

# OMINCLEAR: SOFT EFFECTS REMOVAL FROM IMAGES WITHIN A VERSATILE MODEL SUPPLEMENTARY MATERIALS

**Anonymous authors**

Paper under double-blind review

In this supplementary material, we are going to illustrate i) more details of our data curation details; ii) more details of the haze synthetic pipeline; iii) the detail design of non-reference metrics; iv) more implementation details and v) more visual results and quality analysis.

## A DATA CURATION ON PUBLIC DATASETS.

Our data collection process aggregates established benchmarks from each domain. For lens flare removal, we incorporate the real-world paired dataset FlareReal600 (Dai et al., 2024) for nighttime optical artifacts. For shadow removal, our dataset combines several widely-used benchmarks, including SRD (Qu et al., 2017), ISTD+ (Wang et al., 2018), and the high-resolution WSRD+ (Vasluianu et al., 2023), to cover a wide variety of shadow types and complexities. The most extensive category is haze removal, for which we collected a diverse range of datasets. This includes smaller, real-world datasets captured under controlled conditions, we name this set as Haze-R, including: I-HAZE (Ancuti et al., 2018b), O-HAZE (Ancuti et al., 2018a), Dense-Haze (Ancuti et al., 2019), NH-Haze (Ancuti et al., 2020; 2021; 2023; 2024), and video dehaze dataset REVIDE (Zhang et al., 2021), multi-level haze dataset LM-Haze (Zhang et al., 2024). Large-scale synthetic datasets that provide broad coverage of different haze conditions like RESIDE (Li et al., 2018) and HAZESPACE2M (Islam et al., 2024) are also included. Finally, for reflection removal, we integrated datasets that capture various scenarios, such as general real-world reflections RRW (Zhu et al., 2024), polarization-based captures POLAR-RR (Lei et al., 2020), and flash-induced reflections RFC (Lei & Chen, 2021), and synthetic by overlaying dataset BDN (Yang et al., 2018). However, these publicly available datasets were originally collected for specific tasks. As a result, their overall distribution is imbalanced, including discrepancies across different tasks, between real and synthetic data, as well as between indoor and outdoor scenes, and day and night conditions.

## B DETAILS OF THE HAZE SYNTHESIS PIPELINE

A significant portion of our training dataset, particularly for atmospheric effects like haze, fog, and smoke, was generated using a custom synthesis pipeline. This pipeline was designed to overcome the limitations of existing synthetic datasets, which often lack physical realism and diversity. Our methodology is built upon two core components: (1) a physically-motivated atmospheric rendering engine that applies uniform atmospheric effects based on scene geometry, and (2) a procedural texture generator that creates complex, non-homogeneous patterns to simulate phenomena like patchy fog or smoke plumes.

### B.1 PHYSICALLY-MOTIVATED ATMOSPHERIC RENDERING MODEL

The foundation of our synthesis pipeline is a unified rendering model inspired by the Radiative Transfer Equation (RTE). This model mathematically describes how light interacts with a participating medium (like haze or fog) as it travels from a scene object to the camera. The final color at a pixel  $x$ , denoted  $I_{out,c}(x)$  for a color channel  $c$ , is a composite of the attenuated scene radiance and the in-scattered light from the atmosphere, known as airlight.

The image formation model is expressed as:

$$I_{out,c}(x) = I_{in,c}(x) \cdot T_c(x) + A_c \cdot (\omega_{0,c} \cdot \kappa) \cdot (1 - T_c(x))^\eta \quad (1)$$

where:

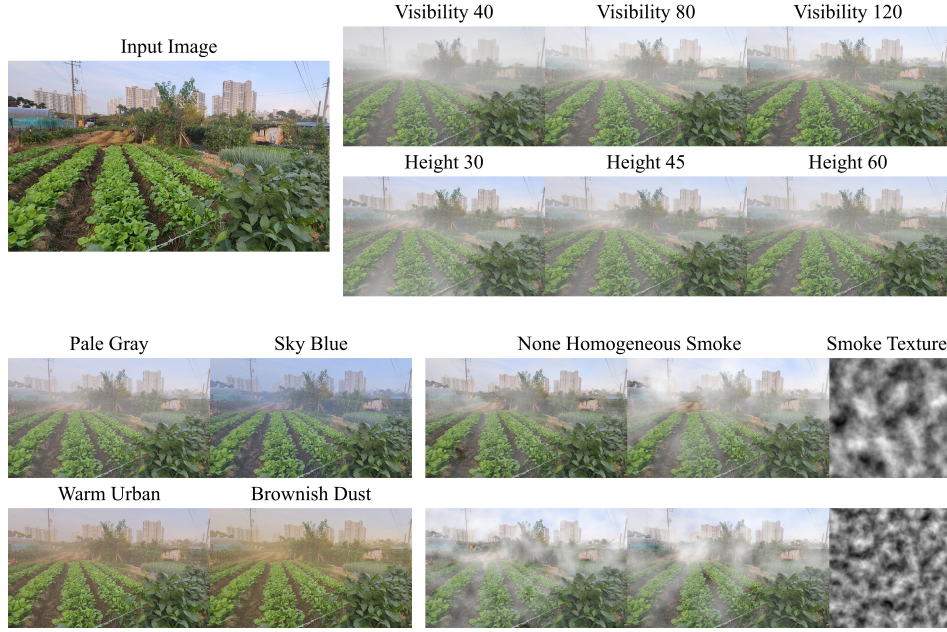


Figure 1: Visualization of our synthetic haze generated by our the proposed pipeline. Our method is capable of synthesizing multiple essences of haze, fog and smoke, within different colors, morphologies and optical properties.

- $I_{in,c}(x)$  is the original, effect-free color of the scene at pixel  $x$ .
- $T_c(x)$  is the **transmittance**, representing the fraction of light that successfully travels from the object to the camera without being scattered or absorbed.
- $A_c$  is the color of the **airlight**, which is the ambient environmental light scattered towards the camera by the atmospheric particles. This parameter is crucial for defining the hue of the haze (e.g., white for fog, sky-tinted for haze, warm gray for smoke).
- $\omega_{0,c}$  is the **single-scattering albedo**, a value in  $[0, 1]$  indicating the proportion of light extinction that is due to scattering versus absorption. For non-absorptive media like fog and haze,  $\omega_0 \approx 1.0$ . For absorptive media like smoke,  $\omega_0 < 1.0$ .
- $\kappa$  is an **anisotropy gain factor**, derived from the Henyey-Greenstein phase function. It accounts for directionality of scattering (i.e., whether particles scatter light more strongly forward or backward). For simplicity in our large-scale synthesis, we set  $\kappa = 1$ , modeling isotropic scattering.
- $\eta$  is a **multiple-scattering boost exponent** ( $0 < \eta \leq 1$ ). This term provides a compact approximation for the effects of multiple scattering events. A lower value of  $\eta$  increases the brightness of the veil, simulating the appearance of denser media where light scatters multiple times before reaching the camera.

#### B.1.1 OPTICAL DEPTH AND TRANSMITTANCE

The transmittance  $T_c(x)$  is determined by the optical depth  $\tau_c(x)$  of the medium along the line of sight, following the Beer-Lambert law:

$$T_c(x) = e^{-\tau_c(x)} \quad (2)$$

The optical depth is the integral of the extinction coefficient  $\beta_{t,c}$  over the distance  $d(x)$  from the camera to the object at pixel  $x$ . To model realistic atmospheres, we assume an exponential decay of particle density with height  $h$ :

$$\beta_{t,c}(h) = \beta_{t0,c} \cdot e^{-h/H} \quad (3)$$

where  $\beta_{t0,c}$  is the base extinction coefficient at a reference height (e.g., sea level), and  $H$  is the **scale height**, which defines how rapidly the atmosphere thins out. For a near-horizontal viewing angle, the optical depth can be approximated as:

$$\tau_c(x) \approx \beta_{t0,c} \cdot e^{-h(x)/H} \cdot d(x) \quad (4)$$

The base extinction coefficient  $\beta_{t0,c}$  is directly related to the meteorological visibility  $V$  by the Koschmieder formula,  $\beta_{t0} \approx 3.912/V$ .

### B.1.2 GEOMETRIC INPUTS: DEPTH AND HEIGHT

Our rendering pipeline requires per-pixel geometric information.

- **Depth:** We use monocular depth maps estimated from the clean input images by Marigold (Ke et al., 2025). These normalized depth maps are converted to distance in meters,  $d(x)$ , using a scene-specific maximum distance  $d_{max}$ .
- **Height:** When a true height map is unavailable, we utilize a **screen-space height proxy**:  $h(x) = h_{max} \cdot (1 - y_{norm})$ , where  $y_{norm}$  is the normalized vertical coordinate of the pixel (0 at the top, 1 at the bottom). This proxy effectively treats pixels near the horizon as being at a higher altitude, enabling the synthesis of effects like low-lying valley fog that is denser at the bottom of the image.

