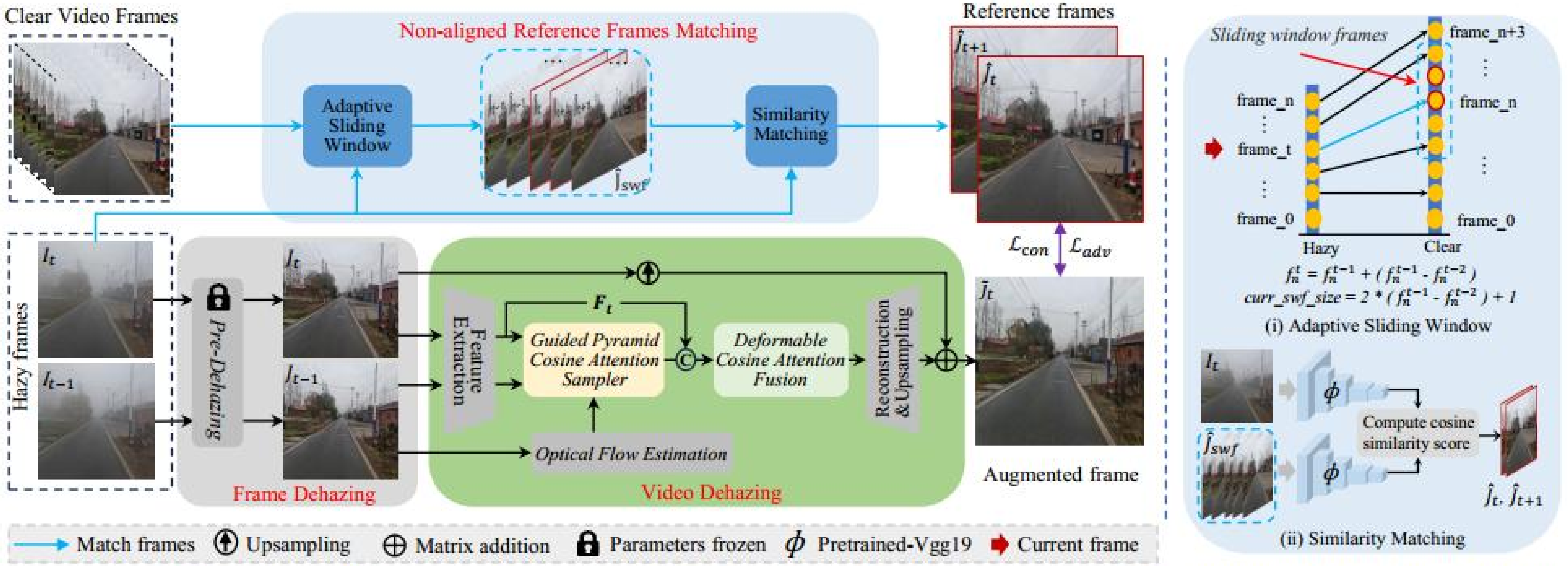
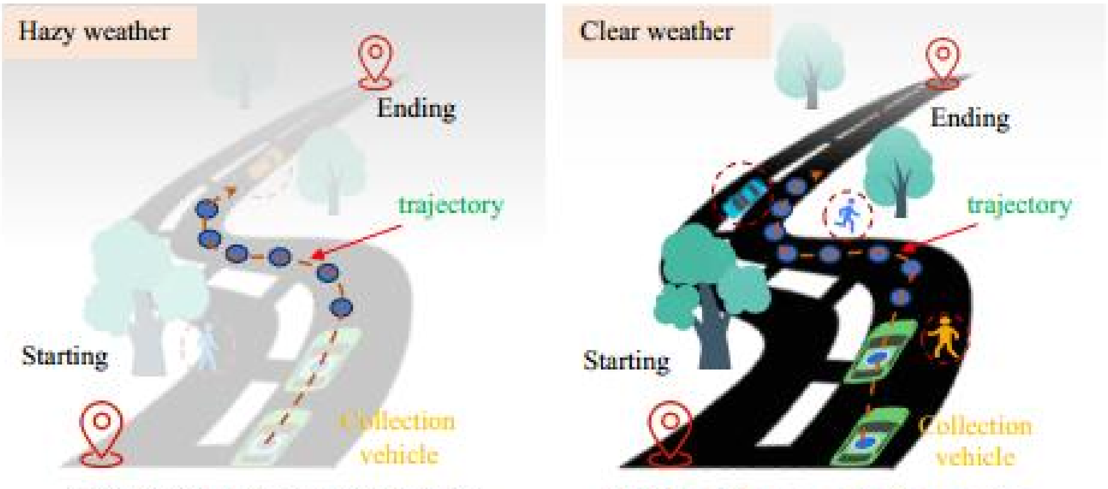
Driving-Video Dehazing with Non-Aligned Regularization for Safety Assistance

Junkai Fan[1], Jiangwei Weng[1], Kun, Wang[1], Yijun Yang[2], Jun Li[1]*, and Jian Yang[1]*
[1]PCA Lab† Nanjing University of Science and Technology, China
[2]The Hong Kong University of Science and Technology (Guangzhou)



(a) Non-aligned video supervision dehazing and visual enhancement framework (b) Non-aligned Reference Frames Matching

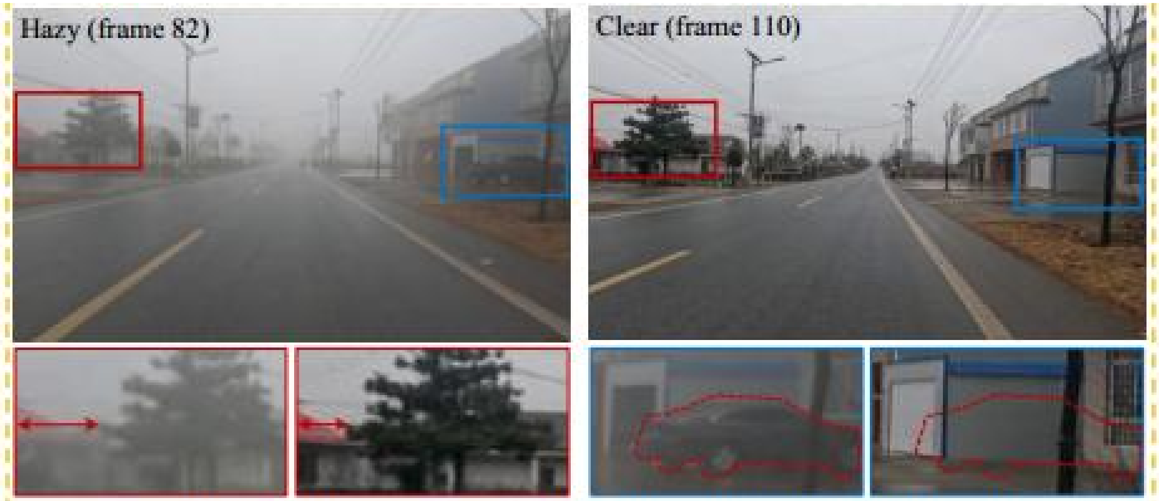
cvpr24



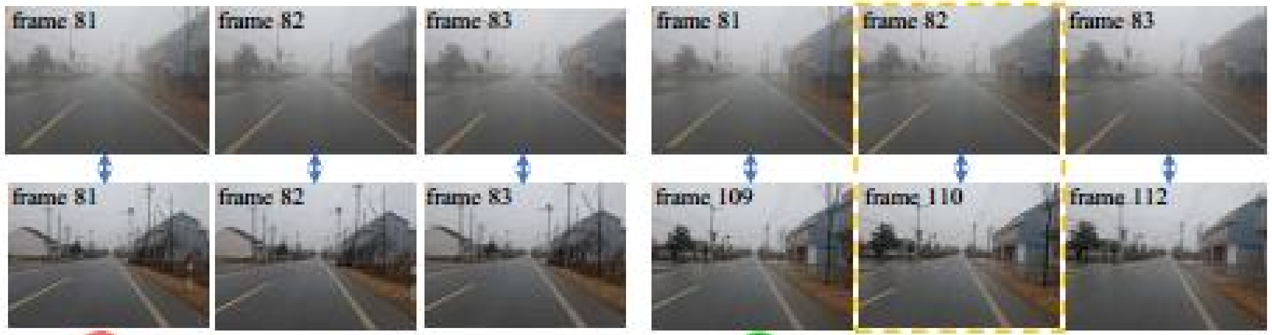
(i) Collecting scenes in hazy weather

(ii) Collecting scenes in clear weather

(a) Temporal-spatial misalignment due to avoiding pedestrians and vehicles.



(c) Non-aligned video frame pairs (i.e., spatial misalignment)

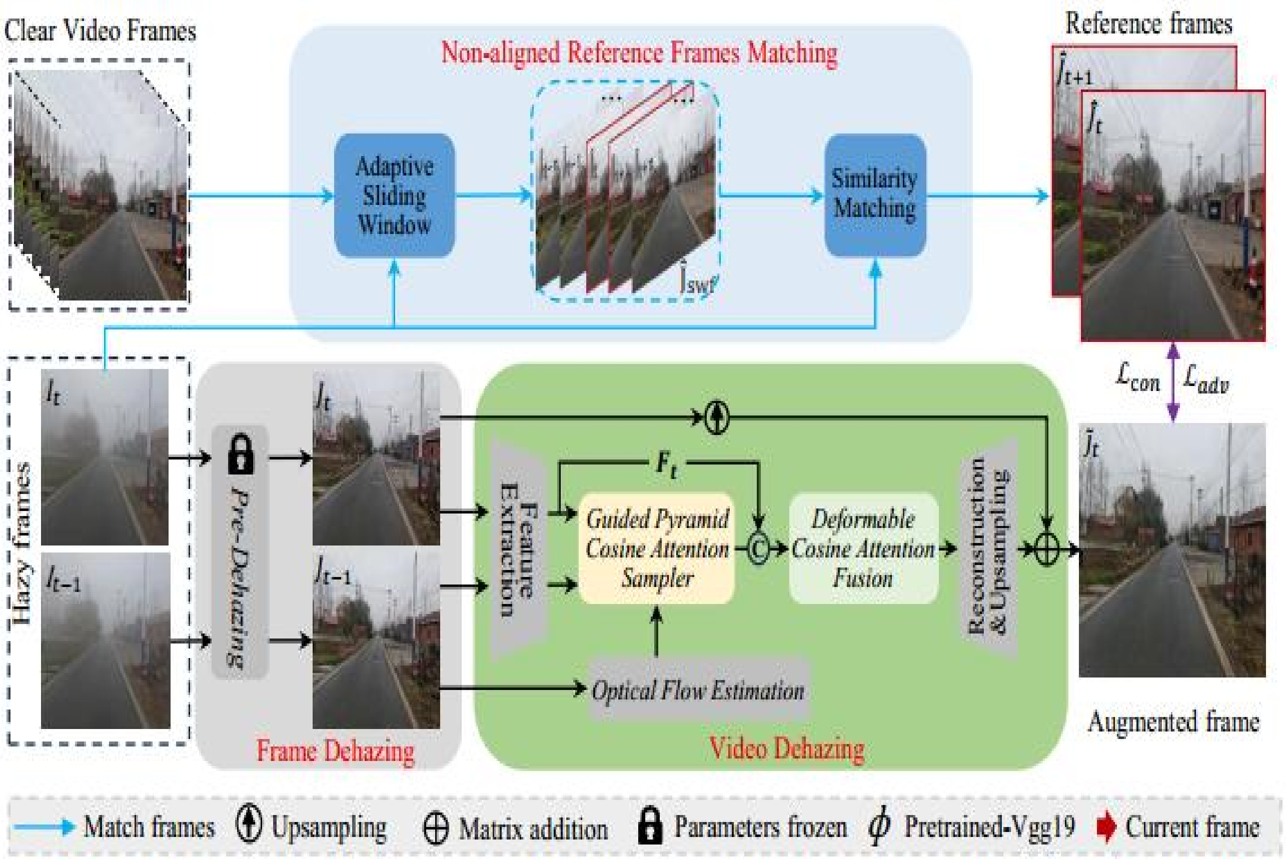


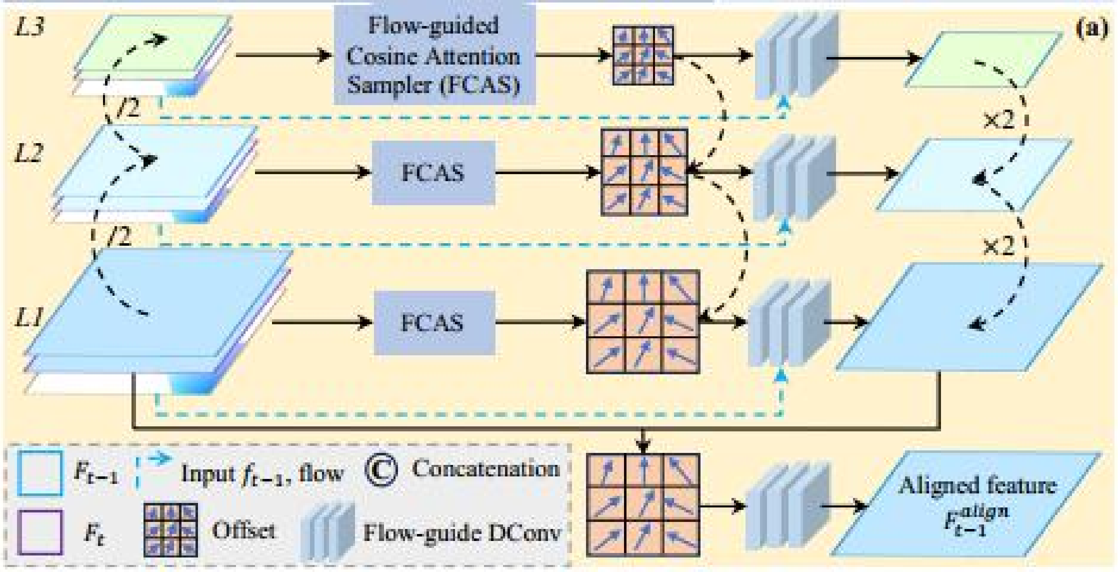
❌ *W/o Matched video frame pairs*　✅ *We matched video frame pairs*

