4D LUT: Learnable Context-Aware 4D Lookup Table for Image Enhancement

Chengxu Liu, Huan Yang, *Associate Member, IEEE*, Jianlong Fu, and Xueming Qian, *Member, IEEE*

***Abstract*— Image enhancement aims at improving the aesthetic visual quality of photos by retouching the color and tone, and is an essential technology for professional digital photography. Recent years deep learning-based image enhancement algorithms have achieved promising performance and attracted increasing popularity. However, typical efforts attempt to construct a uniform enhancer for all pixels' color transformation. It ignores the pixel differences between different content (*e.g.,* sky, ocean, etc.) that are significant for photographs, causing unsatisfactory results. In this paper, we propose a novel learnable context-aware 4-dimensional lookup table (4D LUT), which achieves content-dependent enhancement of different contents in each image via adaptively learning of photo context. In particular, we first introduce a lightweight context encoder and a parameter encoder to learn a context map for the pixel-level category and a group of image-adaptive coefficients, respectively. Then, the context-aware 4D LUT is generated by integrating multiple basis 4D LUTs via the coefficients. Finally, the enhanced image can be obtained by feeding the source image and context map into fused context-aware 4D LUT via quadrilinear interpolation. Compared with traditional 3D LUT, *i.e.,* RGB mapping to RGB, which is usually used in camera imaging pipeline systems or tools, 4D LUT, *i.e.,* RGBC(RGB+Context) mapping to RGB, enables finer control of color transformations for pixels with different content in each image, even though they have the same RGB values. Experimental results demonstrate that our method outperforms other state-of-the-art methods in widely-used benchmarks.**

***Index Terms*— Image enhancement, photo retouching, lookup tables, neural networks.**

I. Introduction

RECENT developments of high precision sensor equipment witness a fast evolution in low-level computer vision fields and digital photography. However, the captured digital photographs still suffer from low quality due to the effects of illumination, weather, camera sensor, photographer skill, and other factors. Image enhancement as an image processing technique to improve the color, contrast, saturation, brightness, and dynamic range can significantly improve the aesthetic visual quality of photos. Compared with manual photo retouching without professional skills and experience, image enhancement algorithms can automatically produce visual-pleasing photos that satisfy visual aesthetics. It can be equipped in smartphones, digital single lens reflex (DSLR) cameras, and professional-grade software (*e.g.,* Photoshop, Lightroom) to provide expert retouching results and has widely promising applications [26], [35], [38].

Traditional image enhancement methods adopt hand-crafted descriptors or filters to adjust the visual quality of an image by feeding a low-quality input image. The hand-crafted global descriptors (*e.g.,* histogram equalization [45], color correction [29], etc.) can only be used to roughly change the tone of the whole image by establishing color mapping relationships, while the selectable local filters (*e.g.,* Laplacian filter [1], Guided filter [10], etc.) can finely adjust the visual quality of the image according to the content differences. For example, pixels belonging to natural landscapes, portraits, ancient architecture, etc. should adopt different local filters according to the differences in their content. However, this traditional manual adjustment depends on professional image retouching skills, and retouching the image pixel-by-pixel is time-consuming and not practical enough.

Benefiting from advances in deep learning, learning-based image enhancement algorithms are gaining rapid development [5], [46]. MIT-Adobe FiveK dataset [2] proposes an experts-retouched dataset containing 5,000 image pairs of natural landscapes, which is the first to establish a benchmark for the entire field. Further, to facilitate this important and high-visibility task, PPR10K [25] proposes a larger-scale portrait photo retouching dataset, where each portrait has been retouched by three experts with professional experience, separately.

Typical learning-based algorithms can be categorized into three main paradigms. Reinforcement learning-based methods [13], [34], image-to-image translation methods [4], [9], [28], and physical modeling-based methods [5], [31], [32], [40], [51]. Among them, the reinforcement learning-based methods [13], [34] attempt to decouple the whole process and enhance the image through a step-wise retouching process. The image-to-image translation methods [4], [9], [28] try to establish a mapping from the input to the enhanced

Manuscript received 4 September 2022; revised 9 May 2023 and 13 June 2023; accepted 16 June 2023. Date of current version 22 August 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101501; in part by the NSFC under Grant 61772407, Grant 61732008, and Grant 62103317; in part by the Science and Technology Program of Xi'an, China, under Grant 21RGZN0017; in part by the Pazhou Laboratory, Guangzhou; and in part by Microsoft Research. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ananda Shankar Chowdhury. *(Corresponding authors: Huan Yang; Xueming Qian.)*

Chengxu Liu was with Microsoft Research Asia, Beijing 100080, China. He is now with the School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: liuchx97@gmail.com).

Huan Yang and Jianlong Fu are with Microsoft Research Asia, Beijing 100080, China (e-mail: huayan@microsoft.com; jianf@microsoft.com).

Xueming Qian is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security and the SMILES Laboratory, School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China, and also with Shaanxi Yulan Jiuzhou Intelligent Optoelectronic Technology Company Ltd., Xi'an 710000, China (e-mail: qianxm@mail.xjtu.edu.cn).

Digital Object Identifier 10.1109/TIP.2023.3290849

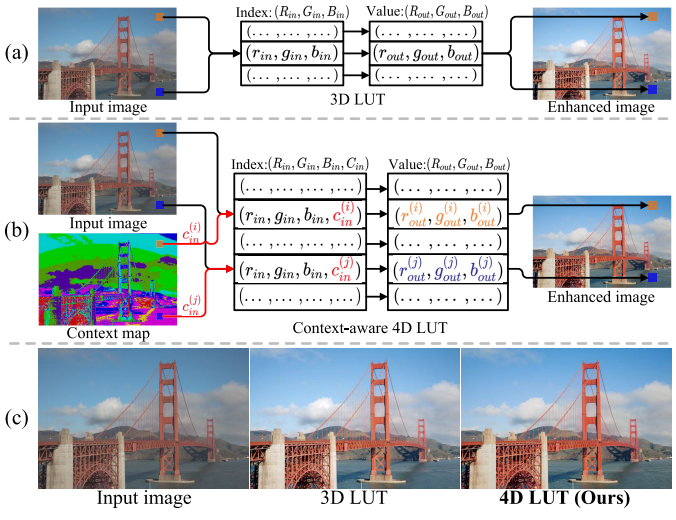


Fig. 1. A comparison between 4D LUT and 3D LUT [51]. (a) and (b) indicate the illustration of the 3D LUT and 4D LUT, respectively. The enhanced pixels (indicated by orange and blue) are retrieved from the index-value correspondence of the defined LUT according to the original input pixels and context (indicated by red). (c) shows a result comparison. They demonstrate that 4D LUT achieves better results by introducing an additional dimension for content-dependent image enhancement.

output directly through a neural network. That method can yield globally optimized results with an end-to-end training manner. However, those algorithms lack interpretability and reliability, like a "black box", and they also consume lots of computational costs, sacrificing their effectiveness in practical applications. To address the limitations of those existing methods, another category of physical modeling-based methods [5], [31], [32], [40], [51] attempt to enhance images using human-interpretable physical models (*e.g.,* Retinex theory [21], bilateral filtering [39], etc). These methods usually adopt a two-step solution, which includes 1) predicting the relevant physics coefficients based on the proposed physical model and assumptions, and 2) adjusting the original pixels to form an enhanced image through physical theory. These efforts not only fail to distinguish the physics coefficients of different content transformations, but also make it difficult to learn in an end-to-end training manner. Therefore, the physical model-based methods do not provide sufficient enhancement capability.

