Empowering Resampling Operation for Ultra-High-Definition Image Enhancement with Model-Aware Guidance

Wei Yu[*], Jie Huang[*], Bing Li, Kaiwen Zheng, Qi Zhu, Man Zhou, Feng Zhao[†]
University of Science and Technology of China
{patrick914y, hj0117, bing0123, kezh, zqcrafts, manman}@mail.ustc.edu.cn, fzhao956@ustc.edu.cn

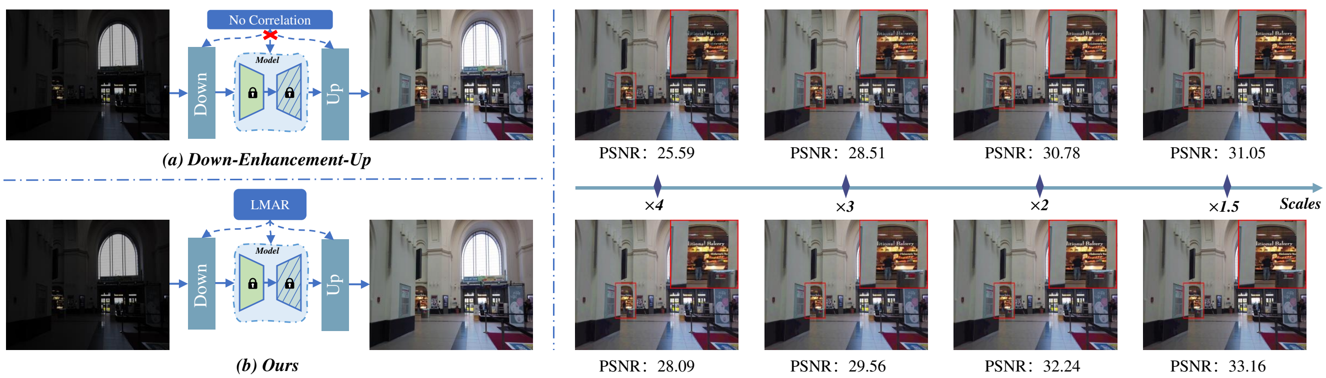


Figure 1. Comparison between previous *Downsampling-Enhancement-Upsampling* paradigm and our proposed method. (a) Previous works treat resampling operators and inner models as separate components, while (b) our method integrates them via empowering model-aware resampling. As shown on the right side, our approach achieves significant performance gains over various resampling scales.

Abstract

Image enhancement algorithms have made remarkable advancements in recent years, but directly applying them to Ultra-high-definition (UHD) images presents intractable computational overheads. Therefore, previous straightforward solutions employ resampling techniques to reduce the resolution by adopting a "Downsampling-Enhancement-Upsampling" processing paradigm. However, this paradigm disentangles the resampling operators and inner enhancement algorithms, which results in the loss of information that is favored by the model, further leading to sub-optimal outcomes. In this paper, we propose a novel method of Learning Model-Aware Resampling (LMAR), which learns to customize resampling by extracting model-aware information from the UHD input image, under the guidance of model knowledge. Specifically, our method consists of two core designs, namely compensatory kernel estimation and steganographic resampling. At the first stage, we dynamically predict compensatory kernels tailored to the specific input and resampling scales. At the second stage, the image-wise compensatory information is derived with the compensatory kernels and embedded into the rescaled input images. This promotes the representation of the newly derived downscaled inputs to be more consistent with the full-resolution UHD inputs, as perceived by the model. Our LMAR enables model-aware and model-favored resampling while maintaining compatibility with existing resampling operators. Extensive experiments on multiple UHD image enhancement datasets and different backbones have shown consistent performance gains after correlating resizer and enhancer, e.g., up to 1.2dB PSNR gain for $\times 1.8$ *resampling scale on UHD-LOL4K. The code is available at https://github.com/YPatrickW/LMAR.*

1. Introduction

Over time, learning-based image enhancement algorithms have achieved progressive performance improvements. However, the majority of existing methods are not compatible with UHD images due to the heavy computational burdens imposed by their megapixel counts.

To address this, previous works have sought to alleviate computation overheads by employing downsampling to reduce the resolution, as shown in Figure 1. Concretely, UHD input images are often resampled to smaller sizes for

*Both authors contributed equally to this research.
†Corresponding author.

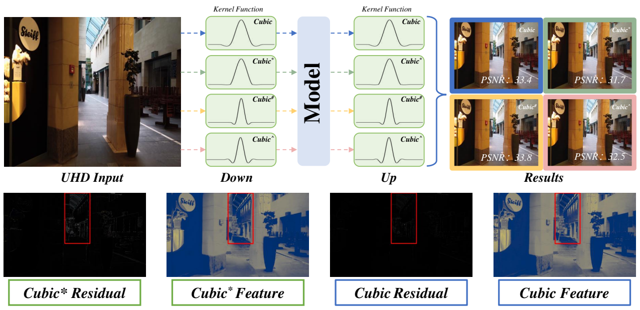


Figure 2. The influence of the front-end resampler. We choose the cubic resampling operator as the baseline and introduce perturbations to its kernel function. Interestingly, manipulating the resampling function can lead to volatile outcomes, with the image quality either being improved or degraded. We also present two more comparisons: the resampling residual of the UHD input image and the feature maps generated by different resizers. It is indicated that the response of the model is highly sensitive to the sampling strategies. This highlights the importance of customizing the resampling process in a model-aware manner.

enhancement, and then subsequently resampled back to the original resolution[22, 39]. Surprisingly, the majority of research efforts have been paid to developing inner processing algorithms, with little attention devoted to the foremost and endmost resampling operators.

Recently, a line of work has proposed the replacement of the conventional interpolation-based resampling operators (e.g., nearest, bilinear, bicubic) with learnable resampling operators [12, 20, 28, 30]. Attributing to the superior modeling capability of deep neural networks, these learnable resampling operators have demonstrated remarkable performance gains. However, the practical application of these learnable resampling operators is hindered by their high computational complexity and reduced efficiency, resulting in a continued preference for interpolation-based resampling operators.