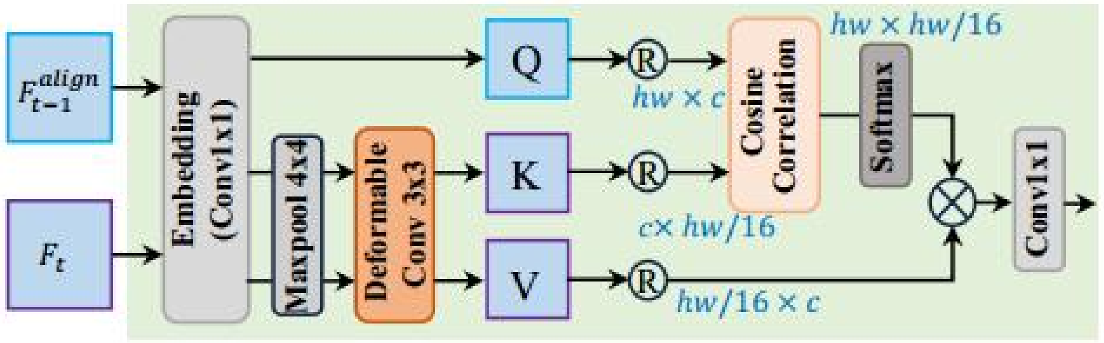
(b) Temporal misalignment in real-world scene hazy video pairs

收集有雾/清晰视频对的两个问题：

- 视频对中的时间错位
- 视频对中的空间错位



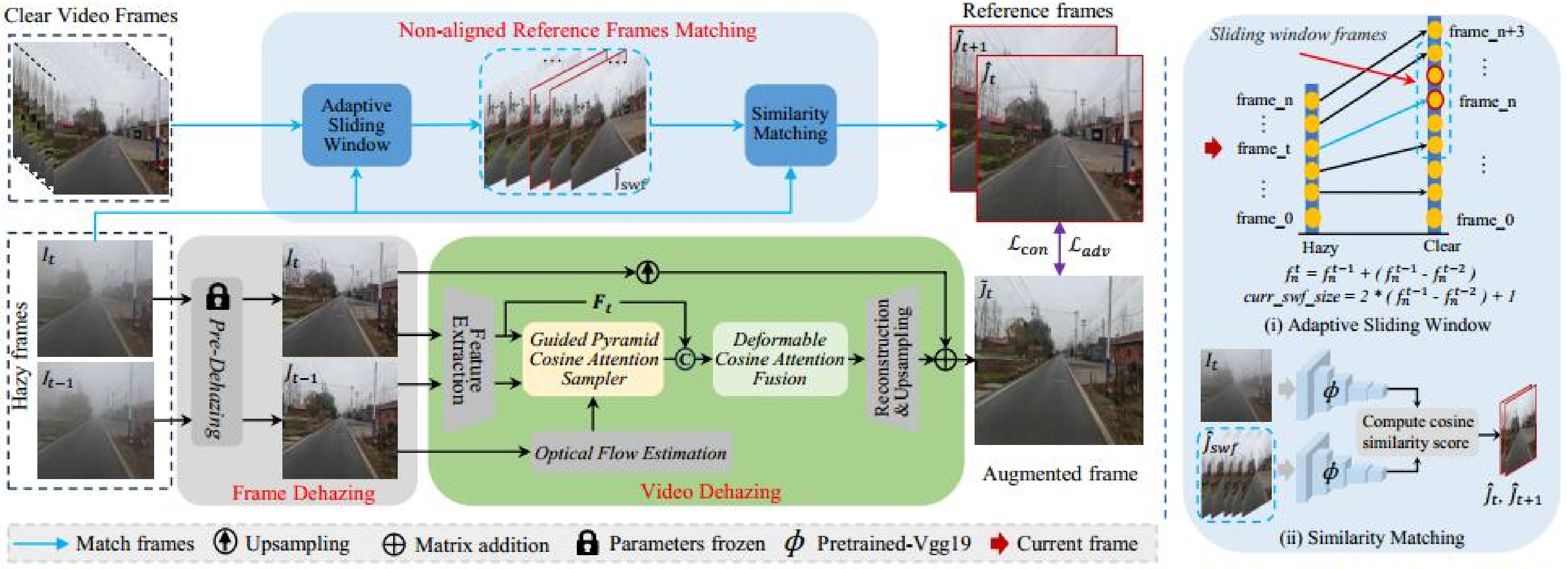




- 第一个为真实驾驶视频去模糊任务提出非对齐正则化策略。

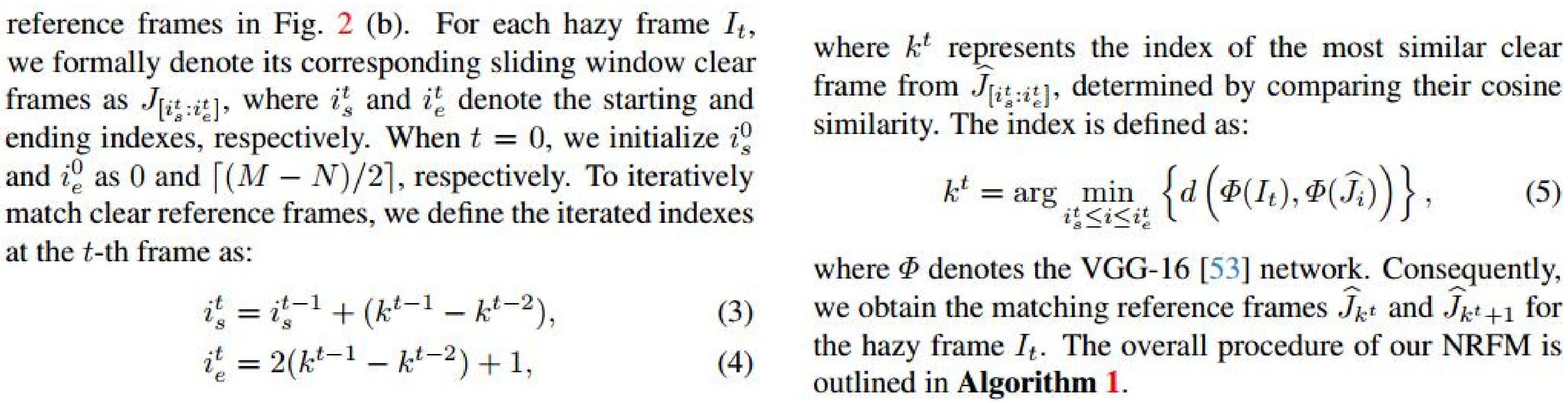
- 提出一种视频去雾网络，该网络设计了光流引导余弦注意力采样器和可变形余弦注意力融合。

- 提供了一个真实的有雾视频数据集，其中包括27个非对齐的有雾/清晰视频对，总共4256个匹配的有雾/清楚帧对。这些配对是在各种真实场景（即乡村和城市道路）中使用GoPro相机收集。



(a) Non-aligned video supervision dehazing and visual enhancement framework

(b) Non-aligned Reference Frames Matching



reference frames in Fig. 2 (b). For each hazy frame I_t, we formally denote its corresponding sliding window clear frames as $J_{[i_s^t:i_e^t]}$, where i_s^t and i_e^t denote the starting and ending indexes, respectively. When $t = 0$, we initialize i_s^0 and i_e^0 as 0 and $\lceil (M - N)/2 \rceil$, respectively. To iteratively match clear reference frames, we define the iterated indexes at the t-th frame as:

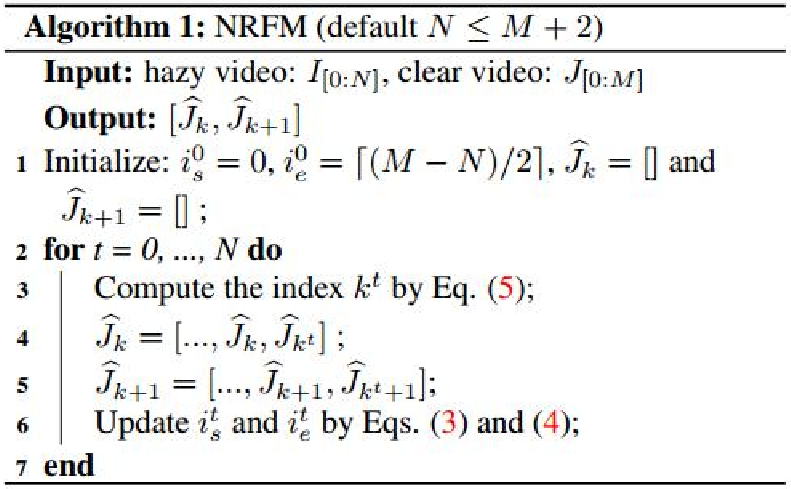
$$i_s^t = i_s^{t-1} + (k^{t-1} - k^{t-2}), \quad (3)$$

$$i_e^t = 2(k^{t-1} - k^{t-2}) + 1, \quad (4)$$

where k^t represents the index of the most similar clear frame from $\hat{J}_{[i_s^t:i_e^t]}$, determined by comparing their cosine similarity. The index is defined as:

$$k^t = \arg \min_{i_s^t \le i \le i_e^t} \left\{ d\left(\Phi(I_t), \Phi(\hat{J}_i)\right)\right\}, \quad (5)$$

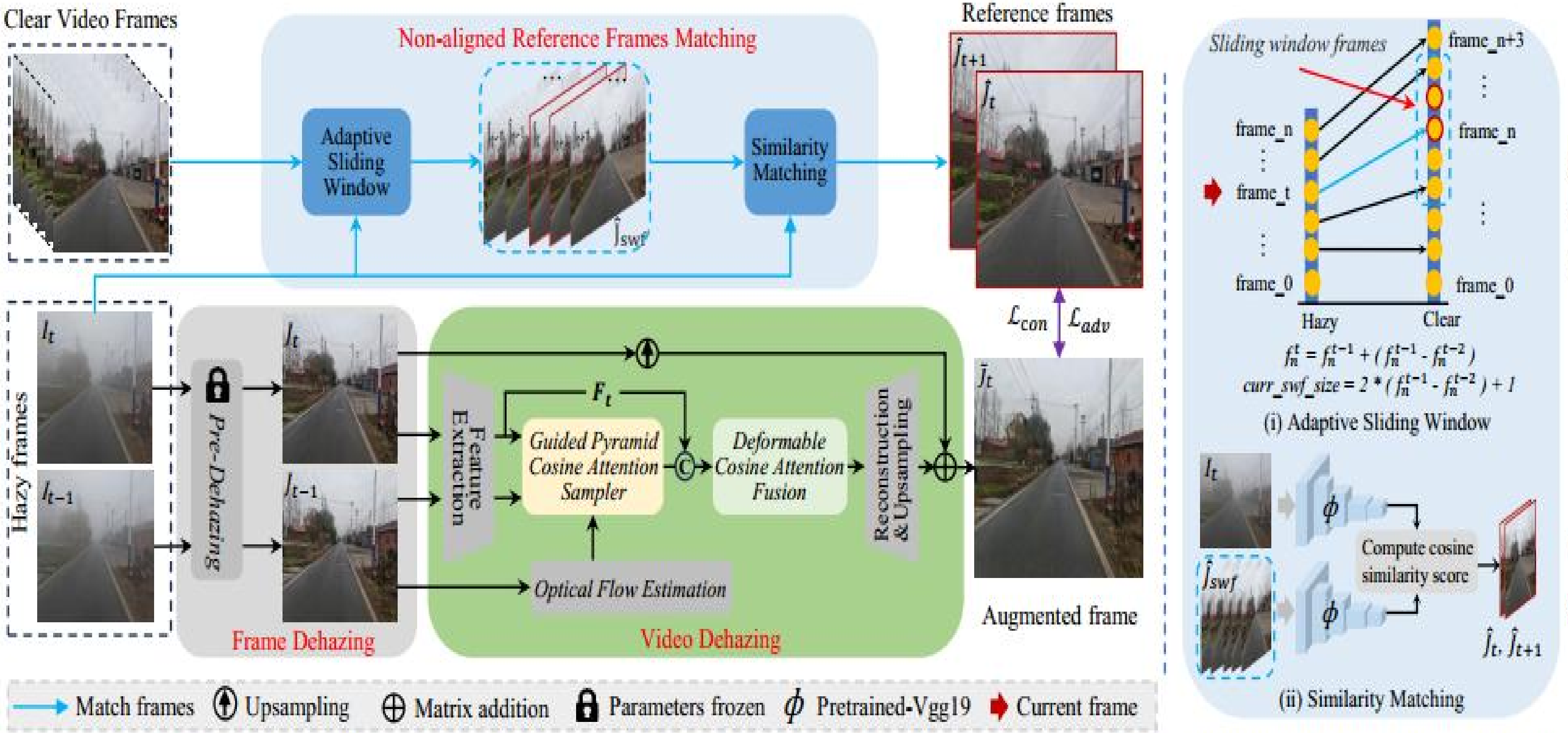
where Φ denotes the VGG-16 [53] network. Consequently, we obtain the matching reference frames $\hat{J}_{k^t}$ and $\hat{J}_{k^t+1}$ for the hazy frame I_t. The overall procedure of our NRFM is outlined in **Algorithm 1**.



