Diff-Plugin: Revitalizing Details for Diffusion-based Low-level Tasks

Supplementary Material

This supplementary material is organized as follows. First, Sec. A details our network architectures. Then, Sec. B discusses some conceptions and method details mentioned in the main manuscript. Lastly, Sec. C elaborates on our experimental details.

A. Network details

Stable Diffusion. Here we briefly review the Stable Diffusion (SD) [22] model we used as the base model. By default, we adopt the widely-used version 1.4 for experiments. SD performs the diffusion process in the latent feature space to reduce the computation cost and enable stable training. It first encodes a *pixel-space* image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ into smaller *latent-space* image $\mathbf{z}_0 \in \mathbb{R}^{H'\times W'\times d}$ via a pre-trained VAE [2] encoder $Enc_V(\cdot)$, such that $\mathbf{z}_0 = Enc_V(\mathbf{I})$. Normally, the H and W of the input $\mathbf{I}$ is 512 in SD 1.4. The $Enc_V(\cdot)$ performs three times down-sampling, resulting in the H' and W' are all 64, and finally apply a quantization operation and a reparameterization on the last layer results in $d = 4$. Then, at time step t, SD adds noise to latent $\mathbf{z}_0$ to form noisy latent $\mathbf{z}_t$, which is fed to the diffusion model ϵ_θ for denoising. The architecture of SD is based on UNet [1] that consists of an encoder, a middle block, and a decoder. The denoised latent $\mathbf{z}_0$ is finally reconstructed to the pixel-space $\tilde{\mathbf{I}}$ via the decoder $Dec(\cdot)$ in VAE, such that $\tilde{\mathbf{I}} \approx \mathbf{I}$.

Task-Plugin. Given an input image $\mathbf{I}$, we first adopt a vision encoder $Enc_I(\cdot)$ (*i.e.*, CLIP-ViT-L/14 [19]) to extract general visual features of $\mathbf{I}$, such that $\mathbf{F}^v_{clip} = Enc_I(\mathbf{I})$ and $\mathbf{F}^v_{clip} \in \mathbb{R}^{256\times 1024}$. The Task-Prompt Branch (TPB) is a Multi-Layer Perception (MLP) network that distills the $\mathbf{F}^v_{clip}$ to $\mathbf{F}^p \in \mathbb{R}^{256\times 768}$. It comprises three MLP layers, each with 1024 units, and the final MLP layer reduces the feature dimension to 768.

To extract the task-specific spatial details effectively, we first capture the full image content information $\mathbf{F}$ from the input image $\mathbf{I}$, utilizing $Enc_V(\cdot)$ from SD. We set the variance in the reparameterization to zero and rely solely on the mean. Then Spatial Complement Branch (SCB) is then utilized to distill the vanilla $\mathbf{F}$. In SCB, we first employ an convolution layer with a kernel of 3×3, referred to as $Conv_{in}(\cdot)$, which is used for dimension adjustment. Then, SCB is structured with two sequential processing units, each comprising a standard ResNet block followed by a standard Cross-Attention Block [22]. The output of the first unit serves as the input for the second. We adopt the same architecture setting as SD for ResNet and Cross-Attention Blocks. The time step t undergoes a dimension adjustment through an MLP layer to align with the spatial features, and then it is directly added into the ResNet and Cross-Attention Blocks. In terms of the output spatial features, SCB combines the output from the $Conv_{in}(\cdot)$ layer, and the outputs from both processing units, into $\mathbf{F}^s$. While in the main manuscript, the spatial feature is represented by a single symbol $\mathbf{F}^s$ for clarity and simplicity, it is important to understand that this symbol encapsulates the rich, multi-source spatial features.

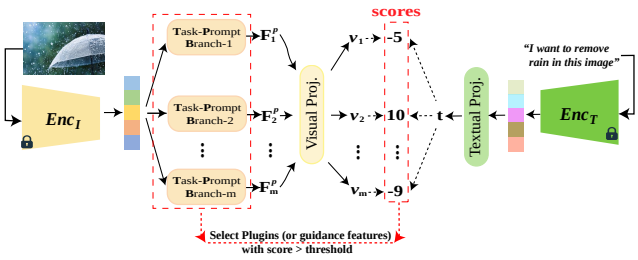


Figure 1. Schematic illustration of the developed Plugin-Selector.

Plugin-Selector. As depicted in Fig. 1, given a natural language sentence $\mathbf{T}$, Plugin-selector employs the pre-trained CLIP text encoder to derive a global textual embedding $\mathbf{F}^t_{clip} \in \mathbb{R}^{768}$. This embedding is then passed through the textual projection head, denoted as $TP(\cdot)$, to align visual embeddings and output text embedding $\boldsymbol{q} \in \mathbb{R}^{768}$. To calculate the similarity score s between the textual embedding $\boldsymbol{q}$ and various Task-Plugin embeddings, we apply Max-Pooling to transform each Task-Plugin embedding from $\mathbf{F}^p_i \in \mathbb{R}^{256\times 768}$ to a reduced form of $\mathbf{F}^p_i \in \mathbb{R}^{1\times 768}$. Then, we feed $\mathbf{F}^p_i$ into a shared visual projection head $VP(\cdot)$, to obtain the textual-visual aligned multi-modality embedding v_i. After that, a cosine similarity function is used to calculate the distance between v_i and $\boldsymbol{q}$. Here, both the text and visual projection heads are a Fully Connected (FC) layer with input and output dimensions of 768.

Placement of the injected spatial feature. Since the last stage of the SD's decoder contains three blocks, and $\mathbf{F}^s$ consists of three parts of spatial features, we treat $\mathbf{F}^s$ as residual features and add to these three blocks in the last stage of the SD's decoder in the order of the three $\mathbf{F}^s$ features. For each residual spatial feature, we append a zero-conv [39] layer for it and train them jointly with the Task-Plugin module.

B. Discussion

*"**Pre-defined mapping table**"* in Line-215 of our main manuscript. As shown in Table 1, a mapping table outlines workflows for complex tasks that require multiple Task-

Tasks	Workflow		
	1	2	3
Old photo restoration	Restoration	Colorization	Super-resolution
Adverse weather removal	Dehazing	Deraining	Desnowing

Table 1. Example of a pre-defined mapping table for complex low-level tasks that cannot be accomplished with a single Task-Plugin. Please note that workflows for complex tasks may vary from person to person, here are just a few examples.

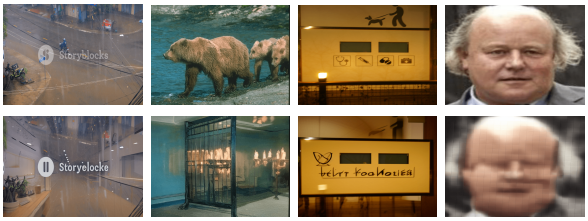


Figure 2. Visualization of the results of first DDIM inversion and then reconstruction. The first and second rows show the input image and the reconstructed image, respectively.

Plugins. For instance, the workflow for old photo restoration may involves a sequence of steps: initially restoring damaged regions like scratches and noises, followed by colorizing the photo, and finally up-scaling the resolution. In adverse weather scenarios, we may first remove dense haze and then remove rain and snow, respectively.

*"**TPB + SCB (Reconstruction)**"* in Line-443 of our main manuscript. We train the Spatial Complement Branch (SCB) using a self-reconstruction denoising loss, where the Ground Truth (GT) is set to the input image during training. Subsequently, we fix the parameters of the reconstruction-based SCB and train only the Task Prior Branch (TPB). This variant tries to preserve the details in the input image through SCB first, and then handle the task-specific degradations via the visual guidance priors provided by TPB.

How *Diff-Plugin* revitalizes details? Providing an initial noise that contains the content of the original image is the most straightforward approach to maintain image details. Specifically, we first convert input images into their corresponding initial noise and then regenerate the images based on this noise. We show some examples in Fig. 2. It is evident that the inversion process, especially in complex images or regions, is unstable and often fails. Therefore, we seek a more stable way to ensure details, which is mainly achieved by SCB in our method.

