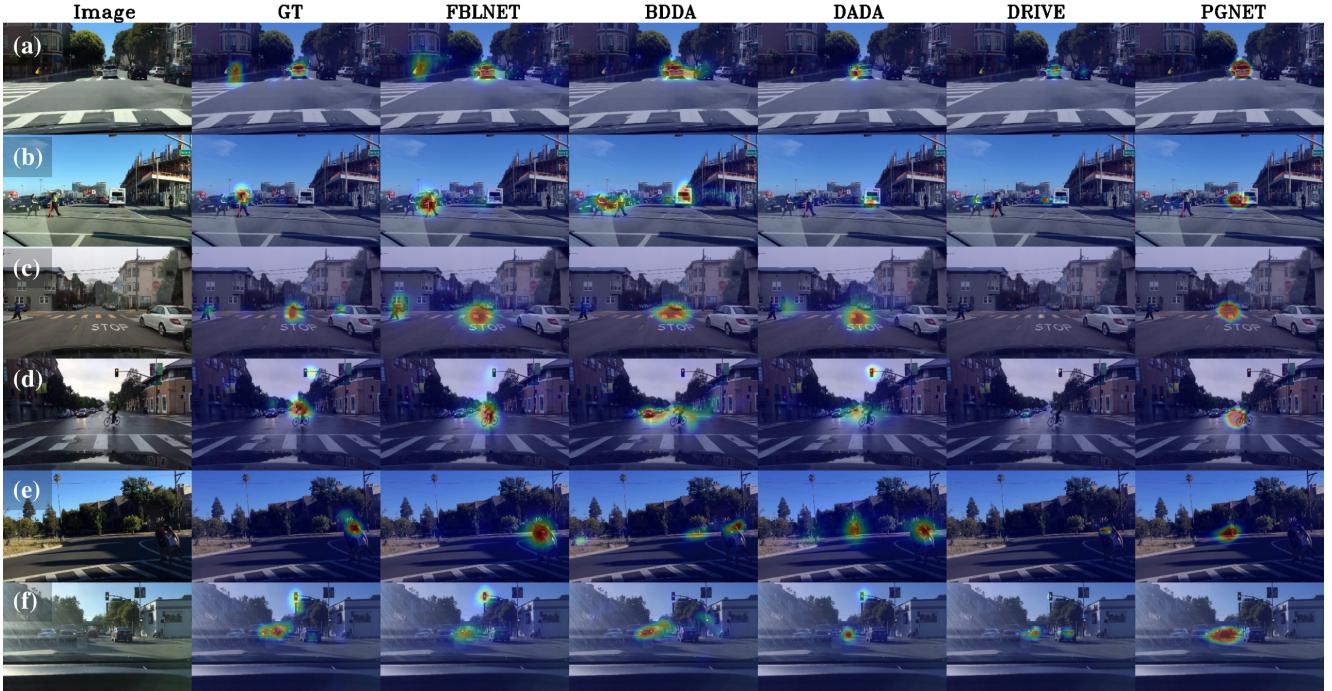


Figure 3: Qualitative comparison of our method (i.e., FBLNet) with baselines (i.e., BDDA [39], DADA [10], DRIVE [4] and PGNet [40]).

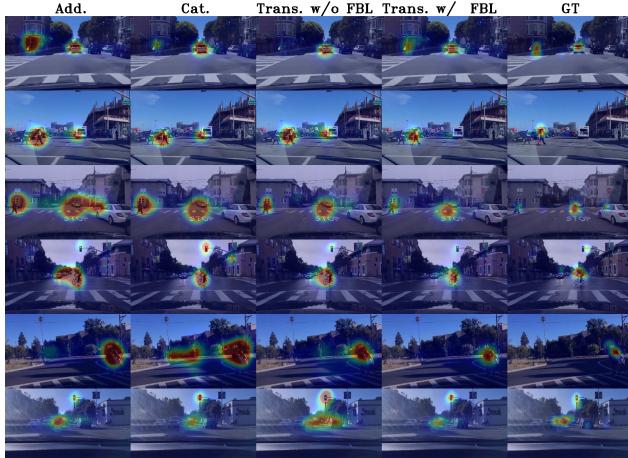


Figure 4: Visualization on the ablation study on our FBL. We can clearly see that the comparison methods present a divergence in the prediction region, inaccurate prediction and even wrong prediction in these complex scenarios

stands out with superior results.