Recently, some outstanding works [25], [41], [51] propose to enhance images by improving the fundamental physical model 3D lookup tables in digital image processing. These methods try to spend fewer runtimes on the enhancement process and focus on learning a uniform enhancer to achieve a globally overall average over all regions of the enhanced result. Although the enhanced results can obtain in real-time, they lack the ability to finely control the color transformation of pixels with different content in each image, and can only obtain globally sub-optimal results. This issue significantly limits the color richness of enhanced images and cannot be handled by 3D lookup tables. For example, as shown in Fig. 1(a), during retouching, the experts often make the sky (indicated by orange) bluer and the sea (indicated by blue) greener in order to improve the color contrast and enhance the aesthetics. That is, even if they have the same RGB values in the unexposed input image, they should be adjusted with different transformations, instead of being treated equally. Intuitively, the content of natural landscapes and portraits should have high color contrast and luminance, respectively, while ancient architectures require lower color temperatures and additional optical effects to describe their history.

Based on the above observation and motivation, we propose a novel learnable context-aware 4D lookup table (4D LUT), which enables content-dependent image enhancement without a significant increase in computational costs, achieving better visually pleasing results (as shown in Fig. 1(c)). As shown in Fig. 1(b), 4D LUT extends the 3-dimensional lookup tables to a 4-dimensional space by introducing an additional contextual dimension, where the input index of 4D LUT (*i.e.,* RGBC) varies with the context map, which increases the color enhancement capability of 4D LUT and enabling finer control of color transformation and stronger image enhancement. In particular, as shown in Fig. 2, it includes four closely-related components. 1) We propose a context encoder to generate context maps through end-to-end learning. The context map represents the pixel-level category in an image based on their content difference, which can extend the image from RGB to RGBC. 2) A parameter encoder for generating a group of coefficients through end-to-end learning. These coefficients can be adaptively changed according to the input image for assisting the final context-aware 4D LUT generation. 3) Based on multiple pre-defined learnable basis 4D LUTs and coefficients obtained above, we propose a 4D LUTs fusion module that crosses different color spaces and integrates them into a final context-aware 4D LUT with stronger enhanced capabilities. 4) We propose to use quadrilinear interpolation, which can convert the input image and context map into four-dimensional spatially indexed for input into the context-aware 4D LUT, and finally outputs the enhanced image after the interpolation operation. Compared with traditional 3D LUT (*i.e.,* RGB mapping to RGB), the design of context-aware 4D LUT (*i.e.,* RGBC mapping to RGB) encourages a content-dependent manner to enable finer control of color transformations, thereby enhancing the pixels' color from an input image to enhanced image.

Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to extend the lookup table architecture into a 4-dimensional space and achieve content-dependent image enhancement without a significant increase in computational costs. More specifically, we propose a learnable context-aware 4-dimensional lookup table (4D LUT), which consists of four closely-related components context encoder, parameter encoder, 4D LUTs fusion, and quadrilinear interpolation.
- The extensive experiments demonstrate that the proposed 4D LUT can obtain more accurate results and significantly outperform existing SOTA methods in three widely-used image enhancement benchmarks.

The rest of the paper is organized as follows. Related work is reviewed in Sec. II. The proposed context-aware 4D LUT is

elaborated in Sec. III. Experimental evaluation, analysis, and ablation study are presented in Sec. IV. The discussions of the related parameters and components are presented in Sec. V. The limitations and failure cases are elaborated in Sec. VI. Finally, we conclude this work in Sec. VII.

II. Related Work

In this section, we first briefly review the traditional algorithms for image enhancement and then review the recent popular study on deep learning-based algorithms.

A. Traditional Algorithms

Traditional image enhancement methods improve the visual quality of images by using hand-crafted global descriptors and local filters. For example, color correction [29] and color histogram equalization [45] adjust image colors by establishing color mapping relationships. Local laplacian filter [1] and Guided filter [10] enhance the image visual quality by operations such as detail smoothing/sharpening. However, only experienced experts can use these hand-crafted feature descriptors or filters, and they are costly in time.

B. Learning-Based Algorithms

Recently, the learning-based image enhancement algorithm has been developed rapidly. These main algorithms can be categorized into three paradigms. Reinforcement learning-based methods, image-to-image translation methods, and physical modeling-based methods. Typical approaches are summarized in Tab. I.

1) Reinforcement Learning-Based Methods: The enhancement methods based on reinforcement learning [13], [20], [34], [49] improve the visual quality by simulating the human step-by-step retouching process. Typically, White-Box [13] proposes to decouple the image enhancement process into a series of suitable parameters. And the deep reinforcement learning approach is used to learn the decision of what action to take next in the current state. Distort-and-Recover [34] casts a color enhancement process as a Markov Decision Process where actions are defined as global color adjustment operations. The agent is then trained to learn the optimal global enhancement sequence of the actions. DeepExposure [49] and UIE [20] employ similar reinforcement learning strategies to learn an unpaired photo-enhanced model in an adversarial manner and severely sacrifice efficiency.

2) Image-to-Image Translation Methods: The Image-to-image translation enhancement methods [3], [4], [15], [16], [27], [28], [50], [53] learn a mapping relationship between input and enhanced images through convolutional networks. Representatively, with the popularity of generative adversarial mechanisms [6], some existing works [14], [27], [53] use the residual network [11] to style transfer and enhance unpaired images. Pix2Pix [15] investigates conditional adversarial networks as a general-purpose solution to image-to-image translation problems and learns a loss function to train the mapping from the input image to the output image. MIEGAN [33] presents a multi-module cascade generative network and an adaptive multi-scale discriminative network to capture both global and local information of a mobile image. DPE [4] improves U-Net [37] into a photo enhancer that transforms an input image into an enhanced image with the characteristics of given a set of photographs. GSGN [19] proposes the first practical multi-task image enhancement network, that is able to learn one-to-many and many-to-one image mappings. Focusing on the fact that subjectively people have diverse preferences for image aesthetics, PieNet [17] proposes the first deep learning method for personalized image enhancement that can enhance images for users by selecting preferences. MIRNet [50] proposes an architecture with the collective goals of maintaining spatially-precise high-resolution representations through the entire network and receiving strong contextual information from the low-resolution representations. CSRNet [9], [28] analyzes the mathematical formulation of image enhancement and proposes a lightweight framework consisting of 1×1 convolutional layer. However, these methods suffer from a lack of transparency in the whole enhancement process and obscure their reliability.