In *Diff-Plugin*, we adopt the output of Task-Prompt Branch (TPB) as the query of the cross-attention layers in the diffusion model, learning task-specific visual guidance priors to handle unwanted degradations such as rain streaks and snowflakes. However, TPB alone cannot maintain intricate details at all. Even with the initial noise from DDIM Inversion, TPB effectively helps to capture large structured scenes but falls short in preserving fine-grained details. To this end, we further introduce (*i.e.*, directly add) the output of Spatial Complement Branch (SCB) to the last stage of the diffusion model's decoder. SCB plays a crucial role in detail preservation yet it struggles to remove unwanted degradations. Finally, combining both SCB and TPB can provide sufficient task-specific priors (*i.e.*, both visual guidance and spatial information) and thus ensures the fine-grained details preservation in the final output.

"Multi-task setting" in Line-476 of our main manuscript. In our experiments, we generate multi-task prompts for the eight low-level tasks by randomly pairing texts from two different tasks. For example: "*I want to remove rain and clear haze.*" and "*clear blur and enhance the lighting for this image*". The use of the '*' symbol denotes that we enumerate the generation of all possible multi-task combinations for each task's text. By default, we test two Task-plugins at the same time during evaluation, because scenarios requiring more than two Task-plugins are uncommon in practical applications. However, note that the Plugin-Selector's functionality is not confined to just two tasks. Owing to our contrastive training paradigm and the strategy of randomly combining multi-task prompts for data augmentation, the Plugin-Selector can still handle combinations that involve more than two tasks.

"Single + Non." in Line-481 of our main manuscript. We generate various text prompts for hypothetical, non-existent Task-Plugins and randomly combine them with prompts for existing Task-Plugins. Here are some examples of these non-existent Task-Plugin prompts: "*I want to conduct **image classification***", "*Can you help to **add some haze** in this photo?*", and "***Introduce some moire pattern** for this image*".

C. Experiments

C.1. Datasets

Task-Plugin: The datasets for training and testing the Task-Plugin are listed in Table 2. For tasks like desnowing, dehazing, and deraining, we specifically excluded samples with very low image quality (*e.g.*, those with extremely low resolution), small degradations that are barely visible, or instances where degradation dominates the entire image, leaving no other discernible content.

Plugin-Selector: To train the Plugin-Selector, we employ Chat-GPT to generate natural language sentences that mimic user input. Our approach is structured as follows:

1. **Sentence Structuring**: We deconstruct sentences into five distinct components: <begin>, <verb>, <article>, <noun>, and <end>. For example, 'I want to remove the rain in this image' is broken down into <be-

Data \ Tasks		Desnowing	Dehazing	Deblurring	Deraining	Low-light Enhancement	Face Restoration	Demoireing	Highlight Removal
Train	Datasets	Snow100K [11]	Reside [10]	Gopro [15]	Merged [36]	LOL [29]	FFHQ [8]	LCDMoire [35]	SHIQ [3]
	Samples	50,000	72,135	2,103	13,712	485	70,000	9,825	10,000
Test	Datasets	Realistic [11]	RTTS [10]	RealBlur-J [21]	Real [26]	Merged	LFW [6, 27]	LCDMoire [35]	SHIQ [3]
	Samples	884	1375	980	122	350	1711	100	1000

Table 2. Dataset description for different low-level vision tasks. The merged refers to test data in these methods [5, 9, 13, 24, 25, 29].

gin>: 'I want to', <verb>: 'remove', <article>: 'the', <noun>: 'rain', and <end>: 'in this image'.

2. **Task-Specific Keywords**: Each task is identified by unique <nouns>, with some overlap in <verbs>across different tasks. For example, 'remove' could apply to both 'rain' and 'snow'. The segments <begin>, <article>, and <end>, are common across all tasks.
3. **Prompt Generation**: We first employ Chat-GPT to create task-specific sentence templates. For those incorrect examples, we eliminate them through manual review. Then, based on these templates, we employ Chat-GPT again, to produce appropriate text prompts for each task (*i.e.*, assemble different components).
4. **Dataset Creation**: For each task, we randomly select 750 prompts for training and 50 prompts for testing.

During training, these task-specific text prompts are combined with image samples corresponding to the task type to construct data pairs.

C.2. Implementations

Task-Plugin. During the training phase of Task-Plugins, we set the training epochs based on the number of available samples for each task. Specifically, for tasks with fewer samples, *e.g.*, deblurring and low-light enhancement, we extend the training to 200 epochs. In contrast, tasks with a sufficient number of samples (*e.g.*, face restoration) are trained for 30 epochs. We employ the standard DDPM sampler with a time step of 1000 for training. For testing, we adopt the UniPC [40] sampler with 20 inference steps for faster inference.

Plugin-Selector. In the training phase of the Plugin-Selector, we randomly selected 5,000 sample pairs for each low-level vision task, conducting training over 200,000 iterations. The inference workflow of the Plugin-Selector is illustrated in Fig. 1. Given an input image and a corresponding text prompt, the Plugin-Selector determines the most appropriate Task-Plugins based on the computed visual-textual similarity score.

Application. Fig. 5 showcases a screenshot of our application interface. Our *Diff-Plugin* framework facilitates prompt-driven processing of low-level tasks and provides users the option to manually select their desired Task-Plugin. Please refer to our **demo video** for more details.

C.3. Metrics

In evaluating our *Diff-Plugin*, we primarily focus on perceptual metrics like FID and KID, which are commonly used in the generative model domain [20, 22, 39]. These metrics are particularly suitable for assessing the visual quality and realism of generated images, especially in scenarios where ground truth (GT) is not available, such as with natural images. In addition, natural images often lack corresponding GT, rendering pixel-based comparison metrics like SSIM and PSNR impractical. To evaluate the FID and KID metrics, we use GT images from the training dataset as reference images for each low-level task. For example, we use 50,000 GT images from Snow100K [11] as the reference images to evaluate the predicted results on Realistic [11] according to FID and KID metrics. Following Parmar *et al.*[17], we opt for the CLIP vision encoder (VIT-B/32) instead of the Inception-V3 model.

Furthermore, when evaluating the accuracy of plugin selectors, in addition to using standard multi-label object classification metrics, we also introduce a strict zero-tolerance accuracy (ZTA) metric. This metric provides a sentence-level classification evaluation from a user-first perspective. It adopts a binary classification approach to ensure the utmost accuracy in determining the relevance of each sentence to the specified task. For instance, consider we have eight low-level vision tasks with assigned task IDs ranging from 0 to 7, corresponding to tasks (0) desnowing, (1) dehazing, (2) deblurring, (3) deraining, (4) low-light enhancement, (5) face restoration, (6) demoireing, (7) highlight removal. When a user inputs a prompt like "*Can you help to remove rain and enhance the brightness for this photo*?", the Plugin-Selector generates similarity scores for each task, such as [-8.4, -15, -7.8, 19.1, 15.4, -3, -21.9, -14.5]. Ideally, the Plugin-Selector should identify 'deraining' and 'low-light enhancement' as relevant tasks, which correspond to task IDs 3 and 4, respectively. The positive and negative classes thus should be [3, 4] and [0, 1, 2, 5, 6, 7], respectively. In the ZTA metric, binary classification is deemed correct only if the similarity scores for all positive classes exceed a threshold (*i.e.*, both 19.1 and 15.4 are greater than $\theta = 0$), and the scores for all negative classes fall below the threshold. If these conditions are not met, the classification is considered incorrect. This metric, thus, em-

CVPR
#170

Figure 3. Visual results of our *Diff-Plugin* on multi-task processing. Here, we simply combine the ***restoration*** and ***colorization*** tasks to mimic the old photo restoration task.

