Boosting Image Restoration via Priors from Pre-trained Models

Xiaogang Xu[1,2,3] Shu Kong[5,6,7] Tao Hu[3,8] Zhe Liu[1*] Hujun Bao[1,4]
[1] Zhejiang Lab [2] CUHK [3] RealityEdge [4] Zhejiang University [5] University of Macau
[6] Institute of Collaborative Innovation [7] Texas A&M University [8] National University of Singapore
xiaogangxu00@gmail.com, skong@um.edu.mo, yihouxiang@gmail.com
zhe.liu@zhejianglab.com, bao@cad.zju.edu.cn

Abstract

Pre-trained models with large-scale training data, such as CLIP and Stable Diffusion, have demonstrated remarkable performance in various high-level computer vision tasks such as image understanding and generation from language descriptions. Yet, their potential for low-level tasks such as image restoration remains relatively unexplored. In this paper, we explore such models to enhance image restoration. As off-the-shelf features (OSF) from pre-trained models do not directly serve image restoration, we propose to learn an additional lightweight module called Pre-Train-Guided Refinement Module (PTG-RM) to refine restoration results of a target restoration network with OSF. PTG-RM consists of two components, Pre-Train-Guided Spatial-Varying Enhancement (PTG-SVE), and Pre-Train-Guided Channel-Spatial Attention (PTG-CSA). PTG-SVE enables optimal short- and long-range neural operations, while PTG-CSA enhances spatial-channel attention for restoration-related learning. Extensive experiments demonstrate that PTG-RM, with its compact size (<1M parameters), effectively enhances restoration performance of various models across different tasks, including low-light enhancement, deraining, deblurring, and denoising.

1. Introduction

Image restoration plays a vital role in real-world scenarios, aiming to reconstruct high-quality images by eliminating degradations. It has broad applications in various fields, such as denoising [43, 44] and low-light enhancement [41, 42] for improving smartphone-captured photos. While effective restoration networks have been proposed [19, 52], the inherently ill-posed nature of image restoration makes it challenging to achieve significant improvements by merely modifying network structures. Simply increasing model parameters does not guarantee better results, as the model may tend to overfit to the training data.

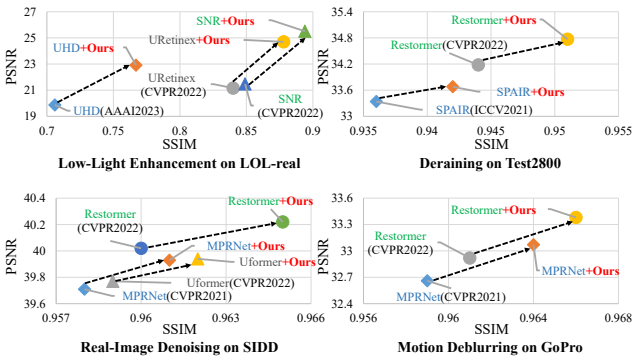


Figure 1. Our method leverages pre-trained models, such as CLIP and Stable Diffusion, and significantly improves image restoration across various tasks. More results on different tasks/models can be seen in experiments. Pre-trained models are involved during the training and not required during the inference.

Restoration performance relies on strong image priors, such as the novel level of denoising [38] or the blur kernel in deblurring [14, 50]. However, estimating these priors is challenging, especially with real-world data. Some approaches utilize physical variables as priors, like depth information [46] and semantic features [1, 36, 41] derived from pre-trained networks. Nevertheless, these physical variables are not robust enough since the dense depth/semantic prediction networks do not have sufficient generalization ability among different scenes in restoration tasks. As a result, employing them requires complex and specific mechanisms, limiting their applicability across various tasks. In this paper, we propose a novel approach that extracts degradation-related information from pre-trained models (with various training objectives) exposed to different degradation during pre-training, all without requiring explicit annotations.

Motivation. Two types of pre-trained models may contain degradation-related information during training: restoration models, and pre-trained models on large-scale data (e.g., CLIP [27], BLIP [16], and BLIP2 [17]). Using

*Corresponding author.

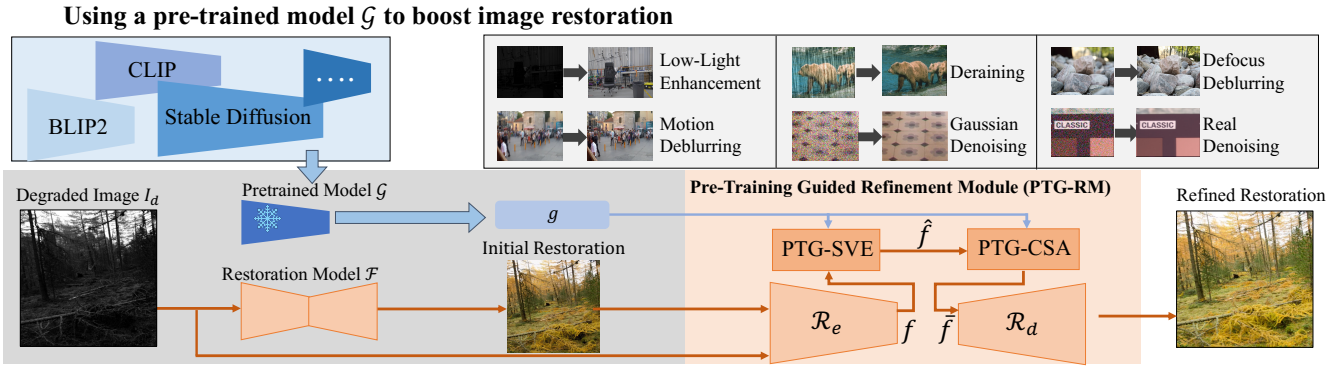


Figure 2. We present a lightweight plugin, *pre-training guided refining module* (PTG-RM), to leverage pre-trained models for enhancing image restoration. The desired prior is the OFS $\mathcal{G}(I_d)$. It has two components, PTG spatial varying enhancement (PTG-SVE), and PTG channel-spatial attention (PTG-CSA). Fig. 3 depicts their details. Our PTG-RM significantly improves restoration in various tasks as listed in the top-right (see quantitative results previewed in Fig. 1).

the former is evident, but models trained with some types of degradation may not effectively help restore images with other types of degradation. Using the latter remains unexplored. CLIP-IQA [33] finds that CLIP features contain degradation-related information and so be useful for image assessment, while no restoration approaches have been proposed yet. Existing pre-trained multi-modality models may have been trained on various degraded images. Presumably, restoration-related annotations are unavailable during pre-training, their resulted features likely contain valuable information for image restoration. The key is to leverage such information to help the target restoration learning. However, the heterogeneity of pre-trained models and restoration models poses difficulties in using the off-the-shelf features extracted from pre-trained models.