3) Physical Modeling-Based Methods: Inspired by the parametric (graduated, radial filters), brush tools, etc. in professional-grade software (*e.g.,* Photoshop, Lightroom), the enhancement based on physical model [5], [7], [8], [23], [31], [32], [40], [43], [51] adjust the image color by predicting the relevant physics parameters and variables based on the proposed physical model and assumptions. Inspired by bilateral grid processing and local affine color transforms, HDRNet [5] predicts the coefficients of a locally-affine model in bilateral space to approximate the desired image transformation. RetinexNet [24], [43] assumes that observed images can be decomposed into reflectance and illumination to enhance the low-light image. DeepUPE [40] introduces intermediate illumination to associate the input with expected enhancement results for learning complex photographic adjustments. Zero-DCE [7] formulates light enhancement as a task

TABLE I
Image Enhancement Methods Based on Deep Learning. RL: Reinforcement Learning-Based. I2I: Image-to-Image Translation. PM: Physical Modeling-Based

Method	Date	Characteristics
White-Box [13]	TOG2018	RL, Decouple Enhancement
Dis-and-Rec [34]	CVPR2018	RL, Markov Decision
DeepExpo. [49]	NeurIPS2018	RL, GAN, Exposure
UIE [20]	AAAI2020	RL, GAN, Unpaired
Pix2Pix [15]	CVPR2017	I2I, Conditional GAN
DPE [4]	CVPR2018	I2I, U-Net, WGAN, Unpaired
GSGN [19]	ECCV2020	I2I, GAN, One2Mang&Many2One
PieNet [17]	ECCV2020	I2I, Personalized
MIRNet [50]	ECCV2020	I2I, Multi-resolution, Multi-scale
MIEGAN [33]	TMM2021	I2I, DSLR camera, GAN
CSRNet [28]	TMM2022	I2I, Lightweight, MLP
HDRNet [5]	TOG2017	PM, Bilateral filtering
RetinexNet [43]	BMVC2018	PM, Retinex theory, LOL dataset
DeepUPE [40]	CVPR2019	PM, Underexposed enhancement
Zero-DCE [7]	CVPR2020	PM, Zero-reference, Curve
DeepLPF [31]	CVPR2020	PM, Parametric filtering
3D LUT [51]	TPAMI2020	PM, LUT-based, high-resolution
CURL [32]	ICPR2021	PM, Multi-colour space, Curve
SA-3D LUT [41]	CVPR2021	PM, LUT-based, Spatial-aware
SCI [30]	CVPR2022	PM, Self-calibrated illumination

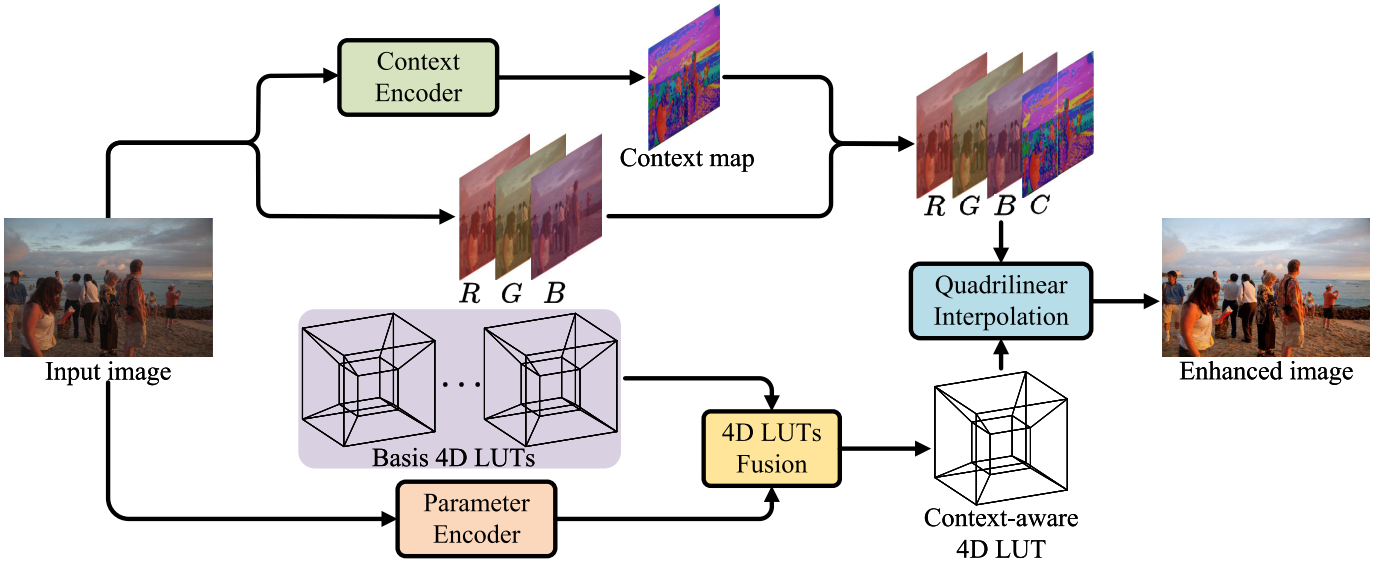


Fig. 2. The overview of 4D LUT. 4D LUT takes an input image and pre-defined basis 4D LUTs as input and generates an enhanced image. Context encoder for generating the pixel-level content-dependent context map. Parameter encoder for generating image-adaptive coefficients. 4D LUTs fusion module integrates the learnable basis 4D LUTs and coefficients into a context-aware 4D LUT. Quadrilinear interpolation module for generating the enhanced image by inputting them. Constrained by the optimization objective, the parameters of the entire network, including the context-aware 4D LUT, can be trained end-to-end.

of image-specific curve estimation and estimates pixel-wise and high-order curves for dynamic range adjustment of an input image. SCI [30] establishes a cascaded illumination learning process with weight sharing to handle the low-light enhancement task. DeepLPF [31] proposes learnable spatially local filters of three different types (Elliptical Filter, Graduated Filter, Polynomial Filter) and regresses the parameters of these filters that are then automatically applied to enhance the image. Inspired by the Photoshop curves tool, CURL [32] designs a multi-color space neural retouching block and adjusts global image properties using human-interpretable image enhancement curves.

Especially, some 3D LUT-based works [25], [41], [47], [51] adopt a two-step solution, which includes 1) predicting the relevant coefficients based on the 3D LUT to obtain an enhancer, and 2) adjusting the color based on the RGB value of each original pixel one by one to form an enhanced image through the uniform enhancer. Although these methods spend less runtime on the enhancement process, they can only achieve a globally overall average over all regions of the enhanced result and lack the ability to finely control the color transformation of pixels with different content in each image. Therefore, in this paper, we extend the 3D LUT into a 4D space and guide the content-dependent image enhancement by the additional dimensions (*i.e.,* context information) introduced.

III. Methodology

In this section, to assist the understanding of the new proposed 4D LUT, we first briefly review the preliminary about 3D LUT and trilinear interpolation. Then we describe the overview and each component of 4D LUT in detail.