### 4.3. Diagnostic Experiment

To examine the detailed impacts of crucial components (i.e., **Feedback Loop**, and **CNN-Transformer Encoder**) in our model, a set of ablative studies are conducted on DADA [10] and BDDA [39] datasets.

**Feedback Loop.** First, we investigate the essential feedback loop network, and the results are shown in Tab. 2. From

Table 2: Ablation study on different types of FBL.

Fusion	DADA			BDDA			avg↑
	SIM↑	CC↑	NSS↑	SIM↑	CC↑	NSS↑	
Cat.	0.3087	0.4721	3.6621	0.4358	0.6134	4.5741	1.6777
Add.	0.3254	0.4869	3.9035	0.4416	0.6116	4.6454	1.7357
w/o FBL	0.3204	0.4864	3.8658	0.4561	0.6352	4.8102	1.7624
w/ FBL	<b>0.3307</b>	<b>0.5020</b>	<b>4.1316</b>	<b>0.4666</b>	<b>0.6447</b>	<b>5.0273</b>	<b>1.8505</b>

the table we can see the performance under different fusion mechanisms on the BDDA [39] and DADA [10] datasets. In the table, Add. and Cat. are two simple ways (i.e., addition and concatenation) we used to fuse the different features from the Transformer path and the CNN path. w/o FBL and w/ FBL represent our incremental knowledge guided fusion module with FBL and without FBL, respectively. An average evaluation (i.e., avg) is adopted to show the overall performance of different methods. Compared with the simple fusion methods, our incremental knowledge guided fusion method exhibits the advantage, averagely obtaining 6.61% and 10.30% performance higher than the addition fusion method and concatenation fusion method, respectively. Meanwhile, compared with the w/o FBL, the method w/ FBL gains performance improvement on all metrics. The results are an average of 6.23% higher than that of configuration without FBL. The reason is explainable. When FBL is disabled, the incremental knowledge  $\mathbf{K}_i$  (Eq. 4) is also disabled, thus the information carried in  $\mathbf{K}_i$  could not

Table 3: Ablation study on feedback node.

Method	DADA			BDDA			
	SIM↑	CC↑	NSS↑	SIM↑	CC↑	NSS↑	avg↑
$\mathbf{B} = \mathbf{d}_0$	0.3244	0.4901	3.9441	0.4492	0.6320	4.7351	1.7625
$\mathbf{B} = \mathbf{d}_1$	0.3232	0.4921	3.9253	0.4649	0.6380	4.8873	1.7885
$\mathbf{B} = \mathbf{d}_2$	<b>0.3307</b>	<b>0.5020</b>	<b>4.1316</b>	<b>0.4666</b>	<b>0.6447</b>	<b>5.0273</b>	<b>1.8505</b>
$\mathbf{B} = \mathbf{d}_3$	0.3303	0.4888	3.9432	0.4616	0.6344	4.8526	1.7852
$\mathbf{B} = \mathbf{d}_4$	0.3219	0.4846	3.8470	0.4615	0.6384	4.8539	1.7679

guide the prediction, and simple fusion methods do not integrate features properly, therefore leading to the performance degradation. By fusing  $\mathbf{K}_i$  with the CNN feature and the Transformer feature, the model makes full use of the historically-accumulative information to predict driver attention, thus bringing performance improvement. These experiments validate the effectiveness of FBL, and prove that  $\mathbf{K}_i$  plays a crucial role in driver attention prediction. They also prove that Transformer mechanism is helpful for performance improvement since it allows the network to learn the potential inter-relationship between different features.