Technical novelty. We introduce a novel pre-training guided refinement module (PTG-RM) that leverages off-the-shelf features (OSF) computed by a pre-trained model $\mathcal{G}$ to improve image restoration tasks. The PTG-RM $\mathcal{R}$ is a lightweight plugin (Fig. 2) (additional $\mathcal{R}$ has $<$1M parameters in total). PTG-RM enables us to determine optimal operation ranges and spatial-channel attention, thus facilitating image restoration. It takes as input the initially enhanced image from $\mathcal{F}$, the input image, and its OSF extracted by a pre-trained model. It is trained with $\mathcal{F}$ (using the same loss as $\mathcal{F}$) and adaptively enhances it. PTG-RM $\mathcal{R}$ consists of two components: Pre-Train-Guided Spatial Varying Enhancement (PTG-SVE), and Pre-Train-Guided Channel-Spatial Attention (PTG-CSA).

PTG-SVE employs spatial-varying operations to refine the initially enhanced results differently from region to region. Unlike previous methods [42] that rely on fixed references to determine optimal operation ranges, we establish a spatial-aware learnable mapping for OSF and utilize the mapped features as spatial-wise guidance. This adaptively fuses the features extracted from short- and long-range operations, allowing different regions to be refined appropriatel and yielding more effective enhancement.

Following PTG-SVE, PTG-CSA further enhances the results by formulating effective channel- and spatial-attention with OSF. We note that different areas may require varying degrees of feature correctness via the attention mechanism. Hence, we propose to generate spatial-varying convolution kernels to synthesize the spatial weights. Our approach tailors the attention process to different regions.

Contributions. We make three major contributions.

- We present a novel and general method that leverages pre-trained models to enhance various restoration tasks. Our work opens up possibilities for improving performance across various domains.
- We propose a novel paradigm that leverages pre-trained priors to formulate effective neural operation ranges and attention mechanisms.
- We validate our method through extensive experiments on different datasets, networks, and tasks, and show remarkable improvements over prior methods (cf. Fig. 1).

2. Related Work

Image Priors for Restoration. Different restoration tasks demand distinct image priors, such as noise levels for denoising and blurring kernels for deblurring. Due to the ill-posed nature of restoration, estimating priors is difficult. In real-world scenarios, these priors are typically intertwined, adding further complexity to the restoration process. Recent literature introduces several methods to improve restoration by leveraging multi-modal maps as unified priors. These methods predominantly rely on pre-computed physical multi-modal maps. For instance, SKF [41] uses semantic maps to optimize the feature space for low-light enhancement. SMG [46] employs a generative framework

to integrate edge, depth, and semantic information, enhancing the initial appearance modeling for low-light scenarios. Additionally, some approaches use Near-Infrared (NIR) information to refine imaging results [12, 32]. These priors are also applied to other restoration tasks, such as image denoising [20] and deraining [18]. However, aligning these priors with the input image can be challenging, and errors in the priors may adversely impact performance. Different from existing works, we propose to leverage pre-trained models as priors to enhance image restoration.

Pre-Trained Models for Downstream Tasks. Recently, a series of pre-trained models with large-scale training datasets have emerged, particularly in the form of multi-modal models such as CLIP [27], BLIP [16], and BLIP2 [17]. The feature space learned by these models offers rich knowledge that can benefit various tasks. While previous work has demonstrated the effectiveness of CLIP in high-level tasks like zero-shot classification [6, 53], image editing [4, 25], open-world segmentation [39, 60], and 3D classification [47, 59], its potential for aiding low-level restoration tasks remains unexplored. Only the capability of employing such for image quality assessment, as demonstrated in CLIP-IQA, has been explored. We propose a general framework to leverage pre-trained models to improve various restoration tasks.

3. Methods

Background. Let I_d represent a degraded image, and I_c denote the corresponding ground-truth (without degradation). A restoration network $\mathcal{F}$ produces restored image $\hat{I}_c = \mathcal{F}(I_d)$. Despite the existence of various effective network structures $\mathcal{F}$ that have been proposed, there are current upper bounds in these tasks. Breaking through these bounds often requires designing more complex networks and training strategies, which can be arduous. Additionally, innovations in network architecture or training strategies for one task might not translate to another. While different priors g have been introduced into the restoration process, including image and physical priors, estimating these priors is difficult.

Motivation. We hypothesize that the prior g can be effectively represented as the feature extracted from various pre-trained models $\mathcal{G}$, as $g = \mathcal{G}(I_d)$. Note that $\mathcal{G}$ is typically not trained with restoration targets but might have been exposed to images with diverse degradations. So it is likely to learn useful information to help image restoration. We propose a novel approach that uses g to improve the initial restoration by $\mathcal{F}$, even if these networks have already reached their current upper bounds.

Challenge. Using g to assist $\mathcal{F}$ is non-trivial. Primarily, the feature g is not inherently aligned with the restoration tasks because they might represent different aspects. For instance, features from CLIP focus more on semantic information, making direct alignment to restoration challenging. Moreover, these priors exhibit varying shapes, such as the one-dimensional (1D) features from the CLIP model, while the features in $\mathcal{F}$ are typically 2D. To reconcile the discrepancies in both representation and shape, we propose a refinement module $\mathcal{R}$ to refine the initial restoration by $\mathcal{F}$. This eliminates the need to align g to distinct features of $\mathcal{F}$ and allows for a unified 1D representation for g. Furthermore, we introduce a novel approach to utilize g to formulate optimal neural operating ranges via an effective attention mechanism in $\mathcal{R}$. This implicitly distills restoration-related information, effectively boosting the final performance.

3.1. Overview of Refinement Module

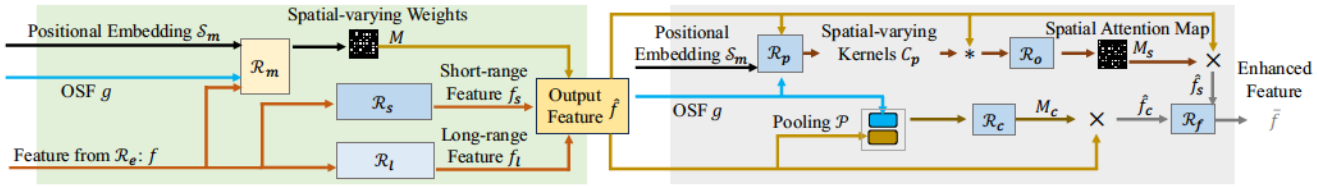
Fig. 2 depicts the restoration pipeline using our method. Given an input image I_d, we have an initial restoration result as $\hat{I}_c = \mathcal{F}(I_d)$. We aim to refine the result using the proposed pre-training guided refinement module (PTG-RM) $\mathcal{R}$, resulting in $\bar{I}_c = \mathcal{R}(\hat{I}_c, I_d, g)$. The key of this approach is to distill restoration-related information from the prior g.

