UniProcessor: A Text-induced Unified Low-level Image Processor

Huiyu Duan[1,2], Xiongkuo Min[1,⋆], Sijing Wu[1],
Wei Shen[2,⋆], and Guangtao Zhai[1,2,⋆]

[1] Institute of Image Communication and Network Engineering,
Shanghai Jiao Tong University
[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
{huiyuduan,minxiongkuo,wusijing,wei.shen,zhaiguangtao}@sjtu.edu.cn

Abstract. Image processing, including image restoration, image enhancement, *etc.*, involves generating a high-quality clean image from a degraded input. Deep learning-based methods have shown superior performance for various image processing tasks in terms of single-task conditions. However, they require to train separate models for different degradations and levels, which limits the generalization abilities of these models and restricts their applications in real-world. In this paper, we propose a text-induced Unified image Processor for low-level vision tasks, termed **UniProcessor**, which can effectively process various degradation types and levels, and support multimodal control. Specifically, our UniProcessor encodes degradation-specific information with the subject prompt and process degradations with the manipulation prompt. These context control features are injected into the UniProcessor backbone via cross-attention to control the processing procedure. For automatic subject-prompt generation, we further build a vision-language model for general-purpose low-level degradation perception via instruction tuning techniques. Our UniProcessor covers 30 degradation types, and extensive experiments demonstrate that our UniProcessor can well process these degradations without additional training or tuning and outperforms other competing methods. Moreover, with the help of degradation-aware context control, our UniProcessor first shows the ability to individually handle a single distortion in an image with multiple degradations. Code is available at: https://github.com/IntMeGroup/UniProcessor.

1 Introduction

During image acquisition, storage, transmission, and rendering, degradations (such as noise, blur, rain, compression, *etc.*) are often introduced, which significantly influence the quality of an image [14]. Image processing, including image

⋆ Corresponding authors.

This work was supported in part by the National Key R&D Program of China 2021YFE0206700; in part by the NSFC 62225112, 62322604, 62176159, 62271312; in part by the Shanghai Pujiang Program 22PJ1407400.

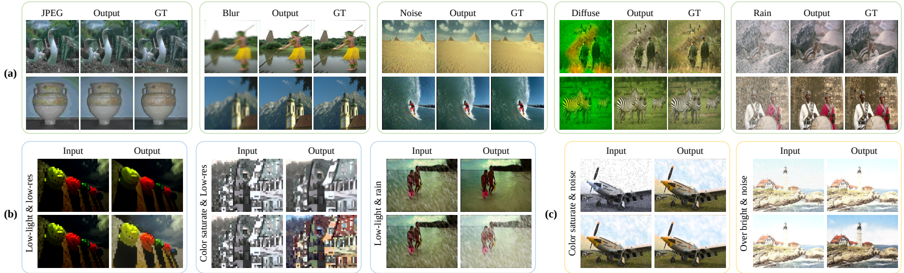


Fig. 1: UniProcessor is capable of processing various degradations in one model with text control. (a) For single degradation, UniProcessor can well restore images. (b) For an image with multiple degradations, our UniProcessor can process individual distortion with text control, which demonstrates the superior distortion perception and disentangling abilities. (c) For images with multiple degradations, Uniprocessor can process each degradation step by step to restore or enhance the images.

restoration, image enhancement, *etc.*, aims at improving the quality of a degraded image and generating a high-quality clean output. Due to the ill-posed nature, this problem is highly challenging and generally requires strong image priors for effective processing [14, 65]. With the availability of large-scale training datasets, deep learning-based image processing methods have been widely developed to tackle various low-level vision tasks, such as denoising [71, 73], deblurring [24, 63], deraining [64, 68], enhancement [57, 62], *etc.*, owing to its strong ability to learn generalizable image priors [4, 13, 54, 65].

Many deep neural networks (DNNs) have been proposed to handle single low-level vision tasks [40, 44, 45]. These methods mainly incorporate task-specific features into the network to process a single problem, such as denoising [44, 72], deblurring [40, 74], *etc.*, which lack the generalization ability to be used on other degradation processing tasks. Some DNN-based methods have focused on designing a robust network architecture to tackle various distortions using one model [4, 14, 53, 60, 65]. Although using one network model, they need to train separate copies with different weights for solving different degradation types or degradation levels. This limits the application of these models in practical scenarios due to the tedious process and complex deployment, and they need to select an appropriate pre-trained weight during the inference process, which requires prior knowledge and additional manipulations.

Recently, several methods towards handling multiple weather degradations using one model with one weight have been proposed [8, 28, 32, 36, 42, 55]. Some of these methods train parallel encoders or decoders for each specific weather degradation, which is hard to scale to more distortion types [32, 36, 55]. AirNet [28] proposes an all-in-one restoration model by utilizing an extra encoder and employing contrastive learning to differentiate various corruption types. However, it can not well disentangle different degradation representations [42]. PromptIR [42] proposes to use learnable prompts to disentangle different degradation representations. However, the learned prompt collection is a complete black-box, which is hard to understand and control. Since in some cases, we may just want to tackle one specific distortion but ignore other degradations, it is important to develop a model that can well disentangle distortion representations and handle each degradation independently. Moreover, these aforementioned models can only handle several degradations, which still have limitations in a wide range of practical applications.

In this paper, we propose an all-in-one text-induced Unified image Processor (**UniProcessor**) to tackle various low-level vision image-processing tasks. Our method supports multimodal control, which first encodes input image and subject prompt to obtain degradation-specific information, *i.e.*, subject prompt embedding, then this embedding and the manipulation prompt are fed into the text encoder to obtain the context control embedding for flexible image manipulation. By interacting context control embedding with the feature representations of the main processor network, we dynamically control and enhance the representations with the degradation-specific knowledge and manipulation-aware information. As shown in Fig. 1, benefiting from the explicit text induction, our UniProcessor can not only well restore images with a single degradation, but also separately or gradually process individual distortions in an image with multiple degradations using text control. In order to facilitate interaction and control, we further develop a vision-language model for general-purpose low-level vision degradation perception via instruction tuning techniques, which can be used to automatically generate the subject prompt for text control. Our UniProcessor is trained on 30 degradation types with various levels to conform to a variety of applications. The main highlights of this work include:

- We present a text-induced framework UniProceesor for all-in-one blind low-level image processing tasks, which has flexible and convenient text control ability.
- We propose a multimodal control module to achieve flexible control ability, which encodes degradation-specific information from the input image and subject prompt, and combines the obtained subject prompt embedding with the manipulation prompt to get the context control embedding.
- An effective processor backbone is developed, which contains a context interaction module to interact with the obtained context control embedding. The multimodal control module and the context interaction module are plug-in modules, which can be easily integrated into any existing image processing network.
- For automatic subject-prompt generation, a vision-language model for general-purpose low-level degradation queries is devised via instruction tuning techniques.
- Beyond the state-of-the-art performance on single distortion processing, our UniProcessor can process multi-degradations individually, which manifests the superior degradation disentangling ability. Moreover, to the best of our knowledge, this is the first work that attempts to solve so many degradation problems using one network.

