

Boosting Night-time Scene Parsing with Learnable Frequency

Zhifeng Xie, Sen Wang, Ke Xu, Zhizhong Zhang, Xin Tan, Yuan Xie, Lizhuang Ma

Abstract—Night-Time Scene Parsing (NTSP) is essential to many vision applications, especially for autonomous driving. Most of the existing methods are proposed for day-time scene parsing. They rely on modeling pixel intensity-based spatial contextual cues under even illumination. Hence, these methods do not perform well in night-time scenes as such spatial contextual cues are buried in the over-/under-exposed regions in night-time scenes. In this paper, we first conduct an image frequency-based statistical experiment to interpret the day-time and night-time scene discrepancies. We find that image frequency distributions differ significantly between day-time and night-time scenes, and understanding such frequency distributions is critical to NTSP problem. Based on this, we propose to exploit the image frequency distributions for night-time scene parsing. First, we propose a Learnable Frequency Encoder (LFE) to model the relationship between different frequency coefficients to measure all frequency components dynamically. Second, we propose a Spatial Frequency Fusion module (SFF) that fuses both spatial and frequency information to guide the extraction of spatial context features. Extensive experiments show that our method performs favorably against the state-of-the-art methods on the NightCity, NightCity+ and BDD100K-night datasets. In addition, we demonstrate that our method can be applied to existing day-time scene parsing methods and boost their performance on night-time scenes.

Index Terms—Night-time Vision, Scene Parsing, Frequency Analysis.

I. INTRODUCTION

SCENE parsing is a fundamental task in computer vision with many downstream applications, such as autonomous driving [1], human parsing [2], and image inpainting [3]. Most representative scene parsing methods [4]–[8] are proposed for day-time scenes. However, while night-time may contribute to half of total working hours (*e.g.*, in autonomous driving), these existing methods do not work well in night-time scenes due to the day-time/night-time scene discrepancies (see Figure 1a).

Zhifeng Xie is with the Department of Film and Television Engineering, Shanghai University, Shanghai 200072, China, and also with Shanghai Engineering Research Center of Motion Picture Special Effects, Shanghai 200072, China. E-mail: zhifeng_xie@shu.edu.cn

Sen Wang is with the Department of Film and Television Engineering, Shanghai University, Shanghai 200072, China. E-mail: wangsen@shu.edu.cn

Ke Xu is with the Department of Computer Science, City University of Hong Kong, HKSAR 999077, China. E-mail: kkangwing@gmail.com

Zhizhong Zhang, Xin Tan, Yuan Xie, and Lizhuang Ma are with the School of Computer Science and Technology, East China Normal University, Shanghai, China. Lizhuang Ma is also with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: zzzhang@cs.ecnu.edu.cn, xtan@cs.ecnu.cn, xieyuan8589@foxmail.com, lzma@cs.ecnu.edu.cn

Manuscript received xx xx, 2022; revised xx xx, 2022

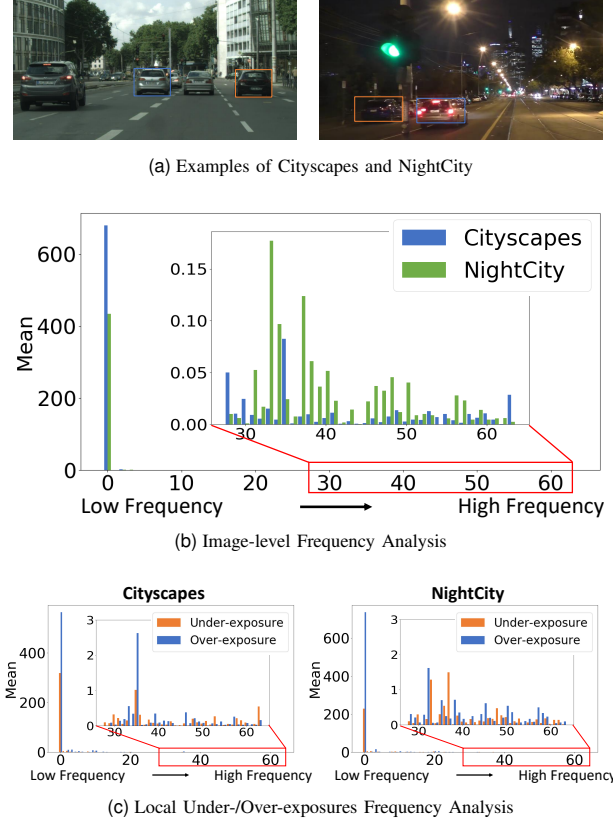


Fig. 1. Image-level frequency statistics. (a) shows one day-time scene from Cityscapes (left) and one night-time scene from NightCity (right). (b) shows image-level frequency distributions of two images of (a). (c) shows local frequency distribution of regions with under-/over-exposures (marked with orange and blue boxes, respectively) of images in (a). The high-frequency components are zoomed in by red box.

Meanwhile, although there are some methods [9]–[13] proposed to transfer the day-time domain knowledge to the night-time domain for scene parsing through domain adaptation, they still cannot achieve practical performances due to the less resolved domain discrepancies.

Recently, Tan *et al.* [14] propose the first large-scale night-time scene dataset (NightCity). They also propose an exposure-guided network for night-time scene parsing (NTSP). Deng *et al.* [15] propose the NightLab, which further boosts the performance of NTSP by learning the image lighting variation and mining hard segmented regions.

However, all these methods typically rely on modeling pixel-intensity-based contextual features, which are not necessarily reliable under uneven night-time lighting conditions. On the other hand, we note that some style transfer-based

segmentation methods [12], [13] assume that the low-level spectrum represents scene lighting information. Hence, two questions are raised: *Can image frequency distributions represent the day-time/night-time domain discrepancies? And are all frequency components important for NTSP?*

To answer the aforementioned two questions, we first conduct an image-frequency based analysis. We first analyze image-level frequency distributions by randomly select one day-time image from the Cityscapes [16] and one night-time image from the NightCity [14] (Figure 1a). We use the Discrete Cosine Transform (DCT) to compute the spectrogram of images as in [17]. Following the JPEG compression process [18], the image is divided into multiple 8×8 blocks. Then, we calculate the mean value of spectrograms of all blocks as shown in Figure 1b. While the frequency distribution of day-time image does differ from that of night-time image and such difference mainly comes from the low frequency components, we can see that night-time images do have different high frequency distribution from day-time image.

We further analyze the local regions of the night-time image where under- and over-exposures happen (marked with orange and blue boxes Figure 1a). For the corresponding comparing regions of day-time image we select the objects with the same semantics (*i.e.*, cars). We compute the spectrograms of those regions as shown in Figure 1c. We can see that the high frequency distribution of day-time image tends to have less peaks due to its relatively even lighting condition, while that of night-time image tends to have more peaks. This demonstrates that high frequency distribution differences reveal the lighting discrepancies of different domains.

Furthermore, we perform quantitative experiments at the dataset level to demonstrate our observation. To better analyze the frequency difference, we divide the spectrogram into four parts, as shown in Figure 2a. First, we calculate the mean values of the spectrogram in each frequency region, and then calculate the variance of the mean values of each frequency region of all night images in the dataset separately. We show the results in Figure 2b that the variance of the night-time scenes in each region is larger than that of the day-time scenes, which indicates that the difference of the dataset-level frequency information for night-time scenes is also significant. This motivates us to design a network for NTSP that is learnable for all frequency information to adjust the frequency components dynamically.

