# Structure Representation Network and Uncertainty Feedback Learning for Dense Non-Uniform Fog Removal

Yeying Jin[1][0000−0001−7818−9534], Wending Yan[1,3][0000−0001−5993−8405], Wenhan Yang[2][0000−0002−1692−0069], and Robby T. Tan[1,3][0000−0001−7532−6919]

[1] National University of Singapore,
[2] Nanyang Technological University,
[3] Yale-NUS College

jinyeying@u.nus.edu, e0267911@u.nus.edu, wenhan.yang@ntu.edu.sg,
robby.tan@{nus,yale-nus}.edu.sg

In this supplementary material, we provide:

1. **More Experimental Results on**
   1-1) SMOKE (Figs. 1 to 2)
   1-2) O-HAZE (Fig. 3)
   1-3) commonly used test foggy image (Figs. 4 to 5)
2. **More Discussion on**
   2-1) Structure Representations (Fig. 6)
   2-2) Grayscale Feature Multiplier
   2-3) Uncertainty Map
3. **Training and Network Architecture**
   3-1) real clean reference images (Fig. 7)
   3-2) synthetic fog images and their clean ground truth (Fig. 8)
   3-3) self-collected smoke images (Fig. 9)

[†] Our data and code is available at: https://github.com/jinyeying/FogRemoval

# 1 More Experimental Results on

Figs. 1 to 2 show our results on real dense and/or non-uniform fog, in comparison with the state-of-the-art CNN-based methods and transformer-based method [1, 2]. Our results are more robust in removing fog, and better preserving the background information than baselines. We collected real dense and/or non-uniform fog data by ourselves. We use a fog machine to generate fog, where we fix the camera pose to record fog data and paired ground truth. Totally we collected 12 pairs of data for evaluation, shown in Fig. 9. The quantitative evaluation is shown in Table. 1.

**Table 1.** Quantitative Evaluation on self-collected daytime smoke data.

| Methods | PSNR↑ | SSIM↑ |
|---|---|---|
| Input | 13.06 | 0.36 |
| Dehamer'22 [1] | 13.27 | 0.36 |
| DehazeFormer'22 [2] | 13.25 | 0.36 |
| D4'22 [3] | 9.62 | 0.10 |
| PSD'21 [4] | 11.01 | 0.29 |
| NLD [10] | 11.81 | 0.33 |
| 4KDehazing'21 [5] | 12.42 | 0.36 |
| EPDN [9] | 12.76 | 0.39 |
| MSBDN [6] | 13.19 | 0.34 |
| DAN [7] | 14.06 | 0.42 |
| GDN [8] | 15.19 | 0.53 |
| **Ours** | **18.83** | **0.62** |