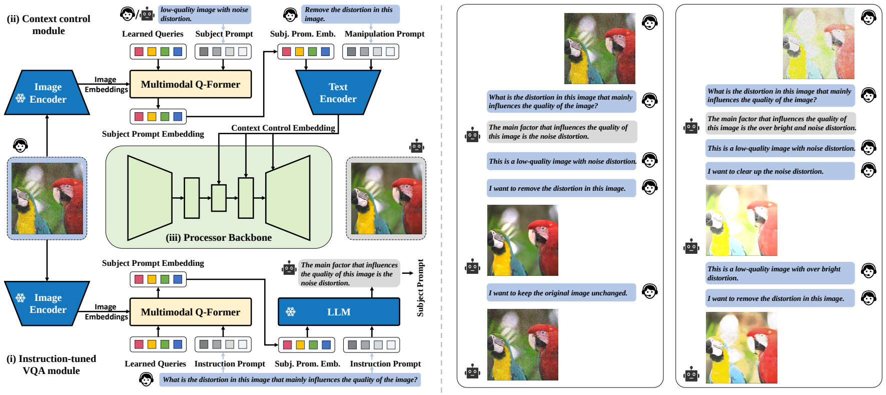
2 Related Work

2.1 Image Processing

With the development of deep learning techniques and the establishment of various databases and benchmarks, many DNN networks have been developed to handle various image restoration and image enhancement tasks, such as denoising [71,73], deblurring [24,63], deraining [64,68], low-light enhancement [57,62], *etc.*, and have achieved state-of-the-art performance. Many previous works have focused on the network design, and have proposed numerous robust architectures. Some early studies have adopted convolutional neural network (CNN) as the backbone [66, 67, 71, 78, 79], and have devised many general-purpose or task-specific modules for various tasks, such as residual and dense connection [26,59,71,79], channel attention [13,41,66,67], spatial attention [13,35,58,67], multi-scale or multi-stage networks [6, 7, 22, 25, 67, 69], *etc.* Recently, with the success of using transformer architecture across various computer vision (CV) tasks, many transformer-based networks have been developed to solve image processing problems, such as IPT [4], SwinIR [34], Uformer [60], Restormer [65], CSformer [14], *etc.* However, the aforementioned models can only solve one image restoration problem with one weight value, which lacks the generalization ability to be applied to various scenarios. Some works have proposed unified models to tackle the images corrupted due to multiple weather conditions, such as snow, rain, haze [32,36,55]. However, they need parallel multiple encoders or decoders for different tasks, which is hard to extend to more degradation types due to the dramatically increasing computational overhead. AirNet [28] and PromptIR [42] are two recent methods towards achieving all-in-one image restoration using contrastive learning or prompt learning. They are still methods that simply learn the mapping from various degraded domains to one clean domain. However, in many cases, we may want to control each degradation separately, which is beyond the capabilities of the above models.

2.2 Vision-language Model

In recent years, many large-scale vision-language models have been proposed and have greatly promoted the development of the CV field. Some vision-language models have tried to build foundation models for vision-language feature alignment. CLIP [43] is an influential method that aligns image features and text features using contrastive learning. FLIP [33] presents a mask pre-training method for the scaling vision-language pre-training process. Based on the multimodal feature alignment pre-training methods, BLIP [31] proposes a pre-training method for vision-language understanding and visual question answering (VQA). Benefiting from the rapid evolution of large language models (LLMs) [52], BLIP-2 [30] presents to use frozen image encoders and LLMs, and only train a lightweight querying transformer to save training costs. InstructBLIP [9] attempts to train general-purpose vision-language models based on BLIP-2 [30] and LLMs [52] with instruction tuning. Based on these general-purpose large-scale multimodal pre-training techniques, many text-to-image generation and editing methods have also been proposed [23, 29, 46, 48].


(ii) Context control module
low-quality image with noise distortion.
Remove the distortion in this image.
Learned Queries
Subject Prompt
Subj. Prom. Emb.
Manipulation Prompt
Image Encoder
Image Embeddings
Multimodal Q-Former
Text Encoder
Subject Prompt Embedding
Context Control Embedding
(iii) Processor Backbone
Subject Prompt Embedding
The main factor that influences the quality of this image is the noise distortion.
Subject Prompt
Image Encoder
Image Embeddings
Multimodal Q-Former
LLM
(i) Instruction-tuned VQA module
Learned Queries
Instruction Prompt
Subj. Prom. Emb.
Instruction Prompt
What is the distortion in this image that mainly influences the quality of the image?
What is the distortion in this image that mainly influences the quality of the image?
The main factor that influences the quality of this image is the noise distortion.
This is a low-quality image with noise distortion.
I want to remove the distortion in this image.
I want to keep the original image unchanged.
What is the distortion in this image that mainly influences the quality of the image?
The main factor that influences the quality of this image is the over bright and noise distortion.
This is a low-quality image with noise distortion.
I want to clear up the noise distortion.
This is a low-quality image with over bright distortion.
I want to remove the distortion in this image.

(a) An overview of the proposed UniProcessor

(b) An illustration of the examples generated by UniProcessor

Fig. 2: An illustration of the overview and the examples of UniProcessor. (a) An overview of the proposed UniProcessor. (i) Our UniProcessor first learns low-level vision-language model via instruction tuning, which can adapt to various degradation-aware visual questions and generate the subject prompt. (ii) The subject prompt and the extracted input image embedding are encoded to obtain the subject prompt embedding, which is then combined with the manipulation prompt to obtain the context control embedding. (iii) The guidance information is injected into the Processor backbone at multiple decoding stages. (b) An illustration of the examples generated by Uniprocessor, which demonstrates the good control ability and degradation disentangling capability.

3 Approach

Given an input degraded image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ with an unknown degradation D, we aim to learn a single model M to process this image and obtain a high-quality output $\hat{I}$. Fig. 2 shows the overview and the examples of our UniProcessor. Our UniProcessor mainly contains three parts, which include an instruction-tuned VQA module, a context control module, and a processor backbone. Our UniProcessor supports human-in-loop manipulation with user-supplied text and automatic manipulation with the prompt generated by the instruction-tuned VQA module. For automatic manipulation, the **overall pipeline** is given as follows. For the degraded input I, UniProcessor first uses the instruction-tuned VQA module to automatically generate a subject prompt. The subject prompt and the extracted input image embedding are then encoded using a multimodal Q-Former to obtain the subjective prompt embedding, which is then combined with the manipulation prompt using a text encoder to obtain the context control embedding. Finally, the context control embedding is injected into the Processor backbone at three decoding stages to control the process procedure and generate the output restored image $\hat{I}$.

3.1 Low-level Vision-language Instruction Tuning

We develop an instruction-tuned VQA module for UniProcessor, which can adapt to various degradation-aware visual questions and generate the subject prompt.

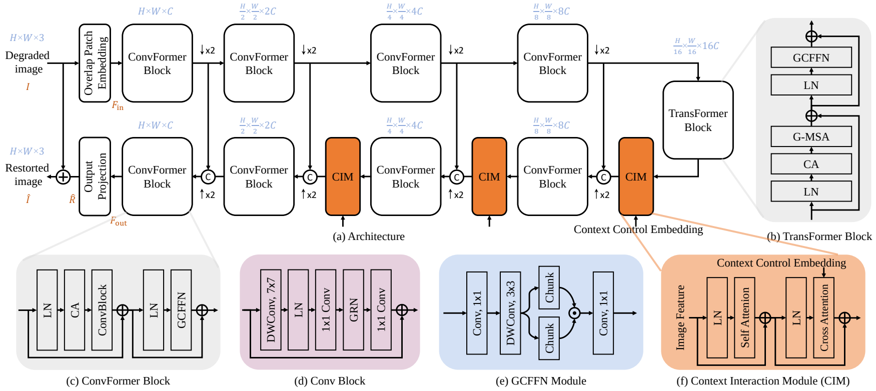
Data preparation. In order to enable the low-level vision perception and instruction ability of our UniProcessor, we first establish a new VQA database for low-level vision. We devise a distortion bank including 30 types of degradation (see supplementary material for more details). Based on this distortion bank, we generate numerous degraded images with various corruption types and levels for over 70000 image patches. Since the degradation types and levels for these images are known, we further generate various degradation-related or quality-related questions and answers for training the low-level vision VQA model. To avoid the risk of model overfitting, we follow the InstructBlip [9] to craft 10 to 15 distinct instruction templates in natural language to articulate the task and the objective (see supplementary material for more details).

Instruction-aware visual feature extraction. UniProcessor first extracts instruction-aware visual features for feasible question answering. As shown in Fig. 2 (a)-(i), the instruction-tuned VQA module encodes the input degraded image with a well pre-trained frozen CLIP image encoder [43] to obtain the image embedding. Moreover, the instruction text is also encoded by a pre-trained text encoder to obtain the instruction prompt. Then a multimodal Query Transformer, *i.e.*, Q-Former, is utilized to extract instruction-aware visual features by jointly interacting image embedding, text prompt, and K learnable query embeddings. The output of the Q-Former consists of K visual vectors, of which the number is the same as the learnable query embeddings.