In this paper, we propose a novel Frequency Domain Learning Network (FDLNet), which first deals with the NTSP in the frequency domain. Specifically, we first propose a Learnable Frequency Encoder (LFE) which fully exploits all frequency components generated by DCT to adjust the channel response of different frequency components dynamically. Since the frequency distribution of different night-time images is diverse, the LFE can adaptively adjust the channel response of frequency components, so the weights of frequency components are unique for each image. Then, we propose a Spatial Frequency Fusion module (SFF), which fuses the spatial features and frequency features in channel-wise. We use both spatial and frequency information to guide the extraction of spatial context features for NTSP. We con-

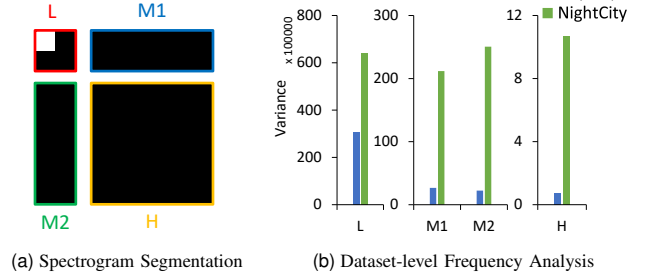


Fig. 2. Dataset-level frequency statistics. (a) The spectrogram is divided into four parts, the low frequency (L), mid frequency (M1, M2) and high frequency (H), in which the low frequency area occupies 1/16 of the frequency map, the two mid frequency areas occupies 3/16, and the high frequency area occupies 9/16. (b) Dataset-level mean variance statistics across the four frequency regions.

duct extensive experiments on night-time datasets (including NightCity, NightCity+ and BDD100K-night), showing that our method plays favorably against state-of-the-art methods. Besides, our method can be easily applied to existing state-of-the-art day-time segmentation methods [5] [6] [7] to adapt them for NTSP.

In sum, our main contributions are:

- 1) We interpret the scene lighting discrepancies between day-time and night-time scenes with the image frequency distributions. We show that a full understanding of image frequency distributions is crucial to NTSP. We propose a novel Frequency Domain Learning Network (FDLNet) for NTSP.
- 2) We propose the Learnable Frequency Encoder (LFE), to dynamically adjust the channel responses of all frequency components. We propose the Spatial Frequency Fusion module (SFF), which leverages the frequency information to model spatial contexts by a fusion of spatial and frequency features.
- 3) We show that our method can easily be applied to state-of-the-art day-time scene parsing methods to boost their performances for NTSP.

II. RELATED WORK

A. Scene Parsing

Scene parsing aims to assign each pixel with its class label. Long *et al.* [4] propose the first fully convolutional network (FCN) to extract deep features for scene parsing. Later, many methods such as PSPnet [5] and Deeplab [19] [20] [6] are proposed to aggregate more spatial features by expanding their reception fields. In order to obtain more effective spatial features, a variety of attention mechanisms have been studied in scene parsing. In [21], Point-wise Spatial Attention is proposed to associate the information of each location with that of other locations. Self-attention mechanism is introduced in DANet [22] and OCNet [23] to capture contextual information. CCnet [7] and Axial-DeepLab [24] apply the Non-local module [25] to model long-range spatial contextual information. Recently, transformer-based methods are proposed to model global contexts for scene parsing. STER [8] uses

the transformer layers to form the encoder for extracting the global context information. Swin Transformer [26] uses the sliding windows with information exchange mechanism to reduce the computational complexity of transformer, while capturing global information. Strudel *et al.* [27] propose a fully transformer architecture with a Mask Transformer as the decoder to generate class masks. Xie *et al.* [28] propose to fuse multi-level features without positional encoding in the encoding stage.

Meanwhile, there are also some methods proposed to encode prior knowledge into scene parsing. HANet [29] models the height distribution statistics of object categories, based on which they propose to learn height-driven attention. SANet [30] factorizes the scene parsing task into two sub-tasks of pixel classification and pixel grouping, and leverages pixel grouping to aggregate contextual information to enhance pixel classification. ISNet [31] learns both image level and semantic level contextual features to model inter- and intra-class correlation for scene parsing. STLNet [32] uses the proposed Quantization and Counting Operator to leverage the low-level texture features for scene parsing. In [33], contextual information beyond a single image is modeled via their proposed MCIBI by dynamically building dataset-level semantic features during training.

All above-mentioned methods are proposed for day-time scene parsing. Due to the lack of large-scale night-time datasets, previous NTSP methods have to resort to semi-supervised learning [34] or domain adaption [9]–[12], which cannot address the domain discrepancies between day-time and night-time scenes. Most recently, Tan *et al.* [14] propose a large-scale real night-time dataset and an exposure-guided network to learn robust semantic features. Deng *et al.* [15] propose the NightLab, which further boosts the performance of NTSP by learning the image lighting variation and mining hard segmented regions.

All previous methods rely on pixel-intensity based spatial contextual information, which may not be reliable due to the existence of over- and under-exposures in night-time scenes. In this paper, we study the NTSP problem from the image frequency perspective, showing that an understanding of frequency distributions facilitates contextual information modeling significantly.

B. Deep Learning in the Frequency Domain

Deep learning in the image frequency domain has many applications of, *e.g.*, image restoration [35] and demoiring [36], model compression [37], image classification [17], [38], [39] and instance segmentation [40]. Their main idea is to select a set of (low-frequency) components to reduce the network computational complexity. To reduce the high-frequency information loss of the downsampling process, a content-aware anti-aliasing module is proposed in [41]. In [17], the Discrete Cosine Transform (DCT) is used to preprocess the image to reduce the loss of important information in the process of downsampling.

Particularly in scene parsing, previous methods [42]–[46] mainly focus on the image boundary information in the

gradient domain. [47] decouples the body (low frequency) and edge (high frequency) features of the image to optimize the boundary details of the prediction results. In [13], style transfer from day-time to night-time is performed in the Fourier domain. However, their assumption of the low-frequency image amplitude component representing the whole scene illumination does not always hold true (*e.g.*, when both over- and under-exposure happen).

Different from previous work, we propose to model the whole image frequency distributions and combine them with pixel intensity-based contextual features for NTSP.

III. THE PROPOSED METHOD

A. Overview

In this paper, we present a novel method that models frequency distribution to facilitate the night-time scene parsing. Figure 3 shows the pipeline of the proposed method. Given an RGB image I , the backbone encodes the images into a spatial feature map, denoted as $f_{spatial}$. Then, we compute frequency features f_{freq} by transforming $f_{spatial}$ into the frequency domain with Discrete Cosine Transform. To fully exploit frequency information, we first propose a Learnable Frequency Encoder (LFE). This module re-weights frequency feature f_{freq} based on the contribution of each frequency component. Second, we propose a novel spatial frequency fusion module to fuse the spatial $f_{spatial}$ and frequency f_{freq} information in channel-wise. After fusion, a standard segmentation head is attached to produce the final parsing results.

B. 2D Discrete Cosine Transform

We employ Discrete Cosine Transform (DCT) to transfer the spatial feature map to frequency domain. First, we simply review the principle of DCT. The basic function of the two-dimensional discrete cosine transform B is:

$$B_{x,y}^{u,v} = \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}, \quad (1)$$

where u and v are the horizontal and vertical frequency components, respectively. N is the size of an image block, and (x, y) represents the spatial locations of the image block. Then the two-dimensional discrete cosine transform can be formulated as:

$$F(u, v) = c(u)c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) B_{x,y}^{u,v}, \quad (2)$$

where $F(u, v)$ is the 2D DCT frequency spectrum, $u, v \in \{0, 1, \dots, n-1\}$, and $f(x, y)$ is a two-dimensional vector element of $N \times N$ in the spatial domain, $x, y \in \{0, 1, \dots, N-1\}$. $c(u)$ and $c(v)$ are compensation factors, written as:

$$c(u), c(v) = \begin{cases} \sqrt{\frac{1}{N}}, & u, v = 0 \\ \sqrt{\frac{2}{N}}, & u, v \neq 0. \end{cases} \quad (3)$$

Following [40], we utilize DCT to record the frequency information in the channel dimension. Given input spatial feature map $f_{spatial} \in \mathbb{R}^{C \times H \times W}$, where C, H and W denote the channel dimension, height and width, respectively.