### B.1.3 COLOR SPACE AND PARAMETERIZATION

All physical calculations are performed in a linear RGB color space to ensure correctness. Input images, which are typically encoded in sRGB, are first decoded to linear space. After the atmospheric effects are composed, the resulting linear image is encoded back to sRGB. For our large-scale data generation, we programmatically varied all key parameters—including *visibility*, *airlight* color, *eta*, and  $H$ —across wide, physically plausible ranges to generate a diverse set of training pairs. We also introduced a random baseline value to the optical thickness  $\tau$  in each render to add further variety.

## B.2 PROCEDURAL GENERATION OF NON-HOMOGENEOUS MEDIA

To simulate complex, turbulent atmospheric effects like patchy fog or smoke, we integrated a procedural texture generator into our pipeline. This process creates realistic, wispy patterns that are used to spatially modulate the density of the rendered haze.

The generation process involves two main steps:

1. **Vector Field Generation:** We first generate a 2D vector field  $\vec{V}(\vec{p})$  for each pixel coordinate  $\vec{p} = (x, y)$ . The components of this field are determined by two independent layers of Perlin noise,  $P(\cdot)$ , distinguished by unique seeds  $(\theta_1, \theta_2)$ , which simulates a turbulent flow field. The resulting vectors are normalized to create a unit vector field  $\hat{V}(\vec{p})$ :

$$\vec{V}(\vec{p}) = \begin{bmatrix} P(\vec{p}; \theta_1) \\ P(\vec{p}; \theta_2) \end{bmatrix}, \quad \hat{V}(\vec{p}) = \frac{\vec{V}(\vec{p})}{\|\vec{V}(\vec{p})\| + \epsilon} \quad (5)$$

where  $\epsilon$  is a small constant to prevent division by zero.

2. **Path Blurring (Advection):** A base noise texture,  $M_0(\vec{p})$ , is iteratively advected along the vector field  $\hat{V}(\vec{p})$  for  $N$  steps. In each step  $k$ , the new texture  $M_{k+1}(\vec{p})$  is a blend of the previous texture  $M_k(\vec{p})$  and a value sampled from a forward-projected position  $\vec{p}'$ . This technique smears the initial pattern, creating characteristic streaks. The update rule is:

$$M_{k+1}(\vec{p}) = (1 - \alpha) \cdot M_k(\vec{p}) + \alpha \cdot M_k(\vec{p}') \quad (6)$$

where  $\vec{p}' = \vec{p} + \hat{V}(\vec{p}) \cdot \delta_s$ . Here,  $\delta_s$  is the step length,  $\alpha$  is a blending factor (we use  $\alpha = 0.5$ ), and  $M_k(\vec{p}')$  is obtained via bilinear interpolation as  $\vec{p}'$  may have non-integer coordinates.

The resulting grayscale texture after  $N$  iterations,  $M_N(\vec{p})$ , is then used as a spatial density modulator,  $M(x)$ , for the extinction coefficient. The final optical depth calculation is modified to incorporate this texture:

$$\tau_c(x) \approx (\beta_{t0,c} \cdot M(x)) \cdot e^{-h(x)/H} \cdot d(x) \quad (7)$$

This allows us to render haze that is not uniform but varies in density and structure across the image, greatly enhancing the realism and challenge of our synthetic dataset. We also illustrate a sample image synthesized with multiple different types of haze, fog or smoke in Fig. 1.

## C NON-REFERENCE EVALUATION METRICS

To rigorously assess the performance of our model on in-the-wild images where a ground-truth reference is unavailable, we employed specialized non-reference evaluation paradigms. These metrics are designed to provide both a quantitative measure of detail recovery and a qualitative score that emulates human perceptual judgment.

### C.1 RESIDUAL CONTRAST GAIN

While local contrast is a well-established indicator of image sharpness and detail, commonly used in non-reference dehazing or similar tasks (Wang et al., 2024). However since the measurements are averaged over the entire image, for localized effects like some types of lens flares or local shadows, the global evaluation is not significant. To overcome this limitation, we measure the **Residual Contrast Gain**, which quantifies the change in local contrast exclusively within the image regions modified by our model. This approach ensures that the evaluation focuses directly on the model’s restoration efficacy. The computation is performed via the following steps:

1. **Identification of Edited Regions.** Given a grayscale input image  $I_{in}$  and the model’s grayscale output  $I_{out}$ , we first identify the edited regions by computing a pixel-wise absolute difference map,  $D(\vec{p}) = |I_{in}(\vec{p}) - I_{out}(\vec{p})|$ , for all pixel coordinates  $\vec{p}$ . A binary edit mask,  $M_{edit}$ , is then generated by applying a threshold to this difference map, isolating the set of modified pixels over which the analysis is performed.
2. **Local Contrast Calculation.** We define the local contrast at a pixel  $\vec{p}$ , denoted  $C(\vec{p})$ , as the standard deviation of pixel intensities within a  $k \times k$  window centered at  $\vec{p}$ . This operation is performed for both the input and output images, yielding local contrast maps  $C_{in}$  and  $C_{out}$ .
3. **Gain Computation.** The final Residual Contrast Gain,  $\Delta C_{res}$ , is the difference between the average local contrast of the output and input images, computed exclusively over the set of edited pixels (where  $M_{edit} = 1$ ). This is formulated as:

$$\Delta C_{res} = \text{mean}_{\vec{p}|M_{edit}(\vec{p})=1} (C_{out}(\vec{p}) - C_{in}(\vec{p})) \quad (8)$$

A positive  $\Delta C_{res}$  value indicates a net increase in detail and texture within the restored regions.

### C.2 QWENQA: VLM-BASED ASSESSMENT

Moreover, we also developed the **QwenQA** evaluation metric, to leverage the powerful Vision-Language Model (VLM) for more human-like visual assessments. Our framework is built upon the **Qwen2.5-VL-72B-Instruct** model (Bai et al., 2025). The evaluation protocol is designed for consistency and automated parsing, involving three key stages:

1. **Input Standardization.** To eliminate resolution as a confounding variable, the model’s prediction image is first resampled to match the exact dimensions of the original input image, ensuring a fair comparison context for the VLM.
2. **Constrained Prompt Engineering.** The core of QwenQA lies in a meticulously engineered prompt designed to elicit a precise and quantitative response. The prompt structure includes:
  - *Role Assignment:* The VLM is instructed to act as a “top-tier image quality assessment expert,” priming it to leverage its most relevant internal knowledge.
  - *Task Definition:* The prompt provides clear context, defining “Image A” as the original with a specific artifact (e.g., ‘haze’, ‘shadow’) and “Image B” as the processed result.



- *Objective Quantization*: The VLM’s objective is narrowly focused on a single quantitative task: “evaluate the percentage by which the '[artifact name]' is reduced in Image B compared to Image A”. This transforms a descriptive task into a quantitative one.
  - *Strict Output Formatting*: The prompt strictly constrains the VLM’s output to a specific format: “Score: [number]%”. This instruction explicitly forbids any additional descriptive text, explanations, or conversational filler, which is critical for reliable automated parsing.
3. **Automated Score Parsing**. The final step is to parse the VLM’s structured textual output. A regular expression is used to robustly extract the numerical percentage score from the response, yielding the final QwenQA score.

## D IMPLEMENTATION DETAILS

### D.1 HAZE SYNTHESIS DETAILS

Our primary objective in data expansion was to generate a challenging and realistic training set that surpasses the limitations of existing synthetic datasets. To achieve this, we developed a high-throughput synthesis pipeline to apply our physically-motivated atmospheric rendering model on a large scale. This section details the parameterization for various haze types, the batch processing architecture, and the datasets involved.

**Parameterization for Diverse Atmospheric Effects.** The versatility of our rendering model allows us to simulate a wide range of atmospheric conditions by adjusting a few key physical parameters. We defined distinct configurations for haze, fog, and smoke, which were systematically varied to ensure a broad data distribution.

- **Haze**: To simulate different environmental conditions, we primarily varied the *airlight* color and *visibility*. For instance, we used sky-tinted colors like (153, 174, 215) for typical haze, warmer tones such as (200, 180, 140) for urban pollution, and grayish colors like (210, 210, 220) for high-altitude conditions. Visibility was typically set in the range of 100m to 1000m to produce varying levels of haze density.
- **Fog**: Fog is characterized by its dense, non-absorptive particles. We simulated this by setting the single-scattering albedo  $\omega_0$  to (1.0, 1.0, 1.0) and using a neutral white airlight. Fog density was controlled by varying *visibility* (from 30m to 1000m) and the multiple-scattering boost exponent  $\eta$  (typically between 0.5 and 1.0). To simulate low-lying or valley fog, we significantly reduced the scale height  $H$  (e.g., to 30-60m) to confine the effect to the lower parts of the scene.
- **Smoke**: Unlike haze and fog, smoke is an absorptive medium. This was modeled by setting  $\omega_0$  to values less than 1.0 (e.g., 0.75 to 0.85). The *airlight* was configured with warm, darker colors like (180, 150, 120) or (160, 120, 90) to represent the tint of the smoke particles. The scale height  $H$  was generally kept low (e.g., 40-50m) to simulate ground-level smoke plumes.