Instruction-tuned low-level vision VQA. The above extracted instruction-aware visual features are then fed into a frozen LLM as soft prompt input to perform instruction-guided VQA. The frozen LLM adopted in UniProcessor is LLaMA [52]. The connection between the Q-Former and LLM is a fully-connected layer, which adapts the output instruction-aware visual features of the Q-Former to the input dimension of the LLM. The Q-Former is pre-trained with InstructBLIP [9], and we instruction-tune the model, especially the Q-Former, with the language modeling loss to generate the response.

3.2 Degradation-aware Subject and Manipulation Representation Learning

With the help of the aforementioned instruction-tuned VQA module, UniProcessor can automatically generate the degradation-aware subject prompt for an input image. However, it should be noted that UniProcessor also supports user-supplied subject prompts for feasible control. With the development of large-scale vision-language models, including text-image feature alignment such as CLIP [43] and FLIP [33], and text-aligned visual representation extraction such as BLIP [31] and BLIP2 [30], it is achievable to obtain text-image alignment features. However, these features are not specifically tailored to serve as the guidance or control information. To achieve text-guided unified image processing, we further devise a context control module to obtain the context control embedding for guidance, which is described in detail as follows.



(a) Architecture

(b) TransFormer Block

(c) ConvFormer Block

(d) Conv Block

(e) GCFFN Module

(f) Context Interaction Module (CIM)

Fig. 3: An overview of the Processor backbone. (a) The architecture of the Processor backbone. (b) The illustration of a Transformer block. (c) The illustration of a ConvFormer block. (d) The illustration of the ConvBlock. (e) The illustration of the Gated Conv Feed-Forward Network (GCFFN). (f) The demonstration of the Context Interaction Module (CIM). LN indicates a LayerNorm layer. CA is a channel-attention layer. G-MSA represents the global multi-head self-attention. GRN means the global response normalization.

Multimodal degradation-aware subject representation extraction. Only injecting text information to the image generation backbone can not well control a generation process [46, 48, 76], thus many recent works have explored multimodal control methods [18, 21, 76]. As shown in Fig. 2 (a)-(ii), we adopt the BLIP2 [30] to acquire multimodal degradation-aware subject representation. Specifically, the input image is first encoded with a frozen pre-trained CLIP image encoder [43], and then passed through a multimodal Q-Former to interact with the learnable queries and subject prompt. The Q-Former produces a degradation-aware subject visual representation aligned to the subject text prompt, which is used to generate context control embedding in the next step.

Context-aware manipulation representation extraction. To achieve context-aware manipulation, the output of the above multimodal encoder is transformed using a feed-forward network containing two fully connected layers with a GELU activation in-between to obtain the subject prompt embedding, which is conformed to the input format of the text encoder. The subject prompt embedding is then appended to the manipulation prompt with the template "[manipulation prmpt], the [subject prompt] is [subject prompt embedding]" to obtain a soft visual subject manipulation prompt. Finally, the combined manipulation and subject embeddings are fed into a CLIP text encoder [43] to produce the context control embedding, which serves as the guidance information for the processor backbone to achieve controllable generation.

3.3 UniProcessor with Context Control

Processor pipeline. Fig. 3-(a) demonstrates the architecture of the processor backbone of UniProcessor. Our UniProcessor follows the design principles of encoder-decoder with skip connections, similar to UNet [47]. For an input degraded image $\mathbf{I}$, UniProcessor first uses a 3×3 convolutional layer to extract low-level feature embeddings $\mathbf{F}_{\text{in}} \in \mathbb{R}^{H \times W \times C}$. Next, these shallow feature maps $\mathbf{F}_{\text{in}}$ are passed through a 5-level encoder-decoder network, then output feature maps $\mathbf{F}_{\text{out}} \in \mathbb{R}^{H \times W \times C}$. Each stage of the encoder-decoder contains multiple ConvFormer or TransFormer blocks, with the number of blocks gradually increasing from the shallow level to the deep level to maintain computational efficiency. The context interaction module (CIM) is injected into three decoding layers to guide and control the image processing procedure. Fig. 3-(a) shows the feature dimensions of each level. The pixel-unshuffle and pixel-shuffle [50] methods are adopted for the down-sampling process and the up-sampling process, respectively. We finally refine the output feature maps $\mathbf{F}_{\text{out}}$ from the encoder-decoder network with a 3×3 convolutional layer to get the estimated residual map $\hat{\mathbf{R}} \in \mathbb{R}^{H \times W \times 3}$, and obtain the restored image by $\hat{\mathbf{I}} = \mathbf{I} + \hat{\mathbf{R}}$. The UniProcessor is optimized using the L_1 loss: $\mathcal{L} = ||\hat{\mathbf{I}} - \mathbf{I}'||$, where $\mathbf{I}'$ is the ground-truth image.

ConvFormer and TransFormer blocks. Transformer [12, 56] has been successfully applied to the image processing tasks [4]. Due to the huge computational costs of transformer architecture when applied among global image pixels, many image processing methods have been proposed to use transformer architecture in the local context or channel dimension [14, 34, 60, 65]. As shown in Fig. 3 (a), our UniProcessor is a hybrid architecture, which applies ConvFormer blocks (Fig. 3 (c)) to perform local context processing and adopts TransFormer blocks (Fig. 3 (b)) to execute global context learning. Both the ConvFormer block and the TransFormer block contain two parts, including an attention part and a feed-forward part. The attention part in the TransFormer block contains a channel-attention (CA) module and a global multi-head self-attention (G-MSA) module, and the feed-forward part contains a gated convolutional feed-forward network (GCFFN). The ConvFormer block is similar to the TransFormer block, but uses a ConvBlock to perform local context awareness rather than the global context perception. We adopt the ConvNext v2 block as the ConvBlock as shown in Fig. 3 (d).

Channel-attention module and gated convolutional feed-forward module. UniProcessor adopts a channel-attention module [5] to perform channel-wise feature refinement. Given an input tensor $\mathbf{X}$, the output of the CA layer can be formulated as: $\text{CA}(\mathbf{X}) = \mathbf{X} * \text{MLP}(\text{Avg}(\mathbf{X}))$, where Avg is an average pooling layer, MLP is a multilayer perceptron, $*$ indicates a channel-wise product operation. As shown in Fig. 3 (e), UniProcessor adopts a gated convolutional feed-forward network (GCFFN) [10, 14, 49, 65] as the feed-forward network. Give an input tensor $\mathbf{X}$, the GCFFN process can be formulated as: $\mathbf{X}_1 = W_d^1(W_p^1(\mathbf{X}))$, $\mathbf{X}_2 = W_d^2(W_p^2(\mathbf{X}))$, $\hat{\mathbf{X}} = W_p^3(\phi(\mathbf{X}_1) \odot \mathbf{X}_2)$, where W_p represents a 1×1 point-wise convolution, W_d indicates a 3×3 depth-wise convolution, ϕ is the GELU operation, $\odot$ denotes element-wise multiplication.

Context Interaction Module. The highlight of our UniProcessor is to achieve text-guided image processing, which is mainly accomplished by the context in-

Table 1: Comparison results for **30 degrations** with heavy level on the CBSD68 dataset [39]. Our model outperforms other state-of-the-art models for almost all degradation types in terms of the three most commonly used evaluation metrics, *i.e.*, PSNR ↑, SSIM ↑ [61], and LPIPS ↓ [77]. The best results are colored in red and the second-best results are colored in blue. The distortion levels in this table and more results for other distortion levels can be found in **supplemental files**.