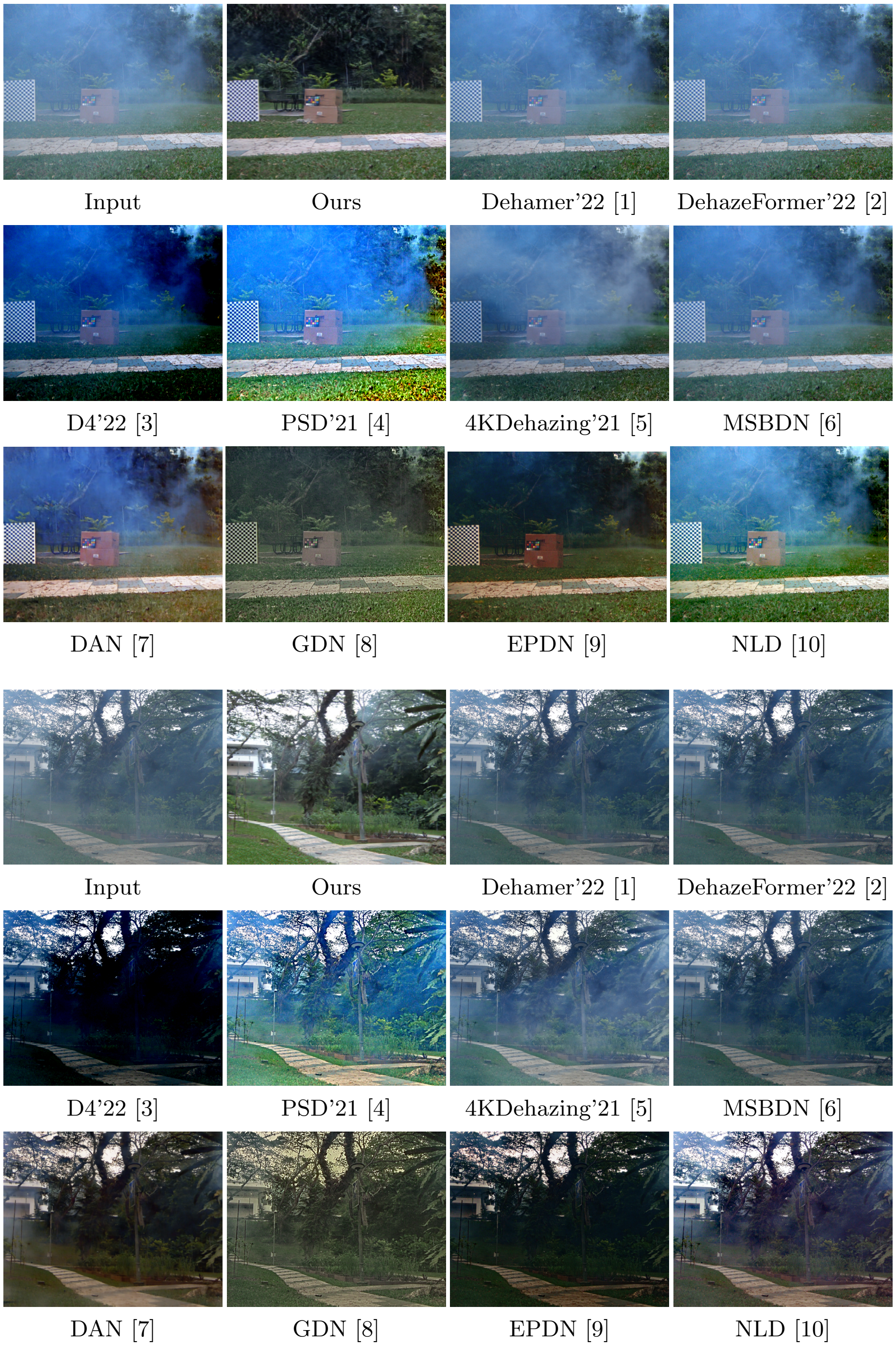
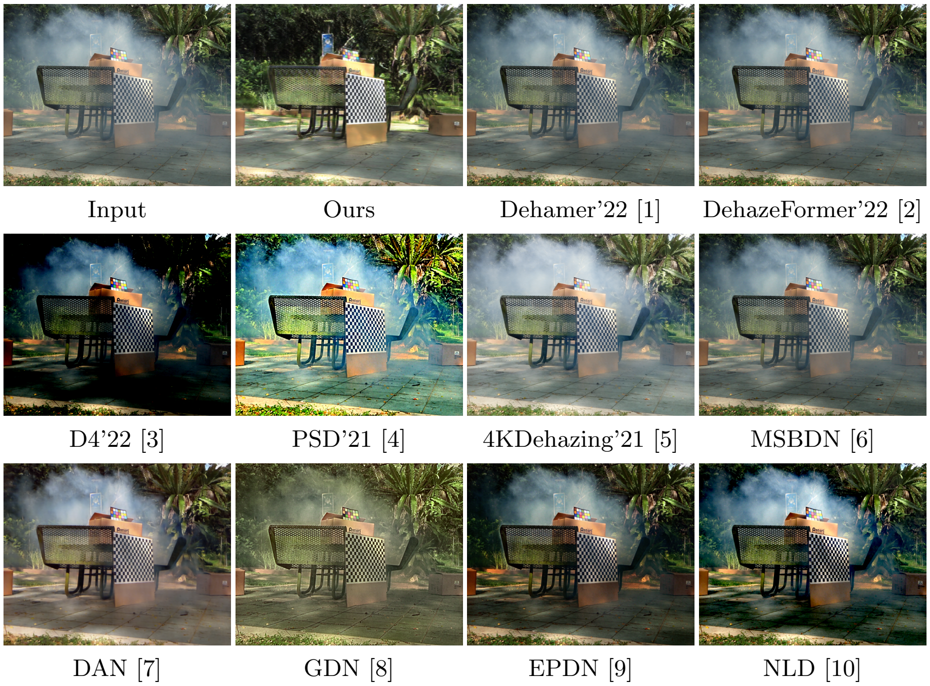


**Fig. 1.** Visual result comparisons of different methods: the state-of-the-art CNN-based methods and transformer-based methods in dense and/or non-uniform fog and smoke.