Algorithm 1: NRFM (default $N \le M + 2$)

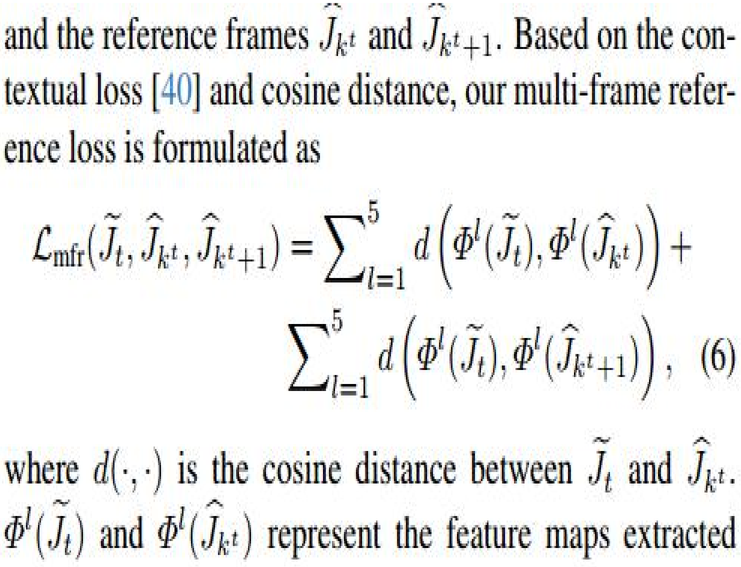
```
Input: hazy video: $I_{[0:N]}$, clear video: $J_{[0:M]}$
Output: $[\hat{J}_k, \hat{J}_{k+1}]$
Initialize: $i_s^0 = 0$, $i_e^0 = \lceil (M - N)/2 \rceil$, $\hat{J}_k = []$ and
    $\hat{J}_{k+1} = []$ ;
for $t = 0, ..., N$ do
    Compute the index $k^t$ by Eq. (5);
    $\hat{J}_k = [..., \hat{J}_k, \hat{J}_{k^t}]$ ;
    $\hat{J}_{k+1} = [..., \hat{J}_{k+1}, \hat{J}_{k^t+1}]$;
    Update $i_s^t$ and $i_e^t$ by Eqs. (3) and (4);
end
```

对于有雾视频，主要目标是建立其相应的清晰和不对齐的参考帧，这些参考帧作为视频去雾网络的约束。



(a) Non-aligned video supervision dehazing and visual enhancement framework

(b) Non-aligned Reference Frames Matching

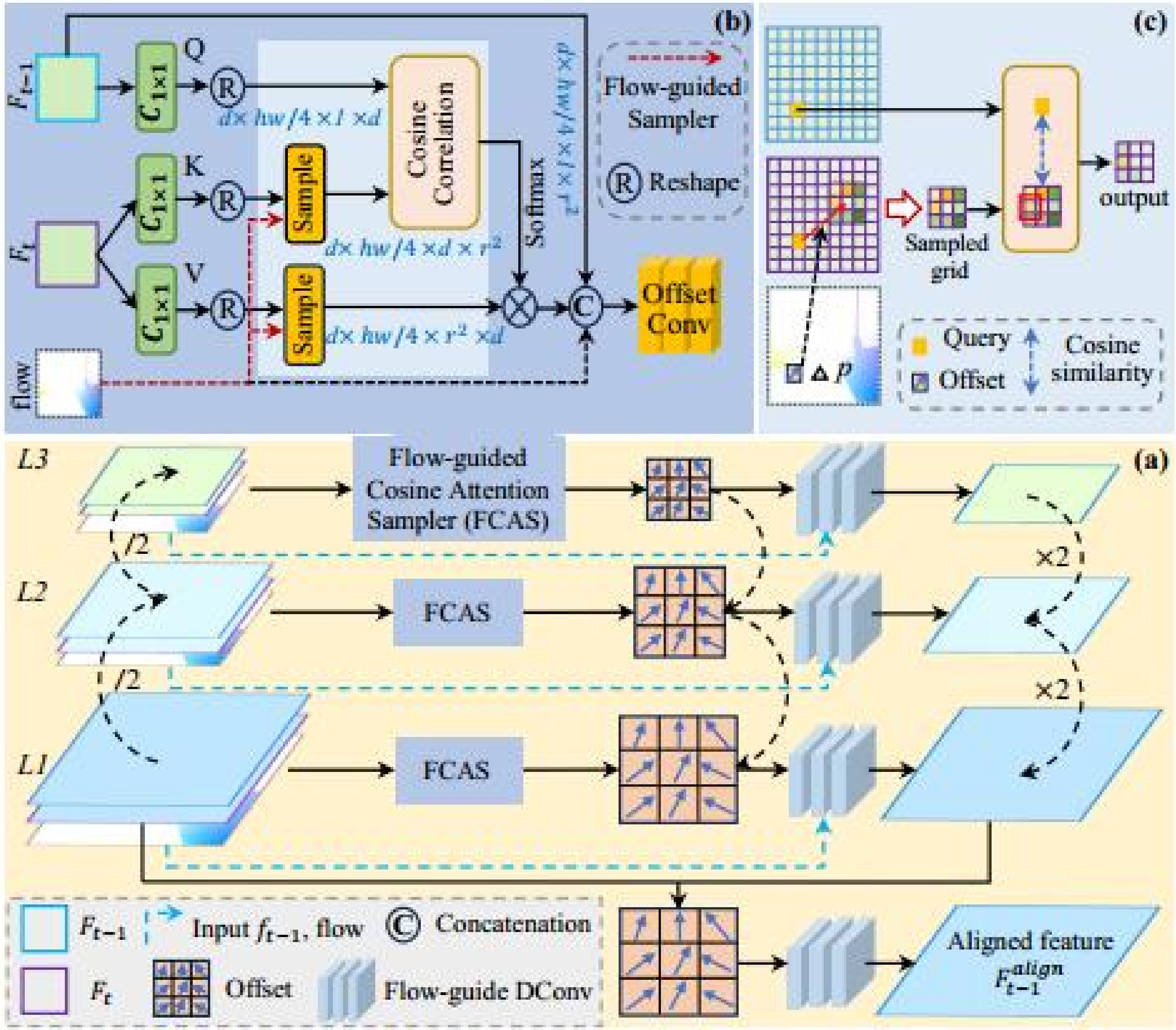


and the reference frames $\hat{J}_{k^t}$ and $\hat{J}_{k^t+1}$. Based on the contextual loss [40] and cosine distance, our multi-frame reference loss is formulated as

$$\mathcal{L}_{\mathrm{mfr}}(\tilde{J}_t, \hat{J}_{k^t}, \hat{J}_{k^t+1}) = \sum_{l=1}^{5} d\left(\Phi^l(\tilde{J}_t), \Phi^l(\hat{J}_{k^t})\right) + \sum_{l=1}^{5} d\left(\Phi^l(\tilde{J}_t), \Phi^l(\hat{J}_{k^t+1})\right), \quad (6)$$

where $d(\cdot,\cdot)$ is the cosine distance between $\tilde{J}_t$ and $\hat{J}_{k^t}$. $\Phi^l(\tilde{J}_t)$ and $\Phi^l(\hat{J}_{k^t})$ represent the feature maps extracted

对于有雾视频，主要目标是建立其相应的清晰和不对齐的参考帧，这些参考帧作为视频去雾网络的约束。



$$\Delta p = (u, v) = \phi_{spy}(I_{t-1}, I_t)(x, y). \tag{7}$$

The set of sampled grid coordinates is expressed as

$$\Omega(p')_k = \{p' + e \mid e \in \mathbb{Z}^2, ||e||_1 \leq (k-1)/2\}, \tag{8}$$

where k represents the sampling kernel size and $\mathbb{Z}^2$ denotes a two-dimensional space. Linear projected query vectors $Q_{x,y} = F_{t-1}W^q$, key vectors $K_{x,y} = F_tW^k$, and value vectors $V_{x,y} = F_tW^v$ at coordinate $p = (x, y)$ of F_{t-1} and F_t are defined using the parameters W^q, W^k, and $W^v \in \mathbb{R}^{C\times d}$, where d is the dimension of the projected vector.

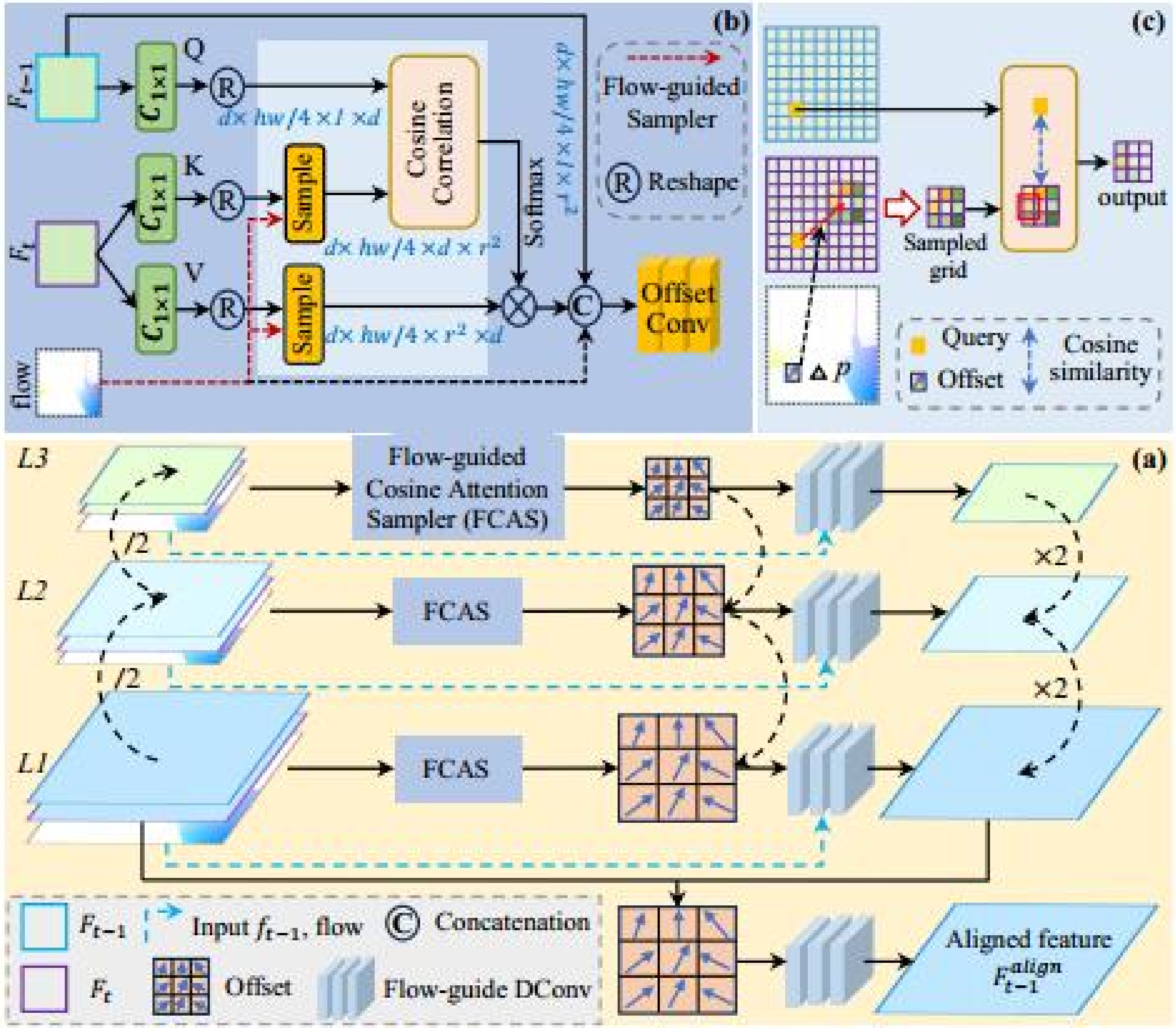


Fig. 3 (c) illustrates the use of the coarse $O_{t-1\to t}$ to guide learnable sampling from $K_{x,y}$ and $V_{x,y}$, expanding the receptive field for cosine correlation calculations to enhance accuracy. Within the sampled grid coordinates, the sampling key and value elements are described as

$$\{K_{i,j}, V_{i,j} \mid (i,j)\in\Omega(p')_k\} = \mathcal{S}(K_{x,y}, V_{x,y}), \quad (9)$$

where $\mathcal{S}$ denotes the interpolation sampling. The cosine attention $F_{\text{attn}} \in \mathbb{R}^{HW/4\times 1\times k^2}$ is then computed by

$$F_{\text{attn}} = \sum_{(i,j)\in\Omega(p')_k} F_{\text{softmax}}\left(\frac{Q_{x,y}^T K_{i,j}}{|Q_{x,y}||K_{i,j}|\sqrt{d}}\right) V_{i,j}, \quad (10)$$

where d is the dimension of the projected vector. Finally, the output offset is computed as

$$o_{t-1\to t} = \text{Conv}\left(\text{Cat}(F_{t-1}, F_{attn}, O_{t-1\to t})\right), \quad (11)$$

where Cat represents the concatenation operation, and $o_{t-1\to t}$ is the offset map between F_{t-1} and F_t.

