PromptFix: You Prompt and We Fix the Photo

Yongsheng Yu[1] * **Ziyun Zeng**[1] * **Hang Hua**[1] **Jianlong Fu** [2] **Jiebo Luo**[1]
[1]University of Rochester, [2]Microsoft Research
{yyu90,zzeng24}@ur.rochester.edu, {hhua2,jluo}@cs.rochester.edu, jianf@microsoft.com

Abstract

Diffusion models equipped with language models demonstrate excellent controllability in image generation tasks, allowing image processing to adhere to human instructions. However, the lack of diverse instruction-following data hampers the development of models that effectively recognize and execute user-customized instructions, particularly in low-level tasks. Moreover, the stochastic nature of the diffusion process leads to deficiencies in image generation or editing tasks that require the detailed preservation of the generated images. To address these limitations, we propose PromptFix, a comprehensive framework that enables diffusion models to follow human instructions to perform a wide variety of image-processing tasks. First, we construct a large-scale instruction-following dataset that covers comprehensive image-processing tasks, including low-level tasks, image editing, and object creation. Next, we propose a high-frequency guidance sampling method to explicitly control the denoising process and preserve high-frequency details in unprocessed areas. Finally, we design an auxiliary prompting adapter, utilizing Vision-Language Models (VLMs) to enhance text prompts and improve the model's task generalization. Experimental results show that PromptFix outperforms previous methods in various image-processing tasks. Our proposed model also achieves comparable inference efficiency with these baseline models and exhibits superior zero-shot capabilities in blind restoration and combination tasks. The dataset and code are available at `https://www.yongshengyu.com/PromptFix-Page`.

1 Introduction

In recent years, diffusion models [19, 57, 66] have achieved remarkable advancements in text-to-image generation. Benefiting from large-scale training on image-text pairs [59], these models can generate highly realistic and diverse images that align with text prompts. They have been successfully applied to various real-world applications, including visual design, photography, digital art, and the film industry. In addition, models trained with instruction-following data [7] have shown promising results in understanding human instruction and performing the corresponding image-processing tasks. Previous studies [21–23, 74, 73] have illustrated that with instruction-following data, we can simply fine-tune a text-to-image generation model to perform various vision tasks such as image editing [7, 21, 74], object detection [22], segmentation [23], inpainting [23, 73], and depth estimation [22, 9]. To follow the success of these methods, we train our model utilizing input-goal-instruction triplet data for low-level image-processing tasks.

We first overcome the challenge of lacking the instruction-following data for low-level tasks. Specifically, we collect image pairs by generating degraded images from the source images and adopting data from existing datasets. Then, we employ GPT4 [51] to generate the diverse text instructions for each task. We obtain $\sim$ 1.01 million input-goal-instruction triplets in the collected dataset. This dataset covers various low-level tasks including image inpainting, object creation, image dehazing, colorization, super-resolution, low light enhancement [8], snow removal, and watermark removal. We enrich the dataset through back-translation augmentation by swapping the inpainted and origi-

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-1263

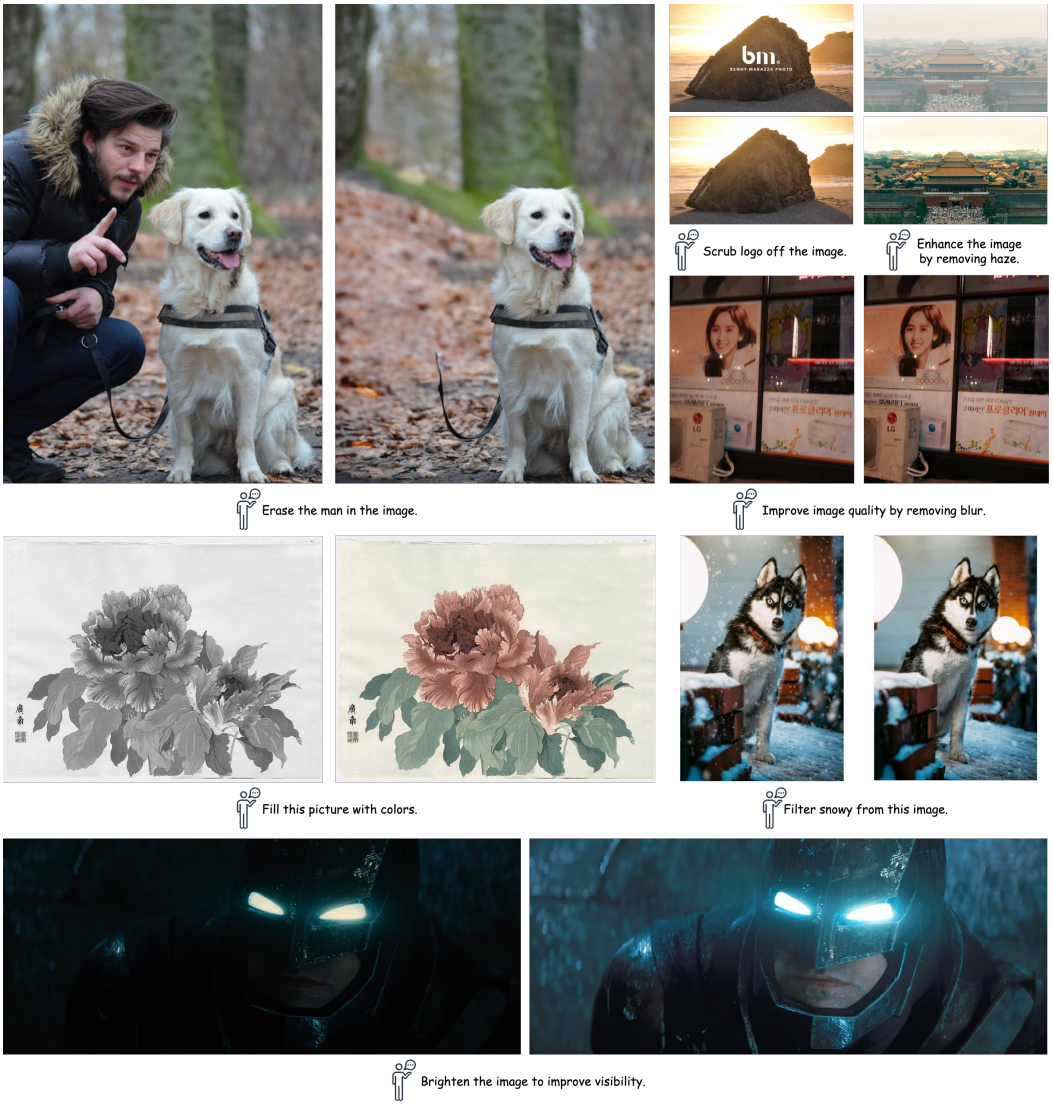

Figure 1: We propose PromptFix, a unified diffusion model capable of performing multiple image-processing tasks. It can understand user-customized editing instructions and perform the corresponding tasks with high quality. One of the key advantages of PromptFix is high-frequency information preservation, ensuring that image details are maintained throughout VAE decoding. PromptFix can handle various images with different aspect ratios.

nal images within the triplet and inverting the semantic orientation of the prompt. This technique effortlessly converts datasets from object removal to object creation. We also provide comprehensive details in Section 3.

With the dataset, we design a new diffusion-based model named PromptFix that can understand user-customized instructions and perform the corresponding low-level image-processing tasks. In PromptFix, we address several challenges that compromise the performance of the model. First, the use of stable-diffusion architecture [57] as a generative prior often faces the issue of spatial information loss, which is caused by VAE compression [20]. Unlike unconditional or text-to-image generation, maintaining spatial detail consistency in image processing poses significant challenges, particularly with high-frequency components like text, as shown in Figure 5. To tackle this problem, we introduce High-frequency Guidance Sampling, in which we use a low-pass filter operator [50, 53] to calculate the *fidelity constraint* and integrate VAE skip-connect features during inference with a lightweight LoRA [27] fusion. Second, since the generative prior is not trained on low-level images, so relying solely on instructions may not always yield the desired outcome, especially when the image degradation is severe. To tackle this degradation adaptation problem, we introduce an *auxiliary prompt*

module to provide models with more descriptive text prompts to enhance controllability for image generation. The auxiliary text prompt can be obtained by VLMs [43]. This approach introduces a semantic caption for a degraded image and the description of its defects, such as blurriness or insufficient lighting. The *auxiliary prompt module* is implemented by an additional attention layer in diffusion U-Net that adapts both instruction and *auxiliary prompts* as conditions and intermittently omits instructional prompts during training. We identify three key advantages of this approach: 1) enabling the model to process images with severe degradation, such as extremely low-resolution images, 2) adapting the model for blind restoration for different types of image degradation, and 3) providing additional pathways for a more precise semantic representation of the target image.

Experimental results demonstrate that our model achieves superior performance in the instruction-based paradigm across three image editing tasks (colorization, watermark removal, object removal) and four image restoration tasks (dehazing, desnowing, super-resolution, and low-light enhancement) in terms of perceptual pixel similarity [75] and no-reference image quality [72]. In summary, our contributions are three-fold:

- We propose a comprehensive dataset tailored for seven image processing tasks. The dataset contains $\sim$ 1.01 million diverse paired input-output images along with corresponding image editing instructions.
- We propose a new all-in-one instruction-guided diffusion model – PromptFix for low-level image-processing tasks. Extensive experimental results show that PromptFix outperforms previous methods in a wide variety of image-processing tasks and exhibits superior zero-shot capabilities in blind restoration and combination tasks.
- We introduce two approaches – high-frequency guidance sampling and auxiliary prompt module to diffusion models to effectively address the issues of high-frequency information loss and the failure in processing the severe image degradation for instruction-based diffusion models in low-level tasks.