Revisiting the above UHD image enhancement pipeline: *Downsampling-Enhancement-Upsampling*, with the internal enhancement algorithm remaining unchanged, a fundamental question arises: *'How significant is the impact of these two resampling operations ?'* The answer to this question is twofold. On the one hand, the quality of the upsampled enhanced results exhibits a decreasing trend as the degree of downsampling aggravates. On the other hand, manipulating the interpolation kernel functions yields a diverse range of outcomes, as illustrated in Figure 2. We attribute this phenomenon to the isolation between the resampling operators and the enhancement algorithm, where both components are disentangled from each other.

To this end, the focus of this paper lies in correlating the resampling operators within the internal enhancement algorithms and unleashing the potential of the resampling operation to make benefits for the aforementioned framework, without modifying the inner processing methods (e.g., pretrained networks). To accomplish this goal, we propose a novel method termed Learning Model-Aware Resampling (LMAR). LMAR correlates the resizer and the enhancer through a compensatory information learning process.

We begin by delving into the impact of resampling from the viewpoint of the enhancer, where the resampling operations directly undermine the representation ability of intermediate features. Therefore, the intermediate features of the UHD image offer valuable insights on how to perform resampling in a model-aware manner. Building upon this recognition, we formulate our method into two steps: Compensatory Kernel Estimation (CKE) and Steganographic Resampling (SR). In the CKE step, we generate image-specific convolution kernels for the input UHD image via implicit neural representation. These kernels are specifically customized for each input and resampling scale. Then, we can obtain compensatory information produced by these predicted filters. Subsequently, in the SR step, the compensatory information is embedded in the downscaled input image, resulting in a newly learnable downscaled input. By utilizing the compensatory information as a medium, we establish the correlation between the resampling operators and the inner enhancer. Finally, the learnable input is passed to the enhancer and optimized to encourage that the intermediate features align more consistently with those of the UHD image. Thanks to these designs, our method not only maintains compatibility with existing interpolation-based resampling operators but also incorporates them into a part of enhancement algorithms under the guidance of model knowledge. Our contributions are summarized as:

- We rethink the disentanglement between the resizer and enhancement model and propose to perform the resampling in a model-aware manner.
- We introduce LMAR, a novel method that facilitates the collaboration between resizer and enhancer via customizing resampling under the guidance of model knowledge, where the representative capabilities of low-resolution images are largely improved.
- Our LMAR delivers two core designs: compensatory kernel estimation and steganographic resampling, which models the interplay between resizer and enhancer by learning embeddable information. Our method is compatible with any interpolation resamplers.
- Extensive experiments are conducted to verify the superiority of correlating resizer and enhancer. Remarkably, our algorithm significantly improves the performance at low resolutions while maintaining equivalent results at the original resolution, without requiring retraining of the enhancer.

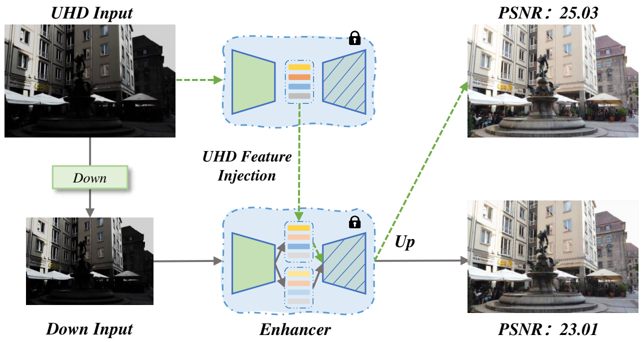


Figure 3. The inspiration for our method design. We replace the compressed bottleneck features of the downsampled input with features extracted from the UHD image (*The green arrow*). Large-margin performance gains can be achieved compared with the vested paradigm (*The gray arrow*). This observation highlights the potential role of feature consistency between UHD input and any downscaled input, which can play as a bridge for establishing correlations between resampling operators and enhancers.

2. Related Work

2.1. Image Resampling

Image resampling has emerged as one of the most commonly employed techniques for altering the resolution of an image. The key to image resampling lies in effectively preserving the quality of image content while maintaining efficiency.

Interpolation operators, such as bilinear, and bicubic [1, 8], have been the popular choice for image resampling for many years. These operators are capable of effectively resampling images at any scale, yet they disregard the spatial variations of distinct image patches, leading to inadequate preservation of the local structure [26].

In recent years, deep neural networks have facilitated advancements in various tasks [5, 10, 13–19, 31, 32, 36, 42–48], including image resampling [4, 11, 12, 20, 28, 30, 34, 35]. Task-aware image downsampling (TAD) [20] presents an encoder-decoder framework that aims to enhance the quality of both downscaling and upscaling processes in a collaborative manner. Though these methods achieve excellent results, none of them focus on correlating interpolation resampler with a pre-trained model.

2.2. UHD Image Processing

A line of studies that specifically focus on high-resolution image processing has been proposed [2, 6, 23, 33, 40, 41]. These methods share a common paradigm: learning from downscaled images to alleviate the computational burdens. In this learning paradigm, the quality of the results is directly determined by the resampled image representation, highlighting the crucial role of resampling strategies. However, they often overlook the impact of resampling approaches and lack in-depth investigation of various resampling operators. In this work, our objective is to explore the interplay between enhancer and resampling operators and establish a collaborative relationship for them to improve performance.

3. Methodology

In this section, we will present a comprehensive illustration of our proposed LMAR. We begin by introducing the motivation behind our approach and describing our settings. Subsequently, we will overview the whole processing pipeline and delve into the design of the Compensatory Kernel Estimation (CKE) and Steganographic Resampling (SR). Finally, we will discuss the optimization strategies employed to enable model-aware resampling.

3.1. Setting and Motivation

Our setting still follows the *Downsampling-Enhancement-Upsampling* paradigm, employing the interpolation-based resampling operators instead of introducing additional burdens with learnable resampling operators. Notably, we will first train an enhancement model for each dataset in our experimental setup and keep unchanged. The main difference lies in the collaboration process between the resizer and enhancer.

