Visual-Instructed Degradation Diffusion for All-in-One Image Restoration

Supplementary Material

6. Additional Details of visual instruction

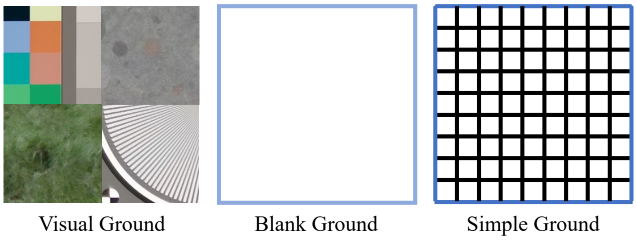
Visual Ground Blank Ground Simple Ground

Figure 5. Replacement of visual ground.

We focus on introducing visual instructions as they are promising in aligning with visual degradations. However, image degradations are “dangling,” meaning that their visual effects only manifest when they exist in the context of degraded images. Therefore, we first apply degradations on some “standard images” to visually demonstrate the degradations to the restoration model. We call this process *grounding* of degradations, the “standard images” are dubbed visual grounds.

The visual grounds should contain a wide range of possible visual constructs, patterns, structures, etc., that may occur in natural images to reveal the full extent of degradation. At the same time, in order to minimize the preference of the visual grounds for degradation and enhance its representation, it should not be in some fixed form. The visual grounds has been carefully designed and consists of regular textures, random textures, standard colors and natural objects. Each part was obtained by random selection from the pool, as shown in Fig. 7. We draw inspiration from image quality assessment [77, 78, 114] and select TE42 [26], a family of charts commonly used for camera testing and visual analysis, to construct a pool of visual grounds. TE42 comprises a rich combination of textures and color palettes.

To better encode degradations themselves (rather than image semantics), we first categorize them into regular and stochastic textures, calibrated colors, and natural images, to comprehensively encode diverse distortions. This is because different visual elements exhibit varying responses to distortions (e.g., flat regions poorly represent blur). Each category includes numerous base elements of the corresponding category sampled from TE42. During training and inference, we sample one element from each category and combine them randomly to constrict a visual ground. Then, according to the degradation we want, multiple degradations of varying intensities are randomly applied to the visual ground. For each degraded image that needs to be recovered, the visual ground is augmented with the same category of degradation as a visual instruction, either individually or as a mixed degradation category. Some samples of visual instructions are shown in Fig. 8. Finally, to better encode degradations themselves (rather than image semantics), we take the residual between the visual ground and its degraded version to facilitate independence from the image content.

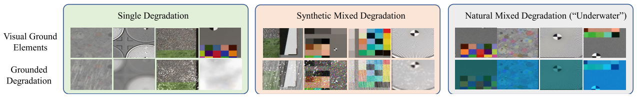
Single Degradation
Synthetic Mixed Degradation
Natural Mixed Degradation (“Underwater”)
Visual Ground Elements
Grounded Degradation

Figure 6. Samples of visually degraded elements from the pool of visual grounds.

In the ablation experiments, we replace visual ground with blank ground and simple ground to verify the effectiveness of the proposed visual ground. The replacement Gound is shown in Fig. 5, where blank ground is a solid color image of equal size to the visual ground, and simple ground adds simple regular shapes. Visualizations of some visually degraded elements from the pool of visual grounds are shown in Section 6. Numerous distortion-sensitive elements ensure targeted responses to specific degradations. The compositional nature of visual instructions improves generalization under compound and real-world distortions. Quantitative results on direct testing are shown in Tables 3 and 4. Training on extensive synthetic distortions enables generalization to real distortions.

7. More Details About Datasets

Our dataset in Sec. 4 consists of All-in-One datasets, mixed distortion datasets, and natural mixture datasets.