**Large-Scale Batch Synthesis Architecture.** To apply these configurations across a massive number of images, we implemented an efficient, parallelized processing pipeline. The core rendering engine was ported to PyTorch (Paszke et al., 2019) to leverage GPU acceleration. We utilized multiprocessing to create a pool of worker processes. In a multi-GPU environment, these workers were assigned to available GPUs in a round-robin fashion, enabling concurrent rendering of multiple image-configuration pairs. Each worker independently handled the data I/O, pre-processing (color space conversion, data normalization), GPU-based rendering, and post-processing of the synthesized hazy image. This architecture allowed us to generate our extensive dataset in a time-efficient manner.

**Datasets for Synthesis.** As stated in our methodology, our goal was to enhance existing large-scale datasets by generating more challenging and realistic haze effects. We leveraged the high-quality, clean ground truth images from public benchmarks, primarily RESIDE (Li et al., 2018) and HAZESPACE (Islam et al., 2024). For each clean image in these datasets, we first estimated a monocular depth map (Ke et al., 2025) and then applied our full suite of atmospheric rendering

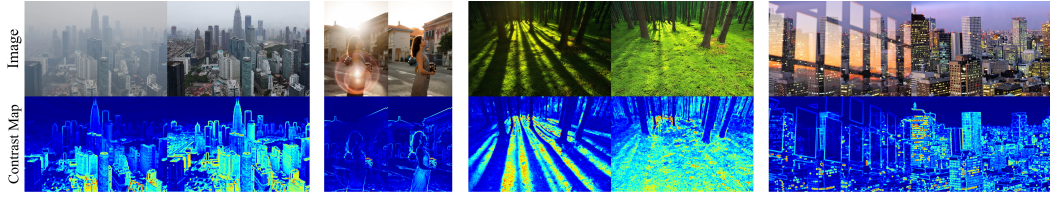


Figure 2: Contrast maps of image before and after edit by OmniClear. Significant enhancements of contrast inside effect regions are observed, indicating our method successfully enhances the degraded image details.

configurations, resulting in a significant expansion of the training data with diverse and physically plausible haze, fog, and smoke effects.

## D.2 TRAINING DETAILS

Our work builds upon an internal pretrained DiT-based image editing model that has demonstrated strong capabilities in general inpainting tasks, such as object addition, removal, and modification. This provides a robust starting point for fine-tuning on our specialized soft-effects dataset. A key aspect of our training methodology is a hierarchical data sampling strategy designed to balance contributions from numerous datasets across multiple tasks. Our data pipeline first groups datasets by their primary task (e.g., shadow removal, dehazing, reflection removal, etc.). During each training step, a task is uniformly sampled, and then a specific dataset within that task group is selected based on a predefined sampling weight. This weighting ratio is configured for each dataset, allowing us to strategically oversample smaller, high-quality real-world datasets to learn the knowledge without domain gaps, while still benefiting from the diversity of larger-scale synthetic data sources to prevent overfitting and enhance the generalization ability. This ensures the model receives a balanced and comprehensive exposure to all types of soft effects.

For the fine-tuning process, our model operates within the DDPM (Ho et al., 2020) framework, which is adapted to use continuous timesteps for increased flexibility. Notably, we employ  $v$ -parameterization instead of the standard  $\epsilon$ -parameterization to improve training stability and sample quality. Our training objective is to predict the noise added to the clean image’s latent representation at a given timestep. The loss function is the mean squared error (MSE) between the predicted noise and the ground truth noise, with a timestep-dependent weighting scheme applied to balance the contribution of different noise levels throughout the training. We train the model for 10k steps at a resolution of 1024x1024. We employ the AdamW optimizer with a learning rate of  $1.2 \times 10^{-5}$ , governed by a linear warmup of 2000 steps followed by a cosine decay schedule. Our OmniClear is trained on all of the data mentioned above with 8 NVIDIA A100 80G for 10k iterations.

## D.3 EVALUATION DETAILS FOR BASELINES

When evaluating the generalist baselines, we provided detailed and specific text prompts to ensure they could achieve their optimal performance. These prompts explicitly described the effect to be removed and the relevant scene context, for instance: *“remove the atmosphere haze completely in this image”* or *“remove the shadow casted by the giraffe on the grass”*. Furthermore, to account for the stochastic nature of generative models, if a model performed poorly or failed to remove the effect on a particular sample, we conducted multiple attempts to ensure we are not using ambiguous or vague text prompts. This is a fair evaluation and mitigates biases arising from individual random outcomes. In contrast, our OmniClear has minimal dependency on text prompts. In our framework, the text serves merely as a high-level task indicator (e.g., *“remove haze”*) without requiring a detailed description of the scene’s content. Consequently, our approach achieves stable and robust results without the need for iterative prompt engineering.