2 Related Work

2.1 Instruction-guided Image Editing

Instruction-guided image editing significantly improves the ease and precision of visual manipulations by adhering to human directions. In traditional image editing, models primarily focused on singular tasks such as style transfer or domain adaptation [24, 55], leveraging various techniques to encode images into a manipulatable latent space, such as those used by StyleGAN [34]. Concurrently, the advent of text-to-image diffusion models [26, 57, 63, 65] has broadened the scope of image editing [7, 25]. Kim et al. [35] showed how to perform global changes, whereas Avrahami et al. [4] successfully performed local manipulations using user-provided masks for guidance. While most works that require only text (i.e., no masks) are limited to global editing [17, 36]. Bar-Tal et al. [6] proposed a text-based localized editing technique without using any mask, showing impressive results. For local image editing, precise manipulations are possible by inpainting designated areas using either user-provided or algorithmically predicted masks [16], all while preserving the visual integrity of the adjacent areas. In contrast, instruction-based image editing operates through direct commands like "add fireworks to the sky," avoiding the need for detailed descriptions or regional masks. Recent approaches utilize synthetic input-goal-instruction triples [7] and incorporate human feedback [76] to execute editing instructions effectively. Despite the advances in using diffusion models for various instruction-guided image editing tasks, there is still a notable gap in research specifically addressing instruction-guided image restoration with these models. Our study aims to bridge this gap by collecting a comprehensive dataset of paired low-level instruction-driven image editing examples and proposing an all-in-one model for low-level tasks and editing.

2.2 Large Language Models for Vision

Recent advancements in the development of Large Language Models (LLMs) have led to the emergence of powerful models with extensive capabilities [15, 31, 39, 43, 77]. These LLMs, pre-trained on large-scale internet-based datasets, are equipped with broad knowledge bases that enhance their zero-shot and in-context learning abilities [5, 29, 51]. Furthermore, there is a growing focus on using LLMs for multimodal tasks [3, 30, 42, 45], incorporating methods like vision-language

https://doi.org/10.52202/079017-1263

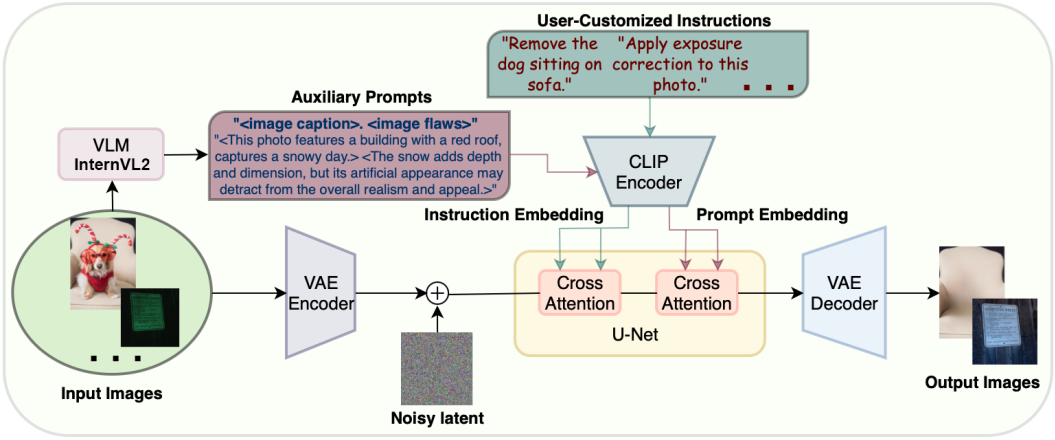

Figure 2: The architecture of our proposed PromptFix.

alignment and adapter fine-tuning. These techniques ensure that the visual data processed by visual encoders is semantically aligned with the textual input of LLMs [43]. This approach has spurred significant advancements in Text-to-Image generation, leading to the development of various LLM-based diffusion models for these tasks [28, 40, 41]. Despite these successes, there remains a relative scarcity of research focusing on using large Vision-Language Models (VLMs) for instructed image editing, particularly in detailed, low-level editing tasks.

3 Data Curation

Current off-the-shelf image datasets [7, 74, 76] with instructional annotations primarily facilitate image editing research, encompassing tasks such as color transfer, object replacement, object removal, background alteration, and style transfer. Nevertheless, their overlap with low-level applications is limited. Moreover, we find it challenging to achieve satisfactory results for existing models in image restoration. Our goal is to construct a comprehensive visual instruction-following dataset specifically for low-level tasks. We obtain $\sim$ 1.01 million training triplet instances.

Paired Image Collection. We initially gather source images from various existing datasets. Subsequently, we produce degraded and inpainted images to create an extensive set of paired image data. We compile approximately two million raw data points across eight tasks: image inpainting, object creation, image dehazing, image colorization, super-resolution, low-light enhancement, snow removal, and watermark removal. For the test set, we randomly select 300 image pairs for each task. More details about the dataset composition are provided in the Appendix A.1.

Instruction Prompts Generation. For each low-level task, we utilized GPT-4 to generate diverse training instruction prompts $\mathcal{P}_{\text{instruction}}$. These prompts include task-specific and general instructions. The task-specific prompts, exceeding 250 entries, clearly define the task objectives. For example, "Improve the visibility of the image by reducing haze" for dehazing. The general instructions include five ambiguous commands that we retained as "negative" prompts to promote adaptive tasks. The specific instruction prompts used for training are detailed in the appendix. For watermark removal, super-resolution, dehazing, snow removal, low-light enhancement, and colorization tasks, we also generate "auxiliary prompts" for each instance. These auxiliary prompts describe the quality issues for the input image and provide semantic captions. More details are discussed in Section 4.2.

4 Methodology

Let $I \in \mathbb{R}^{H \times W \times 3}$ denote the degraded input image. Our PromptFix model aims to enhance I using the prompt $\mathcal{P}$ and the diffusion model $\mathcal{H}$.

https://doi.org/10.52202/079017-1263

Algorithm 1 High-frequency Guidance Sampling.

Input: $\mathcal{H}$, D_θ, I, $\mathcal{P}$
Hyper-parameter: λ, $\{\sigma_t\}_{t=1}^{T}$, $\{\alpha_t\}_{t=1}^{T}$, S_{churn}, S_{noise}, S_{tmin}, S_{tmax}

sample $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I})$
for $t \in \{T, \ldots, 1\}$ **do**
  $\triangleright\ \gamma_t = \begin{cases} \min\left(\frac{S_{\text{churn}}}{N}, \sqrt{2}-1\right) & \text{if } \sigma_t \in [S_{\text{tmin}}, S_{\text{tmax}}] \\ 0 & \text{otherwise} \end{cases}$
  sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}\left(\mathbf{0}, S_{\text{noise}}^2\mathbf{I}\right)$
  $\hat{\mathbf{z}}_t \leftarrow \mathbf{z}_t + \sqrt{\hat{\sigma}_t^2 - \sigma_t^2}\boldsymbol{\epsilon}_t$, $\hat{\sigma}_t \leftarrow \sigma_t + \gamma_t\sigma_t$ ▷ Increase noise temporarily and inject new noise for state transition.
  $\hat{\boldsymbol{\epsilon}}(\mathbf{z}_t, t), \hat{\mathbf{z}}_{t-1} \leftarrow \mathcal{H}(\hat{\mathbf{z}}_t, I, \mathcal{P})$ ▷ Return predicted noise from neural network and denoised latent.
  $\mathbf{z}_{t-1} \leftarrow \hat{\mathbf{z}}_t + (\sigma_{t-1} - \hat{\sigma}_t)\left(\hat{\mathbf{z}}_t - \hat{\mathbf{z}}_{t-1}\right)/\hat{\sigma}_t$ ▷ Execute Euler step moving forward from $\hat{\sigma}_t$ to σ_{t-1}.
  $\mathbf{z}_{t\to 0} \leftarrow (\mathbf{z}_t - \sigma_t\hat{\boldsymbol{\epsilon}}(\mathbf{z}_t, t))/\alpha_t$ ▷ Equation (3)
  $\theta \leftarrow \theta - e^{-\lambda t}\nabla_\theta\mathcal{L}(I, D_\theta(\mathbf{z}_{t\to 0}))$
end for

4.1 Diffusion Model

Diffusion models transform data into noise through gradual Gaussian perturbations during a forward process and subsequently reconstruct samples from this noise in a backward process. In the forward phase, an original data point, denoted as $\mathbf{z}_0$, is incrementally altered towards a Gaussian noise distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, according to the equation:

$$\mathbf{z}_t = q(\mathbf{z}_0, \boldsymbol{\epsilon}, t) = \alpha_t\mathbf{z}_0 + \sigma_t\boldsymbol{\epsilon}, \quad \forall t \in [0, T], \tag{1}$$

where α_t and σ_t are coefficients that manage the signal-to-noise ratio at each interpolation point $\mathbf{z}_t$. This process aims to maintain variance, adopting coefficient strategies as detailed in sources such as [33]. Modeled as a stochastic differential equation (SDE) in continuous time, the forward process can be expressed as $d\mathbf{z} = \mathbf{f}(\mathbf{z}, t)dt + g(t)d\mathbf{w_t}$, where $\mathbf{f}(\mathbf{z}, t)$ is a vector-valued drift coefficient, $g(t)$ is the diffusion coefficient, and $\mathbf{w_t}$ represents Brownian motion at time t.