All-in-One datasets contain images from a variety of different image recovery tasks. Our method and some of the comparison methods are trained and tested uniformly on these datasets. Details of these datasets are given below:

- Motion Deblur: collected from GoPro [81] dataset containing 2103 and 1111 training and testing images, RealBlur [96] dataset containing 7516 and 1961 training and testing images.
- Defocus Deblur: collected from DPDD [2] dataset containing 350 and 76 training and testing images.
- Image Desnowing: collected from Snow100K [67] dataset containing 50000 and 50000 training and testing images, and RealSnow [149] dataset containing 61500(crops) and 240 training and testing images.

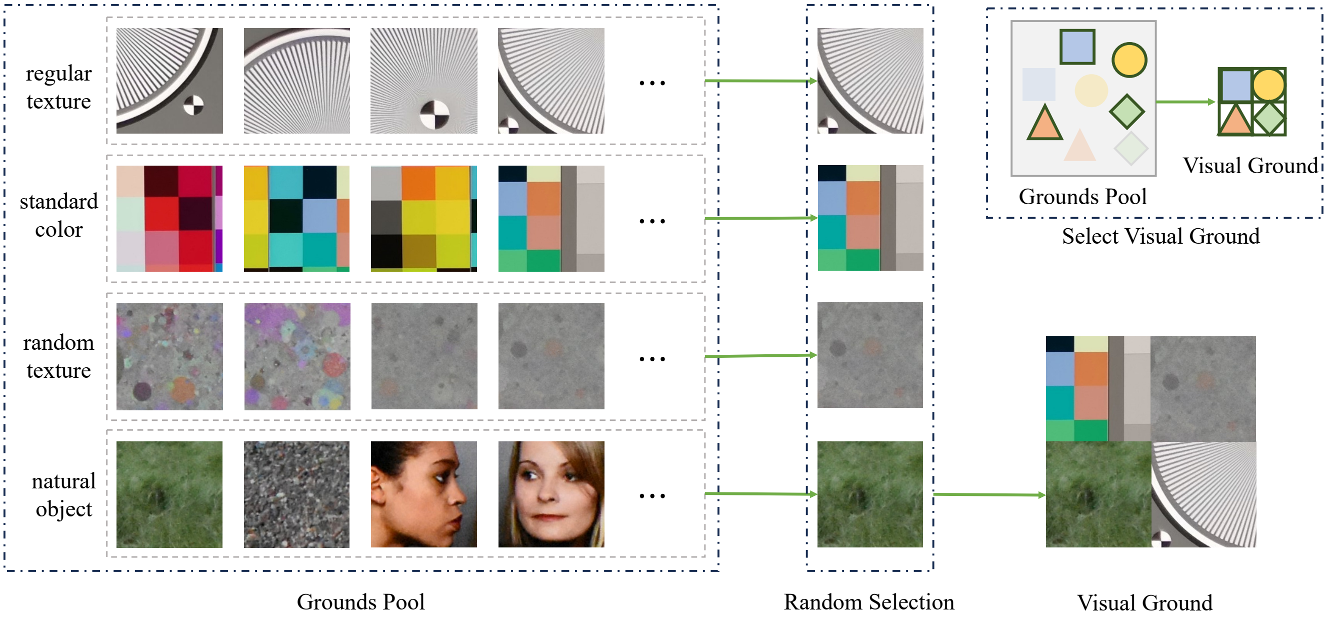

Figure 7. Details of visual ground.