	DRUNet [70]	MPRNet [67]	AirNet [28]	TAPE [36]	SwinIR [34]	Uformer [60]	Restormer [65]	PromptIR [42]	UniProcessor (Ours)
Degradation	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS
JPEG comp.	25.58 / 0.718 / 0.395	25.38 / 0.718 / 0.413	24.90 / 0.709 / 0.407	25.21 / 0.714 / 0.400	24.88 / 0.709 / 0.438	25.19 / 0.722 / 0.378	25.74 / 0.729 / 0.374	25.85 / 0.731 / 0.367	26.03 / 0.737 / 0.367
Gauss. blur	23.39 / 0.578 / 0.580	23.29 / 0.573 / 0.597	22.46 / 0.528 / 0.668	22.56 / 0.532 / 0.624	22.70 / 0.539 / 0.648	22.99 / 0.556 / 0.617	24.17 / 0.621 / 0.528	24.37 / 0.635 / 0.517	24.64 / 0.647 / 0.493
Lens blur	24.08 / 0.633 / 0.447	24.22 / 0.646 / 0.387	22.25 / 0.500 / 0.616	22.15 / 0.491 / 0.561	22.19 / 0.493 / 0.595	23.24 / 0.575 / 0.545	26.39 / 0.759 / 0.278	26.16 / 0.757 / 0.286	27.35 / 0.798 / 0.212
Motion blur	22.09 / 0.548 / 0.537	21.78 / 0.532 / 0.555	20.96 / 0.485 / 0.587	21.12 / 0.499 / 0.592	21.09 / 0.498 / 0.602	21.75 / 0.528 / 0.562	24.81 / 0.720 / 0.306	24.61 / 0.700 / 0.378	25.94 / 0.761 / 0.270
Color diffuse	21.05 / 0.862 / 0.223	23.15 / 0.900 / 0.174	20.45 / 0.870 / 0.215	21.95 / 0.876 / 0.197	21.64 / 0.887 / 0.184	21.92 / 0.890 / 0.187	23.78 / 0.910 / 0.158	24.42 / 0.909 / 0.154	26.03 / 0.922 / 0.140
Color shift	34.59 / 0.986 / 0.055	36.71 / 0.993 / 0.034	36.00 / 0.991 / 0.039	35.77 / 0.991 / 0.038	34.84 / 0.991 / 0.040	36.16 / 0.991 / 0.035	39.29 / 0.995 / 0.020	38.68 / 0.995 / 0.024	41.29 / 0.996 / 0.014
Color saturate	17.41 / 0.881 / 0.288	17.94 / 0.890 / 0.275	17.12 / 0.880 / 0.291	20.66 / 0.881 / 0.264	19.59 / 0.897 / 0.243	19.37 / 0.903 / 0.239	26.04 / 0.944 / 0.101	27.21 / 0.951 / 0.084	33.15 / 0.978 / 0.024
Color saturate2	23.44 / 0.881 / 0.152	25.27 / 0.914 / 0.113	22.24 / 0.883 / 0.167	24.19 / 0.893 / 0.142	23.97 / 0.905 / 0.126	23.18 / 0.901 / 0.126	25.21 / 0.915 / 0.107	25.88 / 0.919 / 0.099	27.39 / 0.929 / 0.084
Gauss. noise	26.40 / 0.723 / 0.297	26.48 / 0.723 / 0.301	26.34 / 0.721 / 0.269	25.87 / 0.677 / 0.303	26.42 / 0.723 / 0.285	26.32 / 0.732 / 0.228	26.45 / 0.739 / 0.258	26.34 / 0.747 / 0.238	26.51 / 0.766 / 0.230
GN (ycbcr)	29.52 / 0.839 / 0.164	29.68 / 0.841 / 0.166	29.52 / 0.841 / 0.144	29.15 / 0.818 / 0.164	29.49 / 0.834 / 0.166	29.44 / 0.845 / 0.128	29.93 / 0.853 / 0.136	29.93 / 0.855 / 0.134	30.40 / 0.868 / 0.125
Impulse noise	39.08 / 0.985 / 0.013	40.38 / 0.988 / 0.009	36.49 / 0.971 / 0.030	38.76 / 0.982 / 0.018	40.44 / 0.986 / 0.012	37.71 / 0.982 / 0.015	42.28 / 0.992 / 0.006	42.06 / 0.992 / 0.006	42.74 / 0.993 / 0.003
Multipli. noise	29.66 / 0.867 / 0.146	30.38 / 0.877 / 0.137	30.02 / 0.873 / 0.127	29.74 / 0.856 / 0.135	30.23 / 0.872 / 0.136	30.28 / 0.883 / 0.099	30.62 / 0.885 / 0.114	30.55 / 0.887 / 0.113	31.25 / 0.901 / 0.102
Denoise	24.78 / 0.657 / 0.553	24.73 / 0.648 / 0.568	24.18 / 0.616 / 0.626	24.14 / 0.623 / 0.593	24.05 / 0.613 / 0.636	24.72 / 0.657 / 0.531	24.76 / 0.646 / 0.586	24.75 / 0.658 / 0.519	25.15 / 0.673 / 0.475
Over bright	15.01 / 0.742 / 0.269	18.01 / 0.800 / 0.228	14.06 / 0.683 / 0.313	16.90 / 0.757 / 0.265	19.08 / 0.842 / 0.178	13.94 / 0.770 / 0.226	20.63 / 0.877 / 0.149	21.64 / 0.882 / 0.141	23.76 / 0.905 / 0.112
Low-light	14.67 / 0.613 / 0.286	19.60 / 0.751 / 0.232	12.59 / 0.466 / 0.363	17.86 / 0.706 / 0.260	18.90 / 0.744 / 0.240	21.27 / 0.801 / 0.175	20.37 / 0.787 / 0.179	24.07 / 0.837 / 0.154	24.23 / 0.846 / 0.138
Mean shift	19.13 / 0.866 / 0.070	22.74 / 0.913 / 0.052	16.91 / 0.777 / 0.072	18.51 / 0.859 / 0.089	19.28 / 0.862 / 0.093	23.26 / 0.926 / 0.041	23.26 / 0.915 / 0.035	24.41 / 0.927 / 0.031	27.99 / 0.947 / 0.023
Bicubic resize/SR	20.98 / 0.462 / 0.734	20.84 / 0.455 / 0.740	20.73 / 0.450 / 0.766	20.72 / 0.449 / 0.748	20.74 / 0.449 / 0.775	20.94 / 0.461 / 0.730	21.12 / 0.468 / 0.709	21.17 / 0.469 / 0.723	21.25 / 0.474 / 0.697
Bilinear resize/SR	20.83 / 0.457 / 0.737	20.68 / 0.451 / 0.736	20.25 / 0.437 / 0.791	20.33 / 0.439 / 0.757	20.47 / 0.442 / 0.767	20.75 / 0.457 / 0.726	21.05 / 0.466 / 0.706	21.10 / 0.468 / 0.725	21.21 / 0.472 / 0.701
Nearest resize/SR	22.08 / 0.574 / 0.481	21.95 / 0.572 / 0.508	21.73 / 0.569 / 0.475	21.81 / 0.563 / 0.534	21.80 / 0.563 / 0.526	22.06 / 0.573 / 0.493	22.07 / 0.579 / 0.483	22.16 / 0.579 / 0.478	22.32 / 0.585 / 0.452
Lanczos resize/SR	21.01 / 0.461 / 0.749	20.91 / 0.456 / 0.748	20.85 / 0.453 / 0.763	20.86 / 0.452 / 0.745	20.87 / 0.452 / 0.774	20.99 / 0.461 / 0.742	21.15 / 0.468 / 0.714	21.17 / 0.468 / 0.728	21.27 / 0.473 / 0.702
Sharpening	25.24 / 0.887 / 0.123	25.03 / 0.884 / 0.123	25.30 / 0.874 / 0.144	25.10 / 0.866 / 0.146	25.39 / 0.896 / 0.106	25.11 / 0.881 / 0.128	25.69 / 0.896 / 0.118	26.65 / 0.917 / 0.088	27.60 / 0.930 / 0.072
Contrast imbal.	21.97 / 0.889 / 0.124	22.01 / 0.887 / 0.122	21.92 / 0.888 / 0.122	26.52 / 0.936 / 0.088	23.66 / 0.872 / 0.160	22.67 / 0.902 / 0.117	30.00 / 0.976 / 0.043	33.13 / 0.982 / 0.023	40.43 / 0.994 / 0.004
Color block	30.83 / 0.958 / 0.072	30.55 / 0.959 / 0.074	24.80 / 0.936 / 0.139	28.20 / 0.951 / 0.095	30.74 / 0.959 / 0.065	32.08 / 0.965 / 0.057	32.01 / 0.964 / 0.061	32.30 / 0.966 / 0.058	33.09 / 0.969 / 0.051
Pixelate	24.22 / 0.697 / 0.331	24.00 / 0.696 / 0.334	23.99 / 0.695 / 0.348	23.85 / 0.689 / 0.383	23.91 / 0.692 / 0.372	23.98 / 0.698 / 0.339	24.14 / 0.702 / 0.332	24.22 / 0.703 / 0.327	24.41 / 0.710 / 0.306
Discontinuous	26.76 / 0.897 / 0.074	26.37 / 0.898 / 0.061	25.16 / 0.886 / 0.076	25.03 / 0.884 / 0.088	25.14 / 0.886 / 0.080	25.00 / 0.887 / 0.077	29.04 / 0.920 / 0.068	27.33 / 0.907 / 0.057	29.65 / 0.929 / 0.062
Jitter	23.85 / 0.634 / 0.387	23.91 / 0.647 / 0.404	23.55 / 0.625 / 0.373	23.27 / 0.618 / 0.418	23.59 / 0.629 / 0.425	23.84 / 0.638 / 0.390	24.07 / 0.652 / 0.405	24.13 / 0.652 / 0.412	24.29 / 0.661 / 0.426
Mosaic	36.46 / 0.984 / 0.017	35.66 / 0.972 / 0.017	34.70 / 0.978 / 0.025	36.45 / 0.977 / 0.025	35.07 / 0.976 / 0.021	11.45 / 0.423 / 0.428	38.18 / 0.983 / 0.016	38.21 / 0.984 / 0.018	40.62 / 0.990 / 0.009
Irregular mask	27.43 / 0.885 / 0.161	28.29 / 0.889 / 0.155	26.74 / 0.868 / 0.177	26.84 / 0.875 / 0.167	27.17 / 0.882 / 0.154	27.74 / 0.894 / 0.143	28.57 / 0.891 / 0.153	29.38 / 0.894 / 0.150	29.97 / 0.900 / 0.138
Block mask	27.72 / 0.916 / 0.129	29.10 / 0.920 / 0.123	27.27 / 0.906 / 0.134	26.35 / 0.905 / 0.140	27.05 / 0.913 / 0.124	29.22 / 0.923 / 0.115	28.69 / 0.920 / 0.122	30.67 / 0.923 / 0.119	32.07 / 0.926 / 0.114
Rain streak	24.34 / 0.769 / 0.203	25.26 / 0.808 / 0.173	17.17 / 0.668 / 0.300	20.32 / 0.756 / 0.216	23.38 / 0.804 / 0.170	16.39 / 0.702 / 0.254	26.89 / 0.847 / 0.123	27.80 / 0.867 / 0.099	28.67 / 0.890 / 0.082
Snow streak	22.89 / 0.644 / 0.382	24.14 / 0.716 / 0.318	24.64 / 0.745 / 0.284	25.19 / 0.781 / 0.233	25.64 / 0.806 / 0.215	26.94 / 0.830 / 0.170	21.47 / 0.620 / 0.398	24.67 / 0.723 / 0.294	30.20 / 0.885 / 0.116
Clean image	41.23 / 0.990 / 0.009	53.36 / 0.999 / 0.001	47.76 / 0.996 / 0.003	41.98 / 0.994 / 0.011	39.31 / 0.995 / 0.006	45.20 / 0.996 / 0.003	49.17 / 0.997 / 0.002	62.24 / 0.998 / 0.001	80.16 / 1.000 / 0.000
Average	25.24 / 0.765 / 0.287	26.31 / 0.779 / 0.277	24.47 / 0.743 / 0.308	25.23 / 0.759 / 0.295	25.40 / 0.769 / 0.293	24.85 / 0.761 / 0.283	27.41 / 0.801 / 0.243	28.35 / 0.809 / 0.236	**30.35 / 0.827 / 0.211**