The backward diffusion process, made possible by notable characteristics of the SDE, is rearticulated via Fokker-Planck dynamics [66] to yield deterministic transitions with consistent probability densities, creating the *probability flow ODE*:

$$d\mathbf{z} = \left[\mathbf{f}(\mathbf{z}, t) - \frac{1}{2}g(t)^2\nabla_{\mathbf{z}}\log p_t(\mathbf{z})\right]dt. \tag{2}$$

This equation outlines a transport mechanism that is learnable through maximum likelihood techniques, applying the perturbation kernel of diffused data samples $\nabla_{\mathbf{z}}\log p_t(\mathbf{z}|\mathbf{z}_0)$, as demonstrated in [32, 66]. Next, we sample $\mathbf{z}_t \sim \mathcal{N}(0, I)$ to initialize the probability flow ODE, and the estimates for the score function via $\hat{\boldsymbol{\epsilon}}(\mathbf{z}_t, t)/\sigma_t$. We employ the Euler method [64, 66] among numerical ODE solvers to obtain the solution trajectory: $\mathbf{z}_0 \approx \mathcal{H}_N \circ \mathcal{H}_{N-1} \circ \cdots \circ \mathcal{H}_1(\mathbf{z}_T)$, where $\mathcal{H}$ denotes the diffusion model and N represents the neural function evaluations (NFEs) for sampling.

During the training phase, a simple diffusion loss [26] is utilized, whereby the neural network still employs forward inference to predict noise. The sample data estimate $\mathbf{z}_0$ can be obtained at any step t by using the current noisy data and the predicted noise and is derived as:

$$\mathbf{z}_{t\to 0} = \frac{\mathbf{z}_t - \sigma_t\hat{\boldsymbol{\epsilon}}(\mathbf{z}_t, t)}{\alpha_t}. \tag{3}$$

To reduce computational costs, the aforementioned diffusion process initiates from isotropic Gaussian noise samples in the latent space [57], rather than the pixel space. This space transformation is facilitated through VAE compression [20]. The VAE autoencoder comprises an encoder $E(\cdot)$ and a left inverse decoder $D(\cdot)$. For instance, an image x can be encoded into a latent code $E(x)$, which can then be approximately reconstructed back into the pixel space as $x \approx D \circ E(x)$.

4.2 VLM-based Auxiliary Prompt Module

Given that low-level image processing focuses on handling degraded images rather than real-world images, we adopt the integration of a VLM to estimate an *auxiliary prompt* for the low-level image I. This auxiliary prompt encompasses both semantic captions and defect descriptions to enhance the semantic clarity of the target image, thereby addressing the instructional gaps inherent in low-level image processing tasks.

https://doi.org/10.52202/079017-1263

Based on text dialogues parameterized by ω within a VLM $\mathcal{V}(\cdot;\omega)$, we employ a frozen VLM, specifically the InternVL2 [15] model, which integrates visual and linguistic modalities as inputs. We facilitate this model to receive paired degraded image I and a textual query $\mathcal{Q}$. To handle the visual input, the InternVL2 first employs a pre-trained encoding model to map each modality into a shared representation space. The visual encoding model ϕ embeds I into the textual space, resulting in $\phi(I)$, which is then combined with the tokenized language embedding $\tau(\mathcal{Q})$. These combined embeddings are fed into the large language model, producing the textual response $\mathcal{R}$.

$$\mathcal{R}(I, \mathcal{Q}) = \mathcal{V}(\phi(I), \tau(\mathcal{Q}); \omega) \tag{4}$$

The visual encoding model of InternVL2 has not undergone extensive fine-tuning in the degradation domain. To acquire an explicit understanding from both semantic and low-level defect perspectives, we meticulously curate the queries $\mathcal{Q}_{\text{semantic}}$ and $\mathcal{Q}_{\text{defect}}$ to guide InternVL2, respectively. Specific query instances are provided in the Appendix. As illustrated in Figure 4 and described by Equation 5, we concatenate the responses related to semantics and degradation textually, forming $\mathcal{P}_{\text{auxiliary}}$, which serves as the *auxiliary prompt*. This acts as a supplement to the instruction prompt $\mathcal{P}_{\text{instruction}}$.

$$\mathcal{P}_{\text{auxiliary}} = [\mathcal{R}(I, \mathcal{Q}_{\text{semantic}}), \mathcal{R}(I, \mathcal{Q}_{\text{defect}})] \tag{5}$$

The conditioned text prompts guide the diffusion model by injecting the embedding into the cross-attention layer [57]. After obtaining the auxiliary prompt, a straightforward approach involves concatenating it with the user-input instruction prompt before feeding the text embedding into the diffusion model. However, this concatenation can render the entire prompt excessively long, leading to forced truncation during tokenization. Therefore, after utilizing the pre-trained CLIP visual encoder ViT-L/14 [54] to extract linguistic features, we process the text embeddings of $\mathcal{P}_{\text{instruction}}$ and $\mathcal{P}_{\text{auxiliary}}$ separately. We introduce additional cross-attention layers identical to the original ones, as depicted in Figure 2. The embeddings of $\mathcal{P}_{\text{instruction}}$ and $\mathcal{P}_{\text{auxiliary}}$ are fed into the Key and Value heads of consecutive attention networks, respectively, thereby achieving an augmented cross-attend adaptation.

4.3 High-frequency Guidance Sampling

There is a fundamental requirement in image restoration and generation tasks: the processed image must maintain high accuracy in semantics. We observe that vanilla VAE reconstructions tend to lose image details such as textual rendering, which contains high-frequency information, as shown in Figure 5. Therefore, we propose high-frequency guidance sampling to balance the quality and fidelity of generation.

The denoising sampling is based on the EDM formulation [33]. To maintain spatial information, we utilize a modified VAE Decoder D_θ to map from the latent space to the pixel space. We modify the VAE decoder by passing skip-connect features from the VAE encoder through additional LoRA convolutions [27] to merge the feature map. The LoRA networks are initialized randomly, with their trainable parameters denoted as θ. Since the parameters of the LoRA convolution are lightweight, merely multi-step backpropagation can maintain high-frequency consistency without requiring extensive fine-tuning.

We propose a *fidelity constraint* to model the spatial discrepancy between the image and the ground truth. We implement two types of high-pass operators to extract high-frequency signals from the degraded image. For the Fourier filtering operator $\mathcal{F}(\cdot)$, we convert the generated image from the spatial to the frequency domain using the Discrete Fourier Transform [50]. High-frequency components are then isolated via high-pass filtering and reconstituted into an image through the Inverse Fourier Transform [50]. Meanwhile, we apply the Sobel [53] edge detection operator $\mathcal{S}(\cdot)$ as a complement. The fidelity constraint evaluates the divergence between the high-frequency components of the ground truth and the processed image, ensuring the preservation of spatial information throughout the sampling process. Additionally, to obtain the image at time step t, we utilize predicted noise ϵ from diffusion model $\mathcal{H}$ to compute the $\mathbf{z}_{t\to 0}$ estimation at any time step. The fidelity constraint is calculated as follows:

$$\mathcal{L}(I, D_\theta(\mathbf{z}_{t\to 0})) = \|\mathcal{F}(I) - \mathcal{F}(D_\theta(\mathbf{z}_{t\to 0}))\|_2^2 + \|\mathcal{S}(I) - \mathcal{S}(D_\theta(\mathbf{z}_{t\to 0}))\|_2^2 \tag{6}$$

Given that $\mathbf{z}_{t>0}$ represents a noisy latent variable, assigning equal weight to each timestep's latent is impractical. To mitigate the cumulative error induced by this practice, we introduce a time-scale weight $e^{-\lambda t}$. The overall sampling algorithm is detailed in Algorithm 1.