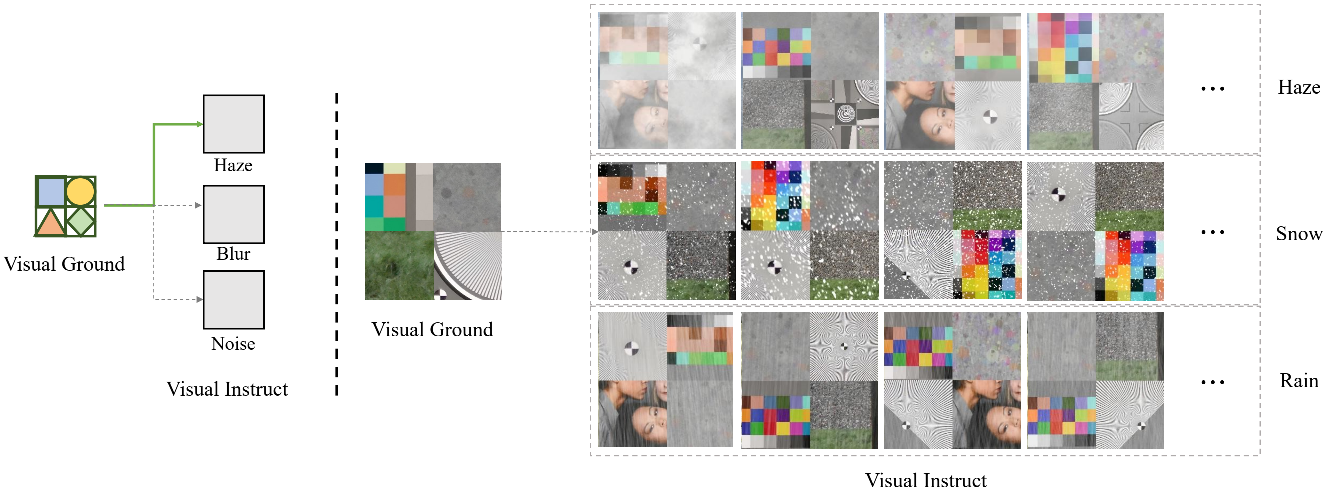

Figure 8. Samples of visual instruct.

- Image Dehazing: collected from RESIDE [49] dataset containing 12591 training images, and Dense-Haze [6] dataset containing 49 and 6 training and testing images.
- Raindrop Removal: collected from RainDrop [87] dataset containing 861 and 307 training and testing images, and RainDS [90] dataset containing 150 and 98 training and testing images.
- Image Deraining: collected from Rain1400 [30] dataset containing 12600 and 1400 training and testing images, Outdoor-Rain [54] dataset containing 8100 and 900 training and testing images, and LHP [33] (only use for testing) dataset containing 300 testing images.
- Real Denoising: collected from SIDD [1] dataset containing 288 and 32 training and testing images.
- JPEG Artifact removal: training dataset collected from DIV2K and FLICKR2K [3] containing 900 and 2650 images. Testing dataset collected from LIVE1 [104] containing 29 testing images.

Mixed distortion datasets' LQ image has three different distortions: rain, snow, and noise. These distortions are superimposed on the image of WED [75] dataset in all 6 orders. The reason rain, snow, and noise were chosen is because they don't conflict with each other (e.g. blur and noise). The variance of the noise is 25 and the size and speed of the rain line and snowflakes are randomized. Our method and all comparison methods train 200 iterations on

Table 6. Comparison of perceptual metrics with state-of-the-art task-specific methods and all-in-one methods on 8 tasks. The best and second-best performances are in red and bold font, with the top 2 with a light black background.