Input

Ours

Dehamer’22 [1]

DehazeFormer’22 [2]

D4’22 [3]

PSD’21 [4]

4KDehazing’21 [5]

MSBDN [6]

DAN [7]

GDN [8]

EPDN [9]

NLD [10]

**Fig. 2.** Visual result comparisons of different methods: the state-of-the-art CNN-based methods and transformer-based methods in dense and/or non-uniform fog and smoke.

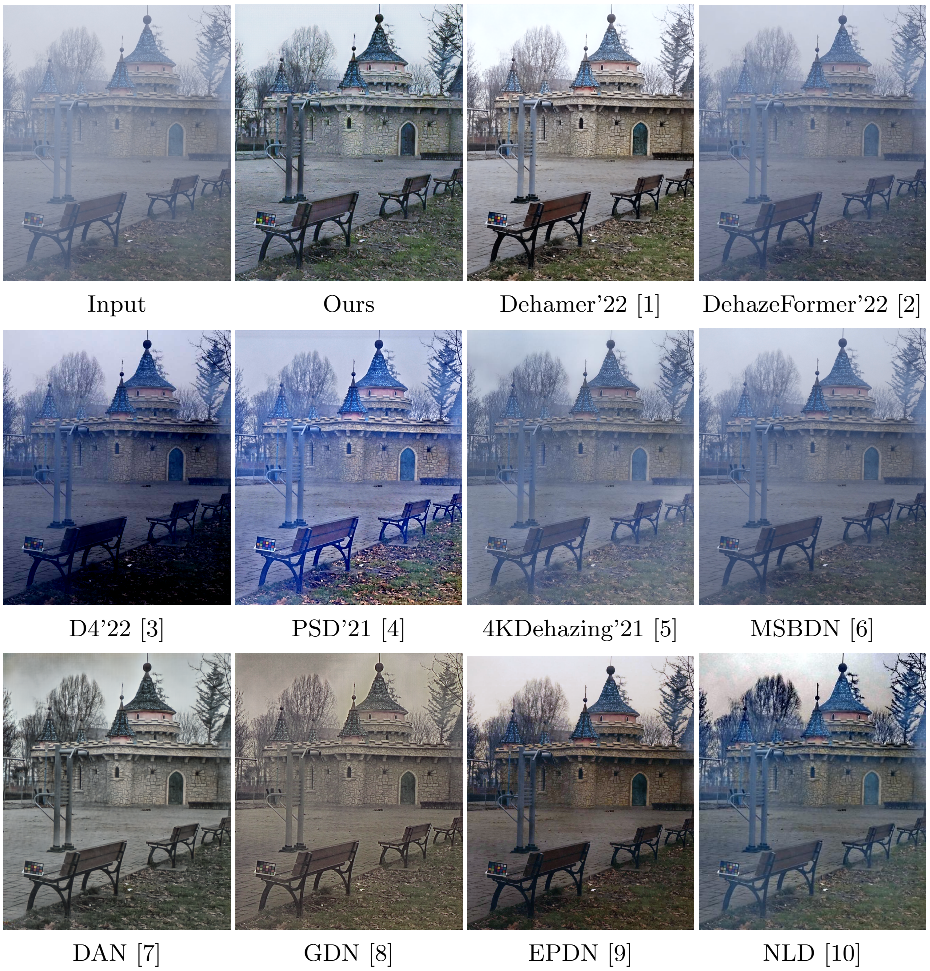
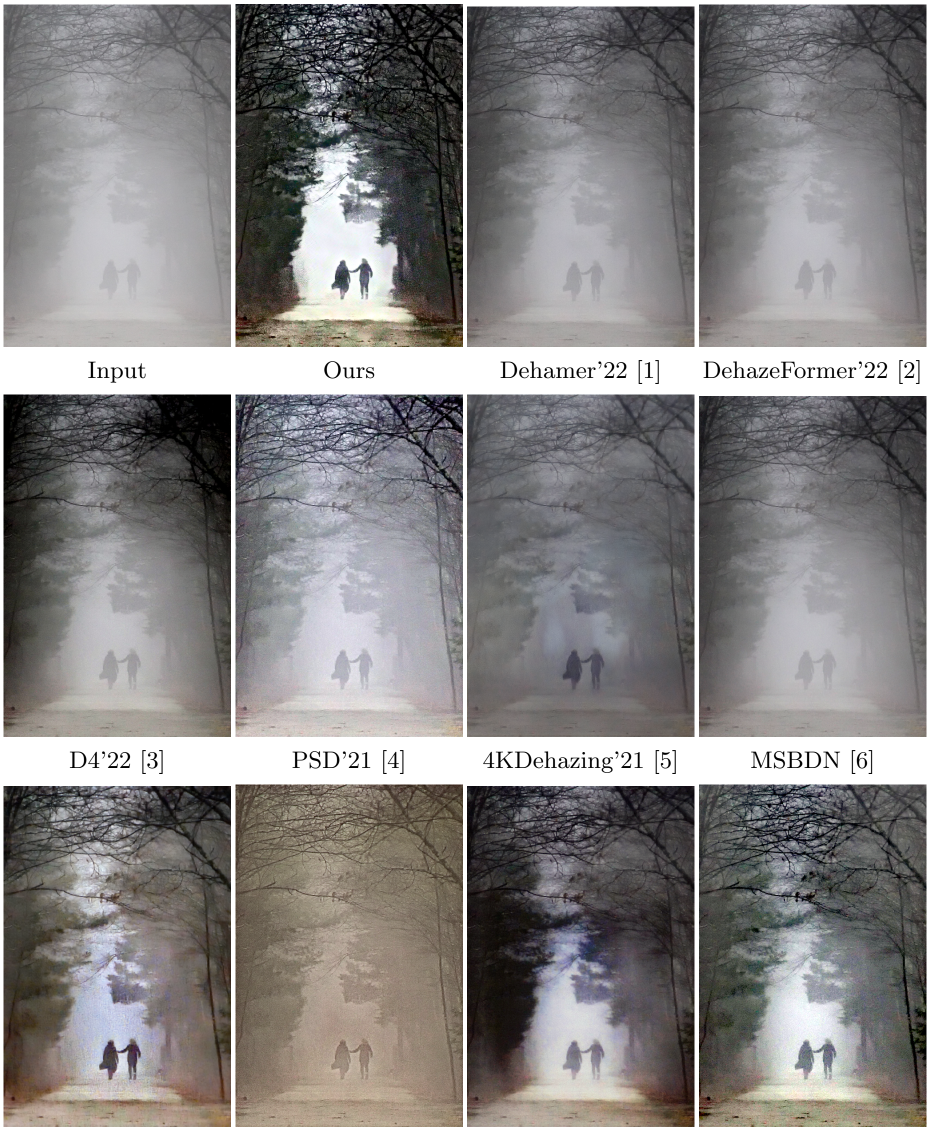