	Desnowing Realistic [11]		Dehazing Reside [10]		Deblurring RealBlur-J [21]		Deraining real test [26]		Low-light Enhanc. merged low.		Face Restoration LFW [27]		Demoireing LCDMoire [35]		Highlight Removal SHIQ [3]	
	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓	FID ↓	KID ↓
Regression-based *specialized* models																
DDMSNET [38]	33.92	5.39	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMNet [32]	-	-	36.40	15.66	-	-	-	-	-	-	-	-	-	-	-	-
FSDGN [33]	-	-	34.58	14.10	-	-	-	-	-	-	-	-	-	-	-	-
MPRNet [36]	-	-	-	-	55.45	15.71	51.17	15.21	-	-	-	-	-	-	-	-
Restormer [37]	-	-	-	-	55.64	15.70	52.78	16.28	-	-	-	-	-	-	-	-
NeRCo [31]	-	-	-	-	-	-	-	-	48.47	10.96	-	-	-	-	-	-
CodeFormer [41]	-	-	-	-	-	-	-	-	-	-	19.94	7.09	-	-	-	-
VQFR [4]	-	-	-	-	-	-	-	-	-	-	19.28	6.72	-	-	-	-
UHDM [34]	-	-	-	-	-	-	-	-	-	-	-	-	29.59	1.45	-	-
SHIQ [3]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	33.74	18.79
Diffusion-based specialized models																
WeatherDiffusion [16]	36.39	5.53	-	-	-	-	53.80	16.59	-	-	-	-	-	-	-	-
IR-SDE [12]	-	-	-	-	47.43	12.42	52.89	16.51	-	-	-	-	-	-	-	-
DiffIR [30]	-	-	-	-	55.78	15.88	-	-	-	-	-	-	-	-	-	-
DiffLL [7]	-	-	-	-	-	-	-	-	51.04	11.74	-	-	-	-	-	-
DR2 [28]	-	-	-	-	-	-	-	-	-	-	23.43/(17.46)	9.29/(5.76)	-	-	-	-
Diff-Plugin (ours)	34.30	5.20	34.68	14.38	51.81	14.63	50.55	13.84	48.98	11.73	20.07/(18.46)	6.91/(6.41)	29.77	1.75	12.58	6.37

Table 3. Quantitative comparisons to SOTAs (both regression-based and diffusion-based methods) on eight low-level vision tasks that need high content-preservation. KID values are scaled by a factor of 100 for readability. In face restoration, to ensure fairness, we apply the same post-processing [4] as DR2 [28] to our *Diff-Plugin* results. Metrics recalculated after this step are indicated in parentheses.

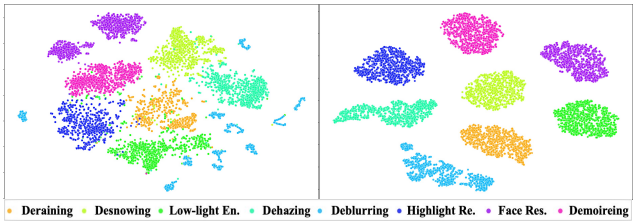


Figure 4. The t-SNE distribution of visual embeddings on the test set of different tasks before (*i.e.*, left sub-figure) and after (*i.e.*, right sub-figure) using the Task-Prompt Branch.

bodies a zero-tolerance policy, strictly assessing the Plugin-Selector's accuracy in identifying the correct task(s) based on user prompts.

C.4. More Internal Analysis

Task-Plugin. Fig. 4 displays a t-SNE plot illustrating the feature distribution map. In this analysis, we sampled 1,000 test images from each low-level vision task[1] and first fed them into $Enc_I(\cdot)$. The output from this encoder was then fed to various Task-Plugins, each responsible for extracting task-specific visual guidance priors based on the task type of the input images. It is obvious to see that the the distribution discrepancy among different low-level tasks is accentuated, indicating that the task-specific guidance priors are distinct and that the TPB has successfully learned these differences. This distinction is also advantageous for the Plugin-Selector, as it enhances the accuracy of visual-textual similarity scoring.

[1]Totalling 8,000 samples. For tasks with fewer than 1,000 test images, we repeated the available samples to reach 1,000.

C.5. Visual Results of Multi-task Processing

In Fig. 3, we showcase additional visual results of multi-task processing using our *Diff-Plugin*. Specifically, we show how the old photo restoration task, particularly in face scenarios, can be simply decomposed into face restoration and colorization sub-tasks. The results illustrate our method's capability to vividly rejuvenate old face photos.

C.6. Comparisons to SOTAs

More quantitative comparisons. We also compare our *Diff-Plugin* to several diffusion-based restoration methods, including WeatherDiffusion [16], IR-SDE [12], DiffIR [30], DiffLL [7] and DR2 [28]. The results are tabulated in Table 3 and we can see that our *Diff-Plugin* can still achieves competitive results compared to these specialized models, verifying its superiority.

More qualitative comparisons. We provide more visual comparisons of different low-level vision tasks in Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11, Fig. 12, Fig. 13.

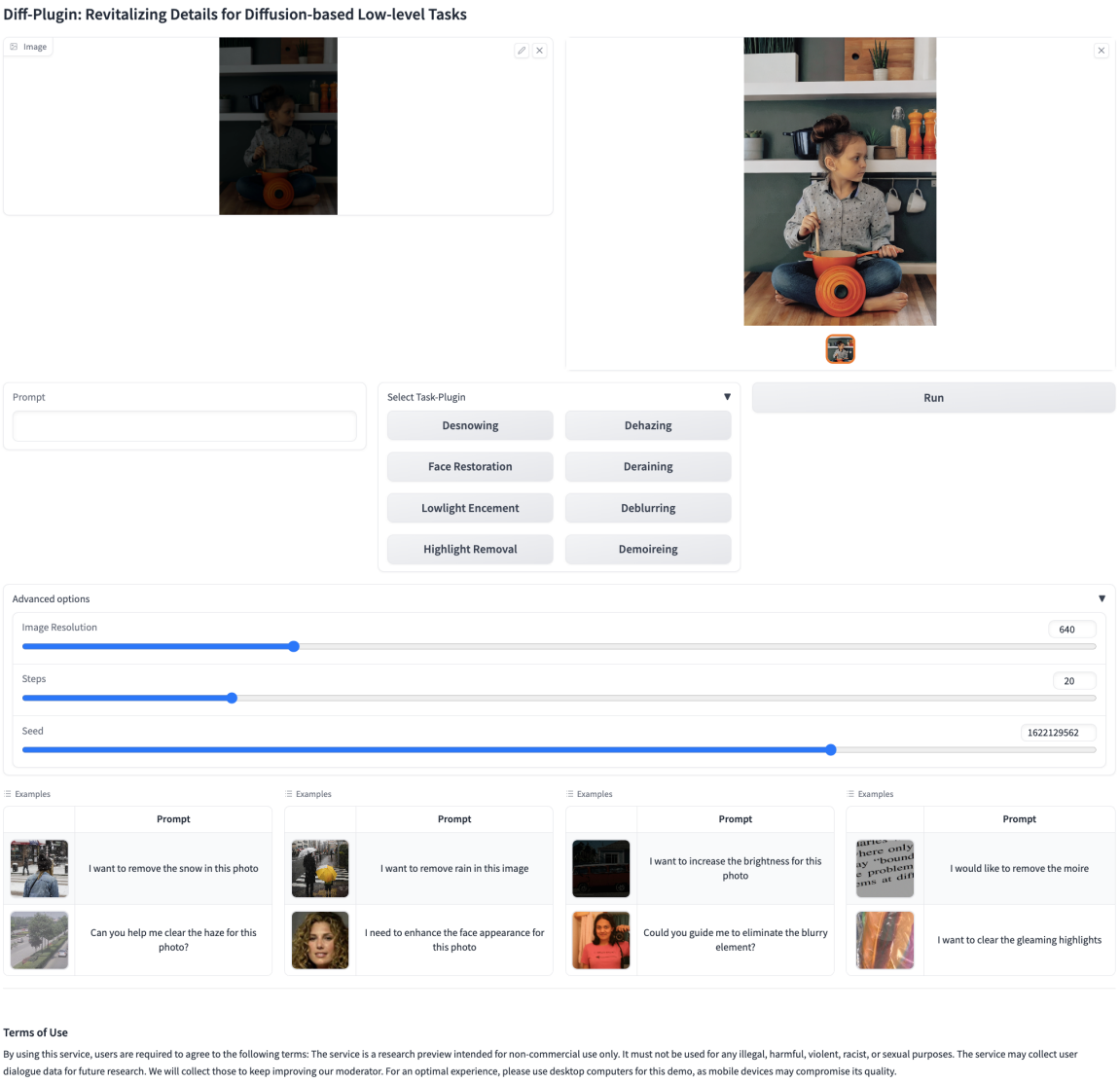


Figure 5. Demo interface diagram of our *Diff-Plugin* framework.