Motion Deblur (GoPro [81])			**Defocus Deblur (DPDD [2])**			**Desnowing (Snow100K-L [67])**		
Method	FID↓	LPIPS↓	Method	FID↓	LPIPS↓	Method	FID↓	LPIPS↓
Task Specific								
MPRNet[133]	10.98	0.091	DRBNet[99]	49.04	0.183	DesnowNet[67]	-	-
Restormer[134]	10.63	0.086	Restormer[134]	**44.55**	**0.178**	DDMSNet[138]	3.24	0.096
Stripformer[111]	**9.03**	**0.079**	NRKNet[92]	55.23	0.210	DRT[58]	8.15	0.135
DiffIR[125]	9.65	0.081	FocalNet[21]	48.82	0.210	WeatherDiff[83]	2.81	0.100
All in One								
AirNet[50]	9.65	0.081	AirNet[50]	58.82	0.193	AirNet[50]	3.92	0.105
PromptIR[86]	15.31	0.122	PromptIR[86]	52.64	0.197	PromptIR[86]	3.79	0.100
DA-CLIP[71]	17.54	0.131	DA-CLIP[71]	57.43	0.201	DA-CLIP[71]	3.11	0.098
MPerceiver[4]	10.69	0.089	MPerceiver[4]	48.22	0.190	MPerceiver[4]	**2.31**	**0.087**
Defusion(Ours)	**8.73**	**0.052**	**Defusion(Ours)**	**20.20**	**0.066**	**Defusion(Ours)**	**0.70**	**0.094**
Raindrop Removal (RainDrop [87])			**Deraining (Rain1400 [30])**			**Real Denoising (SIDD [1])**		
Method	FID↓	LPIPS↓	Method	FID↓	LPIPS↓	Method	FID↓	LPIPS↓
Task Specific								
AttentGAN[87]	33.33	0.056	Uformer[120]	23.31	0.061	MPRNet[133]	49.54	0.200
Quanetal.[91]	30.56	0.065	Restormer[134]	20.33	0.050	Uformer[120]	47.18	0.198
IDT[126]	25.54	0.059	DRSformer[17]	20.06	0.050	Restormer[134]	47.28	0.195
UDR-S^2[16]	27.17	0.064	UDR-S^2[16]	19.89	0.053	ART[136]	42.38	0.189
All in One								
AirNet[50]	33.34	0.073	AirNet[50]	22.38	0.058	AirNet[50]	51.20	**0.134**
PromptIR[86]	35.75	0.073	PromptIR[86]	22.59	0.058	PromptIR[86]	50.52	0.198
DA-CLIP[71]	29.38	0.078	DA-CLIP[71]	35.01	0.116	DA-CLIP[71]	34.56	0.186
MPerceiver[4]	**19.37**	**0.044**	MPerceiver[4]	**17.82**	**0.049**	MPerceiver[4]	**41.11**	0.191
Defusion(Ours)	**10.91**	**0.039**	**Defusion(Ours)**	**12.93**	**0.057**	**Defusion(Ours)**	**32.77**	**0.139**

this dataset for a fair comparison.

Considering the mix distortion described above as a synthetic form, we added the underwater dataset as a natural mix distortion dataset and performed image restoration. Underwater dataset collected from EUVP [39] dataset containing 515 testing images, and TURBID [24] dataset containing 60 testing images. Both our method and the comparison method are trained on All-in-One datasets and tested directly on the underwater dataset.

Sample visualizations for each task and dataset are shown in Fig. 5 to better understand these datasets.

8. Implementation Details

For the visual instruct tokenizer, we follow the implementation of [27][1] and adapt its ImageNet-pretrained [100] VQ-GAN model to our framework. The input size is 224 × 224, randomly cropped during training and center-cropped during inference. The embedding size is 256, and the vocabulary size is 1024. The encoder consists of five levels of channel sizes $[128, 128, 256, 256, 512]$, each level consists of two residual blocks, and the last two layers additionally have an attention block. Each level except the last one