$\mathcal{R}$ is a simple encoder-decoder structure. The encoder and decoder of $\mathcal{R}$ are denoted as $\mathcal{R}_e$ and $\mathcal{R}_d$, respectively. To ensure lightweight implementation, distillation occurs in the latent space, avoiding the need to align g with restoration-related features. The latent feature f is derived through a comparison between the initial enhanced results and the original input images, given as $f = \mathcal{R}_e(\hat{I}_c \oplus I_d)$, where $\oplus$ denotes the concatenation operation. The resulting f is in $\mathbb{R}^{h \times w \times c}$, with h, w, and c representing feature height, width, and channel number, respectively. The priors are used in further learning the latent feature as $\bar{f} = \mathcal{C}(\mathcal{A}(f, g), g)$, where $\mathcal{A}$ and $\mathcal{C}$ represent the Pre-Train-Guided Spatial-Varying Enhancement (PTG-SVE) and Pre-Train-Guided Channel-Spatial Attention (PTG-CSA) modules, respectively. The final enhancement is obtained from the decoder as $[I_m, I_r] = \mathcal{R}_d(\hat{f})$, comprising two components. The first component, I_m, represents the correction mask used to mitigate errors in the initial enhancement results. The second component, I_r, is the residual refinement that addresses artifacts and adds additional details. The final result is denoted as

$$\bar{I}_c = I_d + (\hat{I}_c - I_d) \times I_m + I_r. \tag{1}$$

3.2. Pre-Train-Guided Spatial-Varying Operations

In PTG-SVE, we argue that $g = \mathcal{G}(I_d)$ may contain information reflecting the pixel-level image quality of I_d. For areas with poor quality, long-range operations are used to capture non-local features, while regions with relatively good quality prioritize local features for accurate restoration.



(a) Pre-Train-Guided Spatial-Varying Enhancement $\mathcal{A}$ (b) Pre-Train-Guided Channel-Spatial Attention $\mathcal{C}$

Figure 3. The pipeline of PTG-SVE and PTG-CSA. In PTG-SVE, we use the learnable spatial embedding $\mathcal{S}_m$, OSF g, and input feature f to adaptively formulate spatial weights (M, Eq. 2) for fusing short- and long-range processed features (f_s and f_l) via operations $\mathcal{R}_s$ and $\mathcal{R}_l$, yielding $\hat{f}$ (Eq. 3). In PTG-CSA, OSF g conditions channel attention M_c for $\hat{f}$ through $\mathcal{R}_c$ (Eq. 4). Additionally, g combines with learnable spatial representation $\mathcal{S}_c$ and $\hat{f}$ to generate spatial attention map M_s, using spatial-wise convolutions C_p (obtained via $\mathcal{R}_p$) to derive $\hat{\mathcal{M}}_s$ that is further processed with $\mathcal{R}_o$ (Eqs. 5 and 6). Channel- and spatial-attention outputs ($\hat{f}_c$ and $\hat{f}_s$) merge via $\mathcal{R}_f$ to enhance feature $\bar{f}$ (Eq. 7).

In Fig. 3, the primary objective is to predict the optimal neural operation range for each location of the feature map f, which we refer to as the "range score map", denoted as M. To ensure a general $\mathcal{R}$ with unified 1D priors g from various models, we propose adding location-aware embeddings for the priors, thereby adaptively discovering quality information for different pixels. Let $S = \{(x, y)|x \in [1, w], y \in [1, h]\}$ represent the 2D coordinate map with dimensions $h \times w \times 2$. We use a position embedding module $\mathcal{P}$ to generate spatial representation, denoted as $\mathcal{S}_m = \mathcal{P}(S)$, where $\mathcal{S} \in \mathbb{R}^{h \times w \times c}$. Furthermore, to determine the admired neural operation range for each location of f, we use a learnable mapping function $\mathcal{T}_m$ to transform the priors to another space that can more effectively decide the optimal range. To obtain M, we use a range-learning module $\mathcal{R}_m$, which takes the encoder's feature f, the pre-trained prior g, and the spatial representation $\mathcal{S}_m$ as inputs. The procedure is denoted as

$$M = \mathcal{R}_m(f \oplus \mathcal{T}_m(g) \oplus \mathcal{S}_m). \tag{2}$$

Following [42], we use CNN for the short-range operation, denoted as $\mathcal{R}_s$, and transformer for the long-range operation, represented as $\mathcal{R}_l$. Specifically, we employ the Restormer backbone for $\mathcal{R}_l$ and ResNet for $\mathcal{R}_s$. Suppose the features after the short- and long-range operation are f_s and f_l, respectively. We can obtain the refined feature $\hat{f}$ as

$$f_s = \mathcal{R}_s(f),\ f_l = \mathcal{R}_l(f),\ \hat{f} = M \times f_s + (1 - M) \times f_l. \tag{3}$$

The previous approach [42] relies on pre-computed SNR values, which may not always be accurate and can fail to enhance results, especially when the initial results from $\mathcal{F}$ have reached their upper bound. In contrast, our score range map is learned online based on the input image, restoration-related priors, and explicit spatial features that are learnable. This flexibility allows us to handle various situations, resulting in better performance and generalization (as demonstrated in the ablation study).

3.3. Pre-Train-Guided Attention

As shown in Fig. 3, we further introduce a lightweight component that utilizes pre-trained priors g to create an effective attention mechanism in $\mathcal{R}$. Optimizing the feature attention in $\mathcal{R}$ is crucial for effectively identifying helpful features to enhance the initial results $\hat{f}_c$. This involves both spatial-level and channel-level attentions. The hidden restoration-related information in g can be discovered by using g to improve the restoration features in $\mathcal{R}$ conditioned on them.

We begin by formulating the attention computation at the channel level. We introduce a mapping function $\mathcal{T}_c$ to transform g into the attention-prediction space, and utilize the channel attention computation module $\mathcal{R}_c$. The formulation of the channel attention is

$$M_c = \mathcal{R}_c(\mathcal{O}(\hat{f}) \oplus \mathcal{T}_c(g)),\ \hat{f}_c = \hat{f} \times M_c, \tag{4}$$

where $\mathcal{O}$ is the pooling operation, and $\mathcal{M}_c \in \mathbb{R}^c$.