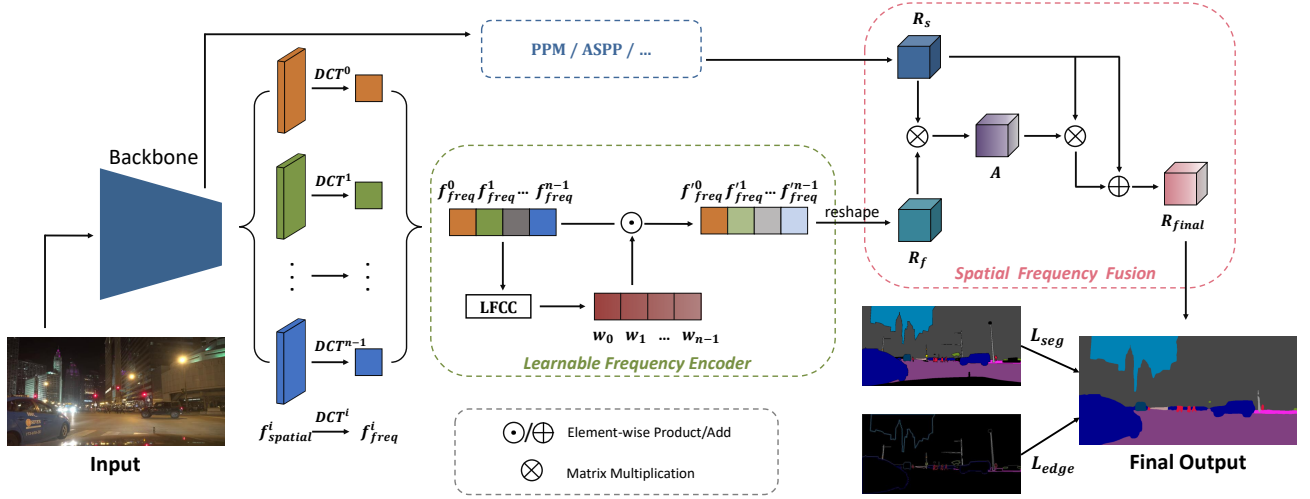


Fig. 3. Illustrating the pipeline of FDLNet. Given an input image, the backbone extracts spatial features. We leverage discrete cosine transform (DCT) in the network to get the frequency features from the output of the backbone. Then, we propose a Learnable Frequency Encoder (LFE) to leverage learnable frequency to guide the network to segment. To this end, a Learnable Frequency Component Convolutional layer (LFCC) is proposed to dynamically adjust the weights of all frequency components and we reshape it to obtain the frequency representations R_f . Meanwhile, we leverage existing spatial context aggregation modules (e.g. PPM [5], ASPP [6]) to extract the spatial representations R_s . We feed both R_f and R_s to the Spatial Frequency Fusion module (SFF) to obtain the affinity representations A . The A guides the R_s as an attention map which adjusts the channel response to get the final representations R_{final} . Finally, we utilize a segmentation head to generate the prediction from the fused feature map. Best viewed in color.

According to the rules of image compression and coding, we reconstruct the size of $f_{spatial}$ into $N \times N$. Then, $f_{spatial}$ is divided into multiple parts in the channel dimension to obtain $f_{spatial}^i \in \mathbb{R}^{\frac{C}{n} \times H \times W}$, where n is the total number of frequency components. Thus, we can obtain each frequency component f_{freq}^i by its corresponding spatial feature component $f_{spatial}^i$ using 2D DCT function DCT^i :

$$\begin{aligned} f_{freq}^i &= DCT^i(f_{spatial}^i(x, y)) \\ &= c(u)c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f_{spatial}^i(x, y) B_{x,y}^{u,v} \\ \text{s.t. } i &\in \{0, 1, \dots, n-1\}. \end{aligned} \quad (4)$$

After that, the multi-spectral frequencies vector $V_{freq} \in \mathbb{R}^C$ is defined as:

$$V_{freq} = \text{cat} \left([f_{freq}^0, f_{freq}^1, \dots, f_{freq}^{n-1}] \right), \quad (5)$$

where cat denotes concatenate operation.

C. Learnable Frequency Encoder

Unlike day-time scenes, the frequency distribution of night images is more discrete (see Figure 1 and 2). Simply using a fixed number of frequency components cannot handle night-time scene parsing. Hence, we propose the Learnable Frequency Encoder (LFE) to learn the importance of each frequency component. To dynamically adjust each frequency component, a Learnable Frequency Component Convolutional layer (LFCC) is used to convert the entire multi-spectral frequency vector V_{freq} into the weight of each frequency component W , as:

$$W = \text{softmax}(LFCC(V_{freq})), \quad (6)$$

where $LFCC$ includes a 1×1 convolutional layer and a batch-norm layer. For training stability, we constrain the weights of LFCC to be positive and sum them to 1 by a *softmax* function. The V_{freq} is reshaped to the size of $n \times \frac{C}{n} \times 1 \times 1$ and $W \in \mathbb{R}^{n \times 1^2 \times 1 \times 1}$, where each weight of the 1^2 channel corresponds to one frequency component $f_{freq}^i \in \mathbb{R}^{\frac{C}{n} \times 1 \times 1}$. This operation can be expressed as:

$$f_{freq}^{'i} = w^i \cdot f_{freq}^i, \quad (7)$$

where w^i is one channel of W corresponding to each frequency component f_{freq}^i . Then we calculate the re-weight multi-spectral frequencies vector V_{freq}' as follows:

$$\begin{aligned} V_{freq}' &= WV_{freq} \\ &= \text{cat} \left(w^0 f_{freq}^0, w^1 f_{freq}^1, \dots, w^{n-1} f_{freq}^{n-1} \right) \\ &= \text{cat} \left(f_{freq}^{'0}, f_{freq}^{'1}, \dots, f_{freq}^{'n-1} \right). \end{aligned} \quad (8)$$

We use n to group the consecutive channels of frequency vector V_{freq} and the output of filter W adjusts the weight of each frequency component f_{freq}^i based on V_{freq} . By multiplying w and V_{freq} element-wise, the encoder is learnable to predict the weight of each frequency component. Finally, the output of the encoder is the re-weight frequency feature V_{freq}' which is rectified at the channel dimension.

Discussion: We note that there are some methods [17], [40] proposed to model image frequency information but only select top k frequency components to represent the whole image. However, as shown in Figure 1 and 2, high-frequency components still contain important information due to the uneven lighting conditions of night-time scenes. Our module models the whole image frequency distribution and adjusts their weights dynamically. The experiment in Table IV

shows that dynamically modeling the whole image frequency distribution facilitates the NTSP performance.

D. Spatial Frequency Fusion

Modeling the frequency distribution helps the network understand the scene illumination. We then use the learned frequency features to guide the network to model spatial context features for night-time scene parsing. Specifically, we propose the Spatial Frequency Fusion module (SFF) to fuse features from two different domains. First, we employ a spatial context aggregation module to enhance the extraction of spatial features $f_{spatial}$, and then utilize a convolutional layer to transform the $f_{spatial}$ into spatial representations $R_s \in \mathbb{R}^{C \times \frac{H}{8} \times \frac{W}{8}}$. Meanwhile, the re-weight frequency feature V'_{freq} is fed into a convolution layer to reduce the dimensionality to generate frequency representations R_f . After that $R_f \in \mathbb{R}^{C \times 1 \times 1}$ extended to $R_f \in \mathbb{R}^{C \times \frac{H}{8} \times \frac{W}{8}}$ so as to keep the same shape with R_s . Then both R_f and R_s are reshaped to $\mathbb{R}^{C \times D}$, where $D = \frac{H}{8} \times \frac{W}{8}$. We conduct matrix multiplication between the transpose of reshaped R_f and R_s , and apply a *Softmax* layer to calculate the affinity map. The affinity operation is then defined as:

$$A(i, j) = \frac{\exp\left(R_s^i \otimes (R_f^j)^T\right)}{\sum_{i=1}^C \exp\left(R_s^i \otimes (R_f^j)^T\right)}, \quad (9)$$

where $A(i, j)$ indicates the effect of i^{th} channel in the spatial representations R_s on the j^{th} channel in the frequency representations R_f and \otimes denotes matrix multiplication. A is the affinity map calculated over the channel dimension. After that, the final fused representation R_{final} is calculated as follows:

$$R_{final} = \alpha(\text{permute}(A \otimes R_s)) + R_s, \quad (10)$$

where *permute* reshapes the result of $A \otimes R_s$ to $C \times \frac{H}{8} \times \frac{W}{8}$ and α is a scale parameter to reduce gradient instability. Note that each channel of R_{final} is the weighted sum of all channels through spatial and frequency features, and effectively captures the long-term dependencies between spatial and frequency domains.