To further explore the significance of FBL, we have done the visualization of the results of this experiment. From Fig. 4, it is evident that when FBL is employed, the model’s prediction results are significantly closer to GT compared to other methods. The concatenation and addition methods show apparent diffusion phenomena, resulting in inaccurate and even incorrect predictions. On the other hand, the incremental knowledge-guided fusion method without FBL fails to achieve a complete prediction. The superiority of FBL in guiding the fusion process is evident from the improved accuracy and reliability of the model’s predictions.

Having demonstrated the effectiveness of the feedback loop, It is also important to diagnose “where to start feedback could achieve optimal performance”. In Tab. 3,  $\mathbf{B} = \mathbf{d}_i$  denote the configuration with the feedback from  $\mathbf{d}_i$ . Intuitively, the configuration  $\mathbf{B} = \mathbf{d}_4$  (Eq. 6) should be the optimal solution since it is in line with our Wiener’s cybernetics inspired design motivation that uses the feedback of output to make the model achieve better performance. However, as shown in Tab. 3, the model presents the best overall performance when  $\mathbf{B} = \mathbf{d}_2$ . The potential reasons are manifold. First, the global information in decoder feature  $\mathbf{d}_i$  gradually weakens while local information gradually strengthens with the increasing of decoder layers, and  $\mathbf{B} = \mathbf{d}_2$  is a trade-off. Second, if the early decoder features (i.e.,  $\mathbf{d}_0$  and  $\mathbf{d}_1$ ) are selected as the feedback feature, the feedback node is excessively near to the fusion node, making the feedback meaningless. Third, if the late decoder feature (i.e.,  $\mathbf{d}_3$  and  $\mathbf{d}_4$ ) is selected as the feedback feature, there exists a large gap among the sizes of the feedback feature, CNN feature, and Transformer feature. Therefore, a necessary down-sampling operation is needed, which leads to the loss of key information.

Table 4: Ablation study on encoder module.

Encoder	DADA			BDDA		
	SIM↑	CC↑	NSS↑	SIM↑	CC↑	NSS↑
CNN	0.3112	0.4832	3.7695	0.4550	0.6376	4.8660
Trans.	0.3117	0.4805	3.7377	0.4513	0.6327	4.7451
CNN + Trans.	<b>0.3307</b>	<b>0.5020</b>	<b>4.1316</b>	<b>0.4666</b>	<b>0.6447</b>	<b>5.0273</b>
						<b>1.8505</b>

**CNN-Transformer Encoder.** We analyze the performance of different encoder types, and the results are summarized in Tab. 4. In the table, three encoder types are compared, and we can observe that the encoder combining both CNN and Transformer generates impressive results, achieving 5.15% and 7.18% average performance improvement compared with the only CNN-based and Transformer-based encoders respectively. The reason is CNN presents the advantage of extracting local features using small size convolution windows but lacks the ability to globally encode long-range dependencies. As a complement, Transformer is powerful at modeling global long-range contexts.

**Discussion.** There exist several typical scenarios that are challenging for existing methods. However, they are common but very important for real driving, thus should be devoted to more research focuses in the future. 1) Salient object is not consistent with the real driver attention. 2) A scene does not have any salient traffic participant in current time. For example, in an intersection with no car or pedestrian, existing visual saliency prediction methods can hardly predict driver attention. 3) Many participants share the similar probability to be predicted as driver attention. It is a common case that several traffic participants in an image have similar saliency.

## 5. Conclusion

This paper handles the problem of driver attention prediction. Different from the existing mainstream methods, the proposed model presents several novelties. First, FBLNet is proposed to enable the model to simulate the human-like driving experience accumulation. Second, the CNN-Transformer encoder is designed to simultaneously extract local and global features from the input image. Third, a Transformer based fusion mechanism is proposed to fuse the CNN feature and the Transformer feature under the guidance of the incremental knowledge. Some conclusions are drawn through extensive experiments. We list the conclusions here and hope they could benefit the related studies in the community. 1) CNN-Transformer encoder is better than the individual CNN or Transformer encoder. 2) Our proposed feedback loop mechanism is effective, because it provides the incremental knowledge containing the accumulated long-range temporal information like human driving experience.