A. Preliminary

1) 3D LUT: 3D LUT is a common tool used in different camera imaging pipeline systems and software for image enhancement, which can be adjusted manually or algorithmically to achieve different kinds of enhancements. A 3D LUT can be represented as a 3D lattice and the value of each element in the lattice can be represented as a triplet $(R_{out}^{(i,j,k)}, G_{out}^{(i,j,k)}, B_{out}^{(i,j,k)})$, where $i, j, k \in \{0, \ldots, N_{bin} - 1\}$, N_{bin} is the number of bins along each of three dimensions. Such a lattice includes a total of N_{bin}^3 sampling points, which form a complete 3D color transformation space. As shown in Fig. 1(a), the element (r_{in}, g_{in}, b_{in}) input to this color space can be mapped to an index by uniformly discretizing the RGB color space. The corresponding transformed output RGB color $(r_{out}, g_{out}, b_{out})$ is the value corresponding to the index. It is worth noting that as the value of N_{bin} increases, the 3D color transformation space becomes more accurate for color transformation and vice versa.

2) Trilinear Interpolation: As described above, the distribution of the elements in the LUT is discrete in space and it cannot be sampled directly by the input index. Therefore, when sampling elements in the 3D LUT, the input color will find its nearest sample point based on its index and calculate its transformed output by trilinear interpolation [25], [51].

Specifically, to locate the nearest 8 adjacent elements around input index, we first construct the input index (x, y, z) to the 3D LUT based on the input RGB color $(r_{in}^{(x,y,z)}, g_{in}^{(x,y,z)}, b_{in}^{(x,y,z)})$, which process can be described as follows:

$$x = r_{in}^{(x,y,z)} \cdot \frac{N_{bin}}{255}, y = g_{in}^{(x,y,z)} \cdot \frac{N_{bin}}{255}, z = b_{in}^{(x,y,z)} \cdot \frac{N_{bin}}{255}, \tag{1}$$

where N_{bin} is the number of bins along each of three dimensions in 3D LUT. We use (i, j, k) to denote the location of the defined sampling point, which can be calculated as follows:

$$i = \lfloor x \rfloor, j = \lfloor y \rfloor, k = \lfloor z \rfloor, \tag{2}$$

where $\lfloor\cdot\rfloor$ represents the floor function. We use (o_x, o_y, o_z) denote the offset of the input index (x, y, z) to defined sampling point (i, j, k), which can be calculated as follows:

$$o_x = x - i, o_y = y - j, o_z = z - k. \tag{3}$$

Then, taking the red color $r_{out}^{(x,y,z)}$ in transformed output RGB color $(r_{out}^{(x,y,z)}, g_{out}^{(x,y,z)}, b_{out}^{(x,y,z)})$ as an example, the interpolation process can be expressed as:

$$\begin{aligned} r_{out}^{(x,y,z)} = {} & (1-o_x)(1-o_y)(1-o_z) r_{out}^{(i,j,k)} \\ & + o_x o_y o_z r_{out}^{(i+1,j+1,k+1)} \\ & + o_x (1-o_y)(1-o_z) r_{out}^{(i+1,j,k)} \\ & + (1-o_x) o_y o_z r_{out}^{(i,j+1,k+1)} \\ & + (1-o_x) o_y (1-o_z) r_{out}^{(i,j+1,k)} \\ & + o_x (1-o_y) o_z r_{out}^{(i+1,j,k+1)} \\ & + (1-o_x)(1-o_y) o_z r_{out}^{(i,j,k+1)} \\ & + o_x o_y (1-o_z) r_{out}^{(i+1,j+1,k)} \end{aligned} \tag{4}$$

where the input $r_{out}^{(\{i,i+1\},\{j,j+1\},\{k,k+1\})}$ is the transformed output red color corresponding to the defined sampling point $(\{i, i+1\}, \{j, j+1\}, \{k, k+1\})$. Similarly, it can also obtain the other colors (*i.e.,* $g_{out}^{(x,y,z)}$ and $b_{out}^{(x,y,z)}$) in the same way. It is worth noting that the interpolation operation is differentiable and can update of LUTs during end-to-end training.

B. Context-Aware 4D LUT

Existing works [25], [41], [47], [51] improve the 3D LUT to enhance the image, and lack a necessary content information in the images. Therefore, we propose the learnable context-aware 4D LUT to achieve content-dependent image enhancement and enable finer control of color transformations for pixels with different content in each image.

As shown in Fig. 2, our method takes the input image and several pre-defined basis 4D LUTs as input, and finally generates an enhanced image. It includes four closely-related components, context encoder, parameter encoder, 4D LUTs fusion module, and quadrilinear interpolation module. Specifically, it involves the following stages: 1) We first use a context encoder to generate a context map that represents the pixel-level category from the input image through end-to-end learning. 2) Parallelly, we use a parameter encoder for generating image-adaptive coefficients applied to fuse the learnable pre-defined basis 4D LUTs. 3) Then, based on the output of the parameter encoder, we use the 4D LUTs fusion module to integrate the learnable basis 4D LUTs into a final context-aware 4D LUT with more enhanced capabilities. 4) Finally, the original input image and the context-aware 4D LUT are input to the quadrilinear interpolation module to obtain the enhanced image. In the following, we will describe them individually. More detailed descriptions also can be found in Alg. 1.

Algorithm 1 4D LUT Algorithm.

Input: Input image $I_{input} \in R^{3\times H\times W}$;
Context encoder $E_{context}(\cdot)$;
Parameter encoder $E_{param}(\cdot)$;
Basis 4D LUTs $\Psi_{n\in\{1,\ldots,N_{lut}\}} = (\psi_n^r, \psi_n^g, \psi_n^b)_{n\in\{1,\ldots,N_{lut}\}}$;
Quadrilinear interpolation $QI_{\hat{\Psi}}(\cdot)$;
Concatenation operation $Concat(\cdot)$;
The number of basis 4D LUTs N_{lut}.
Output: Enhanced image $I_{output} \in R^{3\times H\times W}$.
1: Generate context map C:
$C = E_{context}(I_{input})$, where $C \in R^{1\times H\times W}$;
2: Generate image-adaptive weights $\mathcal{W}$ and biases $\mathcal{B}$:
$\mathcal{W}, \mathcal{B} = E_{param}(I_{input})$, where
$\mathcal{W} = \{w_1, w_2, \ldots, w_{N_w}\}, N_w = 9N_{lut}$ and
$\mathcal{B} = \{b_1, b_2, \ldots, b_{N_b}\}, N_b = N_{lut}$;
3: Integrate the context-aware 4D LUT $\hat{\Psi} = (\hat{\psi}^r, \hat{\psi}^g, \hat{\psi}^b)$:
$\hat{\psi}^r = \sum_{n=1}^{N_{lut}} (w_n \psi_n^r + w_{(N_{lut}+n)} \psi_n^g + w_{(2N_{lut}+n)} \psi_n^b + b_n)$
$\hat{\psi}^g = \sum_{n=1}^{N_{lut}} (w_{3N_{lut}+n} \psi_n^r + w_{(4N_{lut}+n)} \psi_n^g + w_{(5N_{lut}+n)} \psi_n^b + b_n)$
$\hat{\psi}^b = \sum_{n=1}^{N_{lut}} (w_{6N_{lut}+n} \psi_n^r + w_{(7N_{lut}+n)} \psi_n^g + w_{(8N_{lut}+n)} \psi_n^b + b_n)$;
4: Output the enhanced image $I_{output} \in R^{3\times H\times W}$:
$I_{output} = QI_{\hat{\Psi}}(Concat(I_{input}, C))$.