E. Loss Function

We use the standard cross-entropy loss to measure the errors between the network predictions and ground truth labels. In addition, since the high-frequency boundary information is an important cue for scene parsing, we also incorporate edge loss during training. Unlike previous methods [43], [47] that learn edge information in the spatial domain, which are not reliable in night-time scenes due to their complex lighting conditions, we propose to learn edge information in the frequency domain.

Let $s_{i,c}$ and $\hat{s}_{i,c}$ be the ground-truth and prediction results of the i^{th} pixel of class c , respectively. L_{edge} focus on the semantic edge regions of $s_{i,c}$ as:

$$L_{edge} = - \sum_i \sum_c \mathbb{1}_{b_i} \cdot (s_{i,c} \log \hat{s}_{i,c}), \quad (11)$$

where L_{edge} represents cross-entropy loss on semantic edge regions. b_i is the ground-truth semantic edge of the i^{th} pixel and the $\mathbb{1}_{b_i}$ represents indicator function that semantic edge region in the ground-truth $s_{i,c}$.

The overall loss L can be defined as:

$$L = \lambda_1 L_{seg} + \lambda_2 L_{edge}, \quad (12)$$

where L_{seg} is a standard cross-entropy loss. λ_1 and λ_2 are two hyperparameters that control the weighting between the losses.

IV. EXPERIMENTS

To evaluate our proposed method, we conduct extensive experiments on NightCity [14], NightCity+ [15], BDD100K-night [48] and Cityscapes [16]. For all datasets, we use the standard mean Intersection over Union (mIoU) metric as an evaluation criterion.

A. Datasets

NightCity [14] is the first large-scale labeled night-time scene dataset for training and validation. NightCity+ [15] refines some labeling errors in the validation set of NightCity [14]. There is another night-time dataset, BDD100K-night, which selects night-time images with their labels from the BDD100K [48] as described in [14] and [15]. Finally, we also test our model on a day-time dataset Cityscapes [16] to verify its generalization ability.

1) *NightCity*: It includes 4,297 finely annotated images, of which 2,998 images are used for training, and 1,299 images are used for validation. The dataset labels are compatible with Cityscapes and contain 19 categories, and the resolution of the images is 512×1024.

2) *NightCity+*: NightCity+ updates the validation set of NightCity by correcting the labeling errors, and resizes the resolution of the image to 1024×2048.

3) *BDD100K-night*: It has 320 images in the training set and 34 images in the validation set. It also has 19 categories same as Cityscapes and the image resolution is 720×1280.

4) *Cityscapes*: It contains 5,000 annotated images, including 2,975 images for training, 500 images for validation, and 1,525 images for testing. The label contains 19 classes, and the resolution of the images is 1024×2048.

B. Implementation Details

The PyTorch framework is employed to implement our network. In the training phase, our model uses stochastic gradient descent (SGD) optimizer and a poly learning strategy with $(1 - \frac{iter}{total_iter})^{0.9}$. We set the initial learning rate and weight decay coefficients to 5e-3 and 5e-4, respectively. Moreover, we set the batch size to 8, and the crop size is 384×768. We conduct experiments on one TITAN RTX GPU. For data augmentation, we use random scaling with ratio sampled in the range of (0.5, 2.0), random horizontal flip, crop, and Gaussian blur as in [5]. And the training time is set to 260 epochs. For evaluation, we use multi-scale inference with ratios of [0.5, 0.75, 1.0, 1.25, 1.5, 1.75]. We utilize the dilated residual

TABLE I
COMPARISON WITH STATE-OF-THE-ARTS ON NIGHTCITY AND NIGHTCITY+. DTSS REPRESENTS DAY-TIME SEMANTIC SEGMENTATION, NTSP REPRESENTS NIGHT-TIME SCENE PARSING AND MS STANDS FOR MULTI-SCALE INFERENCE. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**.

Method	Years	Original Task	Backbone	Resolution	mIoU(%)	
					NightCity	NightCity+
PSPNet [5]	CVPR 2017	DTSS	ResNet-101	512×1024	51.02	52.24
DeeplabV3+ [6]	ECCV 2018	DTSS	ResNet-101	512×1024	51.99	53.26
DANet [22]	CVPR 2019	DTSS	ResNet-101	512×1024	50.81	52.47
CCNet [7]	ICCV 2019	DTSS	ResNet-101	512×1024	49.81	50.94
GSCNN [43]	ICCV 2019	DTSS	WideResNet38	512×1024	48.92	-
HANet [29]	CVPR 2020	DTSS	ResNet-101	512×1024	51.1	-
STER [8]	CVPR 2021	DTSS	ViT-L	512×1024	43.11	-
UperNet [26]	ICCV 2021	DTSS	Swin-T	512×1024	54.93	-
SegFormer [28]	NeurIPS 2021	DTSS	MIT-B5	512×1024	46.28	-
EGNet [14]	TIP 2021	NTSP	ResNet-101	512×1024	51.8	-
NightLab (DeeplabV3+) [15]	CVPR 2022	NTSP	ResNet-101	1024×2048	-	56.21
FDLNet (PSPNet)	-	NTSP	ResNet-101	512×1024	53.21	54.25
FDLNet (CCNet)	-	NTSP	ResNet-101	512×1024	51.00	52.27
FDLNet (DeeplabV3+)	-	NTSP	ResNet-101	512×1024	54.60	56.20
FDLNet (DeeplabV3+) + MS	-	NTSP	ResNet-101	512×1024	55.42	56.79

network [49] as the backbone with an output stride of 1/8. In the process of SFF, to reduce the amount of computation, we use the projection function to reduce the number of channels to 512. We empirically set $\lambda_1 = 1$ and $\lambda_2 = 0.01$.

C. Comparison on the NightCity and NightCity+

To verify the effectiveness of our method, we train our model and other state-of-the-art methods on the NightCity train set and validate with both the NightCity validation set and the NightCity+ validation set, respectively. For experimental comparison consistency, we rescale the NightCity+ validation set images to 512×1024 .

Methods for Comparisons: To verify the effectiveness of our method, we compare our model with state-of-the-art methods including EGNet [14] and NightLab [15] for Night-Time Scene Parsing (NTSP), PSPNet [5], DeeplabV3+ [6], DANet [22], CCNet [7], GSCNN [43], HANet [29], STER [8], UperNet [26] and SegFormer [28] for Day-Time Semantic Segmentation (DTSS). We report the performances of EGNet and HANet from [14] and NightLab from [15]. PSPNet, DeeplabV3+, DANet and CCNet are trained with the same configurations as ours. Other methods use their official code and configurations for training. Since our method can be applied to day-time segmentation methods for NTSP, we report the results of our model based on PSPNet [5], DeeplabV3+ [6] and CCNet [7].

Quantitative Comparison: From Table I, we can see that the day-time methods cannot achieve satisfying results due to the large gap between day and night scenes, but our proposed method can successfully adapt the day-time model to the night-time scenes. Furthermore, our model based on the PSPNet [5] outperforms the EGNet with a margin of 1.41%. To gain better results, we utilize our model on a stronger baseline DeeplabV3+ [6] and obtain 1.39% improvement on NightCity and 1.95% improvement on NightCity+. We also

TABLE II
COMPARISON WITH STATE-OF-THE-ARTS ON BDD100K-NIGHT. THE BEST RESULT IS MARKED IN **BOLD**.