The main inspiration stems from observing the impact of resampling on the representative feature extraction process of the enhancer. As shown in Figure 3, we inject the full-resolution bottleneck features into the enhancement process of the downsampled input images, where the final results significantly outperform the vested paradigm. It proves that front-most resampling degrades the representative ability of bottleneck features and impacts the following processes. Based on the insight, we correlate the resizer and enhancer through a compensatory information-learning process to achieve consistent feature alignment.

3.2. Overview

The *Downsampling-Enhancement-Upsampling* processing can be mathematically formulated as:

$$\begin{aligned} x_d &= D(x; [h, w]), \\ \tilde{y} &= F(x_d), \\ y &= U(\tilde{y}; [H, W]), \end{aligned} \tag{1}$$

where $D(\cdot)$ is the downsampler that resamples the UHD input image $x \in \mathbb{R}^{H \times W \times 3}$ to lower resolution of $x_d \in \mathbb{R}^{h \times w \times 3}$, $F(\cdot)$ is the pre-trained enhancement model, and $U(\cdot)$ denotes the upsampler which resamples the low-resolution enhanced result $\tilde{y}$ back to the UHD resolution of $H \times W$, termed as y. In LMAR, we rebuild the above paradigm, given as:

$$\begin{aligned} \tilde{y} &= F(x_d, \Theta(x; D, U, F)), \\ y &= U(\tilde{y}; [H, W]). \end{aligned} \tag{2}$$

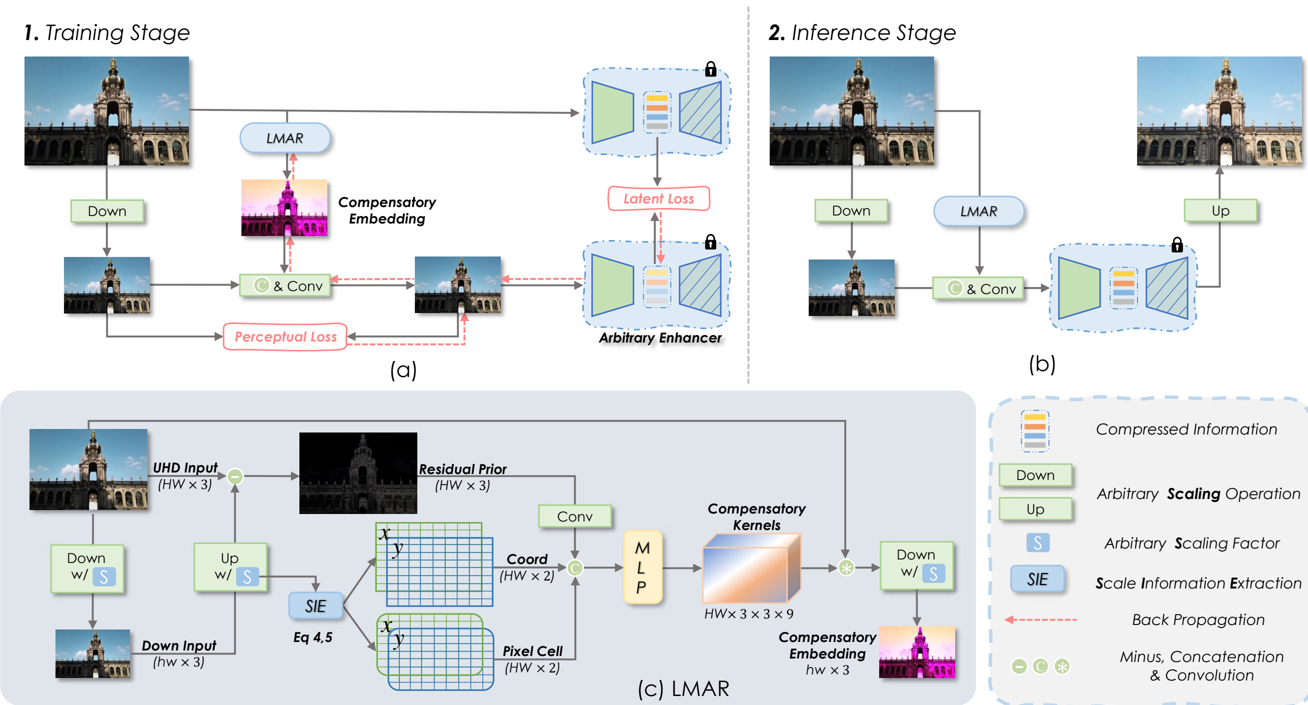


Figure 4. Overview of the proposed LMAR. The sub-graph (a) depicts the training phase of LMAR, which encourages the compensated low-resolution input to maintain representation consistency with the full-resolution UHD input, as perceived by the enhancer. The sub-graph (b) demonstrates the inference pipeline of our LMAR, where the compensated low-resolution input is directly fed into the enhancer and then upsampled to the UHD result. The sub-graph (c) illustrates how LMAR works, where the core lies in estimating compensatory kernels under the guidance of model knowledge to make up for the resampling process.

Here, we introduce additional compensatory information as a medium to facilitate model-aware resampling, denoted as $\Theta(x; D, U, F)$. Our method is presented in Figure 4, where the model-aware resampling is achieved via two core designs, compensatory kernel estimation, and steganographic resampling.