1) Context Encoder: Context encoder can adaptively generate content-dependent context maps under the constraints of the objective function in a learnable manner. We use $E_{context}(\cdot)$ to denote the context encoder, which consists of a series of stacked residual blocks. In detail, it includes four residual blocks with 3×3 kernel size and one residual block with 1×1 size. Among them, the 3×3 residual blocks are applied to extract the high-level image features of the same resolution from the input image, and the 1×1 residual block compresses the image features and is used to output the context map. Suppose that an input image $I_{input} \in R^{3\times H\times W}$ are given. The generated context map $C \in R^{1\times H\times W}$ can be formulated as:

$$C = E_{context}(I_{input}). \tag{5}$$

Intuitively, the context encoder can be viewed as a function that maps the content information for each position in the input image to a compact scalar representation. During training, the context encoder is updated by the back-propagated gradient of the module connected behind. During inference, the more appropriate context map for enhancement is generated adaptively according to the high-level semantic differences of different regions.

2) Parameter Encoder: To facilitate the fusion of multiple pre-defined basis 4D LUTs and increase the 4D LUT enhancement capability, the parameter encoder extracts a group of image-adaptive coefficients for 4D LUT fusion during end-to-end training. We use $E_{param}(\cdot)$ to denote the parameter encoder, which consists of a series of stacked residual blocks and a parameters output layer consisting of a convolutional layer. In detail, suppose that an input image $I_{input} \in R^{3\times H\times W}$ are given. The generated parameters $\mathcal{W} \in R^{N_w\times 1\times 1}$ and $\mathcal{B} \in R^{N_b\times 1\times 1}$ can be formulated as:

$$\mathcal{W}, \mathcal{B} = E_{param}(I_{input}), \tag{6}$$

where $\mathcal{W} = \{w_1, w_2, \ldots, w_{N_w}\}$ and $\mathcal{B} = \{b_1, b_2, \ldots, b_{N_b}\}$ represent the outputted coefficients (*i.e.,* weights and biases)

to fuse the learnable basis 4D LUTs, respectively. N_w and N_b are the number of weights and biases in the coefficients, respectively. If the number of basis 4D LUTs is assumed to be N_{lut}, then the values of N_w and N_b are $9N_{lut}$ and N_{lut}, respectively.

During training, the parameter encoder is updated by the back-propagated gradient of the module connected behind. During inference, the parameter encoder can be viewed as a parameter predictor that integrates the learnable basis 4D LUTs in a soft-weighting strategy to achieve adaptive context-aware 4D LUT generation for better image enhancement.

3) 4D LUTs Fusion: During the end-to-end training process, the elements of per-defined basis 4D LUTs are gradually updated to adapt to the change of color space. To obtain a context-aware 4D LUT with stronger color transformation capabilities, the 4D LUTs fusion module fuses multiple learnable basis 4D LUTs by using the coefficients obtained from the parameter encoder.

Specifically, we use $\Psi_{n\in\{1,\ldots,N_{lut}\}}$ to denote one of the multiple basis 4D LUTs, where N_{lut} is the number of basis 4D LUTs. As described in Sec. III-A.1, the value of each element in Ψ_n can be represented as a triplet (*i.e.,* red, green, and blue color spaces), in which we use ψ_n^r, ψ_n^g, and ψ_n^b to denote the corresponding red, green, and blue color spaces in Ψ_n, respectively. Besides, we use $\hat{\Psi} = (\hat{\psi}^r, \hat{\psi}^g, \hat{\psi}^b)$ to represent the fused context-aware 4D LUT. Taking the fusion process of red space $\hat{\psi}^r$ as an example, it can be generated by:

$$\begin{aligned}\hat{\psi}^r = &\textstyle\sum_{n=1}^{N_{lut}}(w_n\psi_n^r + w_{(N_{lut}+n)}\psi_n^g \\ &+ w_{(2N_{lut}+n)}\psi_n^b + b_n),\end{aligned} \tag{7}$$

where w and b are the weights and biases output from the parameter encoder. Similarly, we can also obtain the other colors space (*i.e.,* $\hat{\psi}^g$ and $\hat{\psi}^b$) in the same way.

In general, the addition of weights allows the different color spaces to interact and fuse with each other, resulting in a more appropriate color temperature (similar to white balance). The addition of biases can adaptively enhance the overall brightness of the image. Such a design on fusion approach makes the fused context-aware 4D LUT with more superior enhancement capability.

4) Quadrilinear Interpolation: Based on the original RGB image, the generated context map, and context-aware 4D LUT, we can obtain the enhanced image via interpolation operation. However, different from the 3D LUT and trilinear interpolation described in Sec. III-A, the index of our proposed 4D LUT is on the 4-dimensional space (*i.e.,* RGB+Context).

To effectively interpolate the values based on the index of 4-dimensional RGBC space, we propose to use a quadrilinear interpolation closely related to the 4D LUT. Suppose that an input image $I_{input} \in R^{3\times H\times W}$ are given, the context map $C \in R^{1\times H\times W}$ and context-aware 4D LUT are generated, the enhanced image $I_{output} \in R^{3\times H\times W}$ can be formulated as:

$$I_{output} = QI_{\hat{\Psi}}(Concat(I_{input}, C)), \tag{8}$$

where $QI_{\hat{\Psi}}(\cdot)$ denotes the quadrilinear interpolation based on the context-aware 4D LUT $\hat{\Psi}$. $Concat(\cdot)$ is the concatenation operation.

Specifically, similar to in Sec. III-A.2 above, the input index to the 4D LUT based on the input RGBC value can be represented as (x, y, z, u), we first locate the nearest 16 adjacent elements around the input index as the sampling point (*i.e.,* the distance from input index satisfies $\Delta \in [-1, 1]$ in each dimension). We use (i, j, k, l) to denote the coordinates of a defined sampling point in 4D LUT, which can be calculated as follows:

$$i = \lfloor x \rfloor, j = \lfloor y \rfloor, k = \lfloor z \rfloor, l = \lfloor u \rfloor, \tag{9}$$

where $\lfloor\cdot\rfloor$ represents the floor function. The defined nearest 16 adjaent sampling point in 4D LUT are $(\{i, i+1\}, \{j, j+1\}, \{k, k+1\}, \{l, l+1\})$. We use (o_x, o_y, o_z, o_u) denote the offset of the input index (x, y, z, u) to defined sampling point (i, j, k, l), which can be calculated as follows:

$$o_x = x - i, o_y = y - j, o_z = z - k, o_u = u - l. \tag{10}$$

Then, the red color $r_{out}^{(x,y,z,u)}$ in transformed output RGB color $(r_{out}^{(x,y,z,u)}, g_{out}^{(x,y,z,u)}, b_{out}^{(x,y,z,u)})$ can be expressed as:

$$\begin{aligned}r_{out}^{(x,y,z,u)} = &(1-o_x)(1-o_y)(1-o_z)(1-o_u)r_{out}^{(i,j,k,l)} \\ &+ o_x(1-o_y)(1-o_z)(1-o_u)r_{out}^{(i+1,j,k,l)} \\ &+ (1-o_x)o_y(1-o_z)(1-o_u)r_{out}^{(i,j+1,k,l)} \\ &+ (1-o_x)(1-o_y)o_z(1-o_u)r_{out}^{(i,j,k+1,l)} \\ &+ (1-o_x)(1-o_y)(1-o_z)o_u r_{out}^{(i,j,k,l+1)} \\ &+ o_x o_y(1-o_z)(1-o_u)r_{out}^{(i+1,j+1,k,l)} \\ &+ (1-o_x)o_y o_z(1-o_u)r_{out}^{(i,j+1,k+1,l)} \\ &+ (1-o_x)(1-o_y)o_z o_u r_{out}^{(i,j,k+1,l+1)} \\ &+ o_x(1-o_y)o_z(1-o_u)r_{out}^{(i+1,j,k+1,l)} \\ &+ o_x(1-o_y)(1-o_z)o_u r_{out}^{(i+1,j,k,l+1)} \\ &+ (1-o_x)o_y(1-o_z)o_u r_{out}^{(i,j+1,k,l+1)} \\ &+ o_x o_y o_z(1-o_u)r_{out}^{(i+1,j+1,k+1,l)} \\ &+ (1-o_x)o_y o_z o_u r_{out}^{(i,j+1,k+1,l+1)} \\ &+ o_x o_y(1-o_z)o_u r_{out}^{(i+1,j+1,k,l+1)} \\ &+ o_x(1-o_y)o_z o_u r_{out}^{(i+1,j,k+1,l+1)} \\ &+ o_x o_y o_z o_u r_{out}^{(i+1,j+1,k+1,l+1)},\end{aligned} \tag{11}$$

where the input $r_{out}^{(\{i,i+1\},\{j,j+1\},\{k,k+1\},\{l,l+1\})}$ is the transformed output red color corresponding to the defined nearest 16 adjacent sampling point $(\{i, i+1\}, \{j, j+1\}, \{k, k+1\}, \{l, l+1\})$ in 4D LUT. Similarly, we can also obtain the other colors (*i.e.,* $g_{out}^{(x,y,z)}$ and $b_{out}^{(x,y,z)}$) in the same way. The quadrilinear interpolation operation is differentiable and can

propagate the gradient to update the weight of the network and the element values of context-aware 4D LUT.

C. Training

To update the elements of 4D LUT and the parameters of the network, in this section, we employ several objective functions to supervise the whole training process.

1) 4D Smooth Regularization: The parameters of 4D LUT correspond to the values in the output color space. The color transformation for closer input colors with unsmoothed 4D LUT may produce extreme color changes, reducing the robustness of the model and producing artifacts. Therefore, we use 4D smooth regularization to ensure that the converting from the input space (*i.e.,* RGBC) to the obtained color space (*i.e.,* RGB) is stable enough. We introduce L_2-norm regularization on the elements of the 4D LUT and the outputted coefficient of the parameter encoder to improve the smoothness of the context-aware 4D LUT.

Specifically, inspired by existing work [51], we extend the 3D smooth regularization into a 4D smooth regularization term on the learning of 4D LUT to ensure the local smoothing of the elements in 4D LUT. The smooth regularization of 4D LUT L_s^{lut} can be calculated as follow:

$$\begin{aligned} L_s^{lut} = \sum_{p\in\{r,g,b\}} \sum_{i,j,k,l=0}^{N_{bin}-1} (&\left\| p_{out}^{(i+1,j,k,l)} - p_{out}^{(i,j,k,l)} \right\|^2 \\ &+ \left\| p_{out}^{(i,j+1,k,l)} - p_{out}^{(i,j,k,l)} \right\|^2 + \left\| p_{out}^{(i,j,k+1,l)} \right. \\ &\left. - p_{out}^{(i,j,k,l)} \right\|^2 + \left\| p_{out}^{(i,j,k,l+1)} - p_{out}^{(i,j,k,l)} \right\|^2), \end{aligned} \tag{12}$$

where N_{bin} is the number of bins along each of the output dimensions in 4D LUT. The input $p_{out}^{(\{i,i+1\},\{j,j+1\},\{k,k+1\},\{l,l+1\})}$ is the transformed output red, green, and blue color corresponding to the defined sampling point $(\{i, i+1\}, \{j, j+1\}, \{k, k+1\}, \{l, l+1\})$ in 4D LUT.

The smooth regularization of outputted coefficient L_s^{coe} can be calculated as follow:

$$L_s^{coe} = \sum_{n=1}^{N_w} \left\| w_n \right\|^2 + \sum_{m=1}^{N_b} \left\| b_n \right\|^2, \tag{13}$$

where N_w and N_b are the numbers of weights and biases in the coefficients, respectively. w_n and b_m represent the outputted image-adaptive weights and biases from the parameter encoder.

The overall smooth regularization term L_s can be represented as:

$$L_s = L_s^{lut} + L_s^{coe}. \tag{14}$$

This design makes the elements in 4D LUT locally smoother and ensures the stability of color transformation.

2) 4D Monotonicity Regularization: The values output from the enhanced image should have the ability to cover the entire RGBC space and preserve the robustness and relative color brightness/saturation in the enhancement process. Therefore, to enable the color space output from 4D LUT satisfy the above requirements and converge rapidly, we followed existing work [51] to expand the 3D monotonicity regularization into 4D monotonicity regularization term L_m as follows:

$$\begin{aligned} L_m = \sum_{p\in\{r,g,b\}} \sum_{i,j,k,l=0}^{N_{bin}-1} [&g(p_{out}^{(i,j,k,l)} - p_{out}^{(i+1,j,k,l)}) \\ &+ g(p_{out}^{(i,j,k,l)} - p_{out}^{(i,j+1,k,l)}) + g(p_{out}^{(i,j,k,l)} \\ &- p_{out}^{(i,j,k+1,l)}) + g(p_{out}^{(i,j,k,l)} - p_{out}^{(i,j,k,l+1)})], \end{aligned} \tag{15}$$

where $g(\cdot)$ denotes the ReLU activation function (*i.e.,* $g(x) = max(0, x)$). N_{bin} is the number of bins along each of the output dimensions in 4D LUT. The input $p_{out}^{(\{i,i+1\},\{j,j+1\},\{k,k+1\},\{l,l+1\})}$ is the transformed output red, green, and blue color corresponding to the defined sampling point $(\{i, i+1\}, \{j, j+1\}, \{k, k+1\}, \{l, l+1\})$ in 4D LUT. This design not only allows the color transformation to cover the entire RGBC space, but also allows the 4D LUT to converge faster during training.