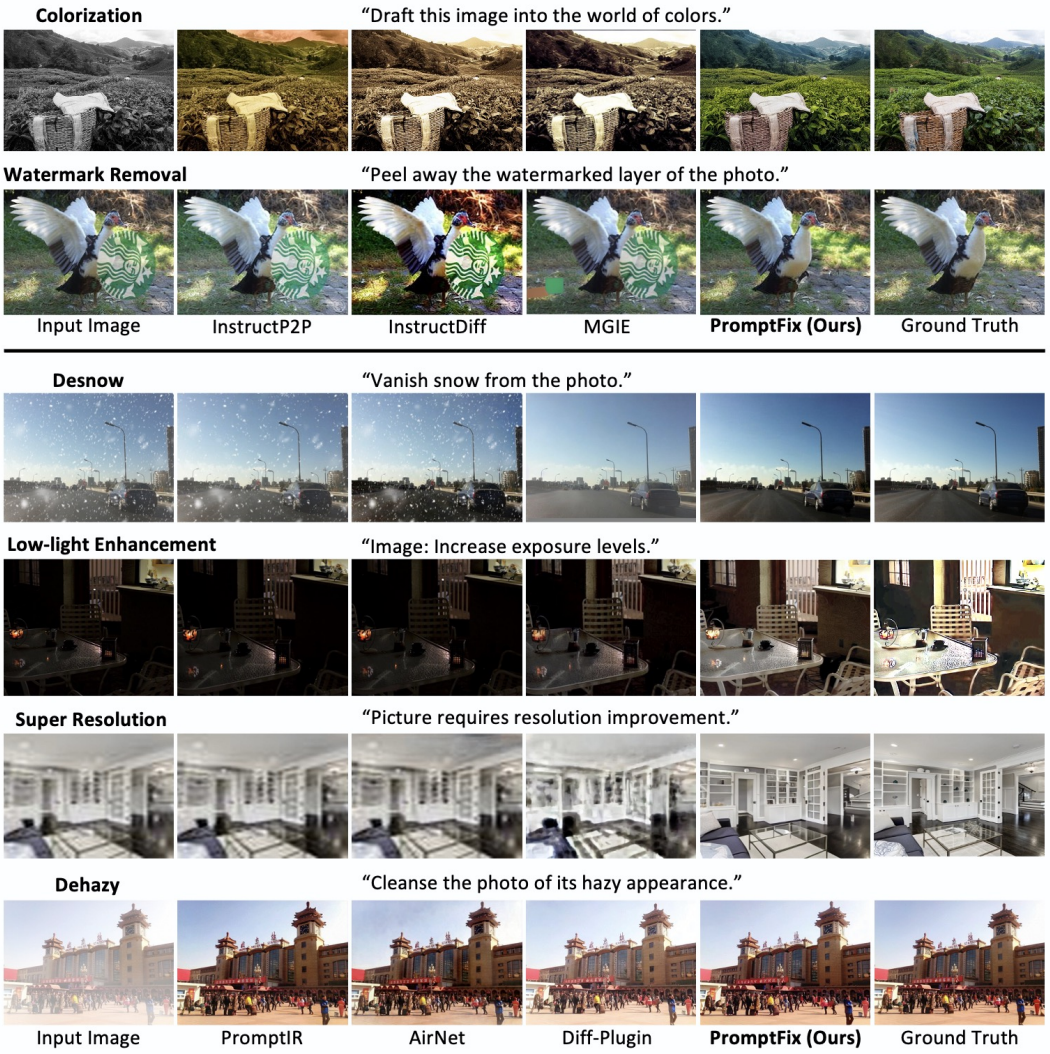

Figure 3: Qualitative comparison between PromptFix and other instruct-driven diffusion methods (InstructP2P [7], InstructDiff [23], and MGIE [21]) for image processing, as well as low-level generalist techniques (PromptIR [52], AirNet [38], and Diff-Plugin [46]) for image restoration.

5 Experiments

5.1 Experimental Setup

Implementation details. We train PromptFix for 46 epochs on 32 NVIDIA V100 GPUs, employing a learning rate of 1×10^{-4} with the Adam optimizer. The training input resolution is set to 512×512, matching the capabilities of our backbone models, InternVL2 [15] and Stable Diffusion 1.5 [57]. To facilitate classifier-free guidance [7], we randomly drop the input image latent, instruction, and auxiliary prompt with a probability of 0.075 during training. The hyperparameter λ for the time-scale weight in Algorithm 1 is empirically set to 0.001. For more implementation details, please refer to the appendix.

Baselines and metrics. We adopt instruction-based generalist models, such as InstructP2P [7], MGIE [21], and InstructDiffusion [23], as our primary comparison. MGIE employs VLM-guided techniques for image editing, while InstructDiffusion addresses overlapping tasks with our training objectives, including watermark removal and inpainting. Additionally, we evaluate all-in-one image restoration methods like AirNet [38] and PromptIR [52] (which do not support instruction input), as well as image restoration expert models, fine-tuned for specific sub-tasks [46, 73]. We assess the

https://doi.org/10.52202/079017-1263

Table 1: Quantitative comparison is conducted across seven low-level datasets with a 512×512 input resolution. Expert models refer to approaches, such as Diff-plugin [46], which use non-generalizable training pipelines and maintain separate pre-trained weights for each of the four restoration tasks. Image Restoration Generalist Methods denote models that integrate multiple low-level tasks into a single framework. Instruct-driven Diffusion Methods represent diffusion generative baselines that follow human language instructions. ↑ indicates higher is better and ↓ indicates lower is better. The **best** and second best results are in bold and underlined, respectively.

Method	LPIPS↓ / ManIQA↑			LPIPS↓ / ManIQA↑			
	Colorization	Object Removal	Watermark Removal	Low-light Enhance.	Desnow	Dehazy	Super Res.
Expert Models							
Diff-Plugin [46]	-	-	-	0.227/0.453	0.133/0.508	0.033/0.758	0.097/0.555
Inst-Inpaint [73]	-	0.227/0.593	-	-	-	-	-
Image Restoration Generalist Methods							
PromptIR [52]	-	-	-	0.330/0.539	0.235/0.553	0.037/0.764	0.105/0.442
AirNet [38]	-	-	-	0.332/0.541	0.245/0.589	0.039/0.780	0.107/0.450
Instruction-driven Diffusion Methods							
InstructP2P [7]	0.394/0.424	0.177/0.791	0.341/0.378	0.581/**0.460**	0.365/**0.560**	0.216/0.625	0.234/0.528
InstructDiff [23]	0.433/0.256	0.071/**0.811**	0.247/0.675	0.368/0.309	0.255/0.530	0.124/0.711	0.233/0.623
MGIE [21]	0.425/0.393	-	0.463/0.506	0.491/0.356	0.483/0.417	0.249/0.704	0.397/0.385
PromptFix (ours)	**0.233/0.489**	**0.054**/0.810	**0.127/0.750**	**0.135**/0.423	**0.103**/0.535	**0.088/0.752**	**0.143/0.642**

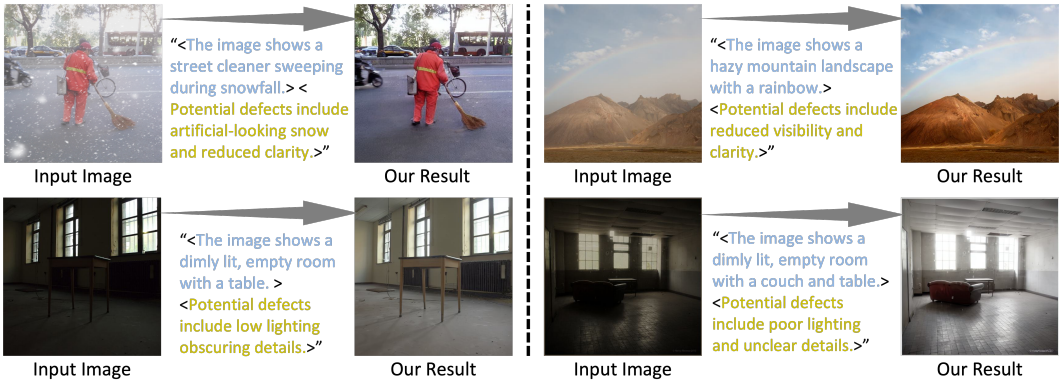

Figure 4: Qualitative analysis of VLM-guided blind restoration for desnowing, dehazing, and low-light enhancement. The results are obtained from PromptFix without explicit task instructions, relying solely on the input image. The auxiliary prompt, automatically generated by a VLM during inference, includes semantic captions and defect descriptions, indicated by <blue> and <yellow> tags, respectively.

similarity of the generated images to the ground truth using metrics such as PSNR, SSIM [69], and LPIPS [75]. For no-reference image quality evaluation, we utilize the ManIQA [72] metric.

5.2 Quantitative and Qualitative Results

Table 1 illustrates the comparative analysis of image restoration and editing techniques, evaluated via LPIPS and ManIQA metrics. The expert model – Diff-Plugin shows limited but notable performance in low-light enhancement (LPIPS/ManIQA: 0.227/0.453) and desnowing (0.133/0.508). Among generalist methods, AirNet demonstrates balanced capabilities in tasks like desnowing and dehazing, achieving LPIPS/ManIQA scores of 0.245/0.589 and 0.039/0.780, respectively. However, the instruction-driven diffusion methods reveal a more nuanced picture, with PromptFix emerging as particularly promising. It excels in colorization (LPIPS/ManIQA: 0.233/0.489), object removal (0.054/0.810), and watermark removal (0.071/0.811), consistently outperforming others. InstructP2P and InstructDiff also perform well in specific tasks, such as low-light enhancement and dehazing, but do not match the overall versatility of PromptFix. MGIE, though effective in certain domains, lacks the consistency seen in "PromptFix (Ours)." This highlights the robustness and superior performance

Table 2: Quantitative comparison of multi-task processing on 200 sampled test images, each paired with a degraded version exhibiting three defects (e.g., desnowing, dehazing and super-resolution) and the corresponding ground truth.

Method	PSNR↑	SSIM↑	LPIPS↓	ManIQA↑
PromptIR [52]	19.30	0.7385	0.2282	0.4310
AirNet [38]	19.15	0.7197	0.2359	0.4576
InstructP2P [7]	14.19	0.5845	0.3046	0.3790
InstructDiff [23]	17.12	0.6387	0.2865	0.4365
PromptFix	**22.05**	**0.7654**	**0.1519**	**0.4889**

Table 3: Quantitative comparison on three low-level tasks. PromptFix † denotes blind restoration without input instruction prompts. The compared baselines specify task objectives explicitly.

Method	LPIPS↓ / ManIQA↑		
	Low-light Enhancement	Desnow	Dehazy
PromptIR [52]	0.331 / 0.539	0.236 / 0.533	0.038 / 0.764
AirNet [38]	0.332 / 0.541	0.245 / 0.589	0.039 / 0.781
InstructP2P [7]	0.581 / 0.461	0.365 / 0.560	0.216 / 0.626
InstructDiff [23]	0.369 / 0.309	0.256 / 0.531	0.124 / 0.712
PromptFix†	0.161 / 0.413	0.115 / 0.531	0.148 / 0.755

of PromptFix across diverse image processing tasks and indicates the potential of PromptFix to set new benchmarks in the field, driven by advanced instruction-driven diffusion methodologies.