## References

- [1] Pierluigi Vito Amadori, Tobias Fischer, and Yiannis Demiris. Hammerdrive: A task-aware driving visual attention model. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5573–5585, 2022. 2
- [2] Sonia Baee, Erfan Pakdamanian, Inki Kim, Lu Feng, Vicente Ordonez, and Laura Barnes. Medirl: Predicting the visual attention of drivers via maximum entropy deep inverse reinforcement learning. In *IEEE/CVF international conference on computer vision*, pages 13178–13188, 2021. 3
- [3] Sang-Woo Ban, Bumhwi Kim, and Minho Lee. Top-down visual selective attention model combined with bottom-up saliency map for incremental object perception. In *International Joint Conference on Neural Networks*, pages 1–8, 2010. 2
- [4] Wentao Bao, Qi Yu, and Yu Kong. Drive: Deep reinforced accident anticipation with visual explanation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7619–7628, 2021. 6, 7
- [5] Ali Borji, Dicky N. Sihite, and Laurent Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 22(1):55–69, 2013. 5
- [6] Ali Borji, Dicky N. Sihite, and Laurent Itti. What/where to look next? modeling top-down visual attention in complex interactive environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(5):523–538, 2014. 2
- [7] Tao Deng, Fei Yan, and Hongmei Yan. Driving video fixation prediction model via spatio-temporal networks and attention gates. In *IEEE International Conference on Multimedia and Expo*, pages 1–6, 2021. 3
- [8] Tao Deng, Kaifu Yang, Yongjie Li, and Hongmei Yan. Where does the driver look? top-down-based saliency detection in a traffic driving environment. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):2051–2062, 2016. 2
- [9] Chiung-Yao Fang, Sei-Wang Chen, and Chiou-Shann Fuh. Automatic change detection of driving environments in a vision-based driver assistance system. *IEEE Transactions on Neural Networks*, 14(3):646–657, 2003. 2
- [10] Jianwu Fang, Dingxin Yan, Jiahuan Qiao, Jianru Xue, and Hongkai Yu. Dada: Driver attention prediction in driving accident scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4959–4971, 2022. 3, 5, 6, 7
- [11] Shun Gan, Xizhe Pei, Yulong Ge, Qingfan Wang, Shi Shang, Shengbo Eben Li, and Bingbing Nie. Multisource adaption for driver attention prediction in arbitrary driving scenes. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):20912–20925, 2022. 3
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6
- [13] Martin Heracles, Gerhard Sagerer, Ursula Körner, Thomas Michalke, Jannik Fritsch, and Christian Goerick. A dynamic attention system that reorients to unexpected motion in real-world traffic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1735–1742, 2009. 2
- [14] Feiyan Hu, Venkatesh G M, Noel E. O'Connor, Alan F. Smeaton, and Suzanne Little. Utilising visual attention cues for vehicle detection and tracking. In *International Conference on Pattern Recognition*, pages 5535–5542, 2021. 1
- [15] Pin-Jie Huang, Chi-An Lu, and Kuan-Wen Chen. Temporally-aggregating multiple-discontinuous-image saliency prediction with transformer-based attention. In *International Conference on Robotics and Automation*, pages 6571–6577, 2022. 3
- [16] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304–1318, 2004. 2
- [17] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998. 2
- [18] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2106–2113, 2009. 5
- [19] Muhammad Monjurul Karim, Yu Li, Ruwen Qin, and Zhaozheng Yin. A dynamic spatial-temporal attention network for early anticipation of traffic accidents. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):9590–9600, 2022. 1
- [20] Fahad Lateef, Mohamed Kas, and Yassine Ruichek. Saliency heat-map as visual attention for autonomous driving using generative adversarial network (gan). *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5360–5373, 2022. 2
- [21] Qiang Li, Chunsheng Liu, Faliang Chang, Shuang Li, Hui Liu, and Zehao Liu. Adaptive short-temporal induced aware fusion network for predicting attention regions like a driver. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2022. 1, 3
- [22] Ailiang Lin, Bingzhi Chen, Jiayu Xu, Zheng Zhang, Guangming Lu, and David Zhang. Ds-transunet: Dual swin transformer u-net for medical image segmentation. *IEEE Transactions on Instrumentation and Measurement*, 71:1–15, 2022. 3
- [23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, pages 9992–10002, 2021. 6
- [24] Cheng Ma, Haowen Sun, Yongming Rao, Jie Zhou, and Jiwen Lu. Video saliency forecasting transformer. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2022. 3
- [25] Zhixiong Nan, Jingjing Jiang, Xiaofeng Gao, Sanping Zhou, Weiliang Zuo, Ping Wei, and Nanning Zheng. Predicting task-driven attention via integrating bottom-up stimulus and top-down guidance. *IEEE Transactions on Image Processing*, 30:8293–8305, 2021. 3