teraction module (CIM). In Section 3.2, we have obtained the degradation-aware context control embedding, which is represented as **E** here. The primary goal of the context interaction module is to enable interaction between the input feature **F** and the context control embedding **E**. As shown in Fig. 3 (f), the CIM contains a self-attention block and a cross-attention block with LayerNorm (LN) layers before. The overall process of the CIM is:

$$\mathbf{F}' = \text{Self-Attn}(\text{LN}(\mathbf{F})) + \mathbf{F}, \tag{1}$$

$$\mathbf{F}'' = \text{Cross-Attn}(\text{LN}(\mathbf{F}'), \mathbf{E}) + \mathbf{F}', \tag{2}$$

where LN, Self-Attn, Cross-Attn represent layernorm, self-attention, and cross-attention, respectively, and $\mathbf{F}'$ is the middle feature, $\mathbf{F}''$ is the output integrated feature of the CIM module. CIM is a plug-in module, which controls the processing procedure by interacting image features and contextual control embeddings through cross-attention layers.

4 Experiments

In this section, we conduct experiments to validate the effectiveness of the proposed method in learning an all-in-one image processing model, and demonstrate the good degradation awareness and disentangling ability of our UniProcessor. Due to the page limitation, more experimental results are shown in the *supplementary material.*

4.1 Experimental Setup

Implementation details. To enable the degradation-aware VQA capabilities of our model, we first adopt the vision language instruction tuning strategy to

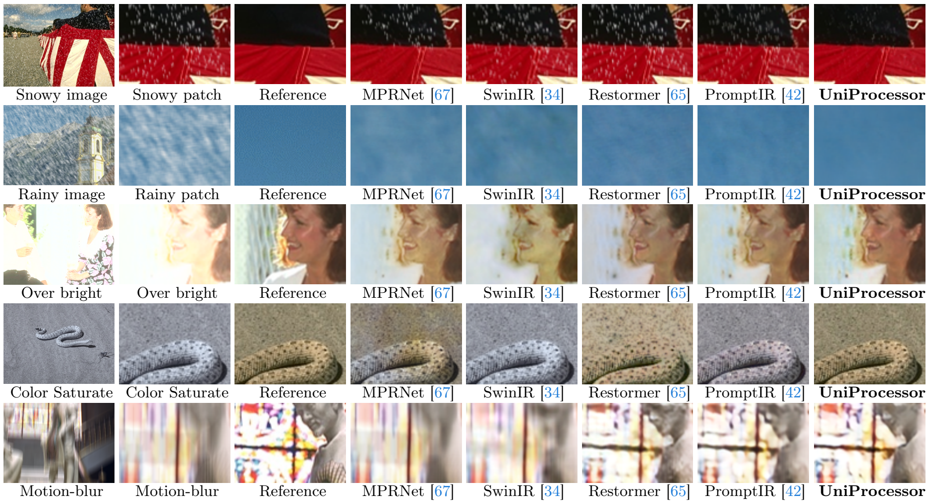


Fig. 4: Visualization results on 5 different degradation types. UniProcessor produces more visually pleasant results.

tune the VQA module of UniProcessor. Specifically, we initialize the weights of the multimodal Q-Former in the instruction-tuned module with pre-trained InstructBLIP [9], which empowers the model with initial ability of complex visual scene understanding. The frozen visual encoder is a ViT-G from EVA-CLIP [16], and the frozen LLM used in UniProcessor is Vicuna [80], which is a decoder-only LLM fine-tuned from LLaMA [52]. We fine-tune it on the constructed low-level VQA dataset to adapt the frozen LLM to give detailed feedback for the degradations of an image. The model is trained using the standard language modeling loss.