Method	Years	Backbone	mIoU(%)
PSPNet [5]	CVPR 2017	ResNet-101	19.62
Deeplabv3+ [6]	ECCV 2018	ResNet-101	23.42
DANet [22]	CVPR 2019	ResNet-101	21.06
CCNet [7]	CVPR 2019	ResNet-101	17.74
SegFormer [28]	NeurIPS 2021	MIT-B5	22.06
AGLN [50]	TIP 2022	ResNet-101	20.16
FDLNet (PSPNet)	-	ResNet-101	25.00
FDLNet (CCNet)	-	ResNet101	23.09
FDLNet (Deeplabv3+)	-	ResNet-101	26.46

use a multi-scale strategy during inference and achieve a performance of 56.79%, which outperforms the NightLab based on DeeplabV3+ with a margin of 0.58%. Noting that the resolution is different between ours and NightLab. Our method requires smaller resolution inputs which reduces computation but achieves higher performance. The results show that our model improves the day-time models to adapt to NTSP and shows superior performance.

Qualitative Comparison: Figure 4 quantitatively compares the prediction results of our model with state-of-the-art methods for DTSS and NTSP. Since NightLab does not show segmentation results on NightCity, we compare our results with EGNet and UperNet. The lighting conditions of night-time images make the frequency distribution quite different. Adjusting the lighting condition cannot allow the network to learn the frequency information, which makes the segmentation effect unsatisfactory. However, our model can handle this problem well. Particularly, in the first row, our model can identify areas of detail, such as distant buildings and poles. In the second row, our model gives more complete poles than EGNet and more complete trees than UperNet. In the

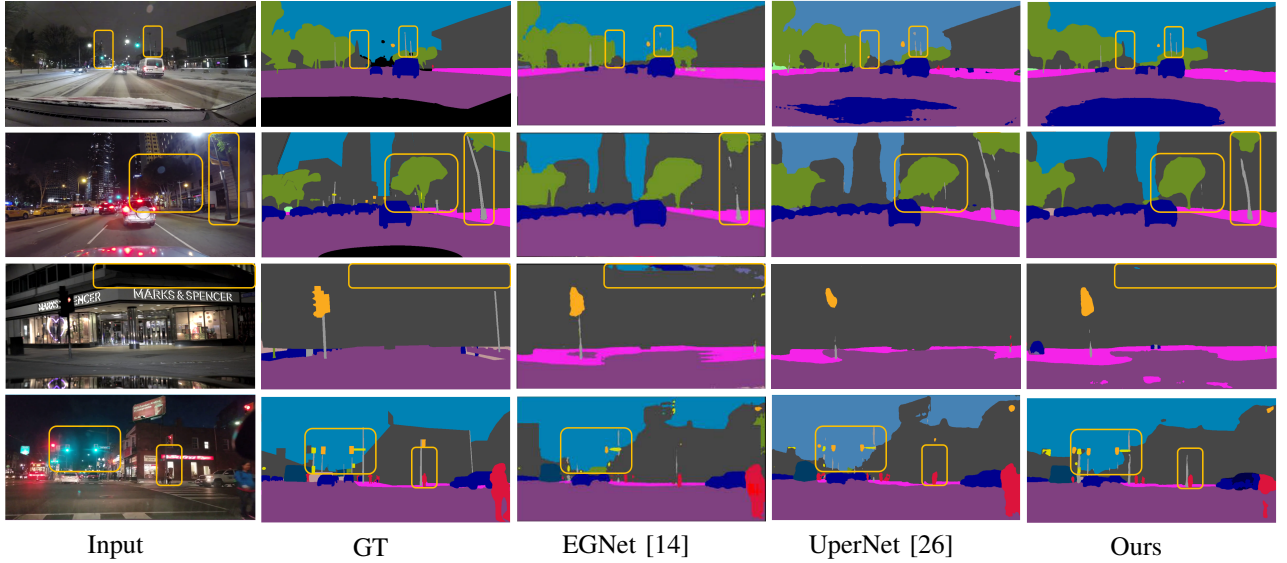


Fig. 4. Qualitative comparison on NightCity. Our advantages are highlighted by orange boxes.

third row, our model segments a more complete building. In the last row, EGNet cannot segment objects such as traffic lights, traffic signs and poles due to overexposure. UperNet cannot segment these objects completely. But our model can recognize small objects with complete fineness. These results demonstrate the superior performance of our proposed model on NTSP.

D. Comparison on the BDD100K-night

We also conducted experiments on another labeled nighttime scene dataset, BDD100K-night, to verify the effectiveness of our method. We compare our model with state-of-the-art methods PSPNet [5], DeeplabV3+ [6], DANet [22], CCNet [7], SegFormer [28], and AGLN [50] for day-time semantic segmentation. The results are reported in Table II. We can see that our model based on DeeplabV3+ achieves the best performance of 26.46%, which shows the generality of our proposed method.

E. Model Analysis

Ablation Study. To verify the effectiveness of different network components, we conduct five ablation studies.

1) *Ablation Studies on the Number of Frequency Components:* The number of frequency components is one of the important factors affecting model performance. The network extracts image features and compresses the information into channel representations, so we use DCT to compress spatial features into $N \times N$ blocks to extract frequency features. Due to the limitation of channel numbers, N could be 2, 4, 8, 16, or 32. We use the ResNet-50 as the backbone and train the network for 120 epochs, as shown in Figure 5. We find that selecting 8×8 frequency components obtain the best performance. In other experiments, we set the number of frequency components to 8×8 .

2) *Ablation Studies on Baseline Model:* We take the PSPNet [5] as the baseline. Due to the limitation of lighting

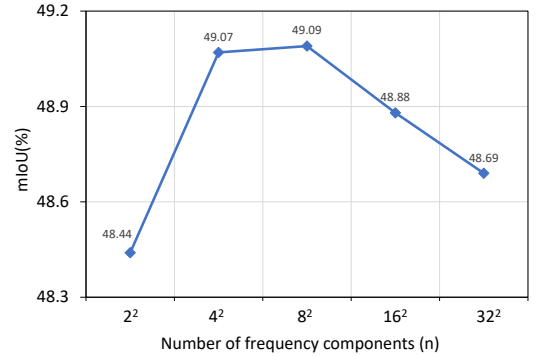


Fig. 5. Ablation Studies on the Number of Frequency Components.

conditions, night images have a lot of hard data [15], so we use OHEM [51] during the training process to improve model performance. In Table III, dilated ResNet-101 is used as the backbone, our baseline achieves 51.02% and 0.88% increase with OHEM (2nd row). A simply way to leverage frequency features is to design a SENet-like [52] module as in FcaNet [40]. So we use the same way on LFE module to adjust the channel weights for the obtained frequency features and improve the performance from 51.90% to 52.14% (3rd row). Then, we leverage SFF module in the network to replace the SENet-like module, in order to introduce the spatial context features extracted by PPM [5]. The performance of the model is improved from 52.14% to 52.84% (4th row). Incorporating the L_{edge} brings about 0.37% improvement (5th row), which shows the importance of edge supervision. The model further uses a multi-scale inference strategy (MS) to achieve a performance of 54.02% (6th row).