As for the spatial-attention computation, we utilize the 1D pre-trained prior g to predict location-wise attention based on the feature distribution of each location in $\hat{f}$. Simply using the spatial location information, as shown in Eq. 2, results in each pixel's feature considering a similar condition for neighboring features, limiting the elimination of spatial artifacts. Therefore, we propose an alternative strategy by predicting the neural operation parameters for each location, optimizing the spatial attention based on the varying location-wise feature distribution. We denote the spatial attention computation module as $\mathcal{R}_p$, and first formulate the location-wise convolution map, as

$$\mathcal{C}_p = \mathcal{R}_p(\hat{f}, \mathcal{T}_c(g), \mathcal{S}_c), \tag{5}$$

where the obtained convolution map $\mathcal{C}_p \in \mathbb{R}^{h \times w \times (k_h \times k_w \times c)}$, k_h and k_w are the convolution kernel size, and $\mathcal{S}_c$ is another learnable position embedding here. The obtained convolution maps can be utilized to optimize the feature, and spatial attention can be obtained as

$$\hat{M}_s = \hat{f} * \mathcal{C}_p,\ M_s = \mathcal{R}_o(\hat{M}_s), \tag{6}$$

Datasets	Methods	Original		+Ours-c		+Ours-b		+Ours-s		+Ours-r		+Ours-f	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
LOL	UHD	19.87	0.706	22.91 (**+3.04**)	0.767 (**+6.1**)	21.83 (**+1.96**)	0.732 (**+2.6**)	22.35 (**+2.48**)	0.758 (**+5.2**)	21.71 (**+1.84**)	0.737 (**+3.1**)	22.74 (**+2.87**)	0.764 (**+5.8**)
	URetinex	21.16	0.840	24.70 (**+3.54**)	0.878 (**+3.8**)	23.57 (**+2.41**)	0.869 (**+2.9**)	24.23 (**+3.07**)	0.866 (**+2.6**)	23.99 (**+2.83**)	0.862 (**+2.2**)	24.56 (**+3.40**)	0.870 (**+3.0**)
	SNR	21.48	0.849	25.50 (**+4.02**)	0.892 (**+4.3**)	25.61 (**+4.13**)	0.891 (**+4.2**)	25.19 (**+3.71**)	0.874 (**+2.5**)	25.24 (**+3.76**)	0.887 (**+3.8**)	24.90 (**+3.42**)	0.888 (**+3.9**)
SID	UHD	20.46	0.614	20.99 (**+0.53**)	0.616 (**+0.2**)	21.06 (**+0.60**)	0.619 (**+0.5**)	22.34 (**+1.88**)	0.625 (**+1.1**)	21.11 (**+0.65**)	0.618 (**+0.4**)	21.08 (**+0.62**)	0.619 (**+0.5**)
	URetinex	21.56	0.619	22.34 (**+0.78**)	0.623 (**+0.4**)	22.02 (**+0.46**)	0.621 (**+0.2**)	22.21 (**+0.65**)	0.623 (**+0.4**)	22.17 (**+0.61**)	0.625 (**+0.6**)	22.40 (**+0.84**)	0.626 (**+0.7**)
	SNR	22.87	0.625	23.34 (**+0.47**)	0.630 (**+0.5**)	23.15 (**+0.28**)	0.627 (**+0.2**)	23.08 (**+0.21**)	0.631 (**+0.6**)	23.06 (**+0.19**)	0.632 (**+0.7**)	23.17 (**+0.30**)	0.636 (**+1.1**)

Table 1. Comparisons on LOL-real and SID dataset. $-c$, $-b$, $-s$, and $-r$ refer to using CLIP, BLIP2, Stable Diffusion, and restoration models trained on SDSD, respectively. $-f$ denotes applying refinement on the features of $\mathcal{F}$. (+) indicates improvements for PSNR and $\text{SSIM}_{(\text{x100})}$.

Methods	SNR	+SKF	+SMG	+SMG(dep)	+Ours-c
PSNR	21.48	23.05	24.84	24.12	**25.50**
SSIM	0.849	0.853	0.880	0.851	**0.892**
Methods	URetinex	+SKF	+SMG	+SMG(dep)	+Ours-c
PSNR	21.16	23.51	23.74	23.25	**24.70**
SSIM	0.840	0.856	0.852	0.849	**0.878**
+Params	0	2.15M	16.76M	16.76M	0.67M

Table 2. Quantitative comparison on the LOL-real dataset. +Params means the additional parameter number compared with original $\mathcal{F}$.

where $*$ is the convolution operation for each location, and $\mathcal{R}_o$ is another learnable operation which mapps the feature channel c to 1, eliminating the influence from the channel-level dependency. Further, the feature after spatial attention can be described as $\hat{f}_s = \hat{f} \times M_s$.

The features after spatial and channel attentions can be merged via a fusion module as

$$\bar{f} = \mathcal{R}_f(\hat{f}_c \oplus \hat{f}_s), \tag{7}$$

where $\mathcal{R}_f$ denotes the fusin module. The obtained feature $\bar{f}$ can be processed via a decoder $\mathcal{R}_d$ to obtain the residual refinement I_r and the mask I_m as indicated in Eq. 1.

3.4. Loss Function

Our designed $\mathcal{R}$ can be jointly trained with the model $\mathcal{F}$. Suppose the paired ground truth for the input image I_d is $\mathcal{I}_c$, and the loss function for the model $\mathcal{F}$ is denoted as $\mathcal{L}_g(\hat{I}_c, \mathcal{I}_c)$ (is usually the reconstruction loss in the pixel level or perceptual loss, and can also be the unsupervised loss), then the loss function for the refinement module can be written as $\mathcal{L}_g(\bar{I}_c, \mathcal{I}_c)$. In summary, the overall loss is

$$\mathcal{L}_g(\hat{I}_c, \mathcal{I}_c) + \lambda_1 \mathcal{L}_g(\bar{I}_c, \mathcal{I}_c), \tag{8}$$

where λ_1 is the loss weight and remains robust across various tasks and networks (in our experiments, $\lambda 1$ is always set as 1).

4. Experiments

We first introduce tasks and datasets used in experiments, followed by a detailed analysis of our methd using low-light image enhancement as an example. We also demonstrate the effectiveness of our method on other tasks.

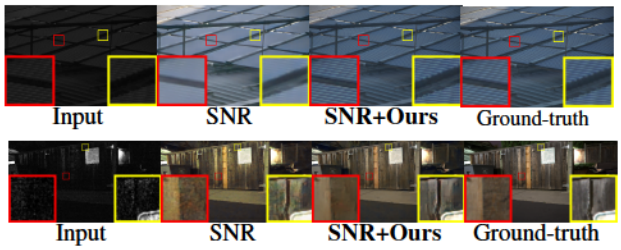


Figure 4. Comparisons on LOL-real (top) and SID (bottom). Results with "Ours" have less noise and clearer visibility.