References

[1] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, pages 8780–8794, 2021. 1

[2] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 1

[3] Gang Fu, Qing Zhang, Lei Zhu, Ping Li, and Chunxia Xiao. A multi-task network for joint specular highlight detection and removal. In *CVPR*, pages 7752–7761, 2021. 3, 5

[4] Yuchao Gu, Xintao Wang, Liangbin Xie, Chao Dong, Gen Li, Ying Shan, and Ming-Ming Cheng. Vqfr: Blind face restoration with vector-quantized dictionary and parallel decoder. In *ECCV*, pages 126–143, 2022. 5

[5] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE TIP*, 26(2):982–993, 2016. 3

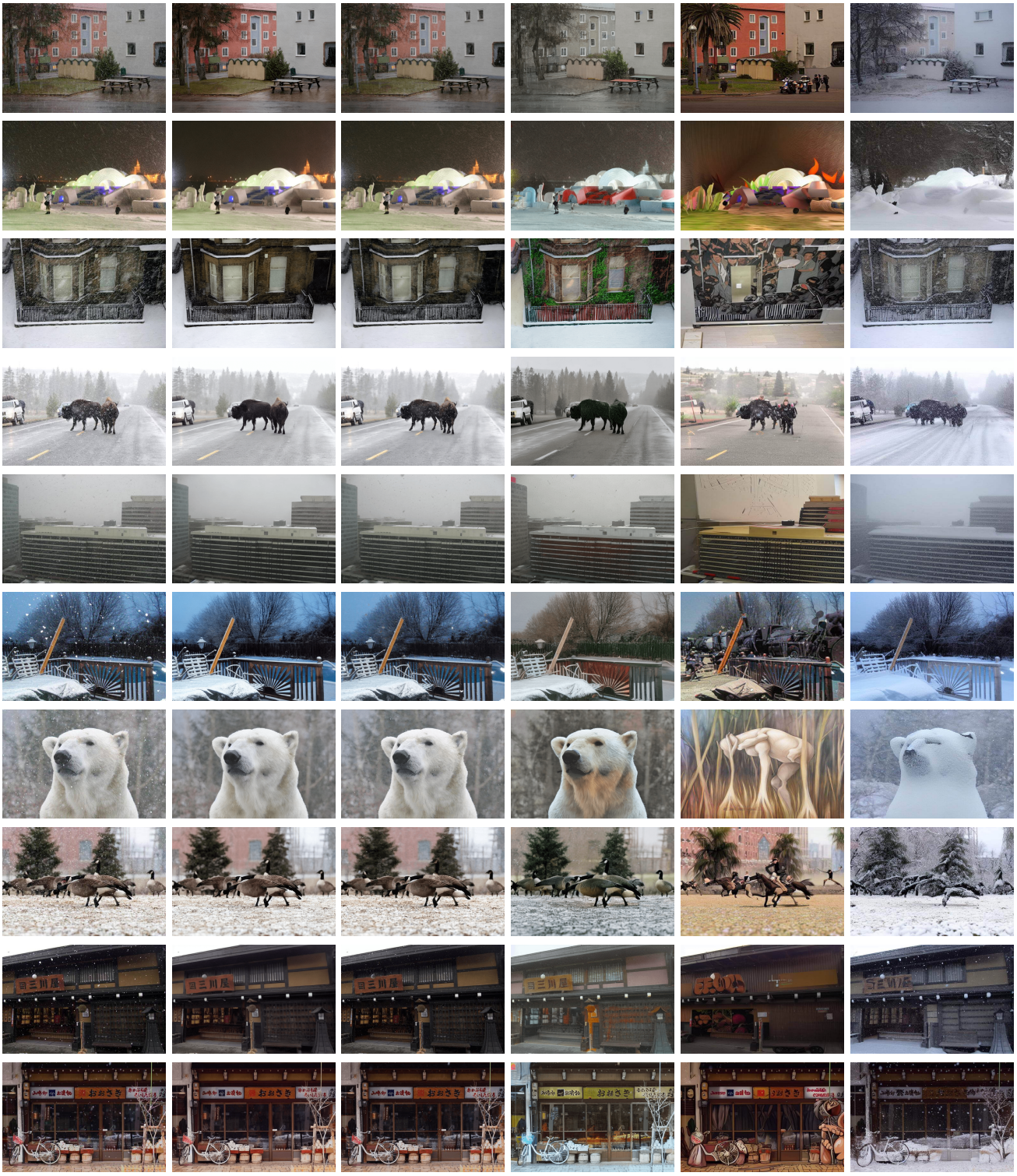
[6] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Technical report, University of Massachusetts, Amherst*, 2007. 3

[7] Hai Jiang, Ao Luo, Songchen Han, Haoqiang Fan, and Shuaicheng Liu. Low-light image enhancement with wavelet-based diffusion models. *arXiv*, 2023. 5

[8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 3

[9] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE TIP*, 22(12):5372–5384, 2013. 3