3.3. Compensatory Kernel Estimation

Interpolation-based image resampling process can be decomposed as projected grid calculation and weighted pixel aggregation [29]. To this end, we estimate the compensatory kernel using a representation that is related to the scale. Inspired by LIFF [4], we employ two scale features, the relative coordinate grid and the pixel cell. The relative coordinate grid depicts the pixel location shift in the resampling processing. We first compute the uniform coordinate of the downsampling scale $C_d \in \mathbb{R}^{h \times w \times 2}$ and the UHD scale $C_u \in \mathbb{R}^{H \times W \times 2}$, which are normalized between $[-1, 1]$. The calculations of the above grids are given as:

$$\begin{aligned} C_d(i,j) &= (-1 + \frac{(2i+1)}{h}, -1 + \frac{(2j+1)}{w}), \\ C_u(i,j) &= (-1 + \frac{(2i+1)}{H}, -1 + \frac{(2j+1)}{W}), \end{aligned} \tag{3}$$

where $i \in [0, h-1] \vee [0, H-1], j \in [0, w-1] \vee [0, W-1]$ are the position indexes for the height and width dimensions. As the sizes of the two coordinates are different, we first project them C_d to C_u and obtain the relative grid shift at the downsampling scale, which is expressed as:

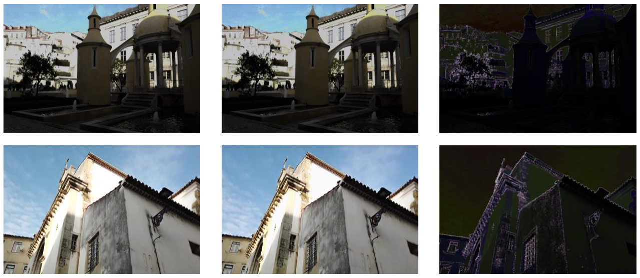
$$\begin{aligned} \tilde{C}_d &= \mathit{grid\_sample}(C_d, C_u), \\ \tilde{C}_r &= C_u - \tilde{C}_d, \\ C_r &= (\tilde{C}_r(i) \times h, \tilde{C}_r(j) \times w), \end{aligned} \tag{4}$$

where the *grid_sample* is a remapping function that remaps the C_d into $\tilde{C}_d$ of size $H \times W$, $\tilde{C}_r$ represents the relative shift and C_r denotes the relative coordinate grid measured at downsampling scale. The pixel cell represents the ratio of pixel area changes between downsampling and upsampling, denoted as:

$$P_c(i,j) = (2 * \frac{h}{H}, 2 * \frac{w}{W}), \tag{5}$$

where $P_c \in \mathbb{R}^{H \times W \times 2}$, and all pixel locations share the same scale ratio.

The aforementioned scale-related representations are instance-same at a given scale, leading to the generation of identical kernels without considering the distinctions between images. Thus, we propose the utilization of back-projection [9] as a constraint to capture the interrelation



(a) x_d (b) x_c (c) $|x_c - x_d|$

Figure 5. Comparison of different downscaled input images. (a) The downscaled input images by the cubic resampling operator, (b) images derived from the cubic resampling operator in a model-aware manner, and (c) the residual between (a) and (b), where obvious differences can be witnessed in high-frequency areas.

between each downscaled input and its corresponding UHD counterpart, formulated as:

$$\tilde{x} = Conv(x - U(x_d; [H, W]))), \tag{6}$$

where $\tilde{x} \in \mathbb{R}^{H \times W \times 3}$ is the back-projection of UHD input image x.

Considering the pixel-wise aggregation nature of the resampling process, we model the generation of compensatory information by predicting the convolution kernels for the UHD input image instead of predicting pixel values for specific locations. Specifically, we flatten the relative coordinate C_r, pixel cell P_c, and back-projection $\tilde{x}$ then concatenate them. Afterward, we feed these concatenated features into a multi-layer perception (MLP) to predict compensatory kernels as follows:

$$K = MLP_\phi([C_r; P_c; \tilde{x}]), \tag{7}$$

where $[;]$ denotes the channel-wise concatenation, and $K \in \mathbb{R}^{H \times W \times s}$ represents the predicted compensatory kernels. The kernel for each pixel location is represented in the channel dimension s, consisting of the multiplication of the input channel, out channel, and kernel size, with all being three in our case.

3.4. Steganographic Resampling

With the estimated compensatory kernel from the above process, we calculate the compensatory information for the input UHD image as follows:

$$\widehat{x} = K \otimes x, \tag{8}$$

where $\widehat{x} \in \mathbb{R}^{H \times W \times 3}$ is the generated UHD compensatory embedding, and $\otimes$ denotes the convolution operation.

To further maintain the intrinsic characteristics of the downscaled input images, we downscale the compensatory information and perform compensatory steganography in the low-resolution space, which is computed as:

$$x_c = Conv([x_d; D(\widehat{x}; [h, w])]). \tag{9}$$

Here, we concatenate the downscaled UHD image and downscaled compensatory information, then perform channel reduction by convolution layer. In this way, the newly derived x_c can be optimized under the guidance of model knowledge, where we utilize the compensatory information to bridge the resizer and enhancer.

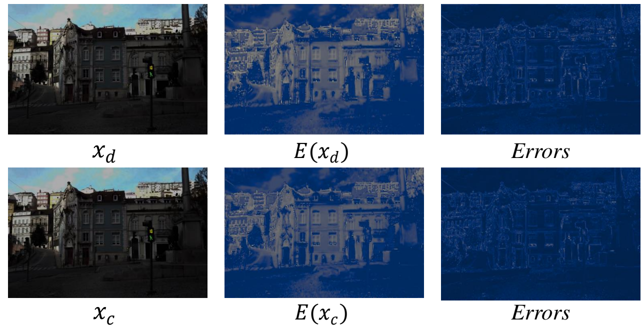


Figure 6. Feature comparison of different downscaled input images. The second column represents the bottleneck features of the downscaled inputs. As observed in the error maps with the UHD features in the third column, downscaled input produced in a model-aware manner (x_c) maintains a higher level of representative consistency with the UHD input, as perceived by the model.

3.5. Optimization Object

Given the pre-trained enhancer, the bottleneck features of the UHD input image possess the most powerful representational capability compared to all of its downscaled counterparts, Therefore, we encourage the consistency of bottleneck features, which is referred to as resampling consistency loss:

$$\mathcal{L}_{rc} = \mathit{SmoothL1}(E(x) - U(E(x_c))), \tag{10}$$

where the $E(x)$ and $E(x_c)$ represent the corresponding bottleneck features of the UHD input x and learnable downscaled input x_c. $U(\cdot)$ is used for size adjustment. In addition, we introduce attention loss [37] for these two bottleneck features, given as:

$$\mathcal{L}_{fa} = \| \frac{E(x)}{\|E(x)\|_2} - \frac{U(E(x_c))}{\|U(E(x_c))\|_2} \|_2, \tag{11}$$

where $\|\cdot\|_2$ is the $\mathcal{L}_2$ normalization and this loss encourages the attention transfer from $E(x)$ to $U(E(x_c))$.