4.1. Tasks and Datasets

For low-light enhancement, we use the SID [5] and LOL-real [49] datasets. For deraining, we use the Rain13K [52] dataset for training and test on Rain100H [48], Rain100L [48], Test100 [55], Test1200 [54], and Test2800 [8] datasets. For gaussian denoising, we use two settings: synthetic noise on Set12 [56], BSD68 [23], CBSD68 [23], Kodak [7], McMaster [58], and Urban100 [10]; and real-world denoising on SIDD [2]. For single-image motion deblurring, we use the GoPro [24] dataset for training and evaluate on synthetic datasets (GoPro [24], HIDE [30]) and real-world datasets (RealBlur-R [28], RealBlur-J [28]). For defocus deblurring, we use the DPDD [3] training data and test on the EBDB [13] and JNB [31] datasets.

4.2. Low-light Image Enhancement

Comparison. We choose current SOTA low-light image enhancement methods as the baselines (UHD [35], URetinex [40], SNR [42]), and apply our refinement module for these baselines to see if their performance can be improved. The priors are chosen from the CLIP [27], BLIP2 [17], Stable Diffusion [29], and pre-trained restoration models (trained on another dataset, as SDSD [34, 45]). We denote these results as $-c$, $-b$, $-s$, and $-r$, respectively. In Table 1, we observe that combining these priors with our refinement module significantly improves the performance of the baselines. Additionally, Fig. 4 provides visual comparisons.

Moreover, we conducted an experiment by adding the refinement module to the intermediate layer of $\mathcal{F}$, refin-

	LOL-real				SID			
	URetinex		SNR		URetinex		SNR	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
w/o SP, with CA and SA	23.45	0.868	24.25	0.886	21.98	0.619	23.02	0.620
with SP, w/o CA, with SA	22.10	0.856	24.05	0.875	22.05	**0.623**	22.93	0.624
with SP and CA, w/o SA	23.76	0.850	23.86	0.879	21.92	0.620	23.07	0.621
Large $\mathcal{R}$ w/o SP/CA/SA	22.74	0.857	24.51	0.881	22.06	0.621	23.04	0.627
w/o Position Embedding $\mathcal{S}$	23.66	0.843	24.13	0.874	22.13	0.620	22.92	0.622
SNR Value as Mask	22.66	0.855	24.77	0.887	22.01	0.617	22.94	0.627
Use 1D Priors via Con.	23.01	0.853	23.83	0.878	22.07	0.622	22.93	0.628
Use 2D Priors via Con.	22.68	0.862	24.11	0.880	22.08	0.618	23.06	0.625
Full Setting	**24.70**	**0.878**	**25.50**	**0.892**	**22.34**	**0.623**	**23.34**	**0.630**

Table 3. Ablation study results. We adopt CLIP as the pre-trained model. "SP" denote PTG-SVE, "CA" and "SA" denote spatial- and channel attentions in PTG-CSA. Con. means Concatenation.

Datasets	LOL-real			SID		
Methods	ZeroDCE	RUAS	SCI	ZeroDCE	RUAS	SCI
PSNR	18.06	18.37	20.28	18.08	18.44	19.09
SSIM	0.580	0.723	0.752	0.576	0.581	0.585
Methods	+Ours-c	+Ours-c	+Ours-c	+Ours-c	+Ours-c	+Ours-c
PSNR	**18.79**	**19.53**	**21.62**	**18.65**	**18.93**	**19.61**
SSIM	**0.614**	**0.747**	**0.781**	**0.593**	**0.590**	**0.598**

Table 4. Quantitative comparison on the LOL-real and SID dataset for unsupervised methods. We adopt CLIP as the pre-trained model here.

Method	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
	Test100		**Rain100H**		**Rain100L**	
SPAIR	30.35	0.909	30.95	0.892	36.93	0.969
SPAIR+Ours-c	**30.62**	**0.917**	**31.20**	**0.901**	**37.26**	**0.973**
Restormer	32.00	0.923	31.46	0.904	38.99	0.978
Restormer+Ours-c	**32.30**	**0.934**	**31.77**	**0.913**	**39.27**	**0.985**
	Test2800		**Test1200**		**Average**	
SPAIR	33.34	0.936	33.04	0.922	32.91	0.926
SPAIR+Ours-c	**33.58**	**0.942**	**33.35**	**0.924**	**33.16**	**0.932**
Restormer	34.18	0.944	33.19	0.926	33.96	0.935
Restormer+Ours-c	**34.47**	**0.951**	**33.48**	**0.929**	**34.24**	**0.943**

Table 5. Image deraining results.

ing features of the target model. The refinement module is added to the deepest feature layer for efficiency, producing the residual feature map and the mask information for refinement. These results are denoted as $-f$. The improvement achieved by this operation is also evident as displayed in Table 1.

Comparison with Other Priors. Some works, such as SKF [41] and SMG [46], utilize additional information like semantic maps, edge maps, and depth maps to enhance low-light image enhancement results. However, these methods require supervision with paired multi-modal information, whereas our method does not. Additionally, as shown in Table 2, our approach achieves better performance improvement for a given target model. Notably, the improvements achieved by other methods are based on large additional parameters, while our approach only uses a lightweight refinement module $<$ 1M.

Ablation Study: Ablation of Different Components. We first set experiments by deleting different components from our framework, including PTG-SVE (abbreviated as "SP"), and spatial-channel attentions with priors that are abbreviated as "CA" and "SA", respectively. As shown in Table 3, deleting any component will lead to a performance drop.

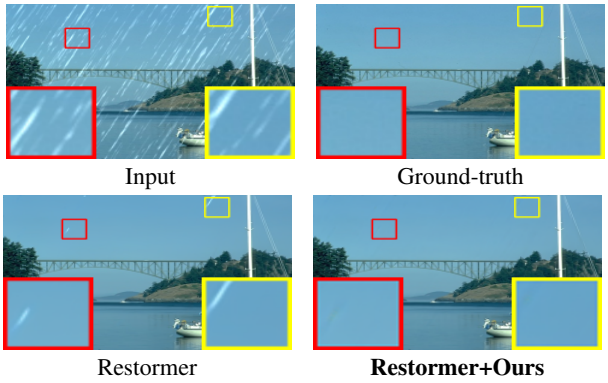


Figure 5. Visual comparison on Rain100H showing the effects of our strategy.

Method	**GoPro**		**HIDE**		**RealBlur-R**		**RealBlur-J**	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MPRNet	32.66	0.959	30.96	0.939	35.99	0.952	28.70	0.873
MPRNet+Ours-c	**32.87**	**0.964**	**31.19**	**0.943**	**36.25**	**0.960**	**28.98**	**0.881**
Restormer	32.92	0.961	31.22	0.942	36.19	0.957	28.96	0.879
Restormer+Ours-c	**33.18**	**0.966**	**31.51**	**0.950**	**36.47**	**0.962**	**29.21**	**0.883**

Table 6. Single-image motion deblurring results.