**Fig. 3.** Visual result comparisons of different methods: the state-of-the-art CNN-based methods and transformer-based methods in O-HAZE.

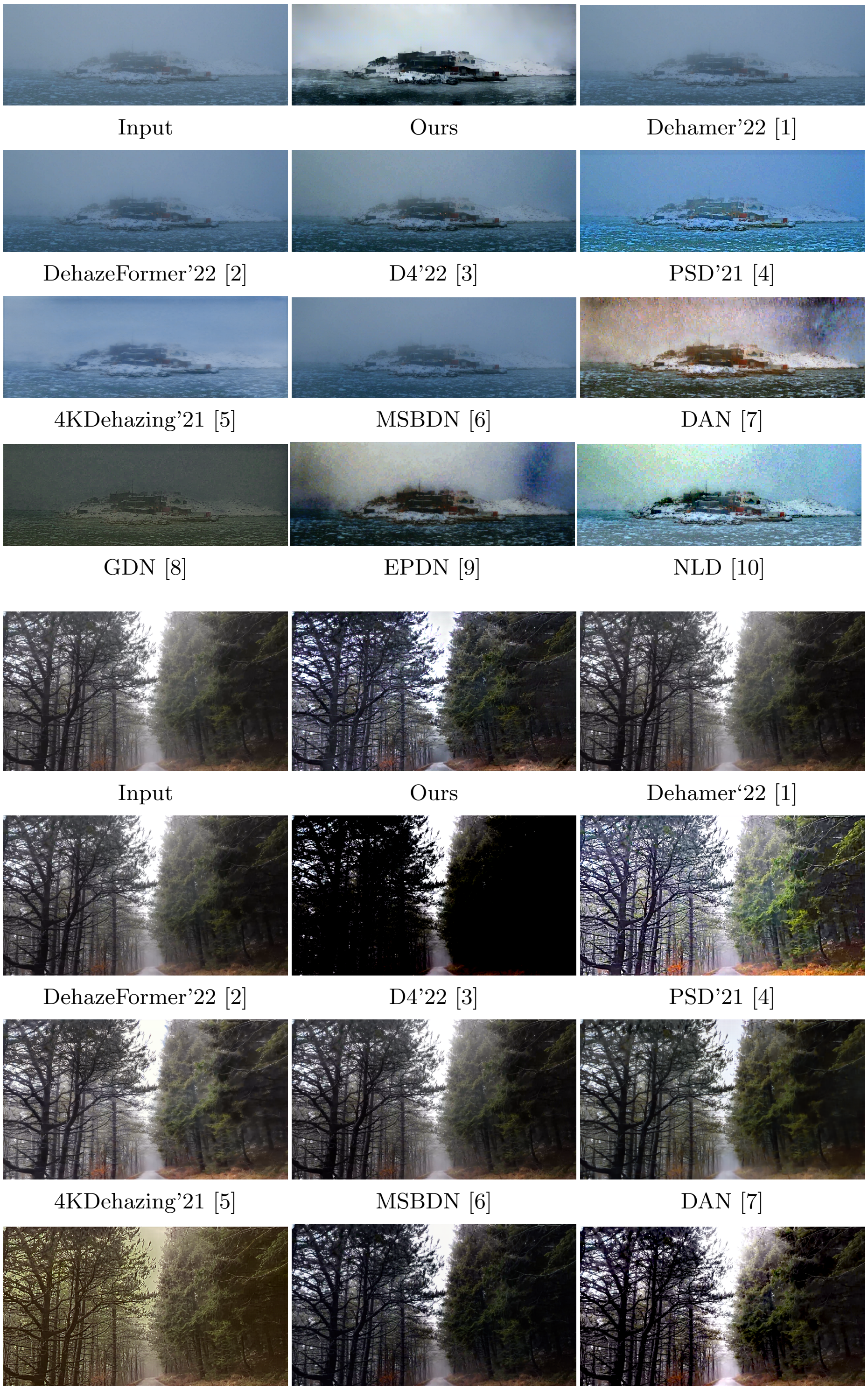


Input Ours Dehamer'22 [1] DehazeFormer'22 [2]

D4'22 [3] PSD'21 [4] 4KDehazing'21 [5] MSBDN [6]

DAN [7] GDN [8] EPDN [9] NLD [10]

**Fig. 4.** Visual result comparisons of different methods: the state-of-the-art CNN-based methods and transformer-based methods in dense and/or non-uniform fog and smoke.



Input

Ours

Dehamer’22 [1]

DehazeFormer’22 [2]

D4’22 [3]

PSD’21 [4]

4KDehazing’21 [5]

MSBDN [6]

DAN [7]

GDN [8]

EPDN [9]

NLD [10]

Input

Ours

Dehamer‘22 [1]

DehazeFormer’22 [2]

D4’22 [3]

PSD’21 [4]

4KDehazing’21 [5]

MSBDN [6]

DAN [7]

GDN [8]

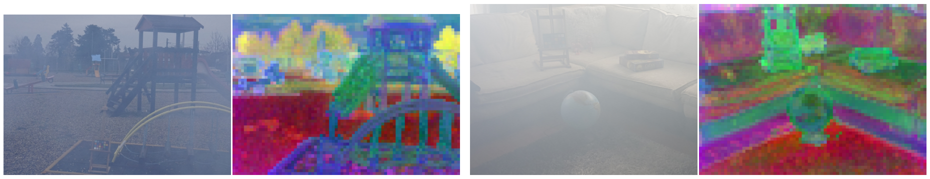
EPDN [9]

NLD [10]

**Fig. 5.** Visual result comparisons of different methods: the state-of-the-art CNN-based methods and transformer-based methods in dense and/or non-uniform fog and smoke.

## 2 More iscussioniscussion on

### 2.1 Structure Representations



**(a)** Input $\mathbf{I}$ **(b)** $S(\hat{\mathbf{J}}_{\mathbf{Y}})$ **(a)** Input $\mathbf{I}$ **(b)** $S(\hat{\mathbf{J}}_{\mathbf{Y}})$

**Fig. 6.** We can observe that (b) DINO-ViT representations capture structure scene/object parts (*e.g.* boxes, balls, trees, buildings).