Figure 3 illustrates the visual comparison across all selected baseline models. In colorization, our PromptFix produces the most visually accurate and vibrant results, closely resembling the ground truth. For the watermark removal task, it effectively restores the original image without introducing artifacts, outperforming MGIE [21] and other methods. In desnowing and low-light enhancement, PromptFix achieves clearer and more natural outputs, significantly reducing noise and enhancing visibility. Additionally, in super-resolution, PromptFix demonstrates remarkable clarity and accuracy, preserving fine details and surpassing all comparative methods. In dehazing, although PromptFix's performance is visually comparable to image restoration experts PromptIR [52] and AirNet [38], PromptFix outperforms the recent stable-diffusion-based method Diff-plugin [46], achieving a clean, sharp appearance, and closely matching the ground truth.

5.3 Ablation Study

Effectiveness of High-frequency Guidance Sampling. The High-frequency Guidance Sampling (HGS) method is introduced to balance *fidelity* and *quality*. To validate the effectiveness of HGS, we conduct qualitative experiments and quantitative experiments. As depicted in Figure 5, in a low-light scenario, the model aims to enhance the visibility (*quality*) of the input image while preserving its original textual details (*fidelity*). For baselines leveraging stable-diffusion as a generative prior, the strong compression capability of VAE also brings the issue of spatial information loss, as demonstrated by InstructDiff [23], MGIE [21], and Diff-Plugin [46] in Figure 5. This issue is independent of the model's ability to effectively follow instructions. As shown by the variant "Ours w/o HGS," our method significantly enhances low-light images compared to the three baselines, yet still fails to retain small-scale textual structures. By incorporating HGS, as seen in "Ours," the proposed framework delivers a high-fidelity solution that also meets the low-light enhancement instruction. The usage of $\mathcal{F}(\cdot)$ and $\mathcal{S}(\cdot)$ improves the quality of the image generated, demonstrated by the quantitative results shown in Table 4.

Figure 5: Preservation of low-level image details using the proposed High-frequency Guidance Sampling (HGS) method, compared to previous VAE-based baselines [21, 23, 46] utilizing stable-diffusion architecture.

VLM-guided blind restoration. We utilize InternVL2 [15] to generate auxiliary prompts and leave the instruction prompt empty. This approach enables users to input an image without the need to provide instructions for its restoration. We evaluate the model's performance on such blind restoration tasks, including low-light enhancement, desnowing, and dehazing. As shown in Table 3, our model achieves performance comparable to four baselines, showing minimal perceptual differences from the ground truth and superior zero-shot capabilities.

https://doi.org/10.52202/079017-1263

Table 4: Quantitative study on HGS and Auxiliary Prompting. $\mathcal{F}(\cdot)$ and $\mathcal{S}(\cdot)$ represent the type of high-frequency operators used in HGS to guide sampling.

HGS ($\mathcal{F}(\cdot)$)	HGS ($\mathcal{S}(\cdot)$)	Auxiliary Prompting	LPIPS↓	ManIQA↑
-	-	✓	0.2068	0.6487
-	✓	✓	0.1707	0.6300
✓	-	✓	0.1795	0.6195
✓	✓	-	0.1990	0.5856
✓	✓	✓	0.1600	0.6274

Table 5: Ablation Study on Different Types of Instruction Prompt. $\mathbb{A}$: instructions used during training; $\mathbb{B}$: out-of-training human instructions with fewer than 20 words; $\mathbb{C}$: out-of-training human instructions with 40-70 words.

Method	Instruction Prompt Type	LPIPS↓	ManIQA↑
PromptFix	$\mathbb{A}$	0.1600	0.6274
	$\mathbb{B}$	0.1639	0.6258
	$\mathbb{C}$	0.1823	0.5958

Multi-task processing. Although PromptFix is not explicitly trained to handle multiple low-level tasks simultaneously within the same image, it demonstrates the capability for multi-task processing. We construct the validation dataset with 200 images, and each image contains 3 restoration tasks such as colorization, watermark removal, low-light enhancement, desnowing, dehazing, and super-resolution. We benchmarked PromptFix against AirNet and PromptIR, both generalist image restoration methods, as well as InstructP2P and InstructDiff, which are instruction-driven diffusion methods. As shown in Table 2, PromptFix outperforms these baselines, achieving superior image quality, structural similarity, and minimal perceptual differences from the ground truth, as evidenced by competitive PSNR, SSIM, and LPIPS scores, along with a higher ManIQA score indicating visually pleasing and high-quality results. Conversely, while methods like InstructP2P and InstructDiff perform well in specific metrics, they do not match the overall balanced performance of PromptFix. These results indicate the robustness and versatility of PromptFix.

Different types of instruction prompts. We verify PromptFix's generalization to various human instructions in Table 5, by conducting ablation comparisons with three types of prompts: instructions used during training and out-of-training human instructions with fewer than 20 words and with 40-70 words. PromptFix's performance slightly declines with out-of-training instructions, but the change is negligible. It indicates that PromptFix is robust for instructions under 20 words, which is generally sufficient for low-level processing tasks. We observe a performance drop with longer instructions, possibly due to the long-tail effect of instruction length in the training data. Although low-level processing tasks usually don't require long instructions, addressing this issue by augmenting the dataset with longer instructions could be a direction for future work.

6 Conclusion

We present PromptFix, a novel diffusion-based model along with a large-scale visual-instruction training dataset, designed to benefit instruction-guided low-level image processing. PromptFix effectively addresses challenges related to spatial information loss and degradation adaptation by high-frequency guidance sampling and a VLM-based auxiliary prompt module. These mechanisms improve the model's performance in the instruction-based image-processing paradigm. Extensive experiment results demonstrate PromptFix's advanced capabilities of generating accurate and high quality images. In addition to the improvement in terms of conventional metrics, we observe that PromptFix is also effective at processing multi-task processing and achieving blind restoration in low-light enhancement, desnowing, and dehazing.

References

[1] Ancuti, C.O., Ancuti, C., Sbert, M., Timofte, R.: Dense-haze: A benchmark for image dehazing with dense-haze and haze-free images. In: ICIP (2019)

[2] Ancuti, C.O., Ancuti, C., Timofte, R., De Vleeschouwer, C.: O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In: CVPRW (2018)

[3] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: Vqa: Visual question answering. In: ICCV (2015)

[4] Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: CVPR (2022)

[5] Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., et al.: Openflamingo: An open-source framework for training large autoregressive vision-language models. arXiv preprint arXiv:2308.01390 (2023)

[6] Bar-Tal, O., Ofri-Amar, D., Fridman, R., Kasten, Y., Dekel, T.: Text2live: Text-driven layered image and video editing. In: ECCV (2022)

[7] Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023)

[8] Cai, Y., Bian, H., Lin, J., Wang, H., Timofte, R., Zhang, Y.: Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In: ICCV (2023)

[9] Cai, Y., Xiao, Z., Liang, Y., Qin, M., Zhang, Y., Yang, X., Liu, Y., Yuille, A.: Hdr-gs: Efficient high dynamic range novel view synthesis at 1000x speed via gaussian splatting. In: NeurIPS (2024)

[10] Chen, C., Chen, Q., Do, M.N., Koltun, V.: Seeing motion in the dark. In: ICCV (2019)

[11] Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: CVPR (2018)

[12] Chen, H., Ren, J., Gu, J., Wu, H., Lu, X., Cai, H., Zhu, L.: Snow removal in video: A new dataset and a novel method. In: ICCV (2023)

[13] Chen, W.T., Fang, H.Y., Ding, J.J., Tsai, C.C., Kuo, S.Y.: Jstasr: Joint size and transparency-aware snow removal algorithm based on modified partial convolution and veiling effect removal. In: ECCV (2020)

[14] Chen, W.T., Fang, H.Y., Hsieh, C.L., Tsai, C.C., Chen, I., Ding, J.J., Kuo, S.Y., et al.: All snow removed: Single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss. In: ICCV (2021)

[15] Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Muyan, Z., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: CVPR (2024)

[16] Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. In: ICLR (2023)

[17] Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., Raff, E.: Vqgan-clip: Open domain image generation and editing with natural language guidance. In: ECCV (2022)

[18] Cun, X., Pun, C.: Split then refine: Stacked attention-guided resunets for blind single image visible watermark removal. In: AAAI (2021)

[19] Dhariwal, P., Nichol, A.Q.: Diffusion models beat gans on image synthesis. In: NeurIPS (2021)

[20] Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021)

[21] Fu, T.J., Hu, W., Du, X., Wang, W.Y., Yang, Y., Gan, Z.: Guiding instruction-based image editing via multimodal large language models. In: ICLR (2024)

[22] Gan, Y., Park, S., Schubert, A., Philippakis, A., Alaa, A.M.: Instructcv: Instruction-tuned text-to-image diffusion models as vision generalists. In: ICLR (2024)

[23] Geng, Z., Yang, B., Hang, T., Li, C., Gu, S., Zhang, T., Bao, J., Zhang, Z., Hu, H., Chen, D., et al.: Instructdiffusion: A generalist modeling interface for vision tasks. In: CVPR (2024)

[24] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM (2020)

[25] Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. ICLR (2023)

 https://doi.org/10.52202/079017-1263

[26] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020)

[27] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: ICLR (2022)

[28] Hu, X., Wang, R., Fang, Y., Fu, B., Cheng, P., Yu, G.: Ella: Equip diffusion models with llm for enhanced semantic alignment. arXiv preprint arXiv:2403.05135 (2024)

[29] Hu, Y., Hua, H., Yang, Z., Shi, W., Smith, N.A., Luo, J.: Promptcap: Prompt-guided image captioning for vqa with gpt-3. In: ICCV (2023)