We conduct experiments without SP, CA, or SA to analyze whether additional parameters or priors take a prominent role. The short-range and long-range results are fused via a simple sum, and the spatial-channel attention is conducted using only the features themselves. Additionally, we increase the feature channel number fourfold to add more parameters. The results, denoted as "Large $\mathcal{R}$ w/o SP/CA/SA" in Table 3, are still lower than our full setting, indicating the effectiveness of our proposed approach over simply increasing parameters.

In addition, we perform an experiment by removing the learnable position embeddings $\mathcal{S}_m$ and $\mathcal{S}_c$, denoted as "w/o Position Embedding for Priors" in Table 3. This comparison highlights the importance of using spatial-aware representations for the pre-trained features.

Ablation Study: SNR Value as Mask. In comparison to previous methods that directly use the SNR value as the mask to fuse the short- and long-range results, our approach utilizes pre-trained priors to automatically discover restoration-related information and formulate the fusion mask adaptively. In this ablation study, we demonstrate that our strategy outperforms the direct SNR-based approach, as shown in Table 3.

Ablation Study: Alternatives of Using Priors. In this study, we demonstrate the difficulty of directly aligning priors to the restoration features. We conduct an experiment where the priors are concatenated with the features in the refinement module to implement different components. However, the improvement obtained with this direct approach

Figure 6. Visual comparison on HIDE.

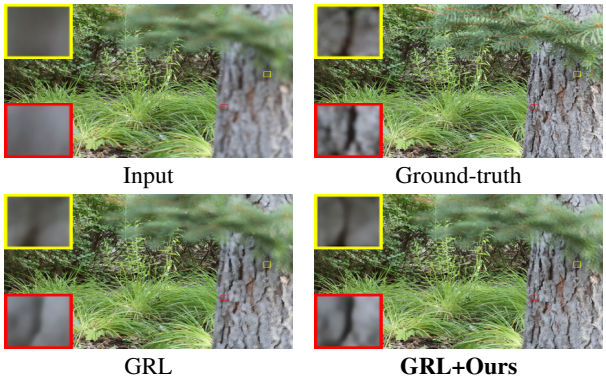


Figure 7. Visual comparison on single-image defocus deblurring.

is not as significant as our proposed method, as shown in Table 3. This is because the different features are heterogeneous with the restoration features, even when the priors are adopted as 2D feature maps. This study highlights the importance of our novel strategy of employing these priors.

$\mathcal{R}$ for Unsupervised Approach. Different from existing refinement methods that need supervision for learning the additional features (e.g., SKF needs the semantic ground truth of the normal-light data, SMG needs the depth and edge information of the normal-light data), our approach does not require the feature of the normal-light data during both training and inference. We only need the feature that is extracted from I_d with the pre-trained model $\mathcal{G}$ during the training. Also, the loss function for training the refinement module can be set the same as that of the target model. Thus, the unsupervised training of the target model can also be adopted in our framework. As shown in Table 4, our method can successfully improve the performance of various unsupervised low-light image enhancement methods with different unsupervised loss terms, including EnGAN [11], ZeroDCE [9], RUAS [21], and SCI [22].

4.3. Other Restoration Tasks

In this section, we conduct experiments using CLIP as the pre-trained model ($-c$). CLIP is chosen for its efficiency and convenience compared to other pre-trained models.

Method	Indoor Scenes			Outdoor Scenes			Combined		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
IFAN$_S$	28.11	0.861	0.179	22.76	0.720	0.254	25.37	0.789	0.217
IFAN$_S$+Ours-c	**28.32**	**0.870**	**0.171**	**23.08**	**0.727**	**0.248**	**25.72**	**0.795**	**0.213**
Restormer$_S$	28.87	0.882	0.145	23.24	0.743	0.209	25.98	0.811	0.178
Restormer$_S$+Ours-c	**29.17**	**0.890**	**0.141**	**23.43**	**0.749**	**0.206**	**26.13**	**0.816**	**0.165**
GRL$_S$-B	29.06	0.886	0.139	23.45	0.761	0.196	26.18	0.822	0.168
GRL$_S$-B +Ours-c	**29.30**	**0.894**	**0.133**	**23.67**	**0.768**	**0.189**	**26.45**	**0.828**	**0.161**
IFAN$_D$	28.66	0.868	0.172	23.46	0.743	0.240	25.99	0.804	0.207
IFAN$_D$+Ours-c	**28.94**	**0.875**	**0.167**	**23.70**	**0.748**	**0.235**	**26.20**	**0.811**	**0.203**
Restormer$_D$	29.48	0.895	0.134	23.97	0.773	0.175	26.66	0.833	0.155
Restormer$_D$+Ours-c	**29.79**	**0.902**	**0.131**	**24.23**	**0.778**	**0.155**	**26.89**	**0.840**	**0.153**
GRL$_D$-B	29.83	0.903	0.114	24.39	0.795	0.150	27.04	0.847	0.133
GRL$_D$-B+Ours-c	**29.96**	**0.911**	**0.110**	**24.62**	**0.803**	**0.145**	**27.27**	**0.855**	**0.128**

Table 7. Defocus deblurring comparisons on the DPDD testset (containing 37 indoor and 39 outdoor scenes). **S:** single-image defocus deblurring. **D:** dual-pixel defocus deblurring.

Method	Set12			BSD68			Urban100		
	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50
DRUNet	33.25	30.94	27.90	31.91	29.48	26.59	33.44	31.11	27.96
DRUNet+Ours-c	**33.51**	**31.18**	**28.27**	**32.20**	**29.73**	**26.84**	**33.65**	**31.34**	**28.16**
Restormer	33.35	31.04	28.01	31.95	29.51	26.62	33.67	31.39	28.33
Restormer+Ours-c	**33.57**	**31.28**	**28.36**	**32.11**	**29.78**	**26.91**	**33.96**	**31.67**	**28.58**
Restormer	33.42	31.08	28.00	31.96	29.52	26.62	33.79	31.46	28.29
Restormer+Ours-c	**33.70**	**31.29**	**28.35**	**32.24**	**29.81**	**26.86**	**33.97**	**31.73**	**28.58**
GRL-B	33.47	31.12	28.03	32.00	29.54	26.60	34.09	31.80	28.59
GRL-B+Ours-c	**33.74**	**31.30**	**28.37**	**32.29**	**29.76**	**26.91**	**34.22**	**31.95**	**28.74**

Table 8. Gaussian grayscale image denoising comparisons. Top super rows: learning a single model to handle various noise levels. Bottom super rows: training a separate model for each noise level.

Deraining. For deraining tasks, we use SOTA methods such as SPAIR [26] and Restormer [52] as baselines. We compute PSNR/SSIM values using the Y channel in the YCbCr color space, similar to existing methods. Table 5 demonstrates that our approach improves the performance of these existing methods and consistently achieves significant performance gains across all five datasets. The qualitative comparison results are shown in Fig. 5.