3) *Ablation Studies on Learnable Frequency Encoder:* To demonstrate the effectiveness of our proposed LFE module, we take PSPNet as a baseline and compare our method with two methods, one is using top-k components in [40] named (TOP), and the other is statically using all frequency components

TABLE III

ABLATION STUDY. LFE STANDS FOR LEARNABLE FREQUENCY ENCODER, SFF STANDS FOR SPATIAL FREQUENCY FUSION MODULE, L_{edge} STANDS FOR SEMANTIC EDGE LOSS, OHM STANDS FOR USING ONLINE HARD EXAMPLE MINING DURING TRAINING. MS STANDS FOR MULTI-SCALE INFERENCE STRATEGY

Method	ohem	L_{edge}	mIoU(%)
PSPNet			51.02
	✓		51.90
+ LFE	✓		52.14
+ LFE + SFF	✓		52.84
	✓	✓	53.21
+ LFE + SFF + MS	✓	✓	54.02

TABLE IV

ABLATION STUDIES ON LFE. TOP STANDS FOR EXPLOIT TOP-K COMPONENTS, SA STANDS FOR STATICALLY EXPLOIT ALL COMPONENTS AND LFE STANDS FOR OUR PROPOSED METHOD LEARNABLE FREQUENCY ENCODER. OHM STANDS FOR USING ONLINE HARD EXAMPLE MINING DURING TRAINING.

Method	Backbone	ohem	mIoU(%)	Δ (%)
PSPNet	ResNet-101	✓	51.90	
+ TOP	ResNet-101	✓	52.67	+0.77
+ SA	ResNet-101	✓	52.34	+0.44
+ LFE	ResNet-101	✓	53.21	+1.31

TABLE V

ABLATION STUDIES ON SFF. FDLNet-SE STANDS FOR USING SENet TO REPLACE OUR SFF MODULE, R_s IS SPATIAL REPRESENTATIONS AND α IS A SCALE PARAMETER IN SFF.

Method	Backbone	ohem	mIoU(%)	Δ (%)
FDLNet	ResNet-101	✓	53.21	
FDLNet-SENet	ResNet-101	✓	52.14	-1.07
w/o R_s	ResNet-101	✓	52.06	-1.15
w/o α	ResNet-101	✓	51.39	-1.82

named (SA). We show the results in Table IV, our learnable frequency encoder strategy achieves the best performance improvement of 1.31%. However, we can see that the performance of TOP is better than SA, which indicates that simply leveraging all frequency components fails to adapt to NTSP due to the diverse frequency distribution. Our method solves this issue by leveraging learnable frequency components.

4) *Ablation Studies on Spatial Frequency Fusion*: To verify the effectiveness of the SFF module, we conduct three experiments. (i) We use the structure of SENet [52] to replace the SFF module, which is named FDLNet-SENet. (ii) We only use frequency information to adjust the channel response without the aid of spatial information (w/o R_s). (iii) We verify the validity of the scale parameter alpha (w/o α). As shown in Table V, other alternative strategies degrade the model performance to varying degrees. Specifically, on the one hand, SENet leverages linear layers to adjust the channel response without the spatial features of the image, which can achieve good results in image classification but is not suitable for the NTSP task of spatial pixel-level classification. On the other hand, SFF leverages convolutional layers to adjust the channel response including the spatial structure information of the image, and achieves a 1.07% improvement over SENet.

The method that only utilizes frequency information has a similar structure to our SFF module, but its guiding effect is limited due to the lack of explicit spatial features. SFF utilizes features from two different domains (spatial and frequency) and outperforms the former by 1.15%. The scale parameter α reduces gradient instability during training since adjusting the frequency component channel responses with information from all channels is a computationally expensive task. α can be changed incrementally to mitigate the drastic gradient changes, without the parameters, the performance of the model drops by 1.82%.

5) *Improvements to Day-time Methods*: Our method can be applied to the existing day-time methods to adapt them for the NTSP task. For consistency comparison, we modify PSP, DeeplabV3+, and CCNet by using our method with the same experimental settings. The results in Table VI show that our method improves the NTSP performance of existing day-time methods while introducing minimum computational overhead.

Comparisons with Frequency Domain Adaption. In some domain adaptation methods [13] [12], the frequency information is used to perform style transformation on the image to reduce the gap between the source and target domains, which is also suitable for the style transformation of day-time and night-time images. For comparison, we use NightCity as the source domain and Cityscapes as the target domain, so night-time images are transformed into daytime-like images by Frequency Domain Adaption (FDA) [13]. Then, we use DeeplabV3+ to obtain the prediction results. To gain more accurate results, we use the resized labels of the validation set of NightCity+ (512×1024) for comparison.

Qualitative Comparison: We report the result in Figure 6 and observe that transforming night-time images to day-time images using FDA can reduce the domain gap between them to a certain extent (blue boxes). However, simply replacing the frequency information of the two images often fails. For example, in the white boxes of the sixth and eighth rows, the transformed images are severely distorted, resulting in incomplete prediction results and chaotic boundaries, while our model uses learnable frequency information to guide the network to predict more complete predictions and the boundaries are clearer. More comparison results are highlighted by orange boxes. These visual results show that our proposed method is more efficient than directly preprocessing the image.

Quantitative Comparison: For better comparison, we also report the results of Deeplabv3+ on the original NightCity, as shown in Table VII. The performance of FDA (51.33%) is even worse than the original method (53.26%), which means that simply preprocessing the image with frequency information does not solve the NTSP problem well. Whereas our method introduces learnable frequency information into the model, the network learns the frequency distribution of night images and achieves better results.

Compare with Day-time Dataset Cityscapes. Our method focuses on night-time scene parsing, because night-time scenes have two characteristics. First, the night scene contains information on all frequency components, including low-frequency areas with rich information and high-frequency areas with relatively little information. Moreover, the information contained

TABLE VI
IMPROVEMENTS TO DAY-TIME METHODS INCLUDING PERFORMANCE COMPARISON ON THREE DIFFERENT VALIDATION SETS AND COMPUTATION COMPARISON.

Method	Backbone	Parameters	FLOPs	NightCity	NightCity+	BDD100K-night
PSPNet	ResNet-101	70.12M	306.04G	51.02	52.24	19.62
FDLNet (PSPNet)	ResNet-101	71.83M	310.89G	53.21	54.25	24.50
DeeplabV3+	ResNet-101	63.98M	314.02G	51.99	53.26	23.42
FDLNet (DeeplabV3+)	ResNet-101	67.46M	335.21G	54.60	56.20	25.15
CCNet	ResNet-101	70.97M	329.55G	49.81	50.94	17.74
FDLNet (CCNet)	ResNet-101	72.68M	334.41G	51.00	52.27	21.82