[1] https://github.com/CompVis/taming-transformers

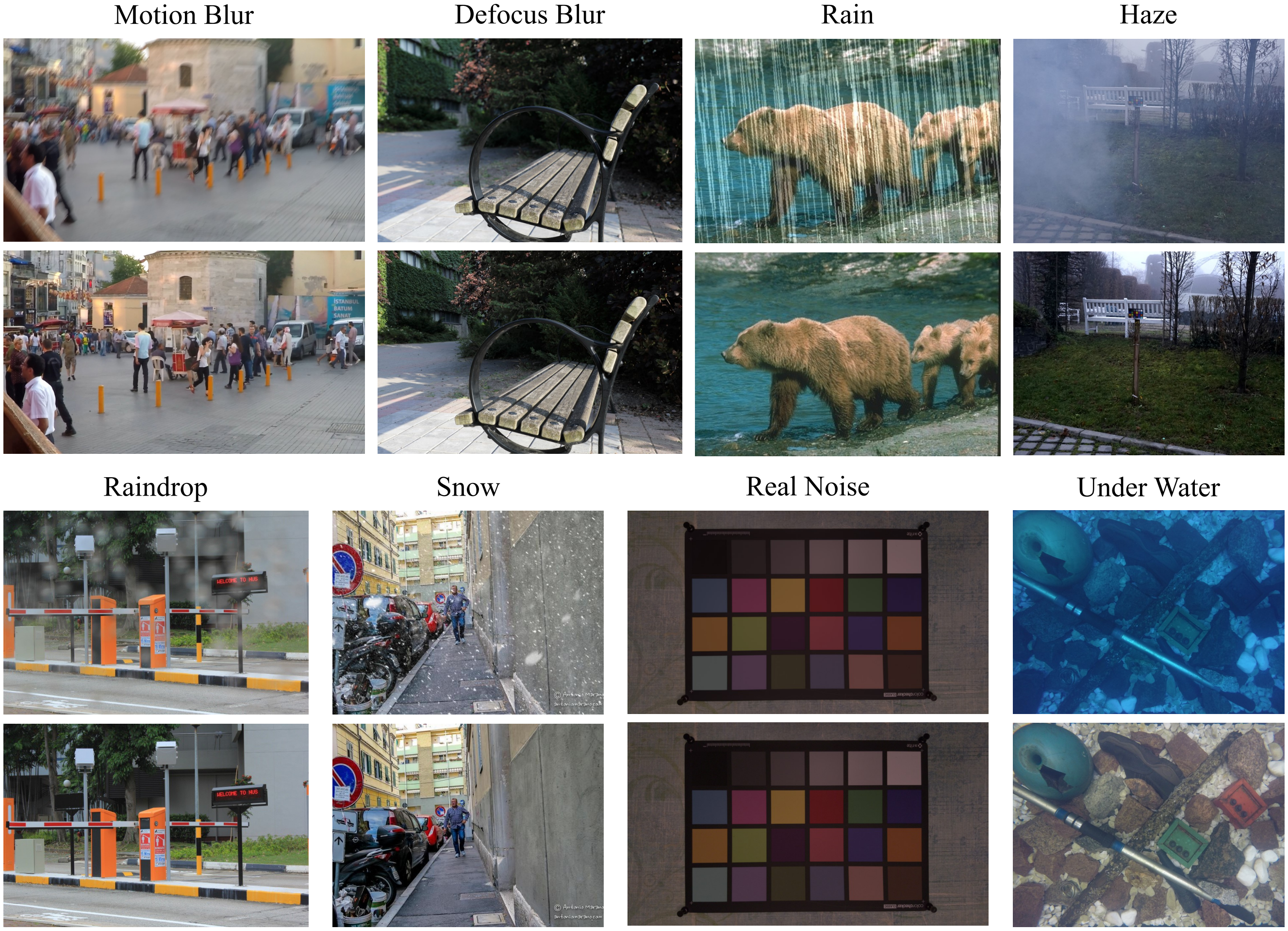

Figure 9. Samples of Datasets.

downsamples the input size by a factor of two. The decoder is symmetric to the encoder. The visual instruct tokenizer is trained by the combination of vector quantization (VQ) loss and a reconstruction loss described by Eqs. (2) to (4), where $\lambda = 1$. Additionally, we adopt a hinge-based adversarial loss [27, 97] with a weighting of 0.8. The discriminator follows the implementation of PatchGAN [40][2]. The visual instruct tokenizer is trained by the Adam optimizer [48] with a learning rate of 4.5e-6, and $\beta_1 = 0.5$, $\beta_2 = 0.9$. The batch size is 8. We adopt random horizontal flip as data augmentation.

The base Diffusion model uses U-Net [22] as the backbone for its restoration process, with weights pre-trained on the LAION-5B dataset [102]. Built upon it, we fine-tune the Defusion on the All-in-One dataset with a batch size of 32 and with an initial learning rate of 1e-4 and decaying to 1e-6 via cosine annealing [69]. We use the AdamW optimizer [68] with β_1 = 0.9 and β_2 = 0.99. In preprocessing, all inputs are normalized in the range $[-1, 1]$ and randomly cropped images to 256×256 for data augmentation. We train the Defusion model on eight NVIDIA A100 GPUs for a total of 600K iterations.