3) Pairwise Reconstruction: The aim of image enhancement is to keep the enhanced image as close as possible to the ground truth image, thus we also incorporate a pixel-level reconstruction loss function. Specifically, for fair comparisons, we follow previous works [5], [28], [51] to define the reconstruction loss L_r between the ground truth I_{GT} and enhanced image I_{output} to train the whole model, it is defined as:

$$L_r = \frac{1}{N_{bs}} \sum_{1}^{N_{bs}} (I_{GT} - I_{output})^2, \tag{16}$$

where N_{bs} represents the batch size during training.

4) Loss Function: As described above, during training, our approach includes a total of three loss components, the 4D smooth regularization loss, the 4D monotonicity regularization loss, and the pairwise reconstruction loss. It is essential for the network to balance these three items. Therefore, we multiply L_s and L_m (as described in Eqn. 14 and 15) with a weight α_s and α_m, respectively, to enable the validity of context-aware 4D LUT while not harming the performance of enhancement. The total loss function is formulated as follows:

$$L_{total} = L_r + \alpha_s L_s + \alpha_m L_m. \tag{17}$$

IV. Experiments

A. Experimental Settings

1) Datasets: We evaluate the proposed 4D LUT and compare its performance with other state-of-the-art (SOTA) approaches on three widely-used challenging benchmarks, derived from two public datasets: **MIT-Adobe-5K-UPE** [40], **MIT-Adobe-5K-DPE** [4], and **PPR10K** [25].

a) MIT-Adobe-5K-UPE: a benchmark is divided from the MIT-Adobe FiveK dataset [2], following the dataset pre-processing procedure of DeepUPE [40]. MIT-Adobe FiveK dataset is a commonly-used landscape photo retouching dataset with 5,000 images captured using various DSLR cameras. Each image contains the corresponding retouched version produced by five experienced experts (A/B/C/D/E). For fair comparisons, we follow previous works [4], [31], [32] to use the photo retouched by expert C as image enhancement

ground truth (GT). We select 4,500 image pairs as the training set and 500 image pairs as the test set in order, and all images are resized to 510 pixels on the long edge.

b) MIT-Adobe-5K-DPE: another benchmark consists of the same image context as the MIT-Adobe FiveK dataset [2] however following the dataset pre-processing procedure of DPE [4]. We follow previous works [31], [32], [40] to use the photo retouched by expert C as ground truth (GT). The difference is that we sequentially select 2,250 image pairs as the training set and 500 image pairs as the test set.

c) PPR10K: a new large-scale portrait retouching dataset to be released in 2021, containing a total of 11,161 high-quality RAW portraits. Each image contains the corresponding retouched version produced by three experienced experts (a/b/c). For fair comparisons, we follow the official split [25] to divide the dataset into 8,875 training pairs and 2,286 test pairs. We compare the results on all expert modifications, and the image size is 360p.

2) Evaluation Metrics: For fair comparisons, we follow previous enhancement works [5], [28], [51] to use peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [42] as a commonly-used metric for evaluating in terms of the color and structure similarity between the enhanced results and the corresponding expert-retouched images. Besides, we also added the widely used metric LPIPS [52] to evaluate perceptual quality, and BRISQUE and NIQE to evaluate objective quality.

3) Implementation Details: Our experiment is conducted on an NVIDIA 2080Ti GPU through PyTorch. For fair comparisons, we follow previous works [25], [41], [47], [51] and use the Adam optimizer [18] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and the batch size of 1. The initial learning rate is set as 1×10^{-4} and then reduce the learning rate by a factor of 0.2 when the losses of the testing set last for 20 epochs without decreasing. We jointly train the entire model for 400 epochs. Except for adding random crop image patches with a scale in the range [0.6, 1.0], horizontal flip, and no other data augmentation methods are used.

Besides, we follow previous works [25], [51] to set the the number of bins N_{bin} in LUT and the number of basis 4D LUT N_{lut} as 33 and 3, respectively. We set the number of weights N_w and biases N_b as 27 and 3, respectively. We empirically set α_s and α_m as 0.0001 and 10 through discussion experiments.

B. Comparison With State-of-the-Art Methods

We compare our 4D LUT with other classical start-of-the-art methods. These methods can be summarized into three categories: reinforcement learning-based methods (*i.e.,* White-Box [13], Dis-Rec [34], and UIE [20]), image-to-image translation methods (*i.e.,* DPED [14], 8Resblock [27], [53], CRN [3], U-Net [37], DPE [4], GSGN [19], MIRNet [50], and CSRNet [9], [28]), and physical modeling-based methods (*i.e.,* HDRNet [5], DeepUPE [40], DeepLPF [31], TED+CURL [32], 3D LUT [51], 3D LUT+HRP [25]). For fair comparisons, we obtain the performance from their original paper or reproduce results with recommended configurations by the authors' officially released models.

TABLE II

QUANTITATIVE COMPARISON (PSNR↑ AND SSIM↑) ON THE MIT-ADOBE-5K-UPE [40] DATASET. RED INDICATES THE BEST AND BLUE INDICATES THE SECOND BEST PERFORMANCE (BEST VIEW IN COLOR)

Method	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓
HDRNet [5]	21.96	0.866	0.0932	13.9294	3.8305
U-Net [37]	22.24	0.850	0.1032	19.8553	4.3021
DPE [4]	22.15	0.850	0.0914	17.6894	4.2424
White-Box [13]	18.57	0.701	-	-	-
Dis-Rec [34]	20.97	0.841	-	-	-
DeepUPE [40]	23.04	0.893	0.0912	-	-
MIRNet [50]	23.73	0.897	0.0869	19.7178	4.2253
TED+CURL [32]	24.20	0.880	0.0702	16.2644	4.1855
DeepLPF [31]	24.48	0.887	0.0871	16.6778	4.5587
CSRNet [9], [28]	24.23	0.900	0.0573	13.6211	3.7012
3D LUT [51]	24.60	0.911	0.0491	13.5290	3.6794
SA-3D LUT [41]	24.68	0.912	0.0403	13.9185	3.7855
4D LUT(Ours)	24.96	0.924	0.0371	13.3830	3.6383

TABLE III

QUANTITATIVE COMPARISON (PSNR↑ AND SSIM↑) ON THE MIT-ADOBE-5K-DPE [4] DATASET. RED INDICATES THE BEST AND BLUE INDICATES THE SECOND BEST PERFORMANCE (BEST VIEW IN COLOR)

Method	PSNR↑	SSIM↑	LPIPS↓	BRISQUE↓	NIQE↓
DPED [14]	21.76	0.871	0.1132	18.2390	4.4646
8RBs [53], [27]	23.42	0.875	0.1024	17.0893	4.3956
CRN [3]	22.38	0.877	-	-	-
U-Net [37]	22.13	0.879	0.1032	19.3325	4.5785
White-Box [13]	21.32	0.864	-	-	-
Dis-Rec [34]	21.60	0.875	-	-	-
DPE [4]	23.80	0.900	0.0910	17.9856	4.3566
UIE [20]	22.27	0.881	-	-	-
DeepLPF [31]	23.93	0.903	0.0884	16.9891	4.5844
TED+CURL [32]	24.08	0.900	0.0722	16.9475	4.2943
GSGN [19]	24.16	0.905	-	-	-
3D LUT [51]	24.33	0.910	0.0419	14.3918	3.6232
SA-3D LUT [41]	24.40	0.909	0.0420	14.5422	3.7893
4D LUT(Ours)	24.61	0.918	0.0386	14.4742	3.7004