[30] Hua, H., Shi, J., Kafle, K., Jenni, S., Zhang, D., Collomosse, J., Cohen, S., Luo, J.: Finematch: Aspect-based fine-grained image and text mismatch detection and correction. In: ECCV (2024)

[31] Hua, H., Tang, Y., Xu, C., Luo, J.: V2xum-llm: Cross-modal video summarization with temporal prompt instruction tuning. arXiv preprint arXiv:2404.12353 (2024)

[32] Hyvärinen, A., Dayan, P.: Estimation of non-normalized statistical models by score matching. Journal of Machine Learning Research (2005)

[33] Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. NeurIPS (2022)

[34] Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019)

[35] Kim, G., Kwon, T., Ye, J.C.: Diffusionclip: Text-guided diffusion models for robust image manipulation. In: CVPR (2022)

[36] Kwon, G., Ye, J.C.: Clipstyler: Image style transfer with a single text condition. In: CVPR (2022)

[37] Li, B., Ren, W., Fu, D., Tao, D., Feng, D., Zeng, W., Wang, Z.: Benchmarking single-image dehazing and beyond. TIP (2018)

[38] Li, B., Liu, X., Hu, P., Wu, Z., Lv, J., Peng, X.: All-in-one image restoration for unknown corruption. In: CVPR (2022)

[39] Li, C., Xu, H., Tian, J., Wang, W., Yan, M., Bi, B., Ye, J., Chen, H., Xu, G., Cao, Z., et al.: mplug: Effective and efficient vision-language learning by cross-modal skip-connections. EMNLP (2022)

[40] Li, Y., Zhang, Y., Wang, C., Zhong, Z., Chen, Y., Chu, R., Liu, S., Jia, J.: Mini-gemini: Mining the potential of multi-modality vision language models. arXiv preprint arXiv:2403.18814 (2024)

[41] Lian, L., Li, B., Yala, A., Darrell, T.: Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. TMLR (2023)

[42] Lin, J., Hua, H., Chen, M., Li, Y., Hsiao, J., Ho, C., Luo, J.: Videoxum: Cross-modal visual and textural summarization of videos. TMM (2023)

[43] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. NeurIPS (2023)

[44] Liu, Y., Zhu, Z., Bai, X.: Wdnet: Watermark-decomposition network for visible watermark removal. In: WACV (2021)

[45] Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv preprint arXiv:2307.06281 (2023)

[46] Liu, Y., Liu, F., Ke, Z., Zhao, N., Lau, R.W.: Diff-plugin: Revitalizing details for diffusion-based low-level tasks. In: CVPR (2024)

[47] Liu, Y.F., Jaw, D.W., Huang, S.C., Hwang, J.N.: Desnownet: Context-aware deep network for snow removal. TIP (2018)

[48] Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015)

[49] Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (2017)

[50] Nussbaumer, H.J., Nussbaumer, H.J.: The fast Fourier transform. Springer (1982)

[51] OpenAI: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)

[52] Potlapalli, V., Zamir, S.W., Khan, S.H., Khan, F.S.: Promptir: Prompting for all-in-one image restoration. In: NeurIPS (2023)

[53] Pratt, W.K., Jr., J.E.A.: Digital image processing. J. Electronic Imaging (2007)

[54] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)

[55] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: ICML (2016)

[56] Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: ECCV (2020)

[57] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)

[58] Ruan, L., Chen, B., Li, J., Lam, M.L.: Aifnet: All-in-focus image restoration network using a light field-based dataset. TCI (2021)

[59] Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5B: an open large-scale dataset for training next generation image-text models. In: NeurIPS (2022)

[60] Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. NeurIPS (2022)

[61] Shen, Z., Wang, W., Lu, X., Shen, J., Ling, H., Xu, T., Shao, L.: Human-aware motion deblurring. In: ICCV (2019)

[62] Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., Hassner, T.: Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In: CVPR (2021)

[63] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015)

[64] Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)

[65] Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. NeurIPS (2019)

[66] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021)

[67] Tudosiu, P.D., Yang, Y., Zhang, S., Chen, F., McDonagh, S., Lampouras, G., Iacobacci, I., Parisot, S.: Mulan: A multi layer annotated dataset for controllable text-to-image generation. In: CVPR (2024)

[68] Wang, R., Xu, X., Fu, C.W., Lu, J., Yu, B., Jia, J.: Seeing dynamic scene in the dark: A high-quality video dataset with mechatronic alignment. In: ICCV (2021)

[69] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP (2004)

 https://doi.org/10.52202/079017-1263

[70] Wei, C., Wang, W., Yang, W., Liu, J.: Deep retinex decomposition for low-light enhancement. In: BMVC (2018)

[71] Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: A face detection benchmark. In: CVPR (2016)

[72] Yang, S., Wu, T., Shi, S., Lao, S., Gong, Y., Cao, M., Wang, J., Yang, Y.: MANIQA: multi-dimension attention network for no-reference image quality assessment. In: CVPRW (2022)

[73] Yildirim, A.B., Baday, V., Erdem, E., Erdem, A., Dundar, A.: Inst-inpaint: Instructing to remove objects with diffusion models. arXiv preprint arXiv:2304.03246 (2023)

[74] Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: A manually annotated dataset for instruction-guided image editing. In: NeurIPS (2023)

[75] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)

[76] Zhang, S., Yang, X., Feng, Y., Qin, C., Chen, C.C., Yu, N., Chen, Z., Wang, H., Savarese, S., Ermon, S., et al.: Hive: Harnessing human feedback for instructional visual editing. In: CVPR (2024)

[77] Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. In: ICLR (2024)

A Appendix

A.1 PromptFix Dataset

The PromptFix dataset contains 1,013,320 triplet instances among 7 different tasks containing object removal, image dehazing, colorization, debluring, low light enhancement, snow removal and watermark removal. Each instance contains an input image, a processed image, an instruction, and an auxiliary prompt generated by InternVL2 [15] which is only available for tasks except object removal. The dataset is available at `https://huggingface.co/datasets/yeates/PromptfixData`.

Object removal: Mulan [67] provides multi-layer image decomposition annotations, inpainting occluded areas and isolating instances in a scene on separate RGBA layers. We combine the background layer and n foreground object layers to obtain the ground truth image and the background layer and $n+1$ foreground object layers to obtain the input image, where $n \in \mathbb{N}$. Different object layers will be placed onto the background image based on their foreground-background relationships.

Image dehazing: We use a combination of synthetic datasets (RESIDE [37], SRRS [13]) and real-world datasets (Dense-Haze [1], O-Haze [2]) for training, comprising 100 pairs of real-world images and 102,230 pairs of indoor and outdoor synthetic images from ITS [37], OTS [37], SOTS outdoor [37], and nyuhaze500 [37].

Colorization: We utilize a subset of Laion-5b [60] and Flickr comprising 317,225 images and generate grayscale images.

Deblurring: We use a combination of GoPro [49], HIDE [61], LFDOF [58], RealBlur [56], TextOCR [62], Wider-Face [71] and a subset of CelebA [48], which collectively contain 70,600 pairs of images.

Low light enhancement: We use a combination of LOL [70], SID [11], SMID [10], SDSD [68], and construct a set of low light synthetic data based on Pexels and Flickr by reducing brightness and adding noise, which collectively contains 212,618 pairs of images.

Snow removal: We use a combination of SRRS [13], CSD [14], RVSD [12], and Snow100K [47], which collectively contains 99,177 pairs of images.

Watermark removal: We use a combination of CLWD [44] and LOGO30K [18], which collectively contain 124,805 pairs of images.

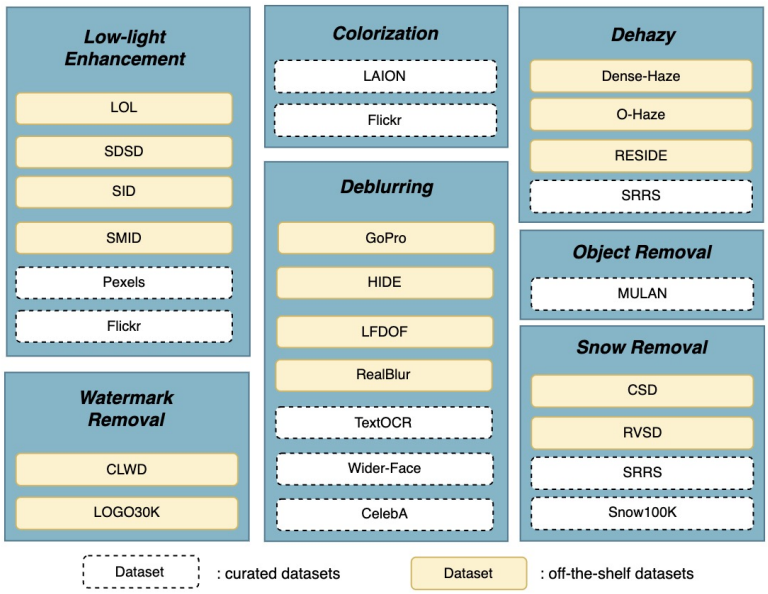

Figure 6: Data composition.

https://doi.org/10.52202/079017-1263

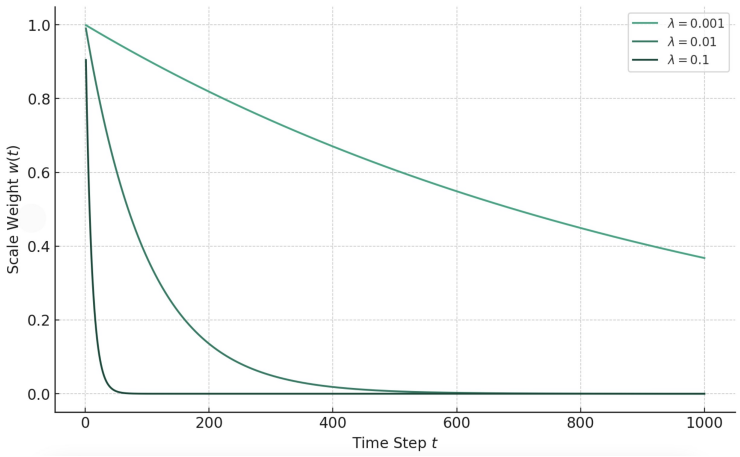

Figure 7: Exponential Decay Weight Functions.