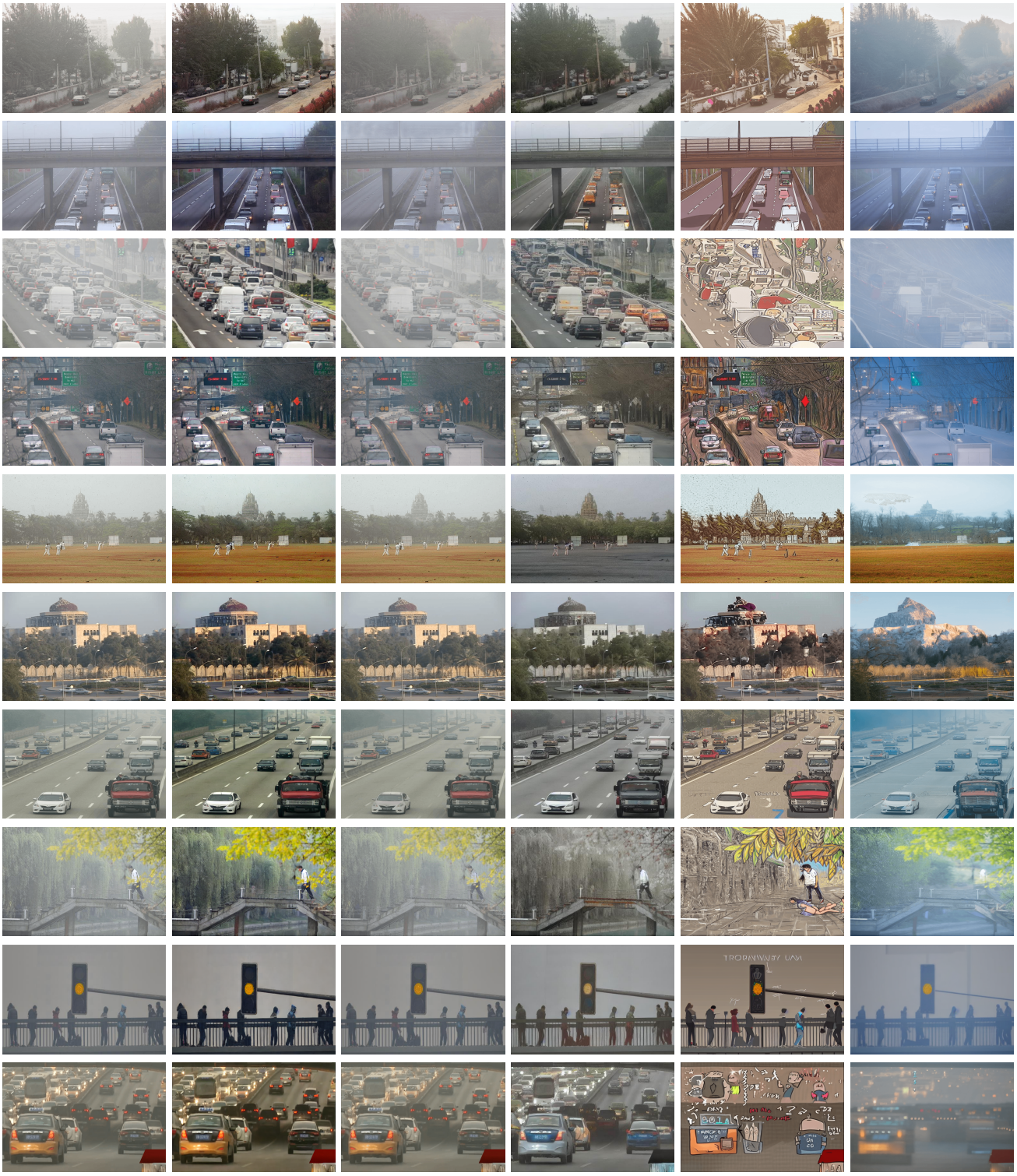
CVPR
#170



(1) Input (2) Ours (3) PromptIR [18] (4) ControlNet [39] (5) Null-Text [14] (6) PNP [23]

Figure 6. Visual comparisons of our *Diff-Plugin* with four representative approaches (one regression-based multi-task model in (3) and three diffusion-based models in (4)-(6)) on the ***desnowing*** task.

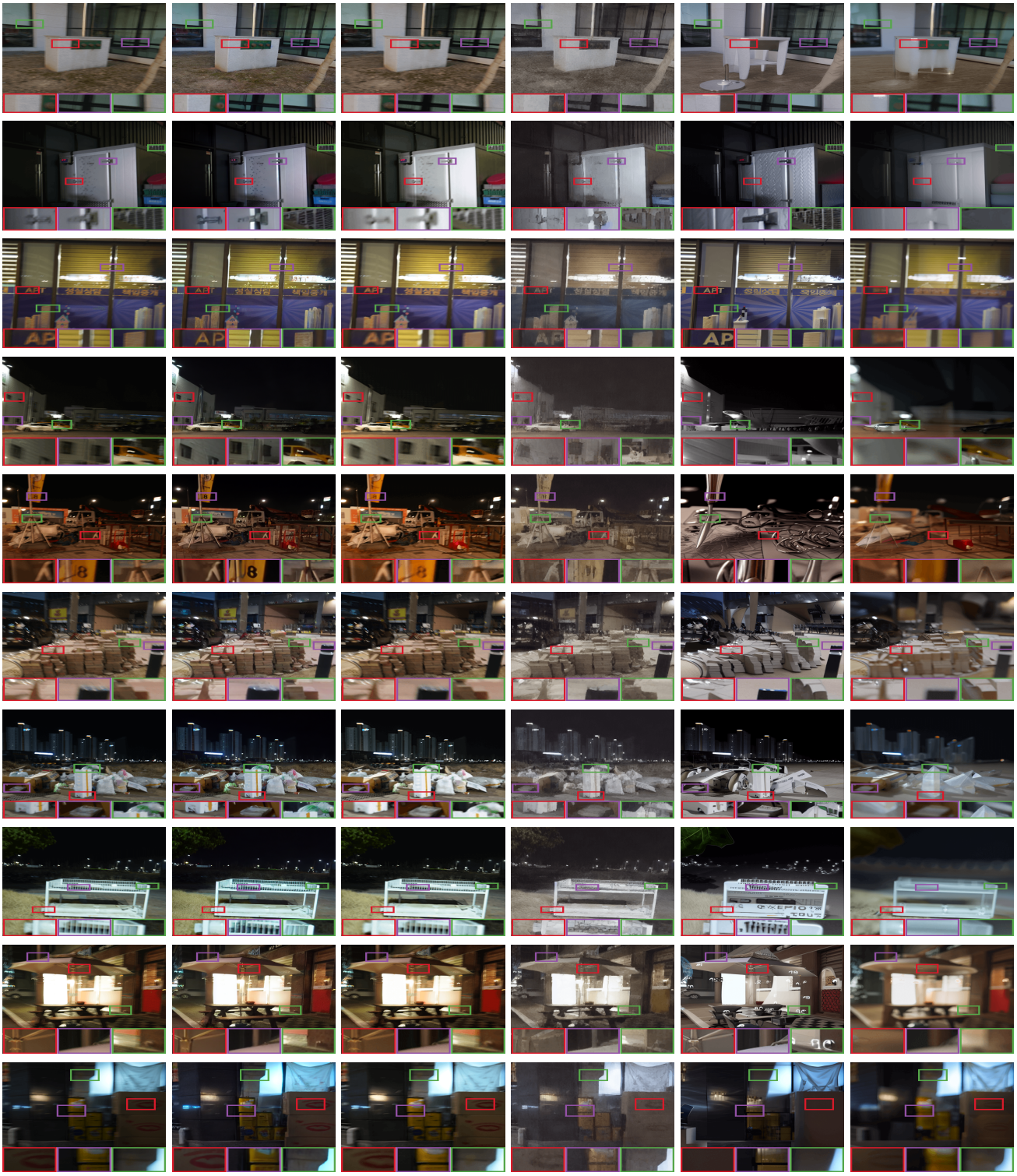
CVPR
#170



(1) Input (2) Ours (3) PromptIR [18] (4) ControlNet [39] (5) Null-Text [14] (6) PNP [23]

Figure 7. Visual comparisons of our *Diff-Plugin* with four representative approaches (one regression-based multi-task model in (3) and three diffusion-based models in (4)-(6)) on the ***dehazing*** task.