### 2.2 Grayscale Feature Multiplier

The consistency between the grayscale feature multiplier and the colorful feature multiplier is based on the Gray World assumption. Under the Gray World assumption, the whole world objects are gray and the color of all objects is just a reflection of the atmospheric light. Therefore, we can rewrite Eq.1 in the main paper into:

$$\mathbf{I}(\mathbf{x}) = (\alpha(\mathbf{x})\mathbf{A})\, t(\mathbf{x}) + (1 - t(\mathbf{x}))\, \mathbf{A}, \tag{1}$$

where $\alpha(\mathbf{x})$ is the reflectivity of the objects in the scene. Note that under the Gray World assumption, the reflectivity of a gray object has the same value for all three RGB channels, which means the colorful image and corresponding grayscale image shall share the same reflectivity.

Next, we can take out the $\mathbf{A}$ from the right side of Eq. 1:

$$\mathbf{I}(\mathbf{x}) = (\alpha(\mathbf{x})\mathbf{t}(\mathbf{x}) + \mathbf{1} - \mathbf{t}(\mathbf{x}))\, \mathbf{A}. \tag{2}$$

Based on Eq. 1, we can write the definition of the feature multiplier to:

$$\mathbf{M}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) + t(\mathbf{x})\mathbf{A} - \mathbf{A}}{\mathbf{I}(\mathbf{x})t(\mathbf{x})}, \tag{3}$$

$$\mathbf{M}(\mathbf{x}) = \frac{(\alpha(\mathbf{x})\mathbf{t}(\mathbf{x}) + \mathbf{1} - \mathbf{t}(\mathbf{x}))\, \mathbf{A} + t(\mathbf{x})\mathbf{A} - \mathbf{A}}{(\alpha(\mathbf{x})\mathbf{t}(\mathbf{x}) + \mathbf{1} - \mathbf{t}(\mathbf{x}))\, \mathbf{A}t(\mathbf{x})}, \tag{4}$$

and we can simplify the Eq. 4 by canceling the term of atmospheric light $\mathbf{A}$:

$$\mathbf{M}(\mathbf{x}) = \frac{(\alpha(\mathbf{x})\mathbf{t}(\mathbf{x}) + \mathbf{1} - \mathbf{t}(\mathbf{x})) + t(\mathbf{x}) - 1}{(\alpha(\mathbf{x})\mathbf{t}(\mathbf{x}) + \mathbf{1} - \mathbf{t}(\mathbf{x}))\, t(\mathbf{x})}. \tag{5}$$

This Eq. 5 can be regarded as the new definition of the feature multiplier under the Gray World assumption [11]. With this definition, we can easily observe that the feature multiplier shall keep consistency after converting to a grayscale domain from a colorful domain.

### 2.3 Uncertainty Map

Inspired by [12, 13], it is a common assumption that the clean domain output $\hat{\mathbf{J}}$ shall follow a Laplace distribution where the mean of this distribution is the ground truth $\mathbf{J}^{gt}$. Under this assumption, we can have a likelihood function as follows:

$$p = \frac{1}{2\theta} \exp(-\frac{\left\| \hat{\mathbf{J}} - \mathbf{J}^{gt} \right\|_1}{\theta}), \tag{6}$$

where $\theta$ is the variance of this Laplace distribution.

In our implementation, we definite this variance as the uncertainty of this output $\hat{\mathbf{J}}$. Therefore, Eq. 6 includes both outputs generated by our multi-task network, and we can design an uncertainty loss to constrain them based on Eq. 6. We Take the natural logarithm of both sides of Eq. 6, we can obtain:

$$\ln p = \ln(\frac{1}{2\theta} \exp(-\frac{\left\| \hat{\mathbf{J}} - \mathbf{J}^{gt} \right\|_1}{\theta})), \tag{7}$$

$$\ln p = -\frac{\left\| \hat{\mathbf{J}} - \mathbf{J}^{gt} \right\|_1}{\theta} + \ln(\frac{1}{2\theta}), \tag{8}$$

$$\ln p = -\frac{\left\| \hat{\mathbf{J}} - \mathbf{J}^{gt} \right\|_1}{\theta} - \ln \theta - \ln 2. \tag{9}$$

Then, an uncertainty loss could be designed such that minimizing this loss is reformulated as maximizing the likelihood in Eq. 9:

$$\arg\max_{\theta} -\frac{\left\| \hat{\mathbf{J}} - \mathbf{J}^{gt} \right\|_1}{\theta} - \ln \theta - \ln 2 = \arg\max_{\theta} -\frac{\left\| \hat{\mathbf{J}} - \mathbf{J}^{gt} \right\|_1}{\theta} - \ln \theta. \tag{10}$$