The aforementioned losses are designed to incorporate model-favored information into the learnable downscaled input image x_c but neglect the distribution consistency between the naive downscaled images x_d and newly derived images x_c. Thus, we need to impose restrictions on the

Table 1. Quantitative results on the UHD-LOL4K datasets with *CNN-I* as the backbone. The results with LMAR are shown in gray with better results highlighted in **bold**.

Method	CNN-I (UHD-LOL4K)						
Scales	(2160, 3840)	(1440, 2560)	(1080, 1920)	(1200, 1600)	(720, 1280)	(540, 960)	(432, 768)
Nearest	31.92 / 0.9843	28.20 / 0.9815	27.01 / 0.9792	26.66 / 0.9763	25.86 / 0.9761	23.99 / 0.9615	23.06 / 0.9445
	31.86 / 0.9841	**28.46 / 0.9824**	**27.69 / 0.9813**	**27.42 / 0.9790**	**26.60 / 0.9806**	**24.38 / 0.9715**	**23.98 / 0.9608**
Bilinear	31.92 / 0.9843	31.07 / 0.9839	29.75 / 0.9828	29.16 / 0.9820	27.01 / 0.9767	25.90 / 0.9744	24.03 / 0.9448
	31.88 / 0.9841	**31.28 / 0.9840**	**30.17 / 0.9835**	**29.85 / 0.9832**	**27.94 / 0.9814**	**26.27 / 0.9795**	**24.85 / 0.9613**
Bicubic	31.92 / 0.9843	31.19 / 0.9840	29.82 / 0.9834	29.26 / 0.9822	27.16 / 0.9767	25.40 / 0.9694	24.04 / 0.9450
	31.89 / 0.9842	**31.42 / 0.9841**	**30.20 / 0.9839**	**29.96 / 0.9836**	**27.97 / 0.9815**	**26.36 / 0.9792**	**24.88 / 0.9617**
Lanczos2	31.92 / 0.9843	31.04 / 0.9840	29.83 / 0.9834	29.22 / 0.9819	26.97 / 0.9761	25.39 / 0.9694	24.03 / 0.9450
	31.88 / 0.9841	**31.39 / 0.9841**	**30.19 / 0.9838**	**29.97 / 0.9834**	**27.90 / 0.9815**	**26.29 / 0.9789**	**24.87 / 0.9617**
Lanczos3	31.92 / 0.9843	30.86 / 0.9838	29.08 / 0.9815	28.88 / 0.9813	27.10 / 0.9764	25.04 / 0.9663	23.86 / 0.9438
	31.90 / 0.9842	**31.59 / 0.9842**	**30.24 / 0.9836**	**30.13 / 0.9834**	**27.93 / 0.9817**	**25.79 / 0.9718**	**24.66 / 0.9611**
Time (ms)	9.7	9.6	13.3	13.4	14.8	15.4	15.0
	10.7	10.2	13.3	13.8	15.1	15.6	15.4

(a) Input (b) × 5 (c) × 4 (d) × 3 (e) × 2 (f) × 1.5 (g) × 1 (h) GT

Figure 7. Qualitative results on the UHD-LOL4K dataset of different scales. The top row is obtained from the cubic operator without LMAR, while the bottom row is with LMAR. Please zoom in for details.

x_c to ensure its similarity to x_d. However, measuring the distances in the pixel space deteriorates the learning of the above process. Inspired by [7], we add a discriminator and impose x_c and x_d to confound it, known as adversarial loss, formulated as:

$$\mathcal{L}_d = log(D(x_d)) + log(1 - D(x_c)), \tag{12}$$

where $D(\cdot)$ is the discriminator, which ensures the interaction between the resizer and enhancer while maintaining distribution consistency between x_c and x_d. In addition, to account for the introduction of the discriminator, a new generation loss is incorporated to constrain x_c, computed as:

$$\mathcal{L}_g = log(D(x_c)). \tag{13}$$

Overall, our optimization objects are twofold. The first optimization focus is on correlating the resizer and enhancer:

$$\mathcal{L}_c = \mathcal{L}_{rc} + \alpha\mathcal{L}_{fa} + \beta\mathcal{L}_g, \tag{14}$$

where α and β are adjustable parameters to balance different losses, which are set empirically. The second optimization object is $\mathcal{L}_d$ to restrict the distribution. These two optimization objects are alternatively optimized. As can be seen in Figures 5 and 6, our downscaled images not only maintain consistency in terms of perceptual distribution but also exhibit a more consistent representation, as perceived by the model.

4. Experiments

4.1. Experimental Setup

Datasets. To evaluate the significance of correlating the enhancer and resizer, experiments are conducted on two UHD low-light enhancement datasets, including the UHD-LOL4K dataset proposed by [33] and the 4KIL dataset proposed by [24]. More details about datasets are provided in the supplementary materials.

Pre-training. We begin by training multiple enhancers $F(\cdot)$ to verify the effectiveness of our method. Specifically, we employ the widely used encoder-decoder architecture in image restoration tasks as the backbone. Regarding the UHD-LOL4K dataset, we leverage the invertible block proposed by [25] as the basic unit for processing, termed CNN-I. For the 4KIL dataset, we replace the basic processing unit with the half instance normalization block proposed

Table 2. Quantitative results on the 4KIL datasets with *CNN-H* as the backbone. The results with LMAR are shown in gray with better results highlighted in **bold**.