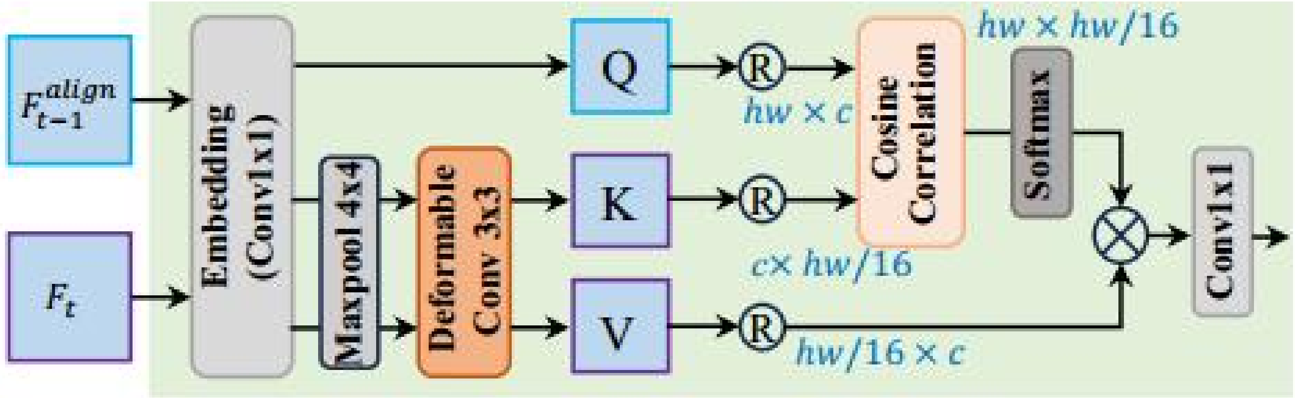


Figure 4. Overview of proposed DCAF. Enhancing cosine correlation for pixel misalignment robustness by expanding the receptive field with DConv, thereby improving cosine fusion performance.

They are computed by

$$\widetilde{Q}_{t-1}^{\text{align}} = \mathcal{M}(C_1(Q_{t-1}^{\text{align}})), \tag{12}$$

$$\widetilde{K}_t = \text{DConv}(\mathcal{M}(C_1(K_t))), \tag{13}$$

$$\widetilde{V}_t = \text{DConv}(\mathcal{M}(C_1(V_t))). \tag{14}$$

Next, we use the Eq. (10) to calculate the cosine correlation, and obtain the fused feature $F_{\text{fusion}} \in \mathbb{R}^{C\times H\times W}$.

3.3. Training Loss

For frame dehazing, we exclusively utilize the pre-trained NSDNet [15], please refer to its training loss for details. Now, let's focus on elucidating the training loss for video dehazing, which is expressed as follows:

$$\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{mfr}} + \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{cr}}, \tag{15}$$

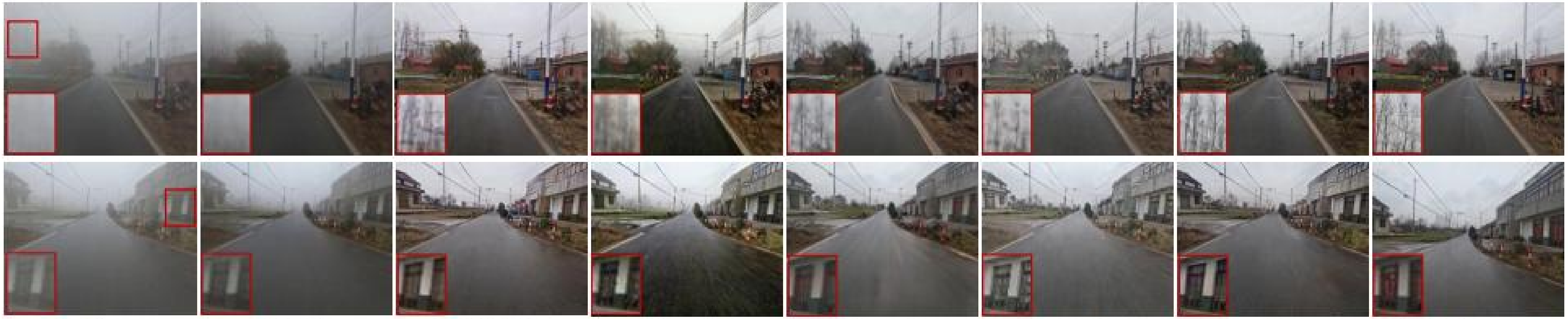
$\mathcal{L}_{\text{adv}}$ represents the adversarial loss [17], and $\mathcal{L}_{\text{mfr}}$ corresponds to the multi-frames reference loss as defined in Eq. (6). Since we lack the ground truth for the aligned feature F_{t-1}^{align}, we optimize the guided pyramid cosine attention sampler (GPCAS) module by using the current frame feature F_t as the label. Our objective is to minimize the discrepancy between F_{t-1}^{align} and F_t, expressed as $\mathcal{L}_{\text{align}} = ||F_{t-1}^{\text{align}} - F_t||_1$. Inspired by [11], we introduce a self-supervised temporal consistency regularization to ensure the consistency (*i.e.*, color and brightness) of pixels between consecutive frames. It can be formulated as:

$$\mathcal{L}_{\text{cr}} = ||M \odot (\mathcal{W}_{t\to t-1}(\widetilde{J}_t, \mathcal{O}_{t\to t-1}) - \widetilde{J}_{t-1}||_1, \tag{16}$$

where M is the occlusion map, $\mathcal{W}$ represents the flow-based image warp [49] for pixel alignment based on optical flow $\mathcal{O}_{t\to t-1}$, and $\widetilde{J}_{t-1}$ is the previous output frame.

Data Settings	Methods	Data Type	GoProHazy		DrivingHazy (NoRef)			InternetHazy (Only testing)			Params (M)	Flops (G)	Ref.
			FADE ↓	NIQE ↓	FADE ↓	NIQE ↓	Votes ↑	FADE ↓	NIQE ↓	Votes ↑			
Unpaired	DCP [20]	Image	0.9835	5.8309	0.9692	5.6799	-	0.9223	6.4744	-	-	-	CVPR'09
	RefineNet [72]	Image	1.5694	5.3693	1.1837	5.5500	-	1.1801	5.8742	-	11.38	75.41	TIP'21
	CDD-GAN [6]	Image	1.1942	4.9787	1.4423	5.0349	-	1.2120	5.1049	-	29.27	56.89	ECCV'22
	D^4 [63]	Image	1.9272	5.7865	1.8658	5.6864	-	1.3277	6.2150	-	**10.70**	**2.25**	CVPR'22
Paired	PSD [7]	Image	1.0529	6.0010	0.9672	5.3520	-	0.9275	5.2187	-	33.11	182.5	CVPR'21
	RIDCP [59]	Image	0.8010	4.6640	1.1077	4.3889	0.315	0.9391	4.6610	0.265	28.72	182.69	CVPR'23
	PM-Net [38]	Video	1.1011	4.1211	0.9434	3.8944	0.220	1.1517	4.0590	0.150	151.20	5.22	ACMM'22
	MAP-Net [60]	Video	1.0611	4.2359	1.0440	4.2542	0.025	1.2130	5.3241	0.030	28.80	8.21	CVPR'23
Non-aligned	NSDNet [15]	Image	0.7996	4.1547	0.9348	4.0529	-	0.8934	4.3835	-	11.38	56.86	arXiv'23
	DVD (Ours)	Video	**0.7598**	**3.7753**	**0.8207**	**3.5825**	**0.440**	**0.8745**	**3.7480**	**0.555**	15.37	73.12	-

Table 1. Quantitative results on three real-world hazy video datasets. ↓ denotes the lower the better. ↑ denotes the higher the better. Due to PM-Net and MAP-Net rely on GT for training, we use $\mathcal{L}_{cx}$ to train them on GoProHazy dataset. Note that we only selected the latest dehazing methods (*i.e.*, RIDCP, PM-Net and MAP-Net) and our DVD for the user study. Moreover, DrivingHazy and InternetHazy were tested on dehazing models trained using GoProHazy and pre-trained dehazing models provided by the authors, respectively.



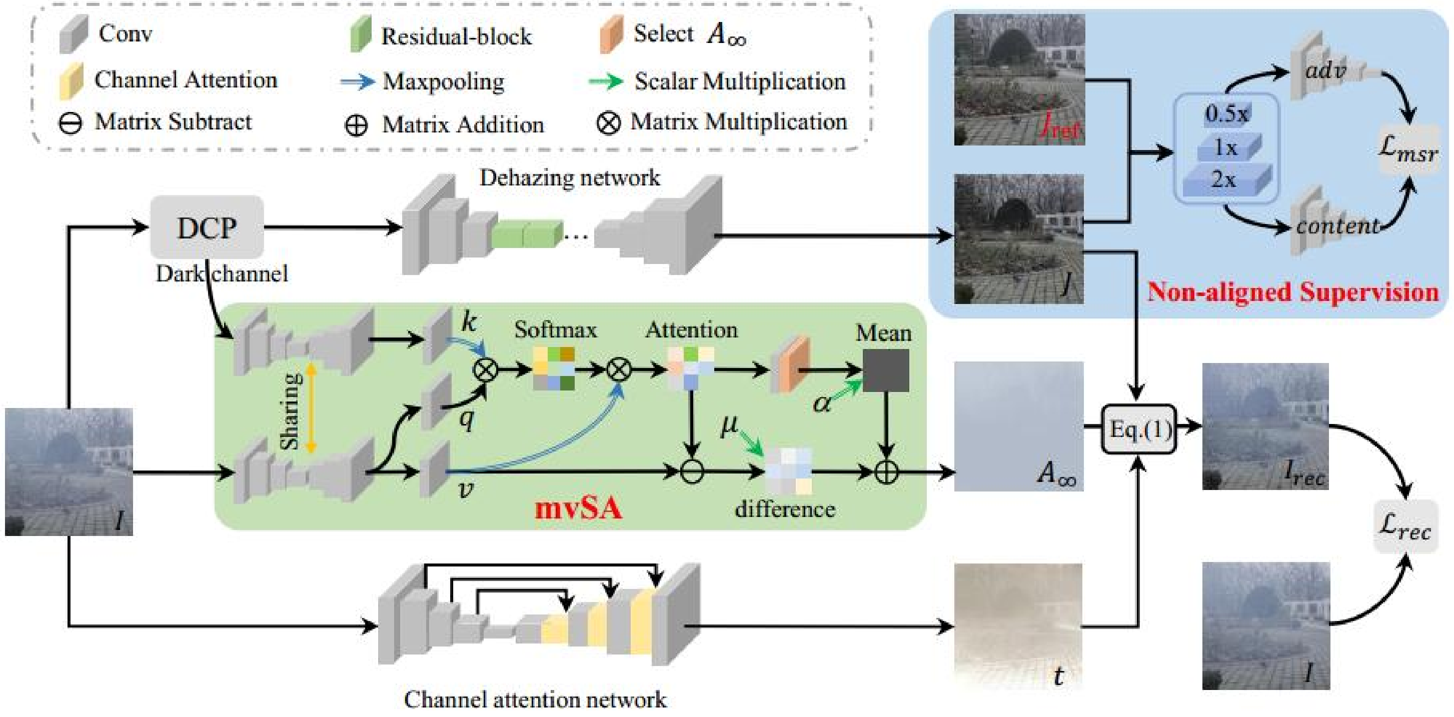
(a) Hazy (b) D4 (c) NSDNet (d) RIDCP (e) PM-Net (f) MAP-Net (g) Ours (h) Reference

Figure 7. Comparison of video dehazing results on GoProHazy. Our method effectively removes distant haze.