## E MORE VISUAL RESULTS AND QUALITY ANALYSIS

### E.1 MORE VISUAL RESULTS

We provide more visual results in Fig. 3 and Fig. 4, by randomly pick in-the-wild photos degraded by soft effects, our OmniClear shows perfect robustness on thoroughly removing the. Besides, OmniClear is also capable of generating or enhancing multiple effects aesthetically.

### E.2 CONTRAST ANALYSIS

To further investigate how OmniClear improves image quality, we visualize the local contrast maps of images before and after editing, as shown in Figure 2. A significant enhancement in contrast is observed within the regions originally degraded by soft effects. This indicates that our method not only removes the obstructive artifacts but also successfully restores and enhances the underlying image details and textures that were suppressed by the effects, leading to a clearer and more vivid output.

## REFERENCES

- Codruta O Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 754–762, 2018a.
- Codruta O Ancuti, Cosmin Ancuti, Mateu Sbert, and Radu Timofte. Dense-haze: A benchmark for image dehazing with dense-haze and haze-free images. In *2019 IEEE international conference on image processing (ICIP)*, pp. 1014–1018. IEEE, 2019.
- Codruta O Ancuti, Cosmin Ancuti, and Radu Timofte. Nh-haze: An image dehazing benchmark with non-homogeneous hazy and haze-free images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 444–445, 2020.
- Codruta O Ancuti, Cosmin Ancuti, Florin-Alexandru Vasluianu, and Radu Timofte. Ntire 2021 nonhomogeneous dehazing challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 627–646, 2021.
- Codruta O Ancuti, Cosmin Ancuti, Florin-Alexandru Vasluianu, Radu Timofte, Han Zhou, Wei Dong, Yangyi Liu, Jun Chen, Huan Liu, Liangyan Li, et al. Ntire 2023 hr nonhomogeneous dehazing challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1808–1825, 2023.
- Codruta O Ancuti, Cosmin Ancuti, Florin-Alexandru Vasluianu, Radu Timofte, Yidi Liu, Xingbo Wang, Yurui Zhu, Gege Shi, Xin Lu, Xueyang Fu, et al. Ntire 2024 dense and non-homogeneous dehazing challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6453–6468, 2024.
- Cosmin Ancuti, Codruta O Ancuti, Radu Timofte, and Christophe De Vleeschouwer. I-haze: A dehazing benchmark with real hazy and haze-free indoor images. In *International conference on advanced concepts for intelligent vision systems*, pp. 620–631. Springer, 2018b.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- Yuekun Dai, Dafeng Zhang, Xiaoming Li, Zongsheng Yue, Chongyi Li, Shangchen Zhou, Ruicheng Feng, et al. Mipi 2024 challenge on nighttime flare removal: Methods and results. *arXiv preprint arXiv:2404.19534*, 2024.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

- Md Tanvir Islam, Nasir Rahim, Saeed Anwar, Muhammad Saqib, Sambit Bakshi, and Khan Muhammad. Hazespace2m: A dataset for haze aware single image dehazing. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 9155–9164, 2024.
- Bingxin Ke, Kevin Qu, Tianfu Wang, Nando Metzger, Shengyu Huang, Bo Li, Anton Obukhov, and Konrad Schindler. Marigold: Affordable adaptation of diffusion-based image generators for image analysis. *arXiv preprint arXiv:2505.09358*, 2025.
- Chenyang Lei and Qifeng Chen. Robust reflection removal with reflection-free flash-only cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14811–14820, 2021.
- Chenyang Lei, Xuhua Huang, Mengdi Zhang, Qiong Yan, Wenxiu Sun, and Qifeng Chen. Polarized reflection removal with perfect alignment in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1750–1758, 2020.
- Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE transactions on image processing*, 28(1):492–505, 2018.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- Liangqiong Qu, Jiandong Tian, Shengfeng He, Yandong Tang, and Rynson WH Lau. Deshadownet: A multi-context embedding deep network for shadow removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4067–4075, 2017.
- Florin-Alexandru Vasluianu, Tim Seizinger, and Radu Timofte. Wsr: A novel benchmark for high resolution image shadow removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1826–1835, 2023.
- Jifeng Wang, Xiang Li, and Jian Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1788–1797, 2018.
- Yongzhen Wang, Xuefeng Yan, Fu Lee Wang, Haoran Xie, Wenhan Yang, Xiao-Ping Zhang, Jing Qin, and Mingqiang Wei. Ucl-dehaze: Toward real-world image dehazing via unsupervised contrastive learning. *IEEE Transactions on Image Processing*, 33:1361–1374, 2024.
- Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In *Proceedings of the european conference on computer vision (ECCV)*, pp. 654–669, 2018.
- Ruikun Zhang, Hao Yang, Yan Yang, Ying Fu, and Liyuan Pan. Lmhaze: intensity-aware image dehazing with a large-scale multi-intensity real haze dataset. In *Proceedings of the 6th ACM International Conference on Multimedia in Asia*, pp. 1–1, 2024.
- Xinyi Zhang, Hang Dong, Jinshan Pan, Chao Zhu, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Fei Wang. Learning to restore hazy video: A new real-world dataset and a new method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9239–9248, 2021.
- Yurui Zhu, Xueyang Fu, Peng-Tao Jiang, Hao Zhang, Qibin Sun, Jinwei Chen, Zheng-Jun Zha, and Bo Li. Revisiting single image reflection removal in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25468–25478, 2024.