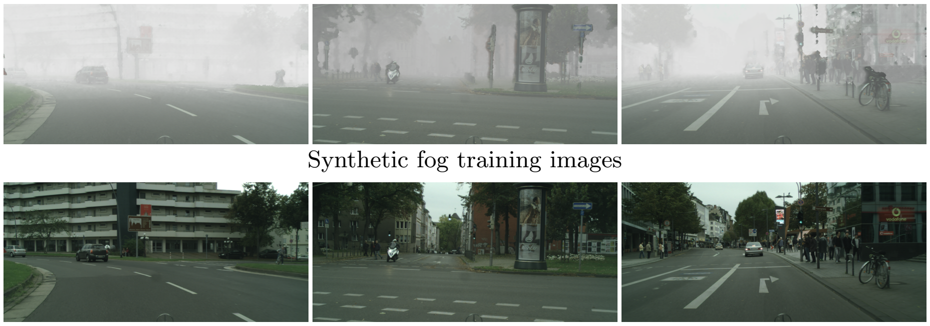
The $\ln 2$ term is a constant that does not affect the maximization of the likelihood, which can be ignored.

For the first term in this likelihood $-\frac{\|\hat{I}_t - I_{gt}\|_1}{\theta_t}$, we simply convert the negative sign to positive and put into the loss function. For the second term $-\ln \theta$, if we just converted the negative sign to positive similar to the first term, the loss value would be negative when the value of the uncertainty map $\theta$ was in the range [0,1], and there would be a negative infinite when the value of the uncertainty map $\theta$ was zero. Therefore, we need to modify this term with a bias. Since we want the uncertainty map $\theta$ to be in the range [0,1], we add a constant threshold after converting the negative sign to positive: $\ln(\theta_t + 1)$.

In summary, this uncertainty loss is designed as follows:

$$\mathcal{L}_{\text{unc}} = \frac{\left\| \hat{\mathbf{J}} - \mathbf{J}^{gt} \right\|_1}{\theta} + \ln(\theta + 1). \tag{11}$$