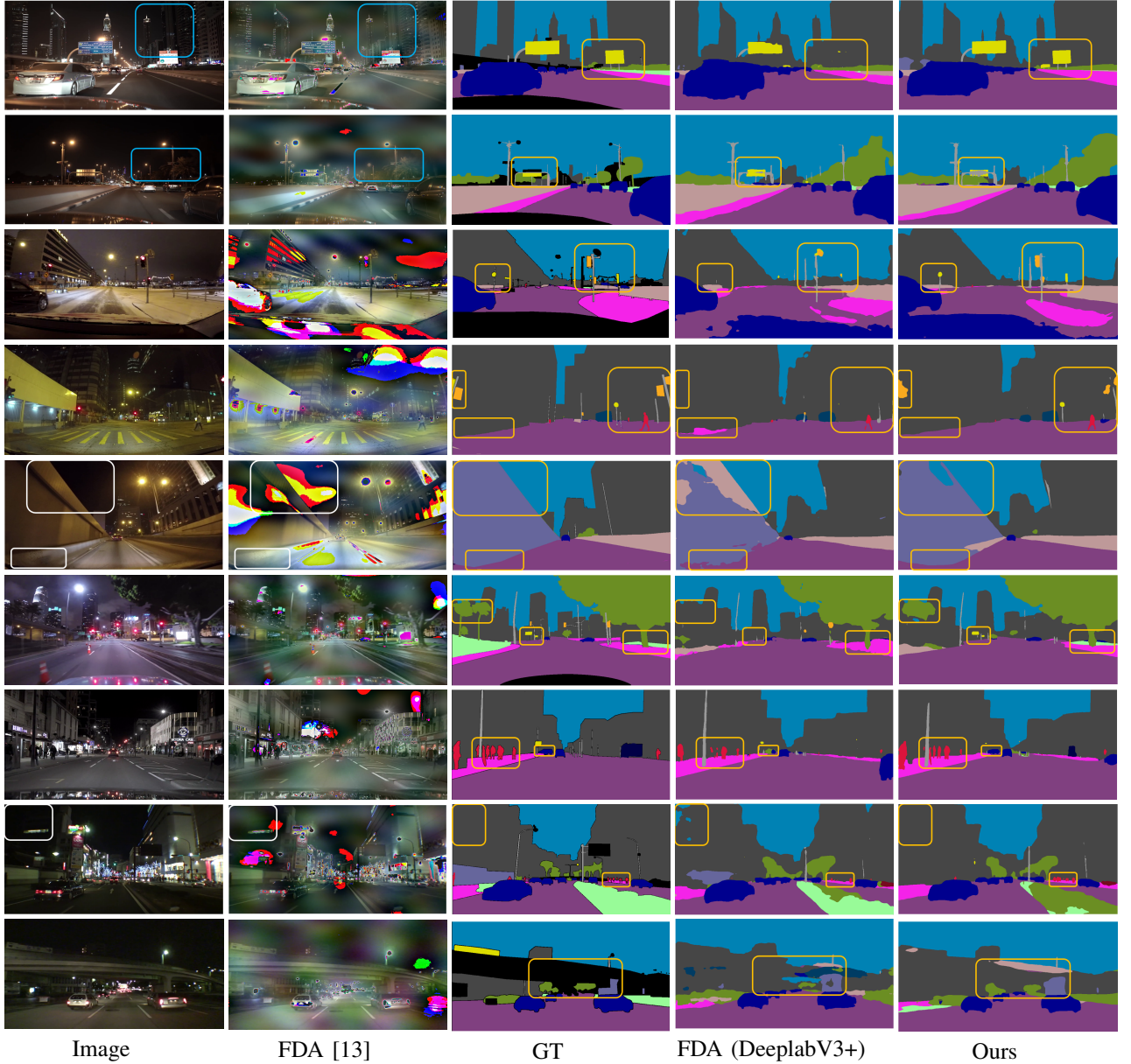


Fig. 6. Qualitative comparison on NightCity. The second column represents the style transformation of the image using Frequency Domain Adaption (FDA) [13], and the fourth column (FDA (DeeplabV3+)) represents the prediction results on the style transformed dataset by using DeeplabV3+ [6]. Successful cases of FDA are highlighted by blue boxes, and failure cases are highlighted by white boxes. Our advantages are highlighted by orange boxes.

in the high-frequency components of night-time images is richer than that of day-time images. Second, the frequency distribution of different night-time images is more different than day-time images, and the network can generate different

component weights by learning each image.

To explore the difference between our method on night-time and day-time images, we train models on Cityscapes and NightCity and perform quantitative comparisons on three dif-

TABLE VII

COMPARISON WITH FREQUENCY DOMAIN ADAPTION (FDA) ON NIGHTCITY. THE METHOD IS EVALUATED ON THE RESIZED NIGHTCITY+ VALIDATION SET. THE BEST RESULTS ARE MARKED IN **BOLD**

Method	road	side.	bulid.	wall	fence	pole	light	sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	moto.	bicy.	mIoU
DeeplabV3+	90.4	51.1	83.2	55.3	53.5	32.0	24.4	52.2	59.0	19.7	88.2	52.2	25.2	82.8	64.9	73.8	59.1	10.2	34.7	53.26
FDA (DeeplabV3+)	90.4	50.6	82.4	53.5	53.1	30.1	23.3	49.0	56.9	20.6	87.4	49.6	17.8	82.3	62.3	72.4	59.5	0	34.1	51.33
FDLNet (PSPNet)	90.5	50.8	83.2	55.9	53.1	28.6	24.8	51.6	59.1	21.1	87.9	50.6	25.2	82.6	63.1	75.1	60.4	28.9	38.3	54.25
FDLNet (DeeplabV3+)	91.2	53.1	83.8	58.3	54.4	34.1	30.1	57.2	60.1	22.2	88.2	55.9	27.6	84.6	61.3	73.8	58.8	29.2	44.0	56.20

TABLE VIII

COMPARISON WITH THE DAY-TIME DATASET.

Method	NighCity	NighCity+	Cityscapes
PSPNet	51.02	52.24	70.86
FDLNet(PSPNet)	53.21	54.25	72.42
$\Delta(\%)$	+2.19	+2.01	+1.56

ferent validation sets of Cityscapes, NightCity and NightCity+. The training settings are the same, except the learning rates are 0.005 and 0.01 for NightCity and Cityscapes, respectively. From Table VIII, we can see that our method achieves better results than baselines on both day-time and night-time datasets. The improvement is 2.19% on NighCity, 2.01% on NighCity+ and 1.56% on Cityscapes, which shows that our model is more effective on night-time images, and also shows the difference in frequency distribution between night-time and day-time images.

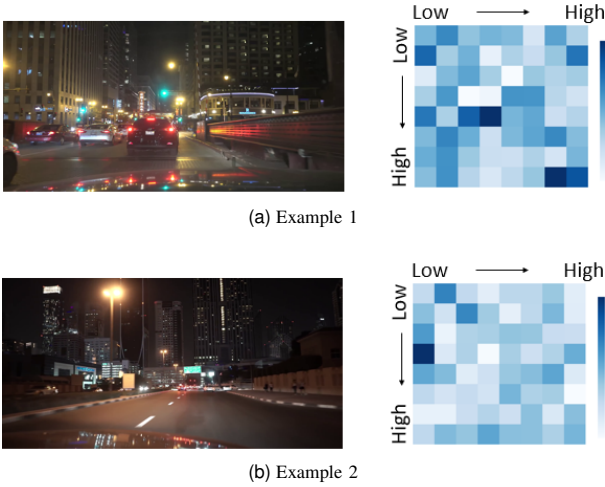


Fig. 7. Image-level LFE heatmap. (left) The source image. (right) In the LFE heatmap, the low frequency components are located in the upper left part and the high frequency components are located in the lower right part. Different images correspond to different frequency affinities.

F. Visual Analysis

To illustrate the capabilities of our proposed Learnable Frequency Encoder (LFE), we visualize the heatmap of LFE on the NightCity validation set.

1) *Image-level LFE*: Our proposed LFE is able to dynamically adjust the weight of each frequency component, which means that there are differences of the frequency component affinity of each image. To illustrate this, we feed different images into the network to obtain the heatmap. Figure 7 shows

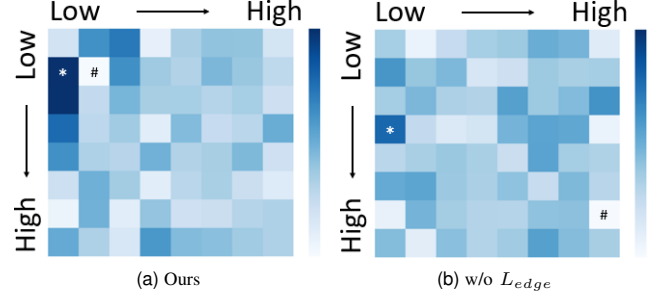


Fig. 8. Dataset-level LFE heatmap. (left) Our model. (right) Our model w/o L_{edge} . * represents the frequency component with the largest weight, and # represents the frequency component with the smallest weight.

that for different images, the weights of frequency components are also different. The frequency components with the largest weight in Figure 7a appear in the high-frequency regions, but the distribution of the frequency weights is relatively loose. While the frequency component with the largest weight is located in the low-frequency regions as shown in Figure 7b, and the distribution of frequency weights is concentrated in low-frequency regions relatively. This shows the frequency distribution in night-time scenes is diverse as we observe in Figure 1 and Figure 2.