A.2 Limitations

Without relying on user input instructions, our PromptFix achieves blind restoration for low-level enhancement, dehazing, and desnowing. However, we observe that this approach occasionally results in out-of-conditioned image control, where the model performs text-to-image generation based on the auxiliary prompt rather than image processing. Although blind restoration is a feature of our VLM-based Auxiliary Prompt Module, for known degradations, we recommend providing user-custom instructions to specify the restoration.

High-frequency Guidance Sampling significantly helps preserve the original image details, counteracting the spatial information loss caused by VAE compression. Nevertheless, we find that adopting HGS makes the restored image slightly resemble the degraded image. While PromptFix remains promising relative to baselines, as shown in Figure 5, results without HGS appear brighter than those with HGS. As we have consistently claimed, employing HGS involves a trade-off between fidelity and quality. In scenarios where VAE reconstruction is sufficiently faithful and user needs prioritize quality, HGS may be omitted.

A.3 More Implementation Details

Below are the more detailed implementation specifics:

- $\mathcal{Q}_{\text{semantic}}$: *"Describe this image and its style in a very detailed manner"*
- $\mathcal{Q}_{\text{defect}}$: *"Introduce the drawback of the image"*

Parameters of the Sobel Operator: This involves applying two 3×3 kernels:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

These kernels calculate gradients in the horizontal and vertical directions, respectively. By convolving these kernels with the image, we obtain the gradient magnitudes at each pixel. The default parameters include a mid-range intensity threshold to discern significant edges.

Time-scale Weight λ: This weight is determined based on the time step of the diffusion model. When the time step is larger, there is more noise, and the fidelity constraint should be decayed. As depicted in Figure 7, the impact of the high-frequency loss exponentially decays with increasing timestep. Empirically, we have set λ to 0.001.

Weight of Fidelity Constraint: The time-scale weight, which is determined by the time step of the diffusion model, decreases as the time step increases, due to the increase in noise. Consequently, the fidelity constraint should be decayed accordingly.

HGS Algorithm Notion Clarification: Using T to represent the neural function evaluations (NFEs) for sampling, and S to denote a series of hyperparameters:

- S_{churn} controls the level of stochasticity.
- S_{noise} adjusts the noise standard deviation.
- S_{tmin} and S_{tmax} define the range of noise levels for stochasticity.

A.4 Efficiency analysis

We conduct an efficiency analysis for PromptFix and other recent baseline models, including Diff-plugin [46] and InstructDiff [23], by profiling the floating-point operations (FLOPs) of their respective diffusion models. As demonstrated in the table below, our method exhibits comparable efficiency to these advanced models. This analysis reveals the capability of PromptFix to maintain high computational efficiency while delivering superior performance in various image-processing tasks.

	TFLOPs
Diff-Plugin [46]	1.1
InstructDiff [23]	0.8
PromptFix (Ours)	0.8

A.5 More Results

General PromptFix Results. We present additional image processing results in Figures 9-15, encompassing tasks including watermark removal, colorization, low-light enhancement, super-resolution, dehazing, desnowing, inpainting, and object removal via bounding boxes. These results are generated by our PromptFix model, utilizing an all-in-one pre-trained weight and guided by user-customized instructions.

Mutli-task processing and blind restoration. In Figure 8, we present the qualitative results of multi-task processing as a complement to the quantitative assessments detailed in Table 2. Additionally, we provide further visualizations of VLM-guided blind restoration outcomes as in Figure 16.

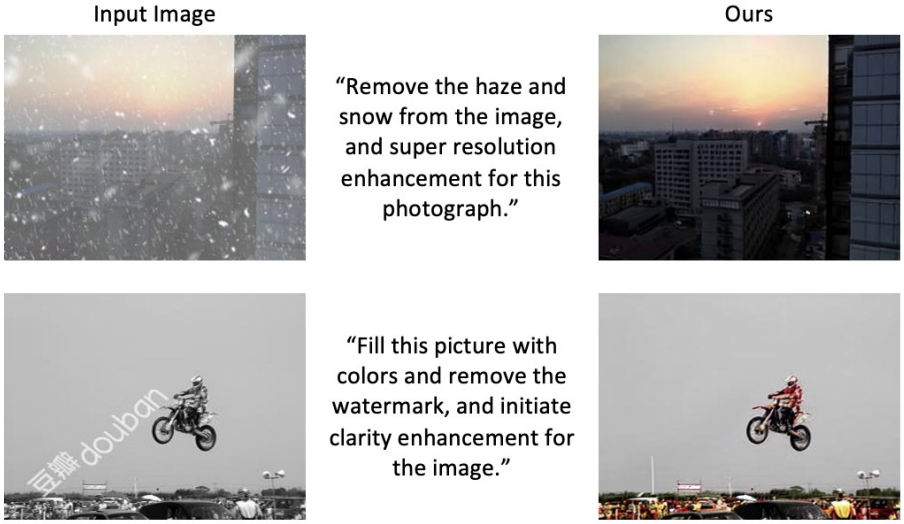

Figure 8: Visual results for Mutli-task processing.

https://doi.org/10.52202/079017-1263

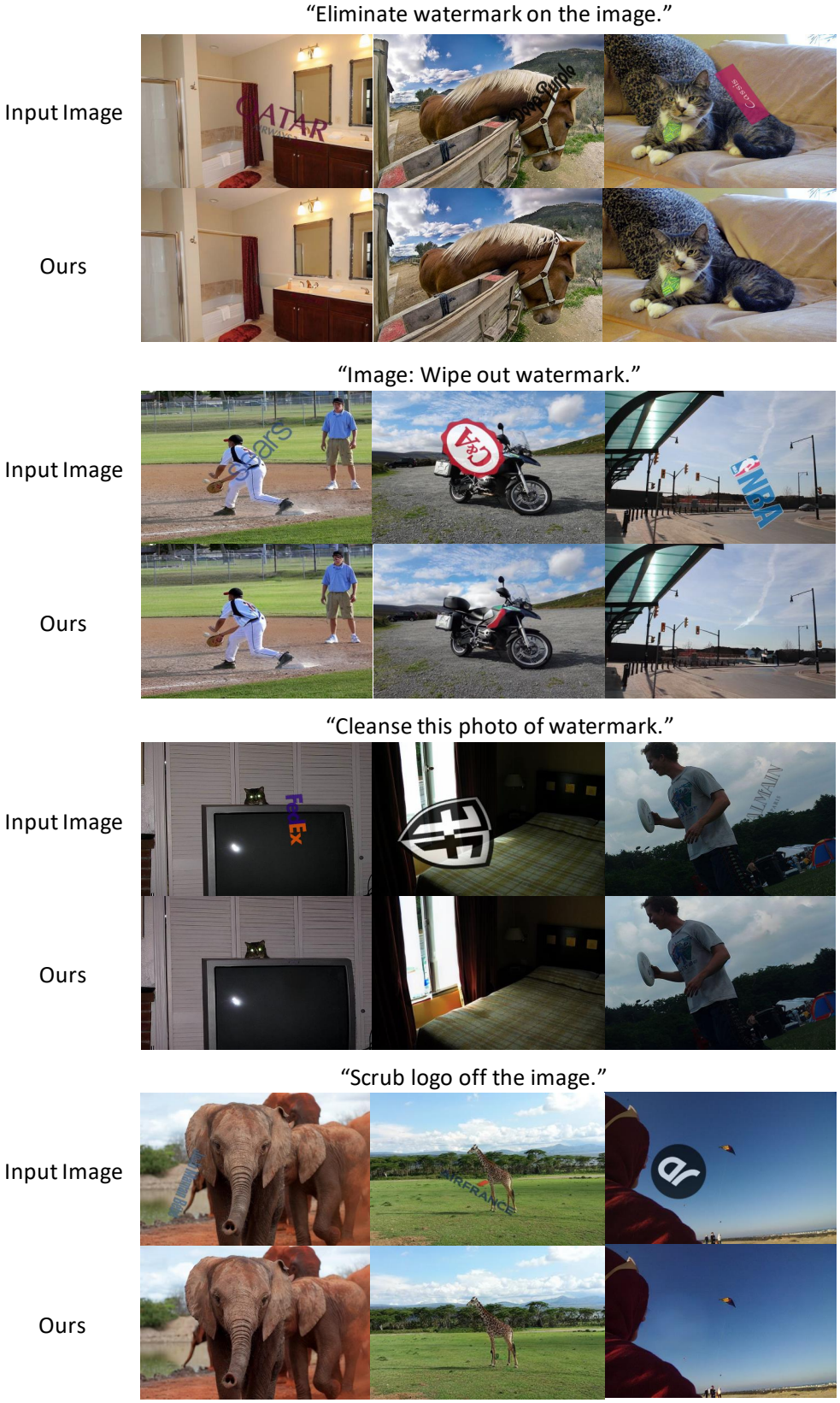

Figure 9: More results for watermark removal.