Method	CNN-H (4KIL)						
Scales	(2160, 3840)	(1440, 2560)	(1080, 1920)	(1200, 1600)	(720, 1280)	(540, 960)	(432, 768)
Nearest	29.80 / 0.9805	28.36 / 0.9779	27.71 / 0.9754	27.67 / 0.9742	26.98 / 0.9719	25.92 / 0.9616	25.37 / 0.9428
	29.77 / 0.9801	**28.53 / 0.9789**	**28.11 / 0.9779**	**27.93 / 0.9762**	**27.21 / 0.9745**	**26.16 / 0.9672**	**25.56 / 0.9493**
Bilinear	29.80 / 0.9805	29.12 / 0.9796	28.47 / 0.9785	28.39 / 0.9779	27.30 / 0.9726	26.38 / 0.9659	25.56 / 0.9439
	29.79 / 0.9804	**29.14 / 0.9804**	**28.55 / 0.9794**	**28.51 / 0.9790**	**27.52 / 0.9750**	**26.57 / 0.9687**	**25.62 / 0.9464**
Bicubic	29.80 / 0.9805	29.25 / 0.9797	28.60 / 0.9787	28.47 / 0.9780	27.30 / 0.9726	26.36 / 0.9650	25.58 / 0.9444
	29.80 / 0.9804	**29.35 / 0.9804**	**28.74 / 0.9797**	**28.96 / 0.9793**	**27.53 / 0.9752**	**26.57 / 0.9685**	**25.79 / 0.9509**
Lanczos2	29.80 / 0.9805	29.24 / 0.9797	28.59 / 0.9787	28.46 / 0.9780	27.29 / 0.9726	26.35 / 0.9649	25.56 / 0.9444
	29.80 / 0.9803	**29.34 / 0.9805**	**28.72 / 0.9796**	**28.63 / 0.9792**	**27.54 / 0.9752**	**26.57 / 0.9684**	**25.78 / 0.9509**
Lanczos3	29.80 / 0.9805	29.32/ 0.9798	28.69 / 0.9798	28.54 / 0.9781	27.34 / 0.9730	26.34 / 0.9645	25.58 / 0.9457
	29.79 / 0.9803	**29.48 / 0.9804**	**28.89 / 0.9799**	**28.76 / 0.9794**	**27.60 / 0.9756**	**26.58 / 0.9686**	**25.80 / 0.9523**
Time (ms)	5.0	5.0	5.8	6.3	6.5	6.9	6.9
	5.9	5.7	6.1	6.5	6.8	8.5	8.0

Table 3. Quantitative results on the 4KIL datasets with *Restormer* as the backbone. The results with LMAR are shown in gray with better results highlighted in **bold**.

Method	Restromer (4KIL)						
Scales	(2160, 3840)	(1440, 2560)	(1080, 1920)	(1200, 1600)	(720, 1280)	(540, 960)	(432, 768)
Nearest	28.86 / 0.9773	27.81 / 0.9750	27.57 / 0.9738	27.45 / 0.9720	26.99 / 0.9714	25.91 / 0.9608	25.31 / 0.9470
	28.78 / 0.9768	**28.36 / 0.9765**	**27.67 / 0.9742**	**27.80 / 0.9739**	**27.27 / 0.9733**	**25.98 / 0.9619**	**25.46 / 0.9527**
Bilinear	28.86 / 0.9773	28.63 / 0.9770	28.50 / 0.9768	28.49 / 0.9767	27.56 / 0.9719	26.71 / 0.9686	25.71 / 0.9459
	28.83 / 0.9770	**28.75 / 0.9771**	**28.52 / 0.9769**	**28.64 / 0.9770**	**27.85 / 0.9739**	**26.77 / 0.9688**	**25.90 / 0.9531**
Bicubic	28.86 / 0.9773	28.83 / 0.9771	28.64 / 0.9770	28.53 / 0.9766	27.57 / 0.9719	26.65 / 0.9668	25.79 / 0.9477
	28.84 / 0.9772	**28.84 / 0.9772**	**28.69 / 0.9771**	**28.76 / 0.9771**	**27.82 / 0.9738**	**26.71 / 0.9678**	**25.94 / 0.9533**
Lanczos2	28.86 / 0.9773	28.83 / 0.9770	28.63 / 0.9769	28.52 / 0.9766	27.56 / 0.9719	26.65 / 0.9668	25.78 / 0.9476
	28.83 / 0.9772	**28.84 / 0.9771**	**28.67 / 0.9770**	**28.74 / 0.9773**	**27.81 / 0.9739**	**26.70 / 0.9678**	**25.93 / 0.9533**
Lanczos3	28.86 / 0.9773	28.84 / 0.9770	28.68 / 0.9768	28.55 / 0.9765	27.56 / 0.9719	26.56 / 0.9652	25.72 / 0.9468
	28.85 / 0.9772	**28.85 / 0.9773**	**28.84 / 0.9771**	**28.85 / 0.9772**	**27.84 / 0.9740**	**26.58 / 0.9665**	**25.89 / 0.9528**
Time (ms)	11.9	11.5	12.5	11.7	12.0	11.7	12.3
	12.8	12.9	12.9	13.0	12.2	12.6	12.5

by [3] to demonstrate the robustness of our method, termed CNN-H. In addition to the above CNN models, we also train a transformer-based enhancer conditioned on Restormer, as proposed by [38], on the 4KIL dataset to further validate the scalability of our method. More details about pre-training are provided in the supplementary materials.

4.2. Collaborative Training

In this stage, we keep the pre-trained enhancers fixed and train to correlate the resizer and enhancer. Considering the randomness of scale variability in the resampling process, we incorporate random scale training. Specifically, we begin by cropping a $H_p \times W_p$ patch from the UHD input image. Next, we randomly sample a scale factor r from a uniform distribution $\mathcal{U}(1, 4)$. Finally, we downsample the high-resolution $H_p \times W_p$ patch by the scale factor r to generate a low-resolution $h_p \times w_p$ patch. The optimization objectives mentioned earlier are applied to these two resolution patches.

Table 4. Comparison of model parameters. Baseline models with LMAR are shown in gray, where the extra parameters are trainable parameters possessed by LMAR.