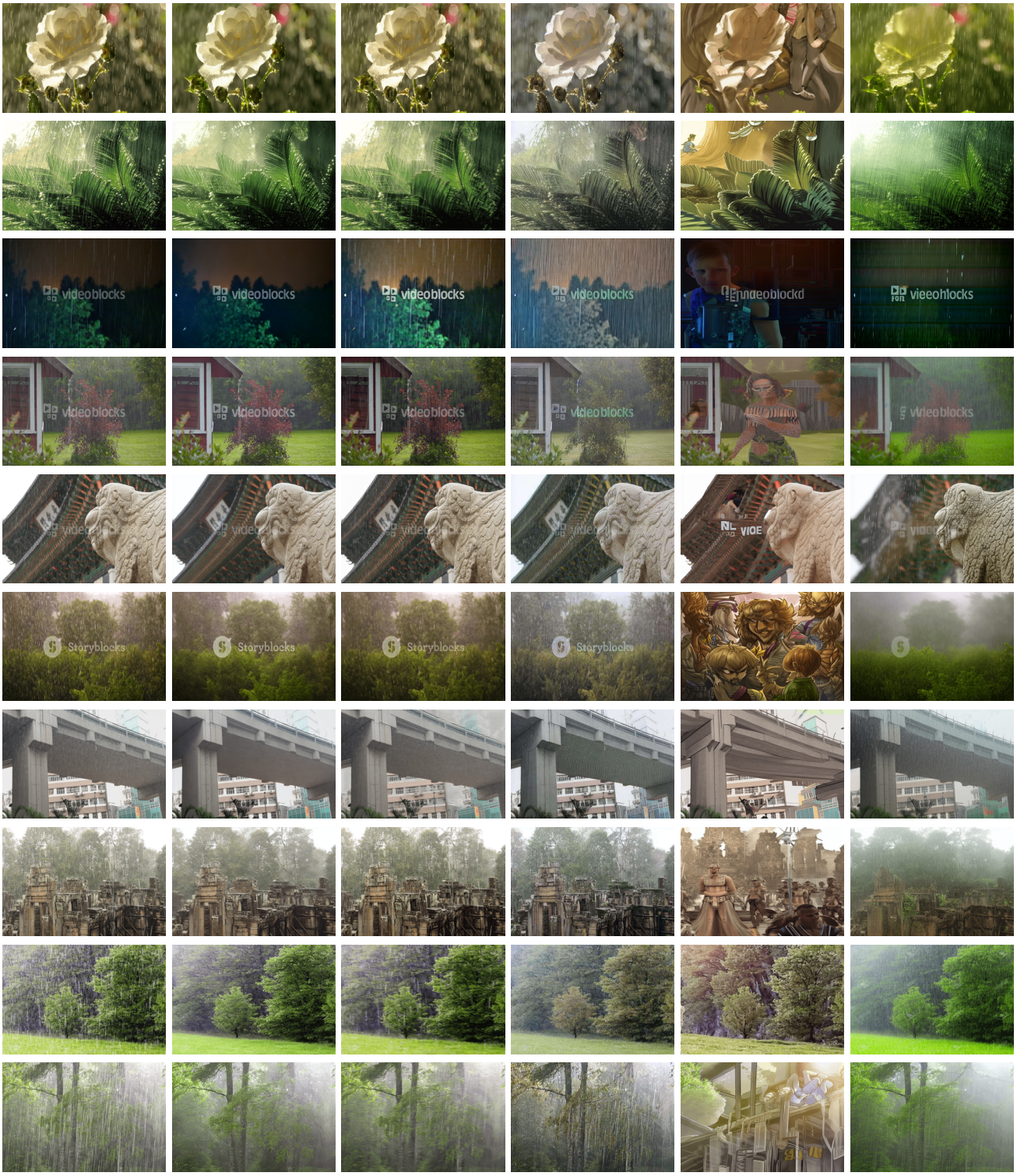
CVPR
#170



(1) Input (2) Ours (3) PromptIR [18] (4) ControlNet [39] (5) Null-Text [14] (6) PNP [23]

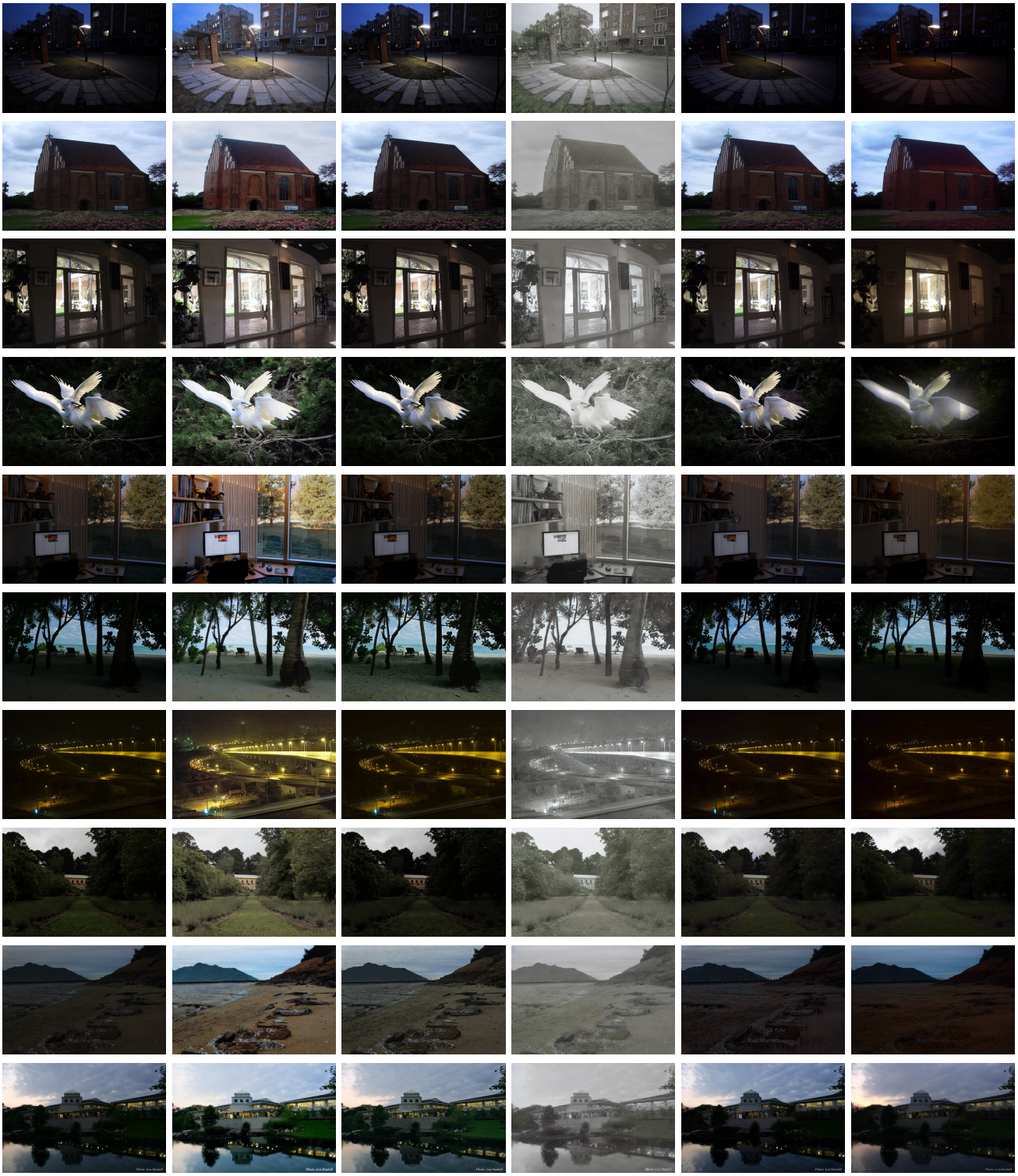
Figure 8. Visual comparisons of our *Diff-Plugin* with four representative approaches (one regression-based multi-task model in (3) and three diffusion-based models in (4)-(6)) on the ***deblurring*** task. Magnified regions are provided for clarity