The image encoder in CIM is same as that in the VQA module, and the text encoder is from BLIP [31]. The architecture of our UniProcessor consists of a five-level encoder-decoder, with [4, 6, 6, 8, 8] ConvFormer or TransFormer layers from level-1 to level-4, respectively. For CIM-plugged decoding layers, each layer contains two CIM blocks, and the total number of CIM components in UniProcessor is 6. The UniProcessor model is trained on 224 × 224 random-cropped patches with random data augmentation methods including horizontal and vertical flips, 90° rotations, *etc.* We use AdamW Optimizer to train the network for 150 epochs with a batch size of 36. The initial learning rate is $2e^{-4}$ and gradually reduces to $1e^{-6}$ with cosine annealing [37].

Database preparation. We adopt a combined set including 900 images from DIV2K [2], 2650 images from Flickr2K, 400 images from BSD500 [3], and 4744 images from WaterlooED (WED) [38], as the training dataset, and use four datasets, including CBSD68 [39], Urban100 [20], Kodak24 [17], and McMaster [75], as the test dataset. We first crop the training images into 512×512 patches with a stride of 416 to generate 71580 small patches [65] for training. During the training process, we randomly crop desired patches from the 512×512 patches prepared above as training patches. The degradations are generated on-the-fly during training process. We first develop a distortion bank, which includes 30 common degradations with various levels (see the *supplementary material* for more details). During training, we randomly adjust the degradation level to cover a wide perception range for improving the generality of the proposed model. For testing, we set three levels of degradations including heavy, middle and slight.

4.2 Multiple Degradation All-in-one Results

We conduct extensive experiments to evaluate the all-in-one image processing results of our proposed UniProcessor as well as six state-of-the-art image restoration methods. These representative methods include DRUNet [70], MPRNet [67], SwinIR [34], Restormer [65], and PromptIR [42]. These competing methods are retrained in the experiments with their publicly released codes and following their original settings, under our data preparation setting. All models are trained for 150 epochs using the same training set and degradation generation methods.

Quantitative comparison results. Table 1 quantitatively demonstrates the performance results of our UniProcessor and six competing models for processing 30 severe degradations on the CBSD68 dataset [39]. It can be observed that our UniProcessor achieves state-of-the-art performance and outperforms other models for almost all degradation types in terms of three commonly used evaluation metrics, *i.e.*, PSNR, SSIM [61], and LPIPS [77], which manifests the effectiveness of the proposed method. Moreover, our method achieves consistent improvement but different amounts for various tasks, *e.g.*, 0.2dB for JPEG but 5.5dB for snow removal compared to PromptIR, indicating saturate improvement for some tasks. More quantitative results can be found in the *supplementary material.*

Qualitative comparison results. Fig. 4 shows the visual comparisons of the results from our UniProcessor and other state-of-the-art restoration models on 5 different degradation types. It qualitatively demonstrates that our UniProcessor can well process these degraded inputs and restore high-quality clean images using an all-in-one model. Moreover, compared to other competing methods, UniProcessor generates more visually-faithful results.

tSNE results. Fig. 5 shows the tSNE plots of the degradation embeddings in UniProcessor and the state-of-the-art all-in-one restoration model PromptIR [42]. Distinct colors represent different degradation types. The embeddings for the three tasks are better clustered in our case, which manifests that UniProcessor can effectively learn discriminative features for recognizing the degradations.

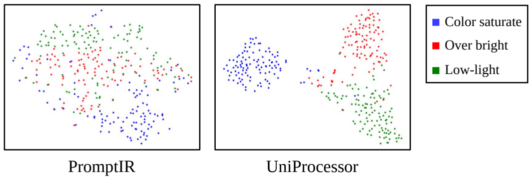
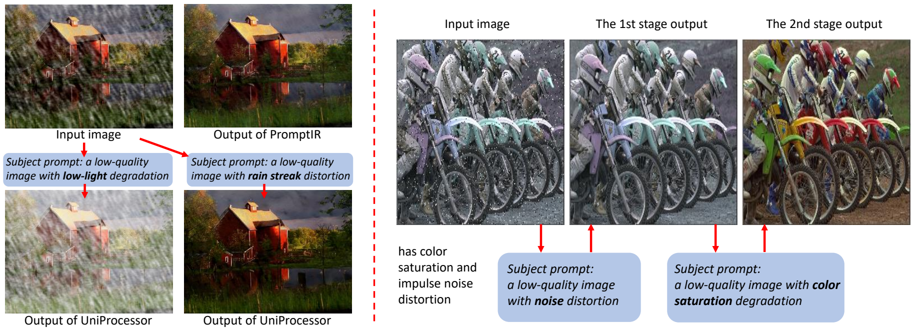


Fig. 5: tSNE plots of the degradation embeddings in UniProcessor (ours) and the state-of-the-art model PromptIR [42]. Our results are better clustered, manifesting the effectiveness of text-induced prompt method for learning discriminative degradation context.


Input image
Output of PromptIR
Subject prompt: a low-quality image with low-light degradation
Subject prompt: a low-quality image with rain streak distortion
Output of UniProcessor
Output of UniProcessor
Input image
The 1st stage output
The 2nd stage output
has color saturation and impulse noise distortion
Subject prompt: a low-quality image with noise distortion
Subject prompt: a low-quality image with color saturation degradation

(a) Separately processing (b) Sequentially processing

Fig. 6: UniProcessor can independently handle individual degradations in an image with multiple distortions through different subject prompt controls, and can gradually process the multiple distortions step by step.

Table 2: Quantitative comparison for multi-degradation separately processing and multi-degradation gradually processing. PromptIR 2-step: process an image using PromptIR twice. Degradation contains ***d1*** and ***d2***. The ***rm d1***, ***rm d2***, ***rm d1+d2***, ***rm d2+d1*** means: remove ***d1***, remove ***d2***, first remove ***d1*** then ***d2***, first ***d2*** then ***d1***, respectively.

	Restormer [65]	PromptIR [42]	PromptIR [42] 2-step	UniProcessor ***rm d1***	UniProcessor ***rm d2***	UniProcessor ***rm d1+d2***	UniProcessor ***rm d2+d1***
Degradation	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS	PSNR / SSIM / LPIPS
Low-light + resize	19.3 / 0.63 / 0.39	19.3 / 0.63 / 0.38	20.1 / 0.64 / 0.38	21.2 / 0.60 / 0.42	19.4 / 0.64 / 0.37	22.9 / 0.68 / 0.36	22.8 / 0.67 / 0.37
Color saturate + resize	18.4 / 0.63 / 0.51	18.4 / 0.63 / 0.50	19.1 / 0.63 / 0.47	20.6 / 0.58 / 0.47	18.5 / 0.63 / 0.49	22.0 / 0.65 / 0.41	22.5 / 0.65 / 0.40
Rain + low-light	21.8 / 0.83 / 0.13	22.2 / 0.85 / 0.11	23.3 / 0.86 / 0.10	23.0 / 0.87 / 0.09	19.3 / 0.58 / 0.38	28.4 / 0.89 / 0.08	28.4 / 0.90 / 0.07
Color saturate + noise	20.8 / 0.93 / 0.20	21.1 / 0.93 / 0.19	21.6 / 0.94 / 0.15	22.6 / 0.87 / 0.26	20.8 / 0.93 / 0.20	24.4 / 0.93 / 0.14	33.9 / 0.98 / 0.02
Over bright + noise	16.1 / 0.79 / 0.17	16.1 / 0.79 / 0.18	18.0 / 0.81 / 0.17	16.0 / 0.78 / 0.18	17.8 / 0.72 / 0.22	19.6 / 0.83 / 0.15	18.0 / 0.73 / 0.21

Multi-degradation separately processing results. Due to the degradation-aware and context manipulation capabilities, our UniProcessor can well disentangle the degradations and achieve the ability to individually process a single degradation in an image with multiple distortions. As shown in Fig. 6 (a), for the input image with low-light and rain streak degradations, the PromptIR [42] model can only output one restored image, and only removes the most influential degradation, *i.e.*, rain streaks. However, for UniProcessor, with different subject prompt control, we can control the process more flexibly and generate the desired output.