Model	CNN-I	CNN-H	Restormer
#Params	1.009MB	1.128MB	1.660MB
	1.314MB	1.495MB	1.965MB

Implementation Details. We conduct experiments based on the PyTorch framework along with the tiny-cuda-nn [27] library with one NVIDIA 3090 GPU. During the collaborative training process, the Adam [21] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is used in our experiments. A batch size of 1 is utilized for all experiments. The cropped patch size ($H_p \times W_p$) is set to 1024×1024 for the CNN enhancers, and 768×768 for the transformer enhancer. For the CNN-I and CNN-H, the collaborative training epoch is set to 12 and 24, respectively. The initial learning rate is $4e^{-4}$, which decays by a factor value of 0.75 every 4 epochs and 8 epochs for the CNN-I and CNN-H enhancers. As for

(a) Input (b) × 5 (c) × 4 (d) × 3 (e) × 2 (f) × 1.5 (g) × 1 (h) GT

Figure 8. Qualitative results on the 4KIL dataset of different scales. The top row is obtained from the lanzcos3 operator without LMAR, while the bottom row is with LMAR. Please zoom in for details.

the transformer enhancer, the number of collaborative training epochs is 60, with a $4e^{-4}$ initial learning rate, which decays by a factor of 0.75 every 20 epochs. The discriminator is composed of several convolution layers, whose initial learning rate is $2e^{-4}$ and shares the same training strategies as mentioned above.

4.3. Performance Comparison

Quantitative Evaluation. We utilize two metrics, the Peak Signal to Noise Ratio (PSNR, dB), and the Structural Similarity (SSIM), to assess the performance gains. To verify the universality of our methods, we adopt five types of resizers and test them on various commonly used scales, including both in-distribution and out-of-distribution scales. As shown in Tables 1, 2, and 3, our method consistently achieves performance improvements across different resolutions, regardless of the choice of resizer or the category of the backbone. Notably, the performance with LMAR at UHD resolution is analogous to the original UHD input, which illustrates the superior continuous modeling capability of LMAR. Additionally, our approach maintains comparable effectiveness with the baselines, where a minor increase in parameters, as shown in Table 4. More importantly, our algorithm only needs to be trained on one type of resizer, which can then be applied to any other resizers.
Qualitative Evaluation. Figures 7 and 8 present the visual comparison on the UHD-LOL4K dataset and 4KIL dataset respectively. These comparisons demonstrate that enabling model-aware resampling leads to enhanced results with improved global lightness and local structural details, across various resampling scales. It verifies that coupling the resizers and enhancers could generate better visual results. More visual results are presented in the supplementary materials.

4.4. Ablation Studies

We conduct ablation studies on the UHD-LOL4K dataset to investigate the rationality of our designs. The ablation results are shown in Table 5. First, we replace the bottleneck features (Full) with the relatively front features (F_f) and relatively end features (F_e) to apply feature constraints, where remarkable performance drops can be observed at 4 times downsampling. The findings indicate that the bottleneck features exhibit the most representative capability in guiding model-aware resampling. Second, we replace the discriminator (Full) with the perceptual loss (VGG) to conduct the distribution restriction. The utilization of discriminator encourages more consistent feature alignment. Lastly, even with sub-optimal constraints, our method outperforms the baseline, emphasizing the effectiveness of coupling resizers and enhancers.

Table 5. Ablation studies on the constraints employed in our designs. We present the PSNR results here.

Scales	Baseline	F_f	F_e	VGG	Full
x2	29.82	29.98	30.05	30.03	30.20
x4	25.40	25.86	25.50	25.47	26.36

5. Conclusion

In this paper, we introduce a novel approach for effectively enhancing the UHD images at different downsampling scales by a model-aware resampling strategy. Our method customizes resampling under the guidance of model knowledge, integrating the resizers and enhancers through compensatory information learning. Notably, our algorithm is fully compatible with existing interpolation resamplers and maintains the efficiency. Moreover, our design promotes performances at low resolutions remarkably while keeping comparable results at UHD resolution, without the need to retrain the enhancer. Extensive experiments have validated the effectiveness and scalability of our method.

6. Acknowledgment

This work was supported by the JKW Research Funds under Grant 20-163-14-LZ-001-004-01, and the Anhui Provincial Natural Science Foundation under Grant 2108085UD12. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Kenneth R Castleman. *Digital image processing*. Prentice Hall Press, 1996. 3

[2] Jiawen Chen, Andrew Adams, Neal Wadhwa, and Samuel W Hasinoff. Bilateral guided upsampling. *ACM Transactions on Graphics (TOG)*, 35(6):1–8, 2016. 3

[3] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 182–192, 2021. 7

[4] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021. 3, 4

[5] Weitao Feng, Deyi Ji, Yiru Wang, Shuorong Chang, Hansheng Ren, and Weihao Gan. Challenges on large scale surveillance video analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 69–76, 2018. 3

[6] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017. 3

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 6

[8] Dianyuan Han. Comparison of commonly used image interpolation methods. In *Conference of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*, pages 1556–1559. Atlantis Press, 2013. 3

[9] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018. 4

[10] Hanzhe Hu, Deyi Ji, Weihao Gan, Shuai Bai, Wei Wu, and Junjie Yan. Class-wise dynamic graph convolution for semantic segmentation. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020. 3

[11] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1575–1584, 2019. 3

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 2, 3

[13] Deyi Ji, Hongtao Lu, and Tongzhen Zhang. End to end multi-scale convolutional neural network for crowd counting. In *Eleventh international conference on machine vision*, pages 761–766. SPIE, 2019. 3

[14] Deyi Ji, Haoran Wang, Hanzhe Hu, Weihao Gan, Wei Wu, and Junjie Yan. Context-aware graph convolution network for target re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1646–1654, 2021.

[15] Deyi Ji, Haoran Wang, Mingyuan Tao, Jianqiang Huang, Xian-Sheng Hua, and Hongtao Lu. Structural and statistical texture knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16876–16885, 2022.