CVPR
#170



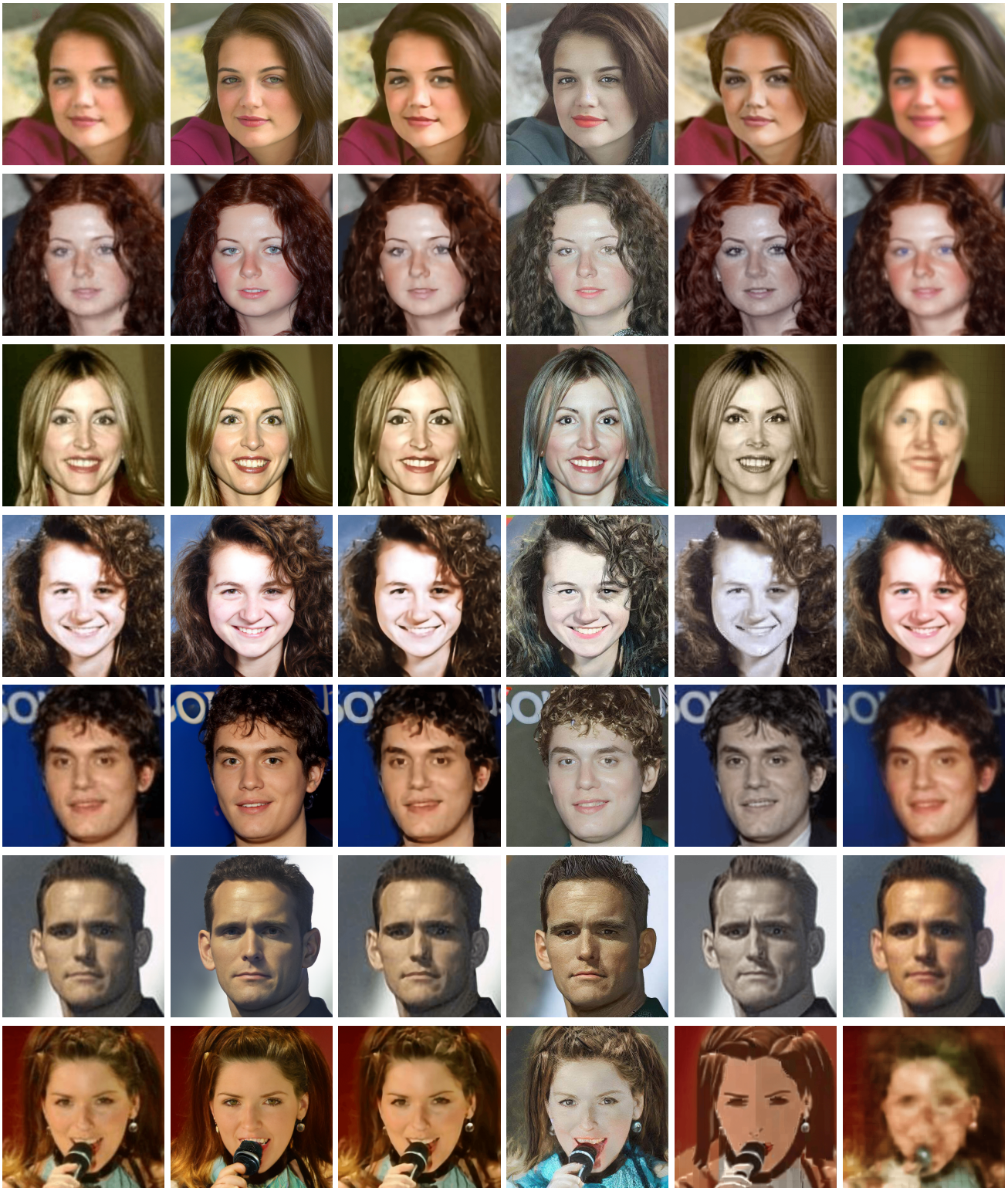
(1) Input (2) Ours (3) PromptIR [18] (4) ControlNet [39] (5) Null-Text [14] (6) PNP [23]

Figure 9. Visual comparisons of our *Diff-Plugin* with four representative approaches (one regression-based multi-task model in (3) and three diffusion-based models in (4)-(6)) on the ***deraining*** task.



(1) Input (2) Ours (3) PromptIR [18] (4) ControlNet [39] (5) Null-Text [14] (6) PNP [23]

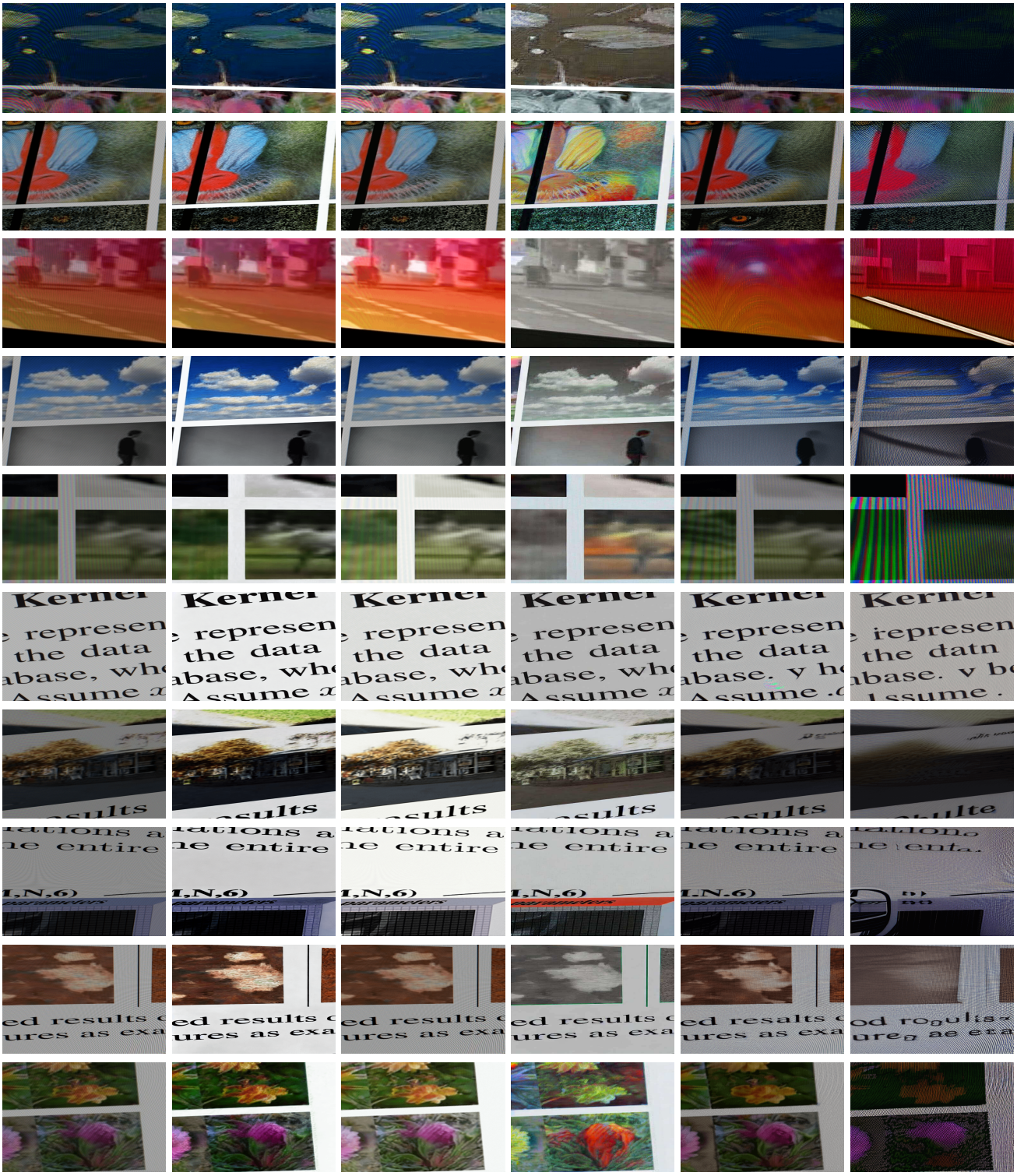
Figure 10. Visual comparisons of our *Diff-Plugin* with four representative approaches (one regression-based multi-task model in (3) and three diffusion-based models in (4)-(6)) on the ***low-light enhancement*** task.