- [26] Zhixiong Nan, Jizhi Peng, Jingjing Jiang, Hui Chen, Ben Yang, Jingmin Xin, and Nanning Zheng. A joint object detection and semantic segmentation model with cross-attention and inner-attention mechanisms. *Neurocomputing*, 463:212–225, 2021. 3
- [27] Zhixiong Nan, Tianmin Shu, Ran Gong, Shu Wang, Ping Wei, Song-Chun Zhu, and Nanning Zheng. Learning to infer human attention in daily activities. *Pattern Recognition*, 103:107314, 2020. 2
- [28] Zhixiong Nan and Tao Xiang. Third-person view attention prediction in natural scenarios with weak information dependency and human-scene interaction mechanism. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2023. 2
- [29] Andrea Palazzi, Davide Abati, simone Calderara, Francesco Solera, and Rita Cucchiara. Predicting the driver’s focus of attention: The dr(eye)ve project. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1720–1733, 2019. 2
- [30] Andrea Palazzi, Francesco Solera, Simone Calderara, Stefano Alletto, and Rita Cucchiara. Learning where to attend like a human driver. In *IEEE Intelligent Vehicles Symposium*, pages 920–925. IEEE, 2017. 1, 2
- [31] Derek Pang, Akisato Kimura, Tatsuto Takeuchi, Junji Yamamoto, and Kunio Kashino. A stochastic model of selective visual attention with a dynamic bayesian network. In *IEEE International Conference on Multimedia and Expo*, pages 1073–1076, 2008. 2
- [32] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 9413–9422, 2020. 2, 6
- [33] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern recognition*, 106:107404, 2020. 2, 6
- [34] Navyasri Reddy, Samyak Jain, Pradeep Yarlagadda, and Vineet Gandhi. Tidying deep saliency prediction architectures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10241–10247, 2020. 2
- [35] Ashish Tawari and Byeongkeun Kang. A computational framework for driver’s visual attention using a fully convolutional architecture. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 887–894, 2017. 2
- [36] Han Tian, Tao Deng, and Hongmei Yan. Driving as well as on a sunny day? predicting driver’s fixation in rainy weather conditions via a dual-branch visual model. *IEEE/CAA Journal of Automatica Sinica*, 9(7):1335–1338, 2022. 1, 2, 6
- [37] Ke Wang, Sai Ma, Fan Ren, and Jianbo Lu. Sbas: Salient bundle adjustment for visual slam. *IEEE Transactions on Instrumentation and Measurement*, 70:1–9, 2021. 1
- [38] Lei Wang and Liping Shen. A convlstm-combined hierarchical attention network for saliency detection. In *IEEE International Conference on Image Processing*, pages 1996–2000. IEEE, 2020. 3
- [39] Ye Xia, Danqing Zhang, Jinkyu Kim, Ken Nakayama, Karl Zipser, and David Whitney. Predicting driver attention in critical situations. In *Asian conference on computer vision*, pages 658–674. Springer, 2018. 1, 3, 5, 6, 7
- [40] Chenxi Xie, Changqun Xia, Mingcan Ma, Zhirui Zhao, Xiaowu Chen, and Jia Li. Pyramid grafting network for one-stage high resolution saliency detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11717–11726, 2022. 3, 6, 7