2) *Dataset-level LFE*: To further analyze the LFE, we summed and averaged the LFE of all images in the validation set, resulting in a dataset-level LFE heatmap. As shown in Figure 7a, we can see that our model prefers low-frequency components. The maximum weight of frequency component (*) is located in the low-frequency part, and the large weights are also generally concentrated in the low-frequency part, which proves that CNN prefers to select the low-frequency region with rich information in extracting features as [17] [40]. However, the minimum weight of frequency component (#) is located in the low-frequency region rather than the high-frequency region, which reflects that the high-frequency information is equally important.

3) *LFE on semantic edge loss*: Since we use the semantic edge loss L_{edge} , which focuses on the prediction of high-frequency related to semantic edges, so we visualize the LFE heatmap for analysis to demonstrate the effectiveness of semantic edge loss. Note that we use the same color numeric intervals in Figure 7a to visualize the results. Figure 7b shows the results w/o L_{edge} . The maximum weight (*) is located in the low-frequency part, and the minimum weight (#) is located in the high-frequency part. In contrast to method w/ L_{edge} , whose minimum weight (#) appears in the low-frequency region. This shows that semantic edge loss strengthens the attention of

edge details to a certain extent. Furthermore, the model w/o L_{edge} overall has a looser selection of frequency components compared to the model w/ L_{edge} , which indicates the semantic edge loss enforces the network to extract frequency features more efficiently and reduce information redundancy.

V. CONCLUSION

In this paper, we propose a Frequency Domain Learning Network (FDLNet) to handle the frequency information distribution diversification of Night-Time Scene Parsing (NTSP). Specifically, the Learnable Frequency Encoder (LFE) adjusts the weights of frequency components generated by the DCT. Since high and low-frequency information is both important for NTSP, the encoder processes all frequency components information. Moreover, the encoder dynamically adjusts each frequency component to adapt to changes in the frequency distribution of night images. Furthermore, the Spatial Frequency Fusion module (SFF) leverages information from two different domains to guide the network segmentation since only utilizing frequency information lacks spatial context features that are important for NTSP. Besides, our method allows a simple modification of the day-time model to adapt it to night-time scenes. Our model achieves state-of-the-art performance on NightCity and competitive results on NightCity+ and BDD100K-night.

ACKNOWLEDGEMENTS

This work is partially supported by the National Natural Science Foundation of China (No. 61972157), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Science and Technology Commission (21511101200). We also appreciate the help of researchers other than the authors of the paper.

REFERENCES

- [1] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, "Deep learning-based image recognition for autonomous driving," *IATSS research*, vol. 43, no. 4, pp. 244–252, 2019.
- [2] P. Li, Y. Xu, Y. Wei, and Y. Yang, "Self-correction for human parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [3] L. Liao, J. Xiao, Z. Wang, C.-W. Lin, and S. Satoh, "Image inpainting guided by coherence priors of semantics and textures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6539–6548, 2021.
- [4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.
- [5] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.
- [6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.
- [7] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 603–612, 2019.
- [8] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6881–6890, 2021.
- [9] D. Dai and L. Van Gool, "Dark model adaptation: Semantic image segmentation from daytime to nighttime," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3819–3824, IEEE, 2018.
- [10] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7374–7383, 2019.
- [11] X. Wu, Z. Wu, H. Guo, L. Ju, and S. Wang, "Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15769–15778, 2021.
- [12] Q. Xu, Y. Ma, J. Wu, C. Long, and X. Huang, "Cdada: A curriculum domain adaptation for nighttime semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2962–2971, 2021.
- [13] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4085–4095, 2020.
- [14] X. Tan, K. Xu, Y. Cao, Y. Zhang, L. Ma, and R. W. Lau, "Night-time scene parsing with a large real dataset," *IEEE Transactions on Image Processing*, vol. 30, pp. 9085–9098, 2021.
- [15] X. Deng, P. Wang, X. Lian, and S. Newsam, "Nightlab: A dual-level architecture with hardness detection for segmentation at night," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16938–16948, 2022.
- [16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.
- [17] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, "Learning in the frequency domain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1740–1749, 2020.
- [18] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [20] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
- [21] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 267–283, 2018.
- [22] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3146–3154, 2019.
- [23] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.
- [24] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," in *European Conference on Computer Vision*, pp. 108–126, Springer, 2020.
- [25] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.
- [26] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.
- [27] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7262–7272, 2021.
- [28] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12077–12090, 2021.
- [29] S. Choi, J. T. Kim, and J. Choo, "Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks," in *Pro-*

- ceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9373–9383, 2020.
- [30] Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, I. B. Daya, Z. Li, W.-S. Zheng, J. Li, and A. Wong, “Squeeze-and-attention networks for semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13065–13074, 2020.
 - [31] Z. Jin, B. Liu, Q. Chu, and N. Yu, “Isnet: Integrate image-level and semantic-level context for semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7189–7198, 2021.
 - [32] L. Zhu, D. Ji, S. Zhu, W. Gan, W. Wu, and J. Yan, “Learning statistical texture for semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12537–12546, 2021.
 - [33] Z. Jin, T. Gong, D. Yu, Q. Chu, J. Wang, C. Wang, and J. Shao, “Mining contextual information beyond image for semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7231–7241, 2021.
 - [34] Z. Feng, Q. Zhou, Q. Gu, X. Tan, G. Cheng, X. Lu, J. Shi, and L. Ma, “Dmt: Dynamic mutual training for semi-supervised learning,” *Pattern Recognition*, vol. 130, p. 108777, 2022.
 - [35] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi-level wavelet-cnn for image restoration,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 773–782, 2018.
 - [36] B. Zheng, S. Yuan, C. Yan, X. Tian, J. Zhang, Y. Sun, L. Liu, A. Leonardis, and G. Slabaugh, “Learning frequency domain priors for image demoirising,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
 - [37] W. Chen, J. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen, “Compressing convolutional neural networks in the frequency domain,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1475–1484, 2016.
 - [38] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, “Faster neural networks straight from jpeg,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
 - [39] M. Ehrlich and L. S. Davis, “Deep residual learning in the jpeg transform domain,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
 - [40] Z. Qin, P. Zhang, F. Wu, and X. Li, “Fcanet: Frequency channel attention networks,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 783–792, 2021.
 - [41] X. Zou, F. Xiao, Z. Yu, and Y. Lee, “Delving deeper into anti-aliasing in convnets,” in *Proceedings of the British Machine Vision Conference (BMVC)*, 2020, 2020.
 - [42] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, “Boundary-aware feature propagation for scene segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6819–6829, 2019.
 - [43] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, “Gated-scnn: Gated shape cnns for semantic segmentation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5229–5238, 2019.
 - [44] Y. Yuan, J. Xie, X. Chen, and J. Wang, “Segfix: Model-agnostic boundary refinement for segmentation,” in *European Conference on Computer Vision*, pp. 489–506, Springer, 2020.
 - [45] M. Haoxiang, Y. Hongyu, and D. Huang, “Boundary guided context aggregation for semantic segmentation,” in *The British Machine Vision Conference (BMVC)*, November 2021.
 - [46] S. Borse, Y. Wang, Y. Zhang, and F. Porikli, “Inverseform: A loss function for structured boundary-aware segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5901–5911, 2021.
 - [47] X. Li, X. Li, L. Zhang, G. Cheng, J. Shi, Z. Lin, S. Tan, and Y. Tong, “Improving semantic segmentation via decoupled body and edge supervision,” in *European Conference on Computer Vision*, pp. 435–452, Springer, 2020.
 - [48] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “Bdd100k: A diverse driving dataset for heterogeneous multitask learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2636–2645, 2020.
 - [49] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
 - [50] J. Li, S. Zha, C. Chen, M. Ding, T. Zhang, and H. Yu, “Attention guided global enhancement and local refinement network for semantic segmentation,” *IEEE Transactions on Image Processing*, vol. 31, pp. 3211–3223, 2022.
 - [51] A. Shrivastava, A. Gupta, and R. Girshick, “Training region-based object detectors with online hard example mining,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.
 - [52] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.