(1) Input (2) Ours (3) PromptIR [18] (4) ControlNet [39] (5) Null-Text [14] (6) PNP [23]

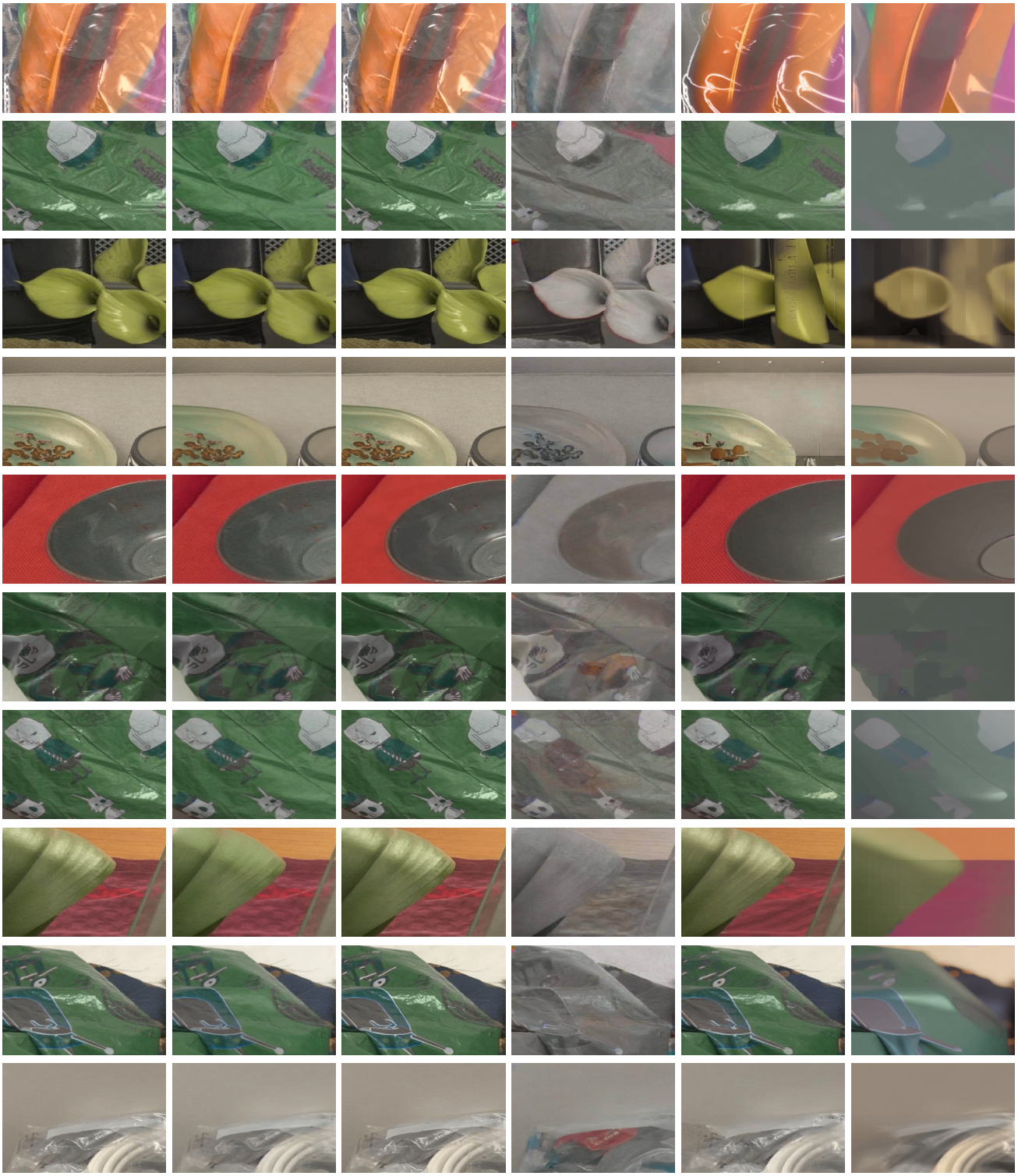
Figure 11. Visual comparisons of our *Diff-Plugin* with four representative approaches (one regression-based multi-task model in (3) and three diffusion-based models in (4)-(6)) on the ***face restoration*** task.

CVPR
#170



(1) Input (2) Ours (3) PromptIR [18] (4) ControlNet [39] (5) Null-Text [14] (6) PNP [23]

Figure 12. Visual comparisons of our *Diff-Plugin* with four representative approaches (one regression-based multi-task model in (3) and three diffusion-based models in (4)-(6)) on the ***demoireing*** task.



(1) Input (2) Ours (3) PromptIR [18] (4) ControlNet [39] (5) Null-Text [14] (6) PNP [23]

Figure 13. Visual comparisons of our *Diff-Plugin* with four representative approaches (one regression-based multi-task model in (3) and three diffusion-based models in (4)-(6)) on the ***highlight removal*** task.

[10] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE TIP*, 28(1):492–505, 2018. 3, 5

[11] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE TIP*, 27(6):3064–3073, 2018. 3, 5

[12] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Image restoration with mean-reverting stochastic differential equations. *ICML*, 2023. 5

[13] Kede Ma, Kai Zeng, and Zhou Wang. Perceptual quality assessment for multi-exposure image fusion. *IEEE TIP*, 24 (11):3345–3356, 2015. 3

[14] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, pages 6038–6047, 2023. 7, 8, 9, 10, 11, 12, 13, 14

[15] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891, 2017. 3

[16] Ozan Özdenizci and Robert Legenstein. Restoring vision in adverse weather conditions with patch-based denoising diffusion models. *IEEE TPAMI*, 2023. 5

[17] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, pages 11410–11420, 2022. 3

[18] Vaishnav Potlapalli, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one blind image restoration. In *NeurIPS*, 2023. 7, 8, 9, 10, 11, 12, 13, 14

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 1

[20] Mengwei Ren, Mauricio Delbracio, Hossein Talebi, Guido Gerig, and Peyman Milanfar. Multiscale structure guided diffusion for image deblurring. In *ICCV*, pages 10721–10733, 2023. 3

[21] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, pages 184–201, 2020. 3, 5

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 3

[23] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, pages 1921–1930, 2023. 7, 8, 9, 10, 11, 12, 13, 14

[24] Vassilios Vonikakis, Rigas Kouskouridas, and Antonios Gasteratos. On the evaluation of illumination compensation algorithms. *MTA*, 77:9211–9231, 2018. 3

[25] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE TIP*, 22(9):3538–3548, 2013. 3

[26] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, pages 12270–12279, 2019. 3, 5

[27] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *CVPR*, pages 9168–9178, 2021. 3, 5

[28] Zhixin Wang, Ziying Zhang, Xiaoyun Zhang, Huangjie Zheng, Mingyuan Zhou, Ya Zhang, and Yanfeng Wang. Dr2: Diffusion-based robust degradation remover for blind face restoration. In *CVPR*, pages 1704–1713, 2023. 5

[29] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *BMVC*, 2018. 3

[30] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. Diffir: Efficient diffusion model for image restoration. In *ICCV*, pages 13095–13105, 2023. 5

[31] Shuzhou Yang, Moxuan Ding, Yanmin Wu, Zihan Li, and Jian Zhang. Implicit neural representation for cooperative low-light image enhancement. In *ICCV*, pages 12918–12927, 2023. 5

[32] Tian Ye, Yunchen Zhang, Mingchao Jiang, Liang Chen, Yun Liu, Sixiang Chen, and Erkang Chen. Perceiving and modeling density for image dehazing. In *ECCV*, pages 130–145, 2022. 5

[33] Hu Yu, Naishan Zheng, Man Zhou, Jie Huang, Zeyu Xiao, and Feng Zhao. Frequency and spatial dual guidance for image dehazing. In *ECCV*, pages 181–198, 2022. 5

[34] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Jiajun Shen, Jia Li, and Xiaojuan Qi. Towards efficient and scale-robust ultra-high-definition image demoiréing. In *ECCV*, pages 646–662, 2022. 5

[35] Shanxin Yuan, Radu Timofte, Gregory Slabaugh, Aleš Leonardis, Bolun Zheng, Xin Ye, Xiang Tian, Yaowu Chen, Xi Cheng, Zhenyong Fu, et al. Aim 2019 challenge on image demoireing: Methods and results. In *ICCVW*, pages 3534–3545, 2019. 3, 5

[36] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, pages 14821–14831, 2021. 3, 5

[37] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pages 5728–5739, 2022. 5

[38] Kaihao Zhang, Rongqing Li, Yanjiang Yu, Wenhan Luo, and Changsheng Li. Deep dense multi-scale network for snow removal using semantic and depth priors. *IEEE TIP*, 30: 7419–7431, 2021. 5

[39] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 1, 3, 7, 8, 9, 10, 11, 12, 13, 14

[40] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *NeurIPS*, 2023. 3

[41] Shangchen Zhou, Kelvin Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer. In *NeurIPS*, pages 30599–30611, 2022. 5