Motion Deblurring. We analyze our approach for deblurring tasks on synthetic datasets (GoPro, HIDE) and real-world datasets (RealBlur-R, RealBlur-J). The baselines include MPRNet [51] and Restormer [52]. Table 6 demonstrates that our approach improves the performance of all these methods on all four benchmark datasets. Although the enhanced network is trained only on the GoPro dataset, it shows more robust generalization to other datasets. Qualitative comparisons are shown in Fig. 6, further supporting our claim.

Defocus Deblurring. Table 7 presents the image fidelity scores of SOTA approaches on the DPDD dataset [3], including IFAN [15], Restormer [52], and GRL [19]. Our refinement module achieves significant performance improvement for these SOTA schemes in both single-image and dual-pixel defocus deblurring settings across all scene categories. The qualitative results are depicted in Fig. 7.

Gaussian Denoising. We conduct denoising experiments on synthetic benchmark datasets with additive white Gaussian noise. We choose DRUNet [57], Restormer [52], and

Method	CBSD68			Kodak24			McMaster			Urban100		
	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50
DRUNet	34.30	31.69	28.51	35.31	32.89	29.86	35.40	33.14	30.08	34.81	32.60	29.61
+Ours-c	**34.54**	**31.97**	**28.76**	**35.58**	**33.15**	**29.97**	**35.71**	**33.50**	**30.25**	**35.10**	**32.82**	**29.83**
Restormer	34.39	31.78	28.59	35.44	33.02	30.00	35.55	33.31	30.29	35.06	32.91	30.02
+Ours-c	**34.63**	**32.04**	**28.88**	**35.65**	**33.26**	**30.15**	**35.86**	**33.64**	**30.63**	**35.26**	**33.22**	**30.21**
Restormer	34.40	31.79	28.60	35.47	33.04	30.01	35.61	33.34	30.30	35.13	32.96	30.02
+Ours-c	**34.76**	**32.05**	**28.94**	**35.72**	**33.27**	**30.21**	**35.80**	**33.63**	**30.55**	**35.32**	**33.14**	**30.27**
GRL-B	34.45	31.82	28.62	35.43	33.02	29.93	35.73	33.46	30.36	35.54	33.35	30.46
+Ours-c	**34.73**	**32.07**	**28.90**	**35.71**	**33.24**	**30.18**	**35.96**	**33.75**	**30.62**	**35.70**	**33.57**	**30.64**

Table 9. Gaussian color image denoising. Equivalent notation meanings (top and bottom rows) as those in Table 8.

Dataset	Method	MPRNet	MPRNet + Ours-c	Uformer	Uformer + Ours-c	Restormer	Restormer + Ours-c
SIDD	PSNR ↑	39.71	**39.93**	39.77	**39.94**	40.02	**40.22**
	SSIM ↑	0.958	**0.961**	0.959	**0.962**	0.960	**0.965**

Table 10. Real image denoising on the SIDD dataset.

GRL [19] as baselines, which are SOTA approaches in denoising. Tables 8 and 9 present PSNR scores of different approaches on grayscale and color image denoising, respectively, for noise levels of 15, 25, and 50. We evaluate two experimental settings: (1) learning a single model to handle various noise levels and (2) learning separate models for each noise level. Our method achieves significant performance enhancement for all these methods under both experimental settings on different datasets and noise levels. The visual results are shown in Fig. 8, showing the effectiveness of our strategy.

Real Denoising. We also conduct denoising experiments on the real-world SIDD dataset, with MPRNet [51], Uformer [37], and Restormer [52] as baselines. Table 10 demonstrates that our refinement method improves both PSNR and SSIM metrics. Notably, on the SIDD dataset, our refinement enables the SOTA approach Restormer to achieve a PSNR surpassing 40.2 dB. The visual comparison is shown in Fig. 8.

User Study. Furthermore, we conducted a large-scale user study with an A/B test strategy involving 80 participants. Each participant is asked to simultaneously see two restored results, i.e., baseline and baseline+ours, and gauge which one is better. As shown in Fig. 9, the results combined with our strategy are more preferred by the participants.

5. Conclusion

In this work, we explore the utilization of features from a pre-trained model to enhance the performance of a restoration model. By unifying the shapes of the pre-trained features, we introduce a novel refinement module PTG-RM that employs PTG-SVE and PTG-CSA mechanisms. Unlike existing strategies, we focus on formulating optimal operation ranges and attention strategies guided by the pre-trained features. The extensive experiments conducted on various tasks, datasets, and networks demonstrate the effectiveness and generalization ability of our approach. We

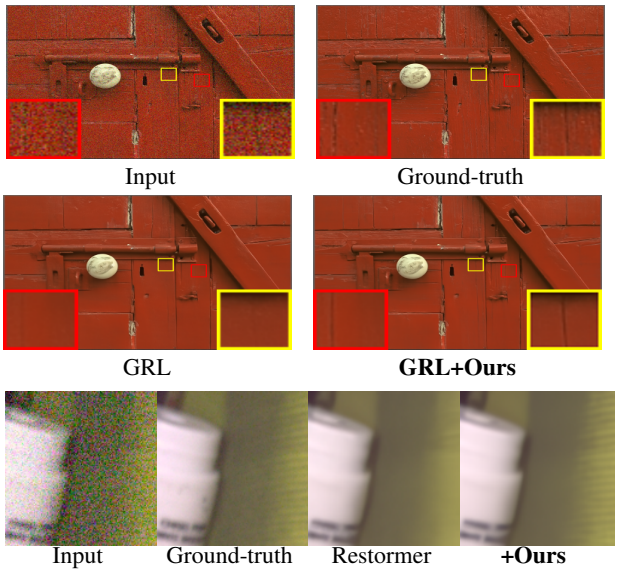


Figure 8. Visual comparisons on Kodak (top) and SIDD (bottom).

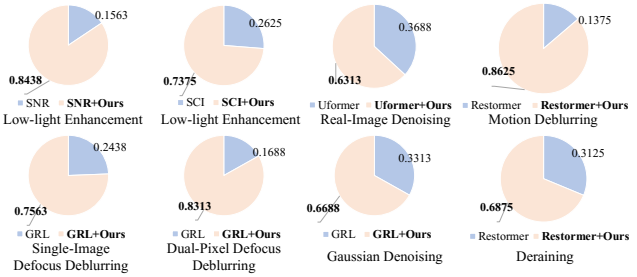


Figure 9. The user study results show that our strategy can effectively improve the performance of restoration approaches in terms of human subjective evaluation.

believe that our proposed principle of discovering hidden useful information in pre-trained models can be applicable to other domains as well.