Non-aligned Supervision for Real Image Dehazing

Junkai Fan[1], Fei Guo[1], Jianjun Qian[1], Xiang Li[2], Jun Li[1*], Jian Yang[1*]

[1]Key Lab of Intelligent Perception and Systems for HighDimensional Information of Ministry of Education
Jiangsu Key Lab of Image and Video Understanding for Social Security,
PCA Lab, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China.
[2]College of Computer Science, Nankai University, Tianjin, China.

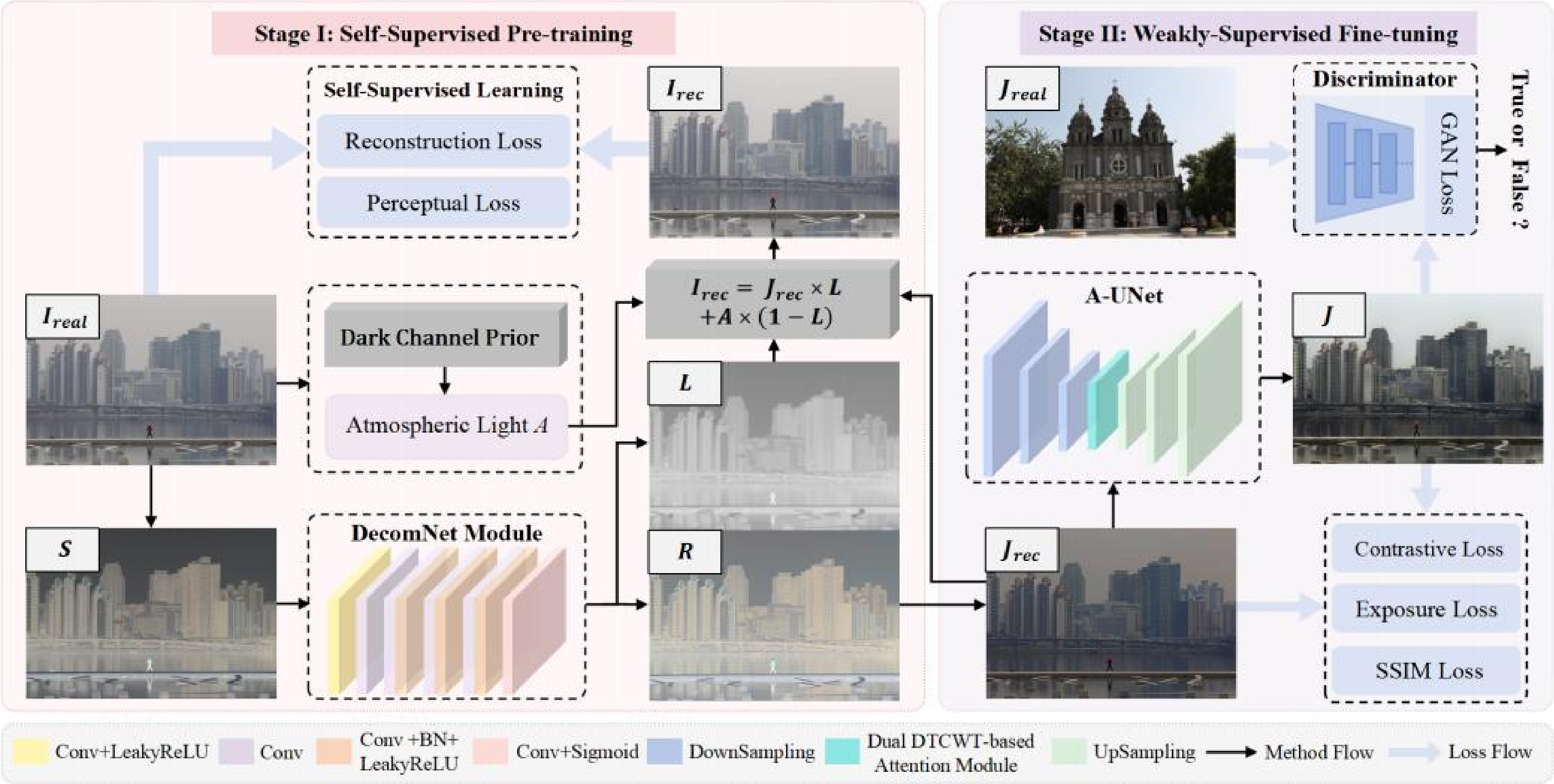


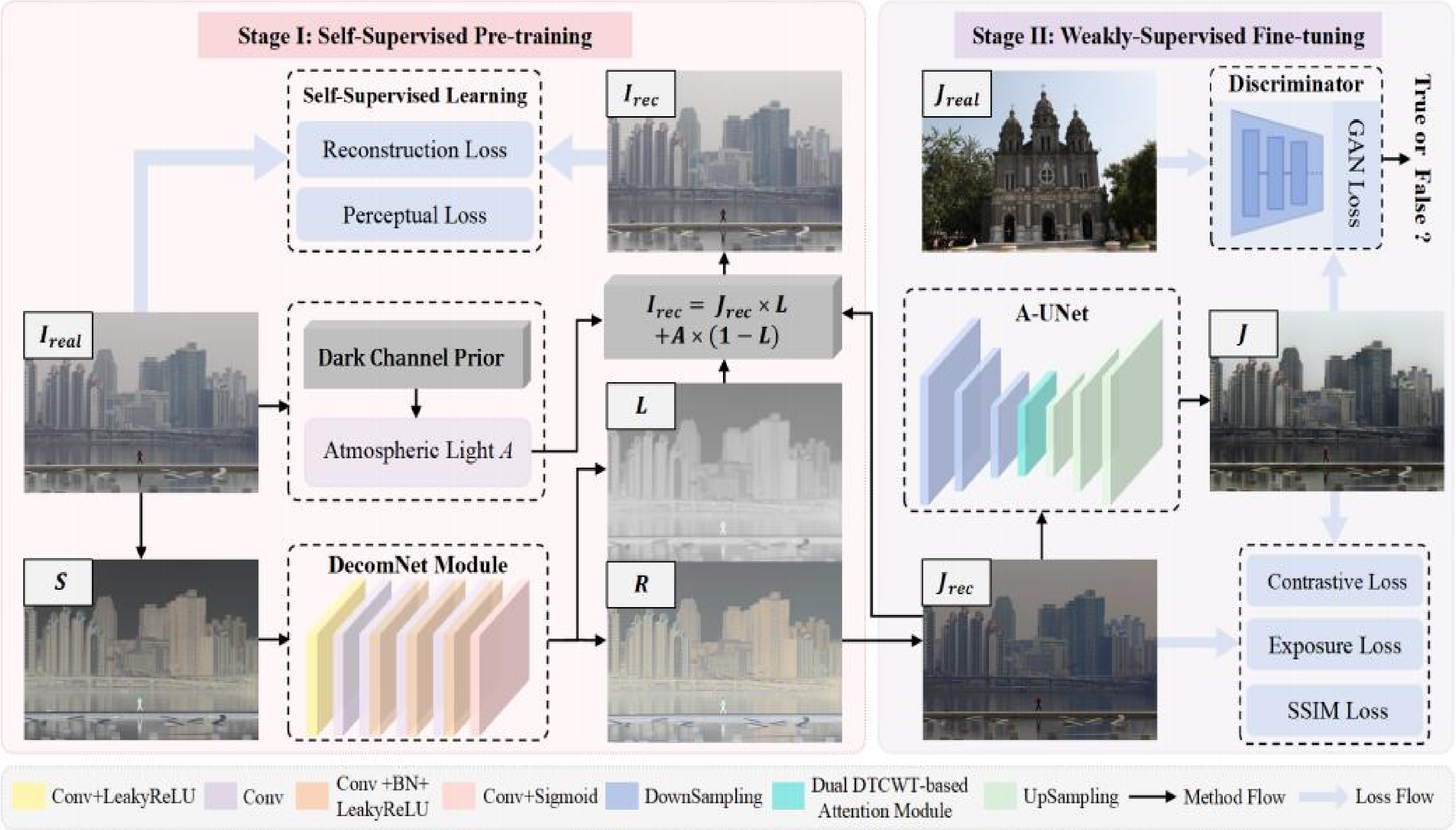
TCSVT25

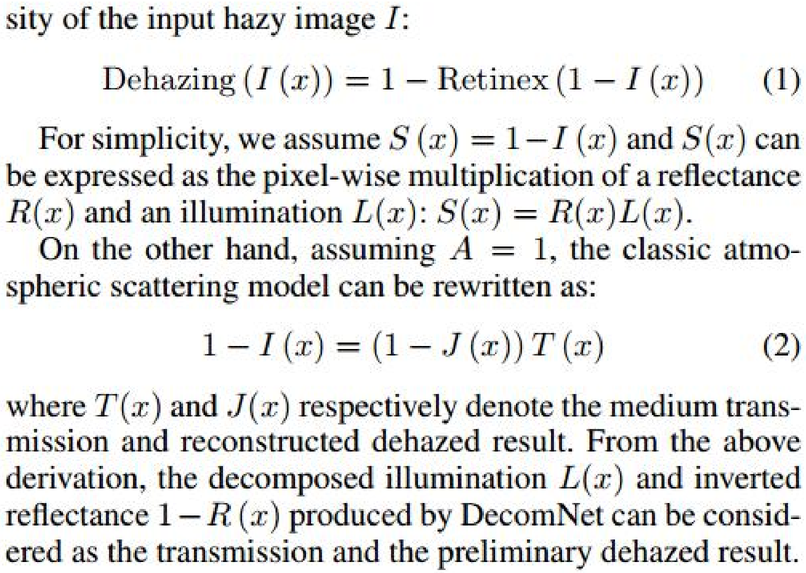
Dehaze-RetinexGAN: Real-World Image Dehazing via Retinex-based Generative Adversarial Network

Xinran Wang[1]*, Guang Yang[1]*, Tian Ye[2], Yun Liu[1,3]†

[1]College of Artificial Intelligence, Southwest University, Chongqing, China
[2]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
[3]College of Computing and Data Science, Nanyang Technological University, Singapore, Singapore
wowwowyahoo@gmail.com, learnforai@gmail.com, owentianye@hkust-gz.edu.cn, yunliu@swu.edu.cn







sity of the input hazy image I:

$$\text{Dehazing}\left(I\left(x\right)\right) = 1 - \text{Retinex}\left(1 - I\left(x\right)\right) \qquad (1)$$

For simplicity, we assume $S\left(x\right) = 1 - I\left(x\right)$ and $S(x)$ can be expressed as the pixel-wise multiplication of a reflectance $R(x)$ and an illumination $L(x)$: $S(x) = R(x)L(x)$.

On the other hand, assuming $A = 1$, the classic atmospheric scattering model can be rewritten as:

$$1 - I\left(x\right) = \left(1 - J\left(x\right)\right) T\left(x\right) \qquad (2)$$

where $T(x)$ and $J(x)$ respectively denote the medium transmission and reconstructed dehazed result. From the above derivation, the decomposed illumination $L(x)$ and inverted reflectance $1 - R\left(x\right)$ produced by DecomNet can be considered as the transmission and the preliminary dehazed result.

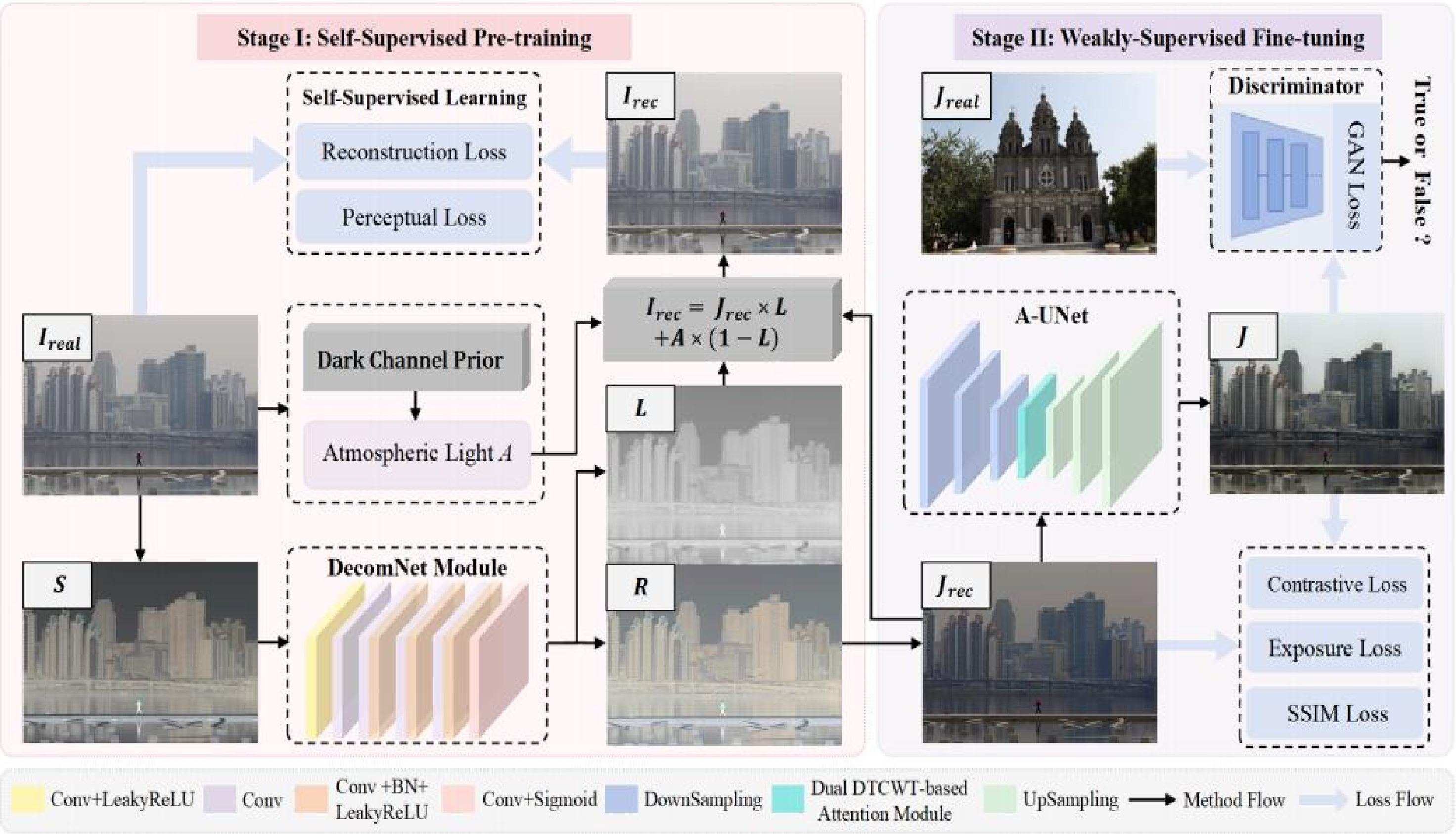
$$I(x) = J(x)T(x) + A(1 - T(x))$$

$$I(x) = R(x)L(x)$$

I=JT+1-T

S=1-I=RL

1-I=(1-J)T



robust and providing more realistic dehazed result.