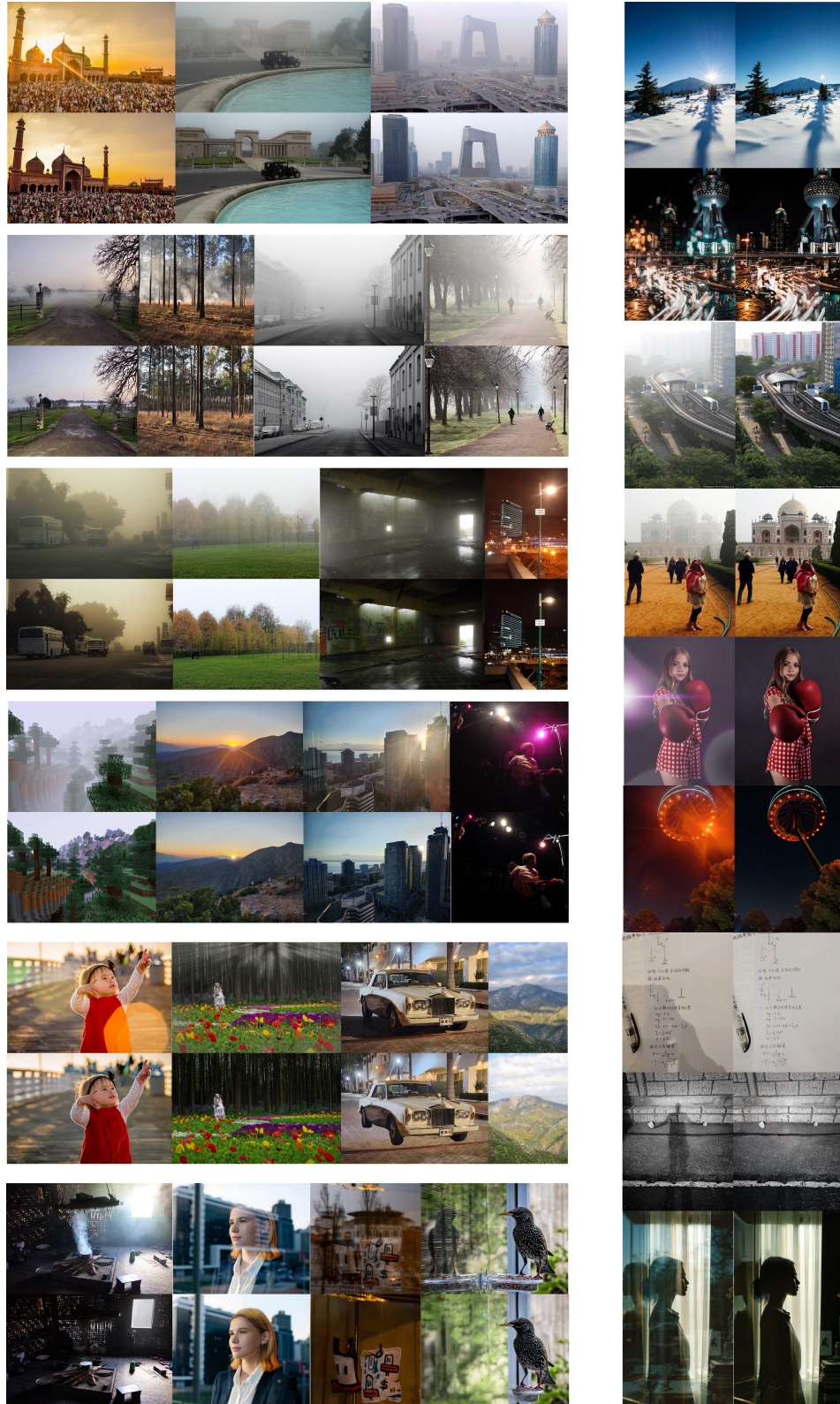


Figure 3: Gallery: Removing effects with OmniClear.