Multi-degradation gradually processing results. The above experiment demonstrates that UniProcessor can well disentangle degradations and individually process them. We further conduct an experiment to demonstrate that UniProcessor has the ability to remove multiple degradations step by step. As shown in Fig. 6 (b), for an input with color saturation and noise distortion, UniProcessor can first remove noise with the noise-related subject prompt. The output is served as the second stage input and we process the image with the color saturation-corresponding subject prompt to obtain the final high-quality image.

Quantitative results of multi-degradation processing. As shown in Table 2, the results of ***rm d1*** and ***rm d2*** have obvious distinctions, due to the remained degradations are different. Moreover, the better results in ***rm d1***

Table 3: Comparisons under All-in-one restoration setting [28,42]: single model trained on a combined set of images originating from different degradation types.

Method	Dehazing on SOTS [27]	Deraining on Rain 100L [15]	Denoising on BSD68 dataset [39]			Average
			$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	
BRDNet [51]	23.23/0.895	27.42/0.895	32.26/0.898	29.76/0.836	26.34/0.836	27.80/0.843
LPNet [19]	20.84/0.828	24.88/0.784	26.47/0.7782	24.77/0.748	21.26/0.552	23.64/0.738
FDGAN [11]	24.71/0.924	29.89/0.933	30.25/0.910	28.81/0.868	26.43/0.776	28.02/0.883
MPRNet [67]	25.28/0.954	33.57/0.954	33.54/0.927	30.89/0.880	27.56/0.779	30.17/0.899
DL [15]	26.92/0.391	32.62/0.931	33.05/0.914	30.41/0.861	26.90/0.740	29.98/0.875
AirNet [28]	27.94/0.962	34.90/0.967	33.92/0.933	31.26/0.888	28.00/0.797	31.20/0.910
PromptIR [42]	30.58/0.974	36.37/0.972	33.98/0.933	31.31/0.888	28.06/0.799	32.06/0.913
UniProcessor (Ours)	31.66/0.979	38.17/0.982	34.08/0.935	31.42/0.891	28.17/0.803	32.70/0.918

Table 4: Ablation study results of UniProcessor. We report the PSNR/SSIM/LPIPS results on two tasks including rain streak removal, and low-light enhancement.

Method	**Rain streak**	**Low-light**
only text encoder	27.76/0.873/0.092	23.85/0.836/0.152
w/o Q-Former	28.23/0.884/0.087	24.05/0.840/0.143
UniProcessor	**28.67/0.890/0.082**	**24.23/0.846/0.138**

(a) Ablation study for the context control module

Method	**Rain streak**	**Low-light**
w/o CIM	27.30/0.862/0.101	23.03/0.832/0.156
level 5	28.30/0.877/0.097	23.65/0.839/0.150
level 5+4	28.44/0.882/0.088	24.10/0.842/0.141
level 5+4+3	**28.67/0.890/0.082**	**24.23/0.846/0.138**

(b) Ablation study for the block position of the context interaction module.

and ***rm d2*** are better than Restormer [65] and PromptIR [42], which manifests the effectiveness of UniProcessor. Furthermore, after sequential processing (***rm d1+d2*** and ***rm d2+d1***), the performance can be greatly improved compared to the smaller improvement of PromptIR with twice inference. The order affects the sequential process depending on tasks and the 1st stage results. For that both the first stage processes are well, the order effect on the results is small. For cases that the first degradation process strongly destroys the feature of another distortion, as processing low-light 1st for low-light+noise, the order strongly affects the second results.

Results on an all-in-one restoration benchmark. We further conduct an experiment following the all-in-one settings proposed in AirNet [28] and PromptIR [42]. As shown in Table 3, our UniProcessor achieves better performance compared to other state-of-the-art models, which further demonstrates the superiority and generality of the proposed method.

4.3 Ablation Study

We further conduct ablation studies for the UniProcessor as shown in Table 4a & 4b. Table 4a demonstrates the ablation results for the context control module. It can be observed that when only using text encoder to extract control information, the performance decreases obviously, and without using Q-Former to encode subject-aligned image representation, the performance also reduces. Table 4b demonstrates the ablation results for the context interaction module. We observe that without CIM, the performance decreases a lot. Adding CIM to shallow level can improve the performance but also increase computational cost. Thus, we only add the CIM to level-5, level-4 and level-3 stages.

Table 5: Comparisons of the computational overhead of UniProcessor.

variant	GMACs	PSNR
u. swin as sa	13.50	39.77
r. ca w/ sa	13.40	39.75
r. sa w/ ca	**11.94**	39.78
UniProcessor	12.67	**39.81**

(a) Computational overhead ablation of the Processor backbone of the UniProcessor.

Model	processor	processor+CCM	processor+CCM+VQA
GMacs	**153.40**	239.83	678.52

Model	MPRNet [67]	PromptIR [42]	UniProcessor (w/o VQA)
GMacs	761.00	**158.40**	239.83

(b) Computational overhead comparisons with other models.

We further conduct ablation experiments for the computational overhead. We first test the performance and the computing overhead of the proposed processor backbone on SIDD dataset [1], and show the results in Table 5a. "u. swin as sa" represents using swin transformer as the spatial attention, *i.e.*, replacing the ConvFormer block with the swin transformer block. "r. ca w/ sa" and "r. sa w/ ca" indicate replacing channel attention with spatial attention and replacing spatial attention with channel attention, respectively, *i.e.*, repeating spatial attention and repeating channel attention twice for each block, respectively. We can observe that, compared to CSformer [14], replacing swin blocks with large-kernel convolutional blocks can effectively improve the performance. Moreover, we observe that the channel attention (ca) and spatial attention (sa) (large-kernel convolution layer) together contribute to the final improvement, and replacing one module with another module will decrease the performance.

Furthermore, we compare the computational overhead for different modules and models. UniProcessor contains three modules, which include a VQA module, a context control module (CCM), a processor backbone. As shown in Table 5b, the GMacs for the processor, processor+CCM, processor+CCM+VQA are 153.4, 239.83, 678.52, respectively. The LLM in the VQA module is the main overhead. Since we can only use processor+CCM for processing, thus the overall processing GMac for UniProcessor is 239.83, which is comparable to other models (GMacs for PromptIR [42], MPRNet [67] are 158.4, 761).

5 Conclusion

In this work, we present a text-induced unified image processor, termed UniProcessor, for all-in-one image processing. UniProcessor first has the ability to perceive low-level degradations and perform quality or degradation-related VQA, which can be used for generating the low-level subject prompt for subsequent processing procedure. Moreover, to achieve controllable and unified image processing, we develop a text-induced processor, which encodes degradation-specific information from input image and subject text prompt, and incorporates the manipulation prompt into the degradation-aware embedding to obtain context control information. The control embedding is interacted with the processor backbone to achieve controllable and unified image processing. Extensive experimental results demonstrate that UniProcessor can well process 30 degradations in one model which outperforms other competing methods, and achieve the ability to process individual distortion in an image with multiple degradations.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1692–1700 (2018) 14
2. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 126–135 (2017) 10
3. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **33**(5), 898–916 (2010) 10
4. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12299–12310 (2021) 2, 4, 8
5. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 17–33. Springer (2022) 8
6. Chen, L., Lu, X., Zhang, J., Chu, X., Chen, C.: Hinet: Half instance normalization network for image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 182–192 (2021) 4
7. Cho, S.J., Ji, S.W., Hong, J.P., Jung, S.W., Ko, S.J.: Rethinking coarse-to-fine approach in single image deblurring. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 4641–4650 (2021) 4
8. Conde, M.V., Geigle, G., Timofte, R.: High-quality image restoration following human instructions. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024) 2
9. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2023) 4, 6, 10
10. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 933–941 (2017) 8
11. Dong, Y., Liu, Y., Zhang, H., Chen, S., Qiao, Y.: Fd-gan: Generative adversarial networks with fusion-discriminator for single image dehazing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10729–10736 (2020) 13
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021) 8
13. Duan, H., Shen, W., Min, X., Tian, Y., Jung, J.H., Yang, X., Zhai, G.: Develop then rival: A human vision-inspired framework for superimposed image decomposition. IEEE Transactions on Multimedia (TMM) (2022) 2, 4
14. Duan, H., Shen, W., Min, X., Tu, D., Teng, L., Wang, J., Zhai, G.: Masked autoencoders as image processors. arXiv preprint arXiv:2303.17316 (2023) 1, 2, 4, 8, 14
15. Fan, Q., Chen, D., Yuan, L., Hua, G., Yu, N., Chen, B.: A general decoupled learning framework for parameterized image operators. IEEE transactions on pattern analysis and machine intelligence **43**(1), 33–47 (2019) 13

16. Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva: Exploring the limits of masked visual representation learning at scale. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19358–19369 (2023) 10
17. Franzen, R.: Kodak lossless true color image suite. http://r0k.us/graphics/kodak/ (1999), online accessed 24 Oct 2021 10
18. Gal, R., Arar, M., Atzmon, Y., Bermano, A.H., Chechik, G., Cohen-Or, D.: Designing an encoder for fast personalization of text-to-image models. arXiv preprint arXiv:2302.12228 (2023) 7
19. Gao, H., Tao, X., Shen, X., Jia, J.: Dynamic scene deblurring with parameter selective sharing and nested skip connections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3848–3856 (2019) 13
20. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5197–5206 (2015) 10
21. Jia, X., Zhao, Y., Chan, K.C., Li, Y., Zhang, H., Gong, B., Hou, T., Wang, H., Su, Y.C.: Taming encoder for zero fine-tuning image customization with text-to-image diffusion models. arXiv preprint arXiv:2304.02642 (2023) 7
22. Jiang, K., Wang, Z., Yi, P., Huang, B., Luo, Y., Ma, J., Jiang, J.: Multi-scale progressive fusion network for single image deraining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8346–8355 (2020) 4
23. Kang, M., Zhu, J.Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T.: Scaling up gans for text-to-image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10124–10134 (2023) 4
24. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: DeblurGAN: Blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8183–8192 (2018) 2, 4
25. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 8878–8887 (2019) 4
26. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4681–4690 (2017) 4
27. Li, B., Ren, W., Fu, D., Tao, D., Feng, D., Zeng, W., Wang, Z.: Benchmarking single-image dehazing and beyond. IEEE Transactions on Image Processing **28**(1), 492–505 (2018) 13
28. Li, B., Liu, X., Hu, P., Wu, Z., Lv, J., Peng, X.: All-in-one image restoration for unknown corruption. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17452–17462 (2022) 2, 4, 9, 13
29. Li, D., Li, J., Hoi, S.C.: Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2023) 4
30. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: Proceedings of the International Conference on Machine Learning (ICML) (2023) 4, 6, 7

31. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 12888–12900. PMLR (2022) 4, 6, 10
32. Li, R., Tan, R.T., Cheong, L.F.: All in one bad weather removal using architectural search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3175–3185 (2020) 2, 4
33. Li, Y., Fan, H., Hu, R., Feichtenhofer, C., He, K.: Scaling language-image pre-training via masking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 23390–23400 (2023) 4, 6
34. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1833–1844 (2021) 4, 8, 9, 10, 11
35. Liu, D., Wen, B., Fan, Y., Loy, C.C., Huang, T.S.: Non-local recurrent network for image restoration. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2018) 4
36. Liu, L., Xie, L., Zhang, X., Yuan, S., Chen, X., Zhou, W., Li, H., Tian, Q.: Tape: Task-agnostic prior embedding for image restoration. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 447–464. Springer (2022) 2, 4, 9
37. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. In: Proceedings of the International Conference on Learning Representations (ICLR) (2017) 10
38. Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., Zhang, L.: Waterloo exploration database: New challenges for image quality assessment models. IEEE Transactions on Image Processing (TIP) **26**(2), 1004–1016 (2016) 10
39. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 416–423 (2001) 9, 10, 11, 13
40. Nah, S., Son, S., Lee, J., Lee, K.M.: Clean images are hard to reblur: Exploiting the ill-posed inverse task for dynamic scene deblurring. Proceedings of the International Conference on Learning Representations (ICLR) (2021) 2
41. Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X., Shen, H.: Single image super-resolution via a holistic attention network. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 191–207. Springer (2020) 4
42. Potlapalli, V., Zamir, S.W., Khan, S., Khan, F.S.: Promptir: Prompting for all-in-one blind image restoration. Proceedings of the Advances in Neural Neural Information Processing Systems (NeurIPS) (2023) 2, 4, 9, 10, 11, 12, 13, 14
43. 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: Proceedings of the International Conference on Machine Learning (ICML). pp. 8748–8763. PMLR (2021) 4, 6, 7
44. Ren, C., He, X., Wang, C., Zhao, Z.: Adaptive consistency prior based deep network for image denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8596–8606 (2021) 2
45. Ren, W., Pan, J., Zhang, H., Cao, X., Yang, M.H.: Single image dehazing via multi-scale convolutional neural networks with holistic edges. International Journal of Computer Vision (IJCV) **128**, 240–259 (2020) 2

46. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695 (2022) 4, 7
47. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proceedings of the Medical Image Computing and Computer Assisted Intervention (MICCAI). pp. 234–241. Springer (2015) 7
48. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22500–22510 (2023) 4, 7
49. Shazeer, N.: Glu variants improve transformer. arXiv preprint arXiv:2002.05202 (2020) 8
50. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1874–1883 (2016) 8
51. Tian, C., Xu, Y., Zuo, W.: Image denoising using deep cnn with batch renormalization. Neural Networks **121**, 461–473 (2020) 13
52. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) 4, 6, 10
53. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: Maxim: Multi-axis mlp for image processing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5769–5780 (2022) 2
54. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9446–9454 (2018) 2
55. Valanarasu, J.M.J., Yasarla, R., Patel, V.M.: Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2353–2363 (2022) 2, 4
56. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2017) 8
57. Wang, W., Wei, C., Yang, W., Liu, J.: Gladnet: Low-light enhancement network with global awareness. In: Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). pp. 751–755. IEEE (2018) 2, 4
58. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7794–7803 (2018) 4
59. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision Workshops (ECCVW). pp. 0–0 (2018) 4
60. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17683–17693 (2022) 2, 4, 8, 9

61. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing (TIP) **13**(4), 600–612 (2004) 9, 11
62. Wei, C., Wang, W., Yang, W., Liu, J.: Deep retinex decomposition for low-light enhancement. arXiv preprint arXiv:1808.04560 (2018) 2, 4
63. Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1107–1114 (2013) 2, 4
64. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Deep joint rain detection and removal from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1357–1366 (2017) 2, 4
65. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5728–5739 (2022) 2, 4, 8, 9, 10, 11, 12, 13
66. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Learning enriched features for real image restoration and enhancement. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 492–511 (2020) 4
67. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14821–14831 (2021) 4, 9, 10, 11, 13, 14
68. Zhang, H., Sindagi, V., Patel, V.M.: Image de-raining using a conditional generative adversarial network. IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) (2019) 2, 4
69. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5978–5986 (2019) 4
70. Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., Timofte, R.: Plug-and-play image restoration with deep denoiser prior. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **44**(10), 6360–6376 (2021) 9, 11
71. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing (TIP) **26**(7), 3142–3155 (2017) 2, 4
72. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3929–3938 (2017) 2
73. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. IEEE Transactions on Image Processing (TIP) **27**(9), 4608–4622 (2018) 2, 4
74. Zhang, K., Luo, W., Zhong, Y., Ma, L., Stenger, B., Liu, W., Li, H.: Deblurring by realistic blurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2737–2746 (2020) 2
75. Zhang, L., Wu, X., Buades, A., Li, X.: Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. Journal of Electronic imaging **20**(2), 023016–023016 (2011) 10
76. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 3836–3847 (2023) 7

77. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 586–595 (2018) 9, 11
78. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 286–301 (2018) 4
79. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **43**(7), 2480–2495 (2020) 4
80. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al.: Judging llm-as-a-judge with mt-bench and chatbot arena. arXiv preprint arXiv:2306.05685 (2023) 10