Specifically, the classic DCP (He, Sun, and Tang 2011) is utilized to estimate the global atmospheric light A. Then, based on the atmospheric scattering model, we combine A, the transmission L and the preliminary dehazed result J_{rec} to produce the reconstructed hazy image I_{rec}:

$$I_{rec}(x) = J_{rec}(x)L(x) + A(1 - L(x)) \qquad (3)$$

In order to construct the relationship between the dehazed result and the input hazy image, we first adopt the reconstruction loss L_{rec} to regularize the reconstructed hazy image I_{rec}, the illumination L, and the reflectance R, simultaneously. L_{rec} is defined as:

$$L_{rec} = \|I_{rec} - I_{real}\|_1 + \lambda_L \|L - \hat{L}\|_1 + \lambda_R \|R - \hat{R}\|_1 \qquad (4)$$

where λ_R and λ_L are weights to control different terms.

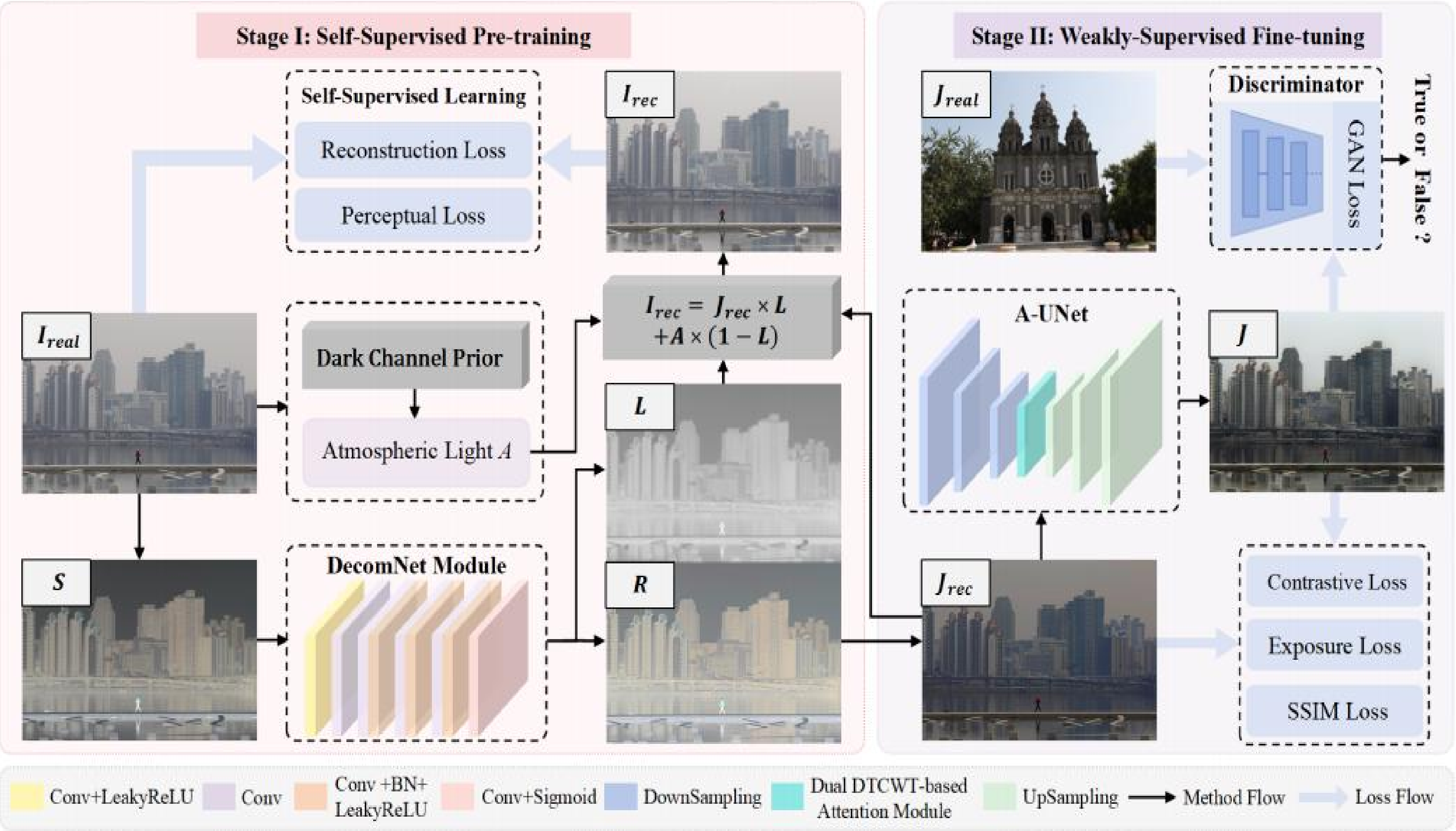
Furthermore, the perceptual loss L_{per} is introduced to improve the visual quality of preliminary dehazed result:

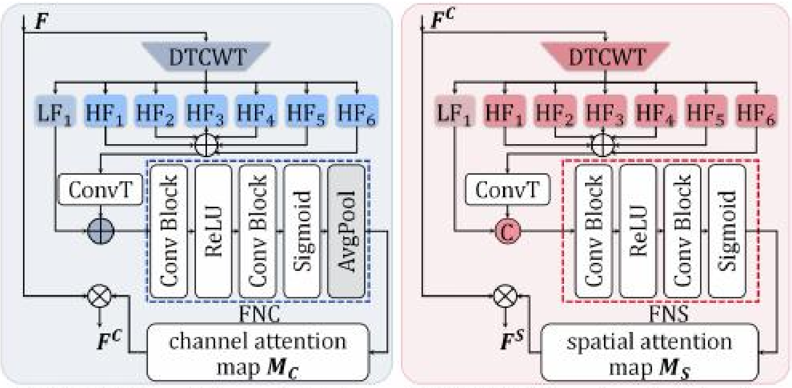
$$L_{per} = \|\phi(I_{rec}) - \phi(I_{real})\|_1 \qquad (5)$$

where ϕ refers to the feature maps obtained from specific layers of VGG-19 (Simonyan and Zisserman 2014).

In summary, the overall loss function in the self-supervised pre-training stage is formulated as:

$$L_{pre} = L_{per} + L_{rec}. \qquad (6)$$





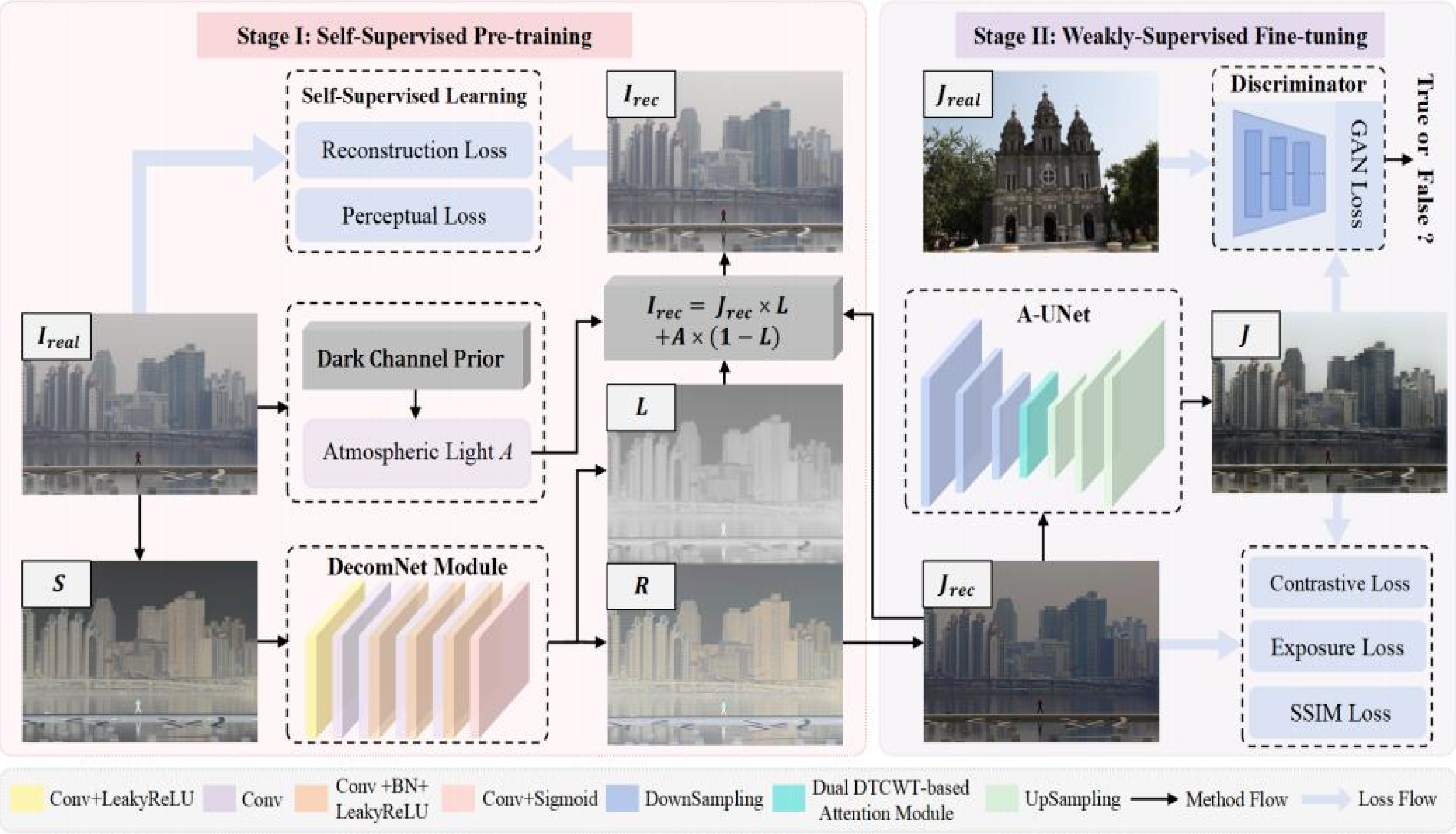
(a) DTCWT-based channel attention (b) DTCWT-based spatial attention

Figure 3: Overall architecture of our designed dual DTCWT-based attention module, consisting of (a) DTCWT-based channel attention and (b) DTCWT-based spatial attention.

$$F_C = M_C \otimes F, \quad F_S = M_S \otimes F_C$$

$$M_C = FNC(LF1 + HFs), \quad M_S = FNS(Cat(LF1, HFs))$$

$$HFs = ConvTranspose(\sum_{i=1}^{6} HF_i)$$



Weakly-Supervised Learning. To fine-tune the A-UNet, we design three losses to improve the quality of the output dehazed result J. Concretely, we first utilize the SSIM loss to constrain the structural similarity between J and J_{rec}:

$$L_{SSIM} = 1 - SSIM(J, J_{rec}), \tag{11}$$

Then, the contrastive regularization loss (Wu et al. 2021) L_{con} is introduced to pull J to the reconstructed dehazed result J_{rec} and push J to its hazy input I_{real}:

$$L_{con} = \rho\left(\phi(I_{real}), \phi(J_{rec}), \phi(J)\right) \tag{12}$$

The exposure loss (Guo et al. 2020) L_{exp} is adopted to eliminate the overexposure of the output dehazed result J.