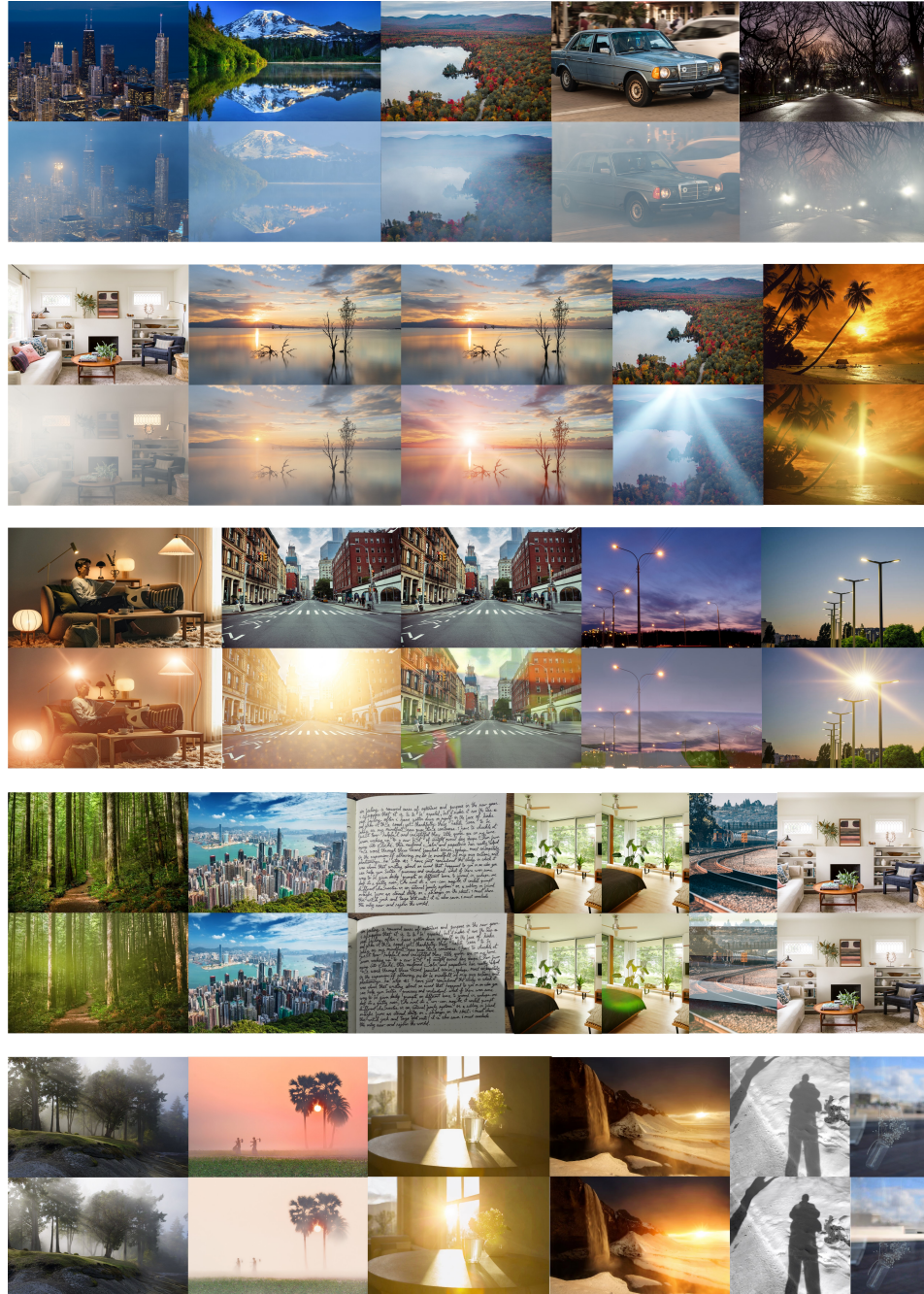


Figure 4: Gallery: Adding or enhancing effects with OmniClear.