9. Addition Experimental Results

9.1. Comparison with SOTAs on Percetual Metrics

We also compare our method with previous SOTAs on perceptual metrics, namely FID [35] and LPIPS [141], which are usually more aligned with human visual preference than reference-based metrics such as PSNR and SSIM [119]. The results are summarized in Table 6. Note that we use the *same* Defusion model and hyperparameters as for Table 1, while showing the best per-dataset results of the other methods. Across all datasets, Defusion achieves comparable or best perceptual qualities than both unified and task-specific methods, usually with large margins. For example, Defusion improves FID by over 100% on defocus deblur and desnowing, while improving LPIPS by over 50% on

[2] https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix/blob/master/models/networks.py

motion deblur and defocus deblur. On other datasets, Defusion also demonstrates advantages over previous methods. It is even more impressive considering that Defusion is optimized for reference-based metrics and generalized directly to perceptual metrics. This clearly shows the promise of diffusion-based methods with regard to human perceptual priors. We believe developing more powerful diffusion-based image restoration methods that are optimized for user preferences is a valuable future research direction.

9.2. Visualization

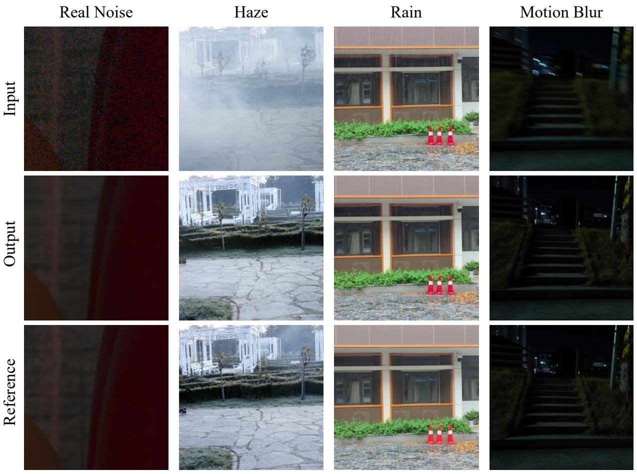

Figure 10. Visual results of real-world datasets.

Figures 10 to 12 provide more visualization results on the all-in-one synthesized/real-world/mixed-degradation datasets.