[16] Deyi Ji, Siqi Gao, Mingyuan Tao, Hongtao Lu, and Feng Zhao. Changenet: Multi-temporal asymmetric change detection dataset. *arXiv preprint arXiv:2312.17428*, 2023.

[17] Deyi Ji, Feng Zhao, and Hongtao Lu. Guided patch-grouping wavelet transformer with spatial congruence for ultra-high resolution segmentation. *International Joint Conference on Artificial Intelligence*, 2023.

[18] Deyi Ji, Feng Zhao, Hongtao Lu, Mingyuan Tao, and Jieping Ye. Ultra-high resolution segmentation with ultra-rich context: A novel benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23621–23630, 2023.

[19] Deyi Ji, Siqi Gao, Lanyun Zhu, Yiru Zhao, Peng Xu, Hongtao Lu, and Feng Zhao. View-centric multi-object tracking with homographic matching in moving uav. *arXiv preprint*, 2024. 3

[20] Heewon Kim, Myungsub Choi, Bee Lim, and Kyoung Mu Lee. Task-aware image downscaling. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–414, 2018. 2, 3

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[22] Yue Li, Dong Liu, Houqiang Li, Li Li, Zhu Li, and Feng Wu. Learning a convolutional neural network for image compact-resolution. *IEEE Transactions on Image Processing*, 28(3): 1092–1107, 2018. 2

[23] Jie Liang, Hui Zeng, and Lei Zhang. High-resolution photorealistic image translation in real-time: A laplacian pyramid translation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9392–9400, 2021. 3

[24] Qiaowanni Lin, Zhuoran Zheng, and Xiuyi Jia. Uhd low-light image enhancement via interpretable bilateral learning. *Information Sciences*, 2022. 6

[25] Yang Liu, Zhenyue Qin, Saeed Anwar, Pan Ji, Dongwoo Kim, Sabrina Caldwell, and Tom Gedeon. Invertible denoising network: A light solution for real noise removal. *arXiv preprint arXiv:2104.10546*, 2021. 6

[26] Don P Mitchell and Arun N Netravali. Reconstruction filters in computer-graphics. *ACM Siggraph Computer Graphics*, 22(4):221–228, 1988. 3

[27] Thomas Müller. tiny-cuda-nn, 2021. 7

[28] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 51–66, 2018. 2, 3

[29] Assaf Shocher, Ben Feinstein, Niv Haim, and Michal Irani. From discrete to continuous convolution layers. *arXiv preprint arXiv:2006.11120*, 2020. 4

[30] Wanjie Sun and Zhenzhong Chen. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing*, 29:4027–4040, 2020. 2, 3

[31] Haoran Wang, Licheng Jiao, Fang Liu, Lingling Li, Xu Liu, Deyi Ji, and Weihao Gan. Ipgn: Interactiveness proposal graph network for human-object interaction detection. *IEEE Transactions on Image Processing*, 30:6583–6593, 2021. 3

[32] Haoran Wang, Licheng Jiao, Fang Liu, Lingling Li, Xu Liu, Deyi Ji, and Weihao Gan. Learning social spatio-temporal relation graph in the wild and a video benchmark. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 3

[33] 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. *arXiv preprint arXiv:2212.11548*, 2022. 3, 6

[34] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. Invertible image rescaling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 126–144. Springer, 2020. 3

[35] Jinbo Xing, Wenbo Hu, Menghan Xia, and Tien-Tsin Wong. Scale-arbitrary invertible image downscaling. *IEEE Transactions on Image Processing*, 2023. 3

[36] Wei Yu, Qi Zhu, Naishan Zheng, Jie Huang, Man Zhou, and Feng Zhao. Learning non-uniform-sampling for ultra-high-definition image enhancement. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1412–1421, 2023. 3

[37] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 5

[38] 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. 7

[39] Yongbing Zhang, Debin Zhao, Jian Zhang, Ruiqin Xiong, and Wen Gao. Interpolation-dependent image downsampling. *IEEE Transactions on Image Processing*, 20(11): 3291–3296, 2011. 2

[40] Zhuoran Zheng, Wenqi Ren, Xiaochun Cao, Xiaobin Hu, Tao Wang, Fenglong Song, and Xiuyi Jia. Ultra-high-definition image dehazing via multi-guided bilateral learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16180–16189. IEEE, 2021. 3

[41] Zhuoran Zheng, Wenqi Ren, Xiaochun Cao, Tao Wang, and Xiuyi Jia. Ultra-high-definition image hdr reconstruction via collaborative bilateral learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4449–4458, 2021. 3

[42] Lanyun Zhu, Deyi Ji, Shiping Zhu, Weihao Gan, Wei Wu, and Junjie Yan. Learning statistical texture for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12537–12546, 2021. 3

[43] Lanyun Zhu, Tianrun Chen, Deyi Ji, Jieping Ye, and Jun Liu. Llafs: When large-language models meet few-shot segmentation. *arXiv preprint arXiv:2311.16926*, 2023.

[44] Lanyun Zhu, Deyi Ji, Tianrun Chen, Peng Xu, Jieping Ye, and Jun Liu. Ibd: Alleviating hallucinations in large vision-language models via image-biased decoding. *arXiv preprint arXiv:2402.18476*, 2024.

[45] Qi Zhu, Zeyu Xiao, Jie Huang, and Feng Zhao. Dast-net: Depth-aware spatio-temporal network for video deblurring. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022.

[46] Qi Zhu, Naishan Zheng, Jie Huang, Man Zhou, Jinghao Zhang, and Feng Zhao. Learning spatio-temporal sharpness map for video deblurring. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[47] Qi Zhu, Man Zhou, Naishan Zheng, Chongyi Li, Jie Huang, and Feng Zhao. Exploring temporal frequency spectrum in deep video deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12428–12437, 2023.

[48] Qi Zhu, Jie Huang, Naishan Zheng, Hongzhi Gao, Chongyi Li, Yuan Xu, Feng Zhao, et al. Fouridown: Factoring downsampling into shuffling and superposing. *Advances in Neural Information Processing Systems*, 36, 2024. 3