**Fig. 7.** Examples of clean reference images for training.



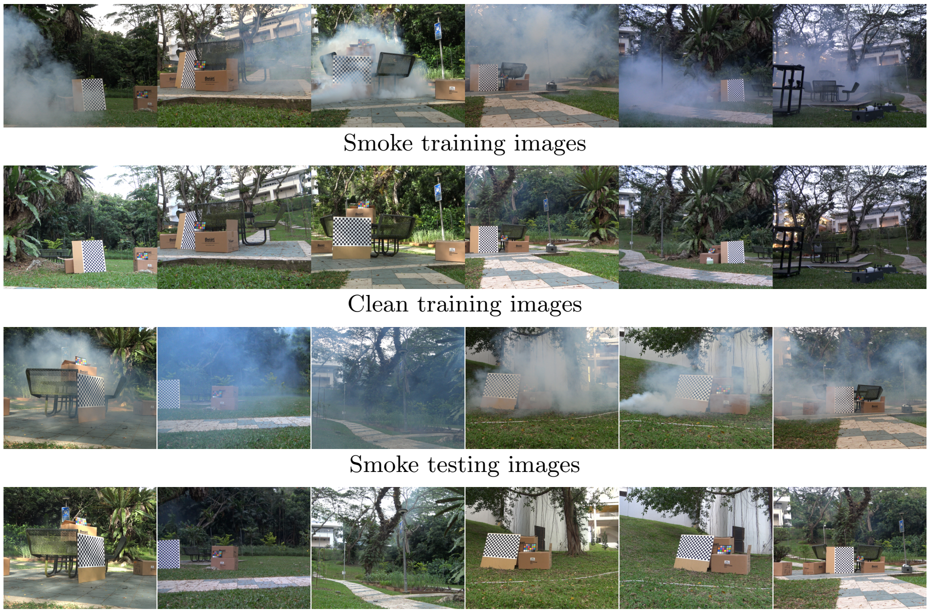
Synthetic fog training images

Synthetic clean training images

**Fig. 8.** Examples of synthetic fog/clean image pairs used for supervised loss in training.

## 3 Training and Network Architecture

The generators of our gray network and color network have similar encoder-decoder network architectures, the only difference is the input and output channel numbers of the first and the last layers (one for the gray network and three for the color network). Our generators are based on the ResNet architecture [15], which is effective for regression problems. The encoder has nine ResNet blocks, and the input size is $512 \times 512$ due to the memory constraint of our GPUs. It contains a sub-network to generate the feature multiplier from the encoded features, which is built with one ResNet block and located behind the encoder. Our generators have two decoders that generate output and uncertainty map respectively, which have three deconv layers. The discriminators are multi-layer networks consisting of three stride layers, a step size of two, whose final feature dimension is reduced to $(H/8, W/8)$. Thus, our discriminator follows the practice of [16]. We initialize the convolution layers with random values and train all the weights for 50K iterations using a mini-batch size of 16. The network is optimized using the Adam method [18] with learning rate $2 \times 10^{-4}$ and $\beta_1 = 0.9$.



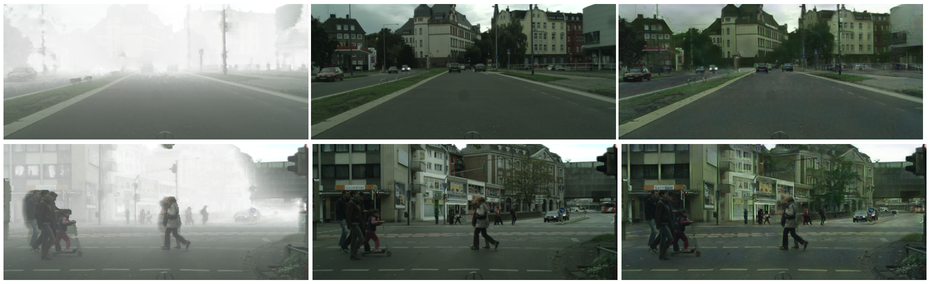
Smoke training images

Clean training images

Smoke testing images

Clean testing clean images

**Fig. 9.** Examples of self-collected smoke images.



**(a)** Input **I** **(b)** Ground Truth **(c)** Ours

**Fig. 10.** Testing results on synthetic fog images.

## References

1. Guo, C.L., Yan, Q., Anwar, S., Cong, R., Ren, W., Li, C.: Image dehazing transformer with transmission-aware 3d position embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 5812–5820
2. Song, Y., He, Z., Qian, H., Du, X.: Vision transformers for single image dehazing. arXiv preprint arXiv:2204.03883 (2022)
3. Yang, Y., Wang, C., Liu, R., Zhang, L., Guo, X., Tao, D.: Self-augmented unpaired image dehazing via density and depth decomposition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 2037–2046
4. Chen, Z., Wang, Y., Yang, Y., Liu, D.: Psd: Principled synthetic-to-real dehazing guided by physical priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 7180–7189
5. Zheng, Z., Ren, W., Cao, X., Hu, X., Wang, T., Song, F., Jia, X.: Ultra-high-definition image dehazing via multi-guided bilateral learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2021) 16180–16189
6. Dong, H., Pan, J., Xiang, L., Hu, Z., Zhang, X., Wang, F., Yang, M.H.: Multi-scale boosted dehazing network with dense feature fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 2157–2167
7. Shao, Y., Li, L., Ren, W., Gao, C., Sang, N.: Domain adaptation for image dehazing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 2808–2817
8. Liu, X., Ma, Y., Shi, Z., Chen, J.: Griddehazenet: Attention-based multi-scale network for image dehazing. In: Proceedings of the IEEE/CVF international conference on computer vision. (2019) 7314–7323
9. Qu, Y., Chen, Y., Huang, J., Xie, Y.: Enhanced pix2pix dehazing network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 8160–8168
10. Berman, D., Treibitz, T., Avidan, S.: Single image dehazing using haze-lines. IEEE transactions on pattern analysis and machine intelligence **42** (2018) 720–734
11. Buchsbaum, G.: A spatial processor model for object colour perception. Journal of the Franklin institute **310** (1980) 1–26
12. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? Advances in neural information processing systems **30** (2017)
13. Ning, Q., Dong, W., Li, X., Wu, J., Shi, G.: Uncertainty-driven loss for single image super-resolution. Advances in Neural Information Processing Systems **34** (2021) 16398–16409
14. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 3213–3223
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
16. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1125–1134

17. Lai, W.S., Huang, J.B., Yang, M.H.: Semi-supervised learning for optical flow with generative adversarial networks. In: Advances in Neural Information Processing Systems. (2017) 354–364
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)