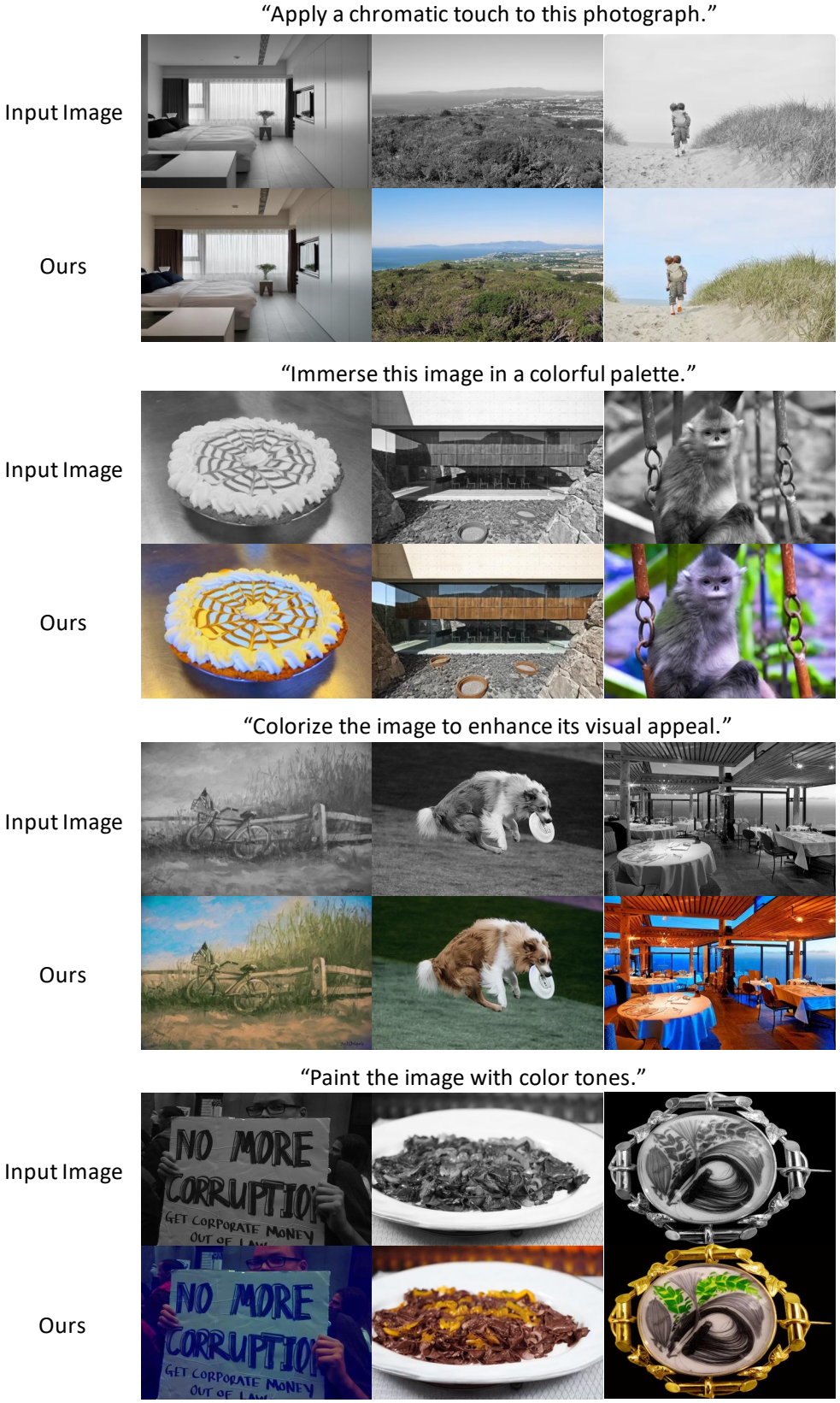

Figure 10: More results for image colorization.

https://doi.org/10.52202/079017-1263

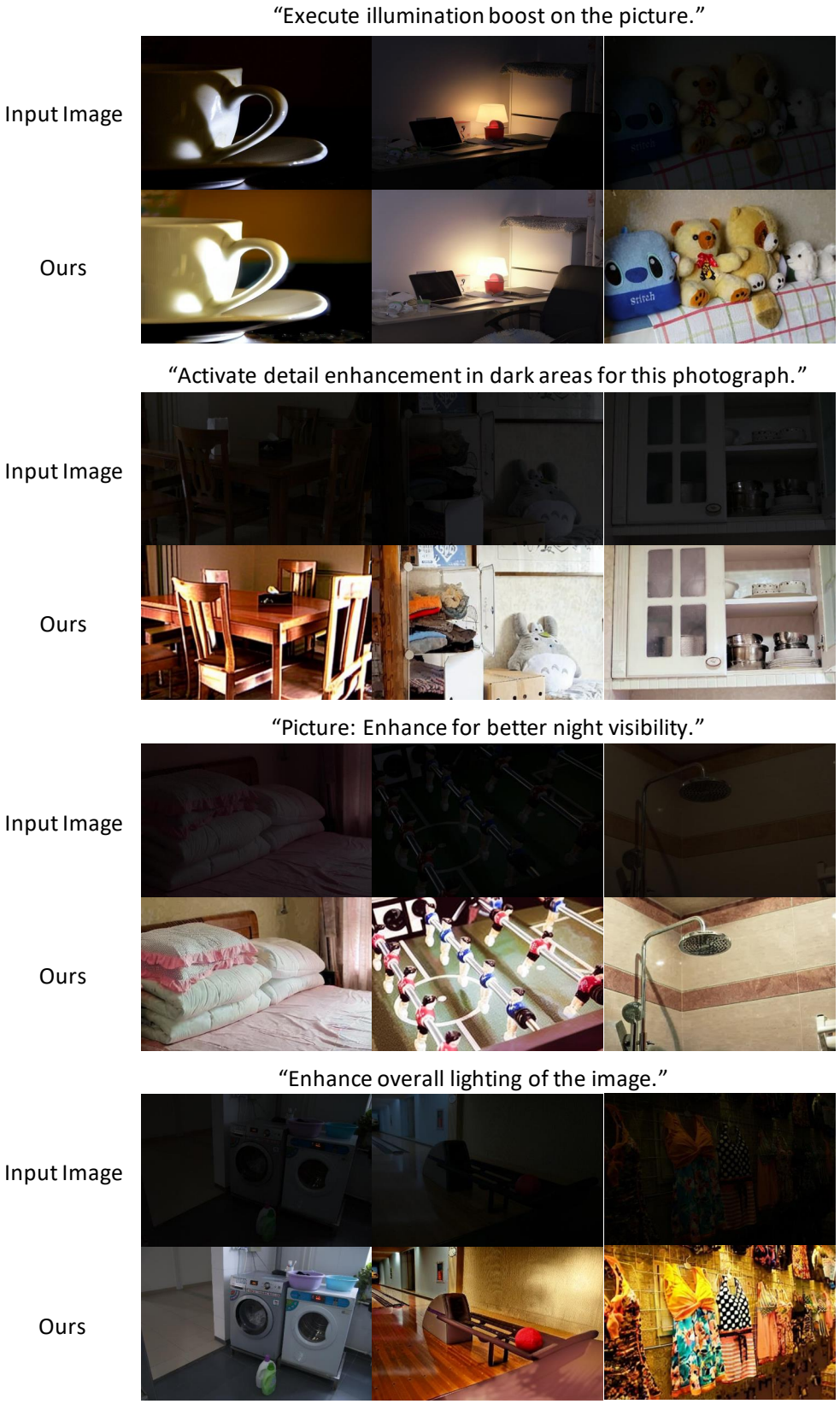

Figure 11: More results for low-light enhancement.

Figure 12: More results for image dehazing.

https://doi.org/10.52202/079017-1263

Figure 13: More results for desnowing.

Figure 14: More results for image inpainting.

https://doi.org/10.52202/079017-1263

Figure 15: More results for object removal by a bounding box.

https://doi.org/10.52202/079017-1263

Figure 16: Visual results for VLM-guided blind restoration.

https://doi.org/10.52202/079017-1263

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction align with the detailed findings and discussions presented in the paper, accurately representing its scope and contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Please see appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper includes all necessary assumptions for each theoretical result and provides thorough and accurate proofs, ensuring clarity and correctness.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of the experimental setup, methodologies, and parameters, ensuring that the main results can be reproduced accurately.

   Guidelines:

https://doi.org/10.52202/079017-1263

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: We are ready to release data and code, along with comprehensive instructions, enabling faithful reproduction of the main experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification:The paper provides comprehensive details on data splits, hyperparameters, selection methods, and the type of optimizer used, ensuring a thorough understanding of the experimental setup and results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not provide information on the statistical significance of the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper details the type of compute workers, providing all necessary information for reproduction.

https://doi.org/10.52202/079017-1263

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS Code of Ethics, ensuring ethical standards are met in all aspects, including data use, transparency, and the potential impact of the findings.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper does not focus on potential societal impacts, either positive or negative, of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: The paper does not mention any specific safeguards for the responsible release of data or models that might be susceptible to misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: They are cited appropriately in the paper, with licenses and terms of use clearly mentioned and respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide new visual instruction datasets for low-level tasks, and the documentation is comprehensive and available alongside the datasets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

https://doi.org/10.52202/079017-1263

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: We do not conduct crowdsourcing experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The study does not involve human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.