In addition, the adversarial learning is also introduced to establish the relevance between the output dehazed result J and unpaired real-world clean data J_{real}, which guarantee that the obtained J conforms more closely to the feature distribution of clean data. To achieve this goal, the classic GAN loss (Goodfellow et al. 2014) L_{GAN} is utilized to update the A-UNet and the discriminator D in a weakly supervised manner:

$$\begin{aligned} L_{GAN}(AUNet, D) = {} & \mathbb{E}_{J_{real}\sim P_{J_{real}}}\left[\log D(J_{real})\right] \\ & + \mathbb{E}_{J_{rec}\sim P_{J_{rec}}}\left[\log(1 - D(AUNet(J_{rec})))\right] \end{aligned} \tag{13}$$

where $P_{J_{real}}$ and $P_{J_{rec}}$ respectively stand for the set of all possible J_{real} and J_{rec}.

Overall, the joint loss function in the weakly supervised fine-tuning stage is formulated as:

$$\begin{aligned} L_{\text{fine}} = {} & \lambda_{SSIM} L_{SSIM} + \lambda_{con} L_{con} + \lambda_{exp} L_{exp} \\ & + \arg \min_{AUNet} \max_{D} \lambda_{GAN} L_{GAN} \end{aligned} \tag{14}$$

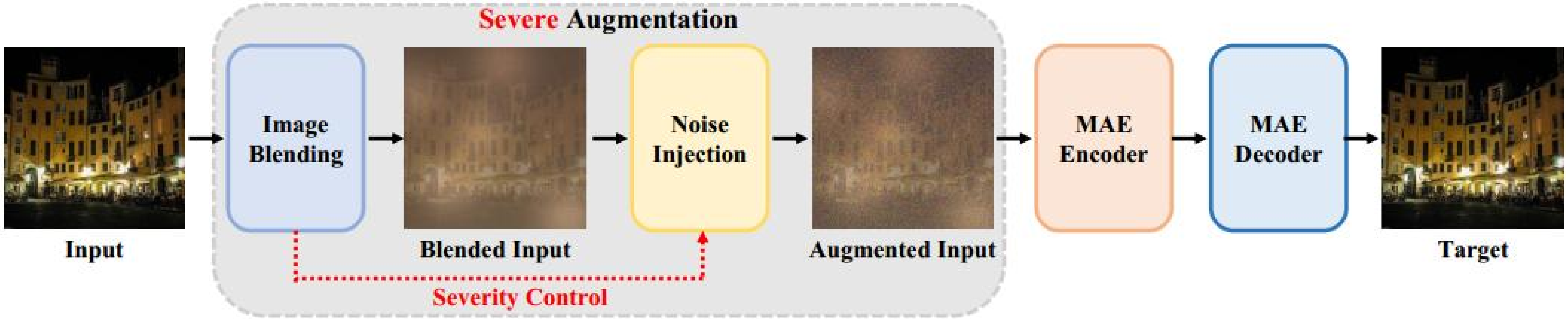
where λ_{SSIM}, λ_{con}, λ_{exp}, and λ_{GAN} are trade-off weights.

The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

NightHaze: Nighttime Image Dehazing via Self-Prior Learning

Beibei Lin[1]*, Yeying Jin[1]*, Wending Yan[2], Wei Ye[2], Yuan Yuan[2], Robby T. Tan[1]

[1]National University of Singapore
[2]Huawei International Pte Ltd
{beibei.lin, e0178303}@u.nus.edu, {yan.wending, yewei10, yuanyuan10}@huawei.com, robby.tan@nus.edu.sg



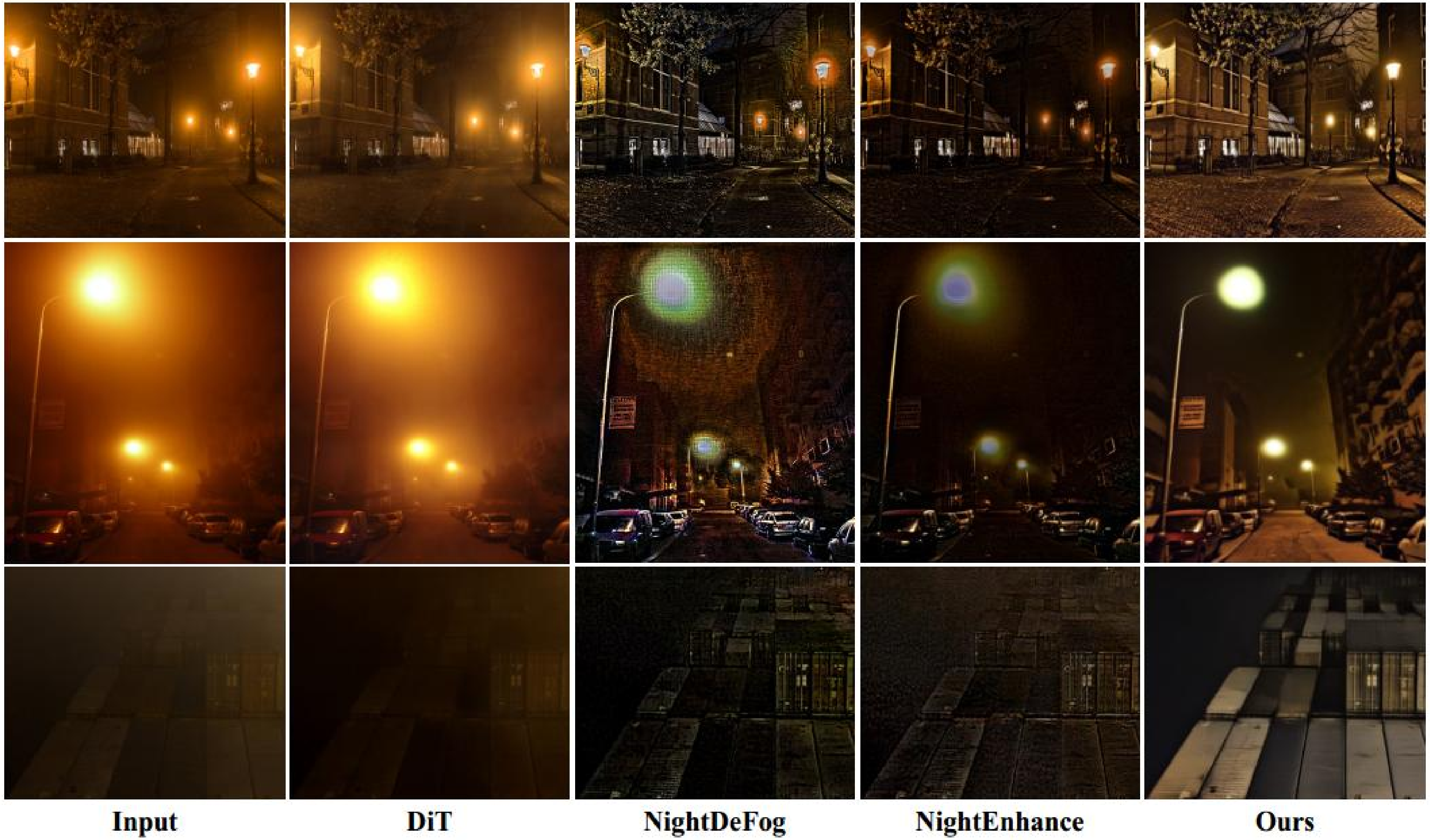
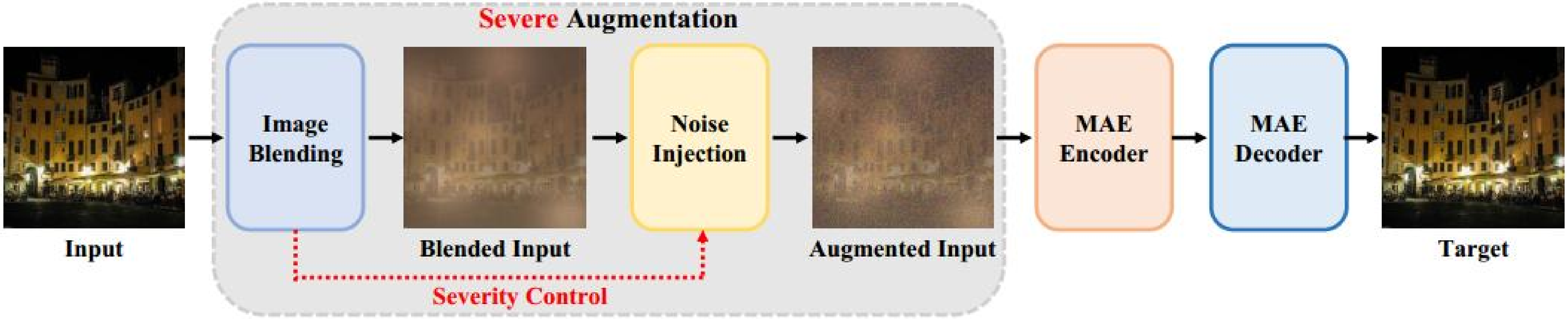
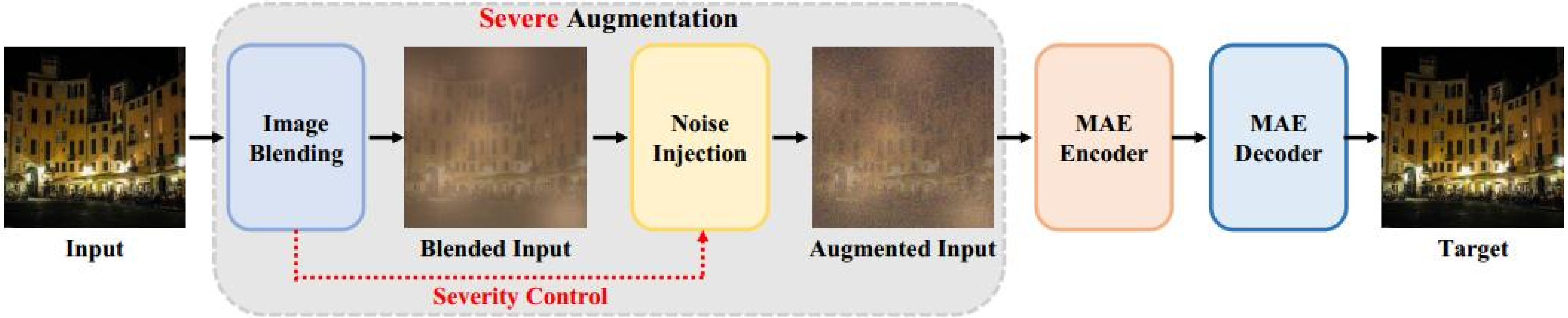


Figure 1: Qualitative results from NightEnhance'23 (Jin et al. 2023), NightDeFog'20 (Yan, Tan, and Dai 2020), DiT'23 (Peebles and Xie 2023) and our method, on the real-world dataset. Ours not only suppress glow but also reveal the detailed textures of the night scenes, including those under low light and strong glow.



自先验学习：
提出了基于严重增强的自我先验学习方法，模仿了掩码自动编码器(MAE)的框架，但不同于MAE使用掩码策略，NightHaze利用夜间图像的光影效果和噪声作为增强手段。

自细化模块：
针对自我先验学习过程中可能出现的过抑制等伪影问题，引入了一个基于教师-学生框架的自细化模块。



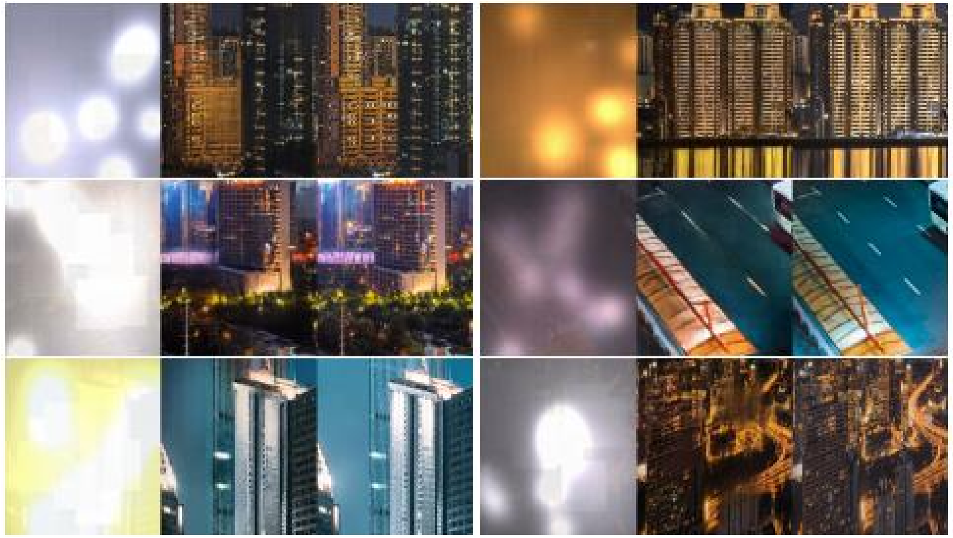
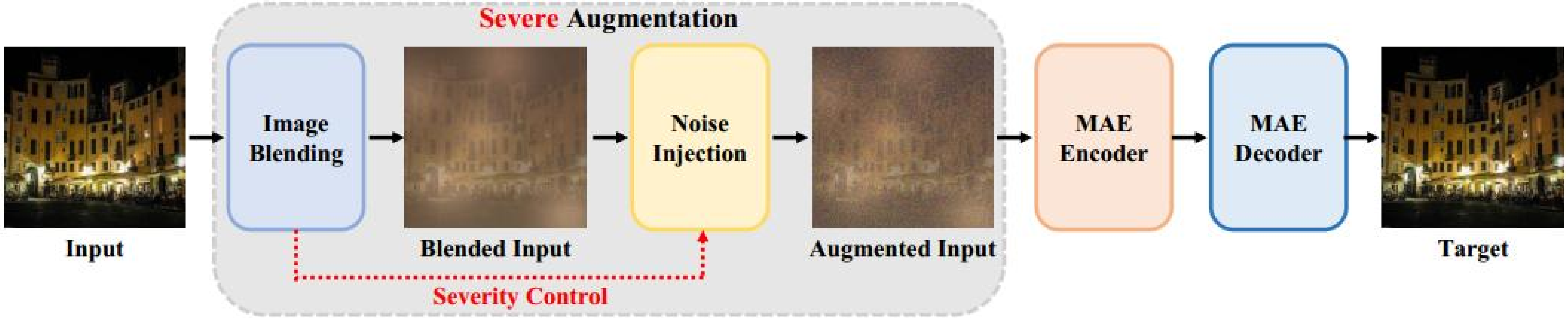