Limitation and Future Work. While our proposed strategy has exhibited significant effects in enhancing the performance of diverse restoration networks across various architectures with its lightweight module, the extent of improvement appears to vary across different experiments. Some instances showcase noticeable enhancement, while others do not. Such differences correlate with the capacity of the target network and the difficulty/complexity of the target task. In future endeavors, we intend to delve into more effective approaches that specifically aid target restoration tasks. We aim to employ a tailored distillation framework to derive refined restoration feature priors, ultimately making significant strides beyond existing upper boundaries. We also aim to develop corresponding technical products.

Acknowledgements. This work is supported by the Natural Science Foundation of Zhejiang Pvovince, China, under No. LD24F020002. SK is partially supported by University of Macau (SRG2023-00044-FST).

References

[1] Andreas Aakerberg, Anders S Johansen, Kamal Nasrollahi, and Thomas B Moeslund. Semantic segmentation guided real-world super-resolution. In *WACV*, 2022. 1

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 5

[3] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *ECCV*, 2020. 5, 7

[4] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022. 3

[5] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *CVPR*, 2018. 5

[6] Sepideh Esmaeilpour, Bing Liu, Eric Robertson, and Lei Shu. Zero-shot out-of-distribution detection based on the pre-trained model clip. In *AAAI*, 2022. 3

[7] Rich Franzen. Kodak lossless true color image suite. http://r0k.us/graphics/kodak/, 1999. Online accessed 24 Oct 2021. 5

[8] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *CVPR*, 2017. 5

[9] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. In *CVPR*, 2020. 7

[10] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 5

[11] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *TIP*, 2021. 7

[12] Shuangping Jin, Bingbing Yu, Minhao Jing, Yi Zhou, Jiajun Liang, and Renhe Ji. Darkvisionnet: Low-light imaging via rgb-nir fusion with deep inconsistency prior. In *AAAI*, 2022. 3

[13] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *TIP*, 2017. 5

[14] Shu Kong and Charless Fowlkes. Image reconstruction with predictive filter flow. *arXiv preprint arXiv:1811.11482*, 2018. 1

[15] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *CVPR*, 2021. 7

[16] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 1, 3

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint*, 2023. 1, 3, 5

[18] Yi Li, Yi Chang, Changfeng Yu, and Luxin Yan. Close the loop: a unified bottom-up and top-down paradigm for joint image deraining and segmentation. In *AAAI*, 2022. 3

[19] Yawei Li, Yuchen Fan, Xiaoyu Xiang, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. Efficient and explicit modelling of image hierarchies for image restoration. In *CVPR*, 2023. 1, 7, 8

[20] Ding Liu, Bihan Wen, Xianming Liu, Zhangyang Wang, and Thomas S Huang. When image denoising meets high-level vision tasks: A deep learning approach. In *IJCAI*, 2018. 3

[21] Risheng Liu, Long Ma, Jiaao Zhang, Xin Fan, and Zhongxuan Luo. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *CVPR*, 2021. 7

[22] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *CVPR*, 2022. 7

[23] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 5

[24] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 5

[25] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 3

[26] Kuldeep Purohit, Maitreya Suin, AN Rajagopalan, and Vishnu Naresh Boddeti. Spatially-adaptive image restoration using distortion-guided networks. In *ICCV*, 2021. 7

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 3, 5

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

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 5

[30] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, 2019. 5

[31] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *CVPR*, 2015. 5

[32] Renjie Wan, Boxin Shi, Wenhan Yang, Bihan Wen, Ling-Yu Duan, and Alex C Kot. Purifying low-light images via near-infrared enlightened image. *TMM*, 2022. 3

[33] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, 2023. 2

[34] Ruixing Wang, Xiaogang Xu, Chi-Wing Fu, Jiangbo Lu, Bei Yu, and Jiaya Jia. Seeing dynamic scene in the dark: A high-quality video dataset with mechatronic alignment. In *ICCV*, 2021. 5

[35] Tao Wang, Kaihao Zhang, Tianrun Shen, Wenhan Luo, Bjorn Stenger, and Tong Lu. Ultra-high-definition low-light image enhancement: A benchmark and transformer-based method. In *AAAI*, 2023. 5

[36] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018. 1

[37] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, 2022. 8

[38] Zejin Wang, Jiazheng Liu, Guoqing Li, and Hua Han. Blind2unblind: Self-supervised image denoising with visible blind spots. In *CVPR*, 2022. 1

[39] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *CVPR*, 2022. 3

[40] Wenhui Wu, Jian Weng, Pingping Zhang, Xu Wang, Wenhan Yang, and Jianmin Jiang. Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement. In *CVPR*, 2022. 5

[41] Yuhui Wu, Chen Pan, Guoqing Wang, Yang Yang, Jiwei Wei, Chongyi Li, and Heng Tao Shen. Learning semantic-aware knowledge guidance for low-light image enhancement. In *CVPR*, 2023. 1, 2, 6

[42] Xiaogang Xu, Ruixing Wang, Chi-Wing Fu, and Jiaya Jia. Snr-aware low-light image enhancement. In *CVPR*, 2022. 1, 2, 4, 5

[43] Xiaogang Xu, Yitong Yu, Nianjuan Jiang, Jiangbo Lu, Bei Yu, and Jiaya Jia. Pvdd: A practical video denoising dataset with real-world dynamic scenes. *arXiv preprint*, 2022. 1

[44] Xiaogang Xu, Hengshuang Zhao, Philip Torr, and Jiaya Jia. General adversarial defense against black-box attacks via pixel level and feature level distribution alignments. *arXiv preprint*, 2022. 1

[45] Xiaogang Xu, Ruixing Wang, Chi-Wing Fu, and Jiaya Jia. Deep parametric 3d filters for joint video denoising and illumination enhancement in video super resolution. In *AAAI*, 2023. 5

[46] Xiaogang Xu, Ruixing Wang, and Jiangbo Lu. Low-light image enhancement via structure modeling and guidance. In *CVPR*, 2023. 1, 2, 6

[47] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning unified representation of language, image and point cloud for 3d understanding. In *CVPR*, 2023. 3

[48] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *CVPR*, 2017. 5

[49] Wenhan Yang, Wenjing Wang, Haofeng Huang, Shiqi Wang, and Jiaying Liu. Sparse gradient regularized deep Retinex network for robust low-light image enhancement. *TIP*, 2021. 5

[50] Yan Yang, Liyuan Pan, Liu Liu, and Miaomiao Liu. K3dn: Disparity-aware kernel estimation for dual-pixel defocus deblurring. In *CVPR*, 2023. 1

[51] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021. 7, 8

[52] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 1, 5, 7, 8

[53] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *CVPR*, 2022. 3

[54] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, 2018. 5

[55] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *TCSVT*, 2019. 5

[56] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 5

[57] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *TPAMI*, 2021. 7

[58] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *JEI*, 2011. 5

[59] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *CVPR*, 2022. 3

[60] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *CVPR*, 2023. 3