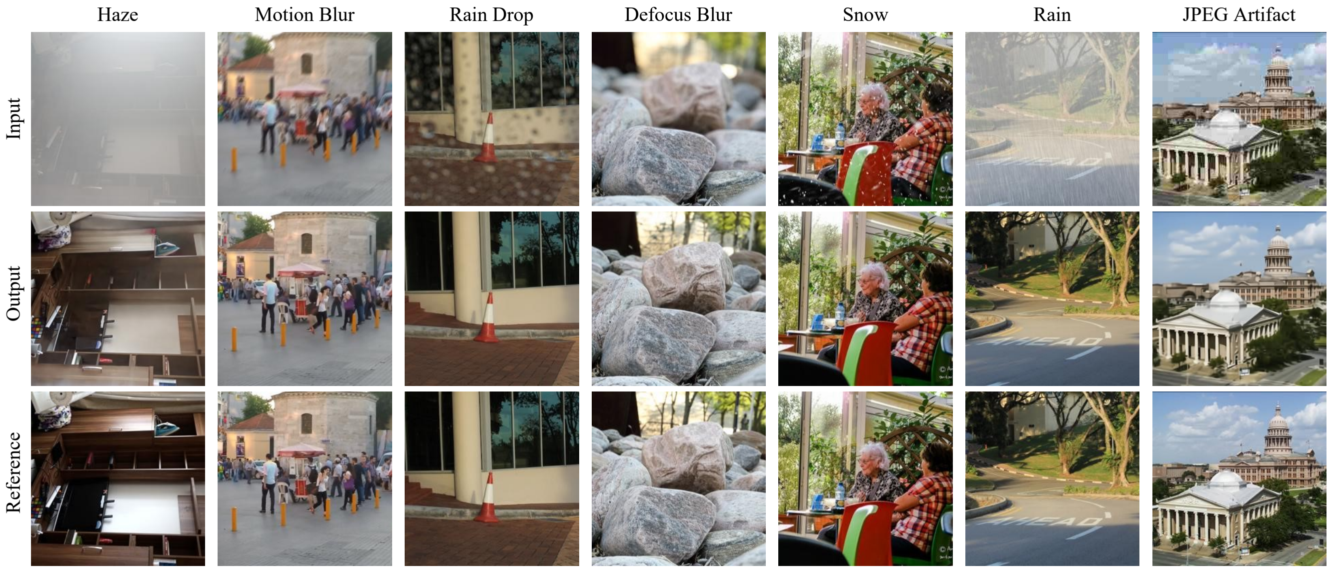

Figure 11. Visual results of synthesized datasets.

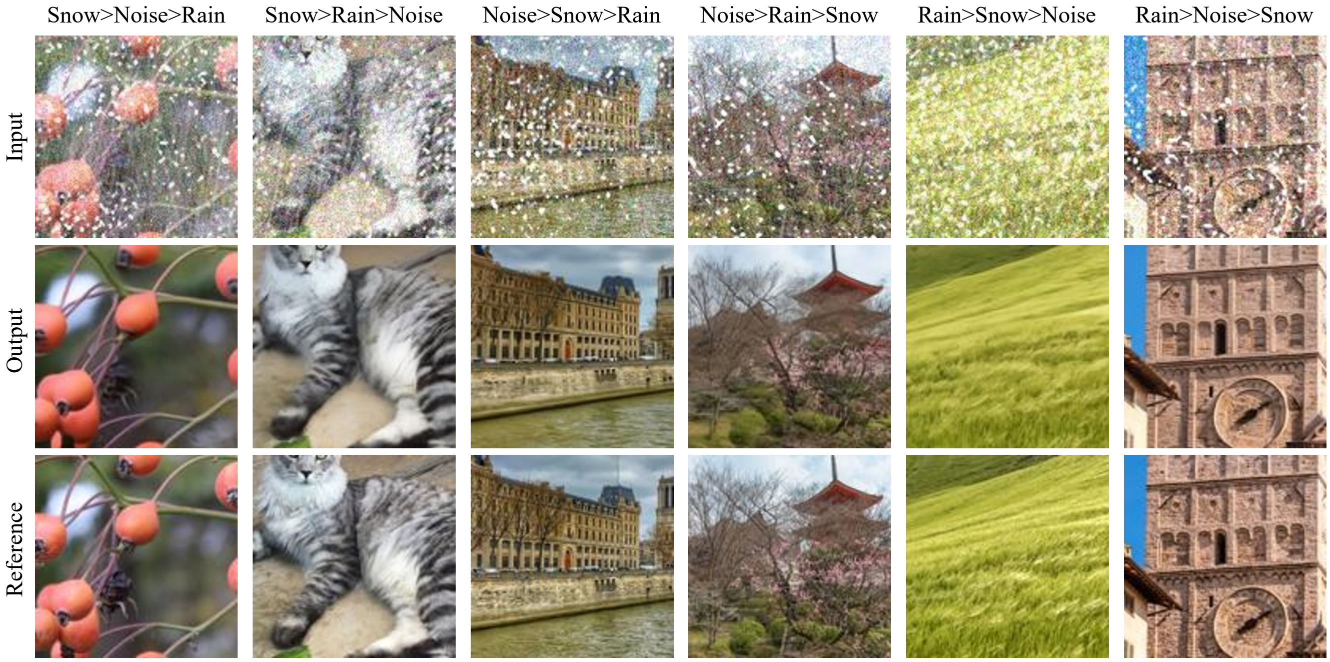

Figure 12. Visual results of mix distortion datasets.