Figure 3: Example results on validation sets. For each triplet, we show the augmented image (left), our restoration (middle), and the ground-truth (right).

$$I = W_b * J + (1 - W_b) * L + \epsilon, \tag{1}$$

where I is the augmented image, J is the clear image, W_b is the blend weight map, L is the light map, ϵ is the noise. We explain the details of each term in Eq. (1) as follows.



$$I = W_b * J + (1 - W_b) * L + \epsilon, \tag{1}$$

where I is the augmented image, J is the clear image, W_b is the blend weight map, L is the light map, ϵ is the noise. We explain the details of each term in Eq. (1) as follows.

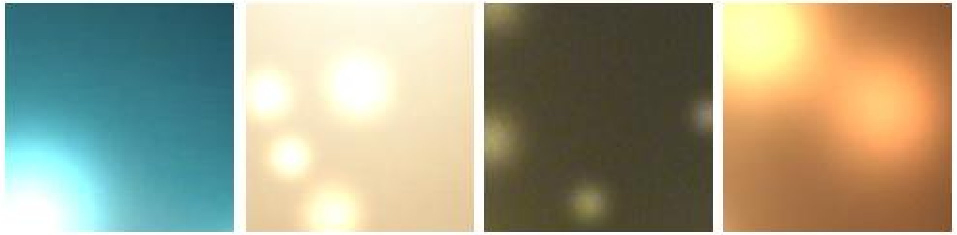
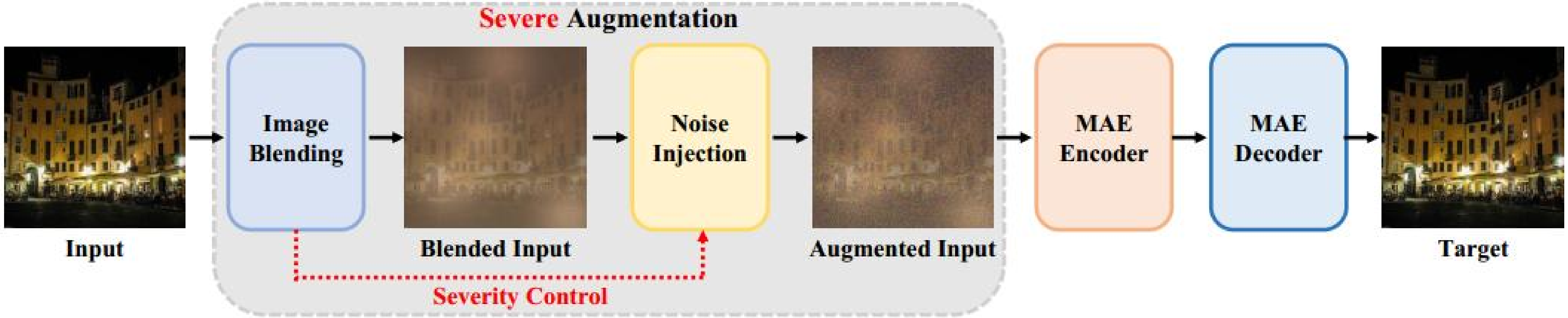
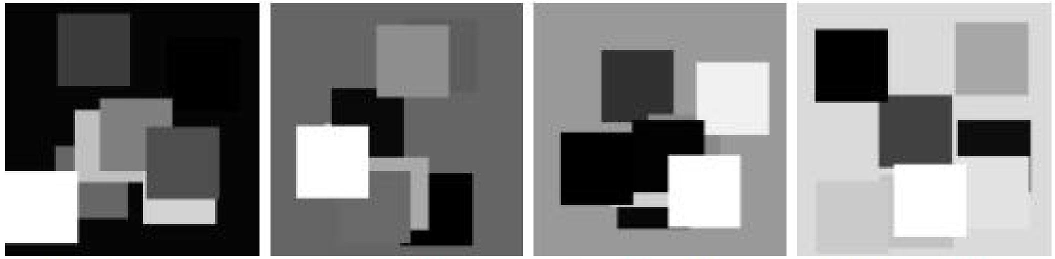


Figure 4: Visualization of different light maps that contain glow effects. We use Gaussian kernels to simulate glow effects. By adjusting the parameters of kernels, we can control the number, size, and brightness of glow regions.



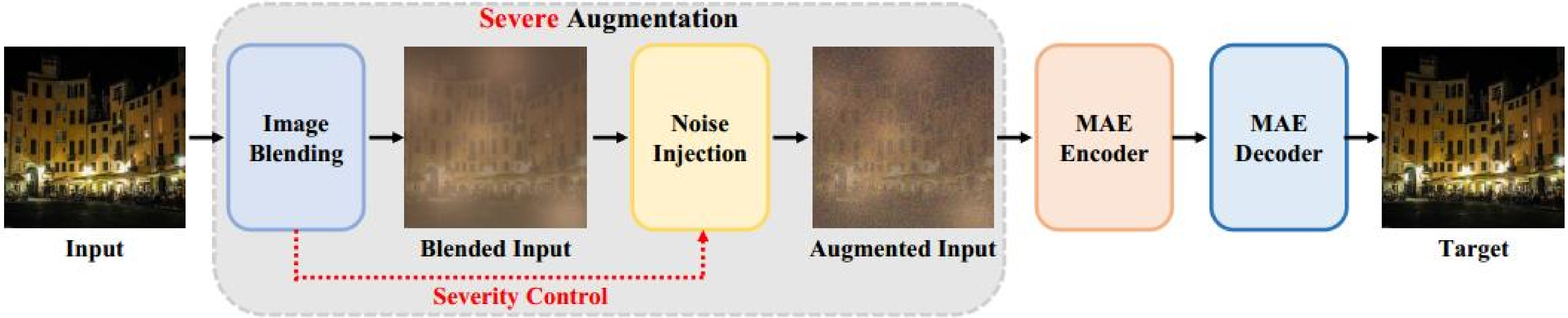
$$I = W_b * J + (1 - W_b) * L + \epsilon, \tag{1}$$

where I is the augmented image, J is the clear image, W_b is the blend weight map, L is the light map, ϵ is the noise. We explain the details of each term in Eq. (1) as follows.



(a) $t = 0.2$ (b) $t = 0.4$ (c) $t = 0.6$ (d) $t = 0.8$

Figure 5: Visualization of blending weight maps, where the black and white regions represent low and high blending values. By adjusting the values of blending weights and the noise term, we can control the severity of our augmentation.



$$I = W_b * J + (1 - W_b) * L + \epsilon, \tag{1}$$

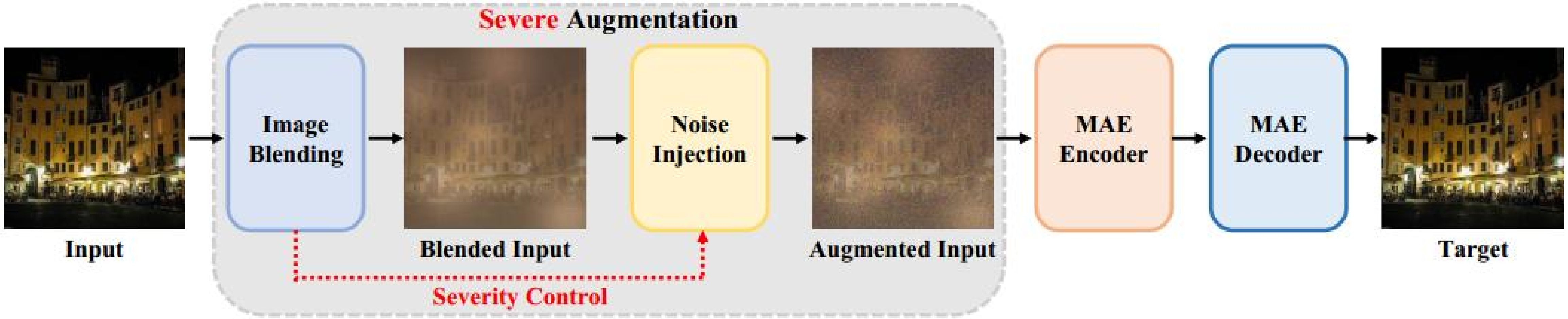
where I is the augmented image, J is the clear image, W_b is the blend weight map, L is the light map, ϵ is the noise. We explain the details of each term in Eq. (1) as follows.

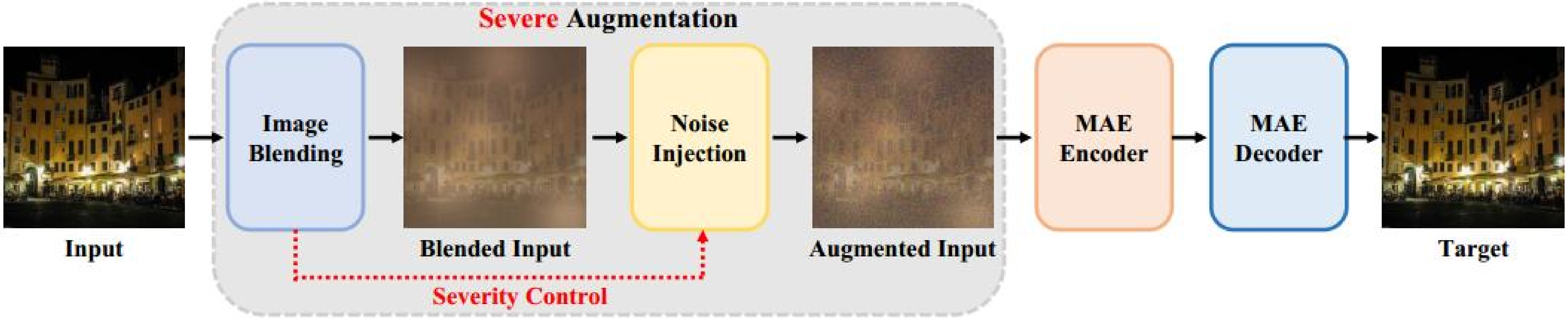
Noise ϵ We take Gaussian noise as the noise ϵ, which can be formulated as $W_\epsilon * \mathcal{N}(0,1)$, where W_ϵ is the weight of the Gaussian noise. Theoretically, $\mathcal{N}(0,1)$ ranges from negative infinity to positive infinity, with about approximately 99.7% of the values fall within the range of -3 to 3. We remove the rest 0.3% values and the negative values of $\mathcal{N}(0,1)$. Thus, the range of our noise term ϵ is (0, 3*W_ϵ).

tivated by this, we define "severe augmentation" as ϵ approaching $W_b * J$ according to Eq. (1). The main reason is

$W_b * J$ becomes to (0, T_{High}).

value of T_{High}, we further define W_ϵ as $W_n * T_{\text{High}}$ and thus the range of ϵ becomes to (0, 3*W_n*T_{High}). We empirically find that when $W_n = 0.1$ and $\epsilon \in [0, 0.3 * T_{\text{High}}]$, our augmentation is difficult enough.


Severe Augmentation
Input
Image Blending
Blended Input
Noise Injection
Augmented Input
Severity Control
MAE Encoder
MAE Decoder
Target



$$\mathrm{Loss} = \frac{1}{N}\sum_{i=1}^{N} |\hat{\mathbf{y}}_i^{\mathrm{uh}} - \mathbf{y}_i^{\mathrm{uh}}| * \mathbf{m}_i^{\mathrm{uh}}, \qquad (2)$$

$$\mathrm{Score} = F_{\mathrm{IQA}}(F(w_s^{t+1}, \mathbf{x}_i^{\mathrm{uh}})) + F_{\mathrm{IQA}}(F(w_s^{t}, \mathbf{x}_i^{\mathrm{uh}})), \quad (3)$$