1) Quantitative Comparison: The results of each algorithm evaluated on datasets MIT-Adobe-5K-UPE [40] and MIT-Adobe-5K-DPE [4] are shown in Tab. II and Tab. III, respectively. Benefit from the strong capabilities of a pure CNN structure based on an image-to-image translation methods (*i.e.,* DPED [14], 8Resblock [27], [53], CRN [3], U-Net [37], DPE [4], GSGN [19], MIRNet [50], and CSRNet [9], [28]) perform image enhancement by designing huge and complex network models, whose performance is often positively correlated with the model size. The latest algorithm CSRNet [9], [28] designs a lightweight enhancement model using the 1×1 convolutional kernels, but still lacks enhancement capability. Besides, algorithms based on reinforcement learning (*i.e.,* White-Box [13], Dis-Rec [34], and UIE [20]) improve the enhancement capability by decoupling multiple steps and also achieve pleasant results, but the computational cost is too large. The physical model-based methods (*i.e.,* HDRNet [5], DeepUPE [40], DeepLPF [31], TED+CURL [32], 3D LUT [51], 3D LUT+HRP [25]) is based on theoretical physical models and assumptions that are transparent in the enhancement process. Representatively, the latest algorithm 3D LUT [51] generally performs better than the other methods. However, this method focus on learning

TABLE IV

QUANTITATIVE COMPARISON (PSNR↑ AND SSIM↑) ON THE PPR10K [25] DATASET. RED INDICATES THE BEST AND BLUE INDICATES THE SECOND BEST PERFORMANCE (BEST VIEW IN COLOR)

Method	Runtime	#Param	PPR10K-a			PPR10K-b			PPR10K-c		
			PSNR(dB)↑	SSIM↑	LPIPS↓	PSNR(dB)↑	SSIM↑	LPIPS↓	PSNR(dB)↑	SSIM↑	LPIPS↓
HDRNet [5]	6.03ms	482K	21.435	0.905	0.0796	21.609	0.907	0.0814	21.841	0.903	0.0745
DeepLPF [31]	51.3ms	1.7M	23.961	0.930	0.0409	22.556	0.919	0.0491	22.763	0.907	0.0553
TED+CURL [32]	82.3ms	1.4M	23.651	0.914	0.0583	23.324	0.911	0.0574	23.869	0.903	0.0557
CSRNet [9], [28]	2.50ms	36.4K	24.039	0.935	0.0389	24.066	0.939	0.0381	24.257	0.930	0.0377
3D LUT [51]	1.99ms	593.5K	24.632	0.937	0.0363	24.101	0.937	0.0382	24.515	0.924	0.0374
3D LUT+HRP [25]	1.99ms	593.5K	24.416	0.942	0.0366	23.985	0.941	0.0370	24.291	0.930	0.0363
SA-3D LUT [41]	7.31ms	4.52M	24.641	0.944	0.0352	24.211	0.940	0.0370	25.421	0.933	0.0360
4D LUT(Ours)	5.75ms	924.4K	24.915	0.944	0.0349	24.398	0.942	0.0361	24.733	0.933	0.0354

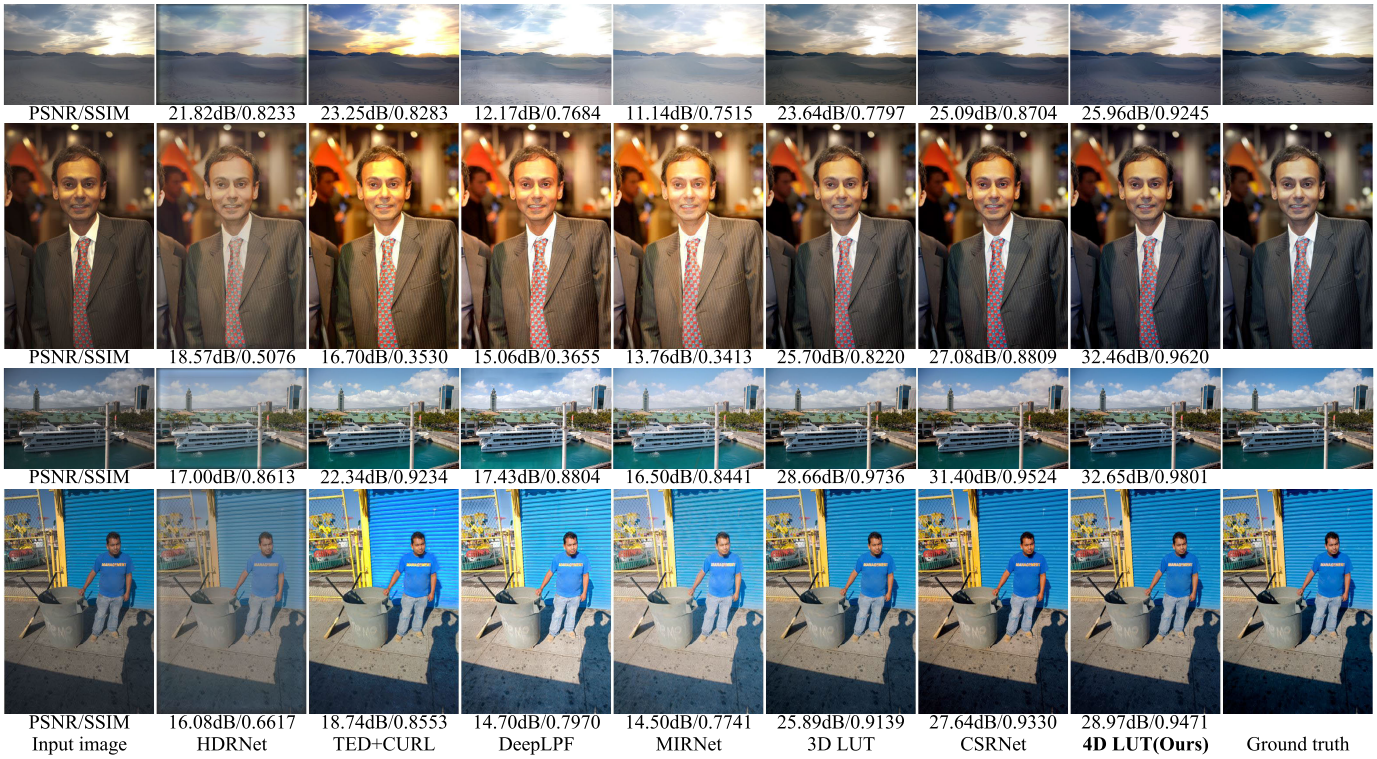


Fig. 3. Visual comparison with state-of-the-arts on MIT-Adobe-5K-UPE [40] dataset. The quantitative comparison (PSNR↑ and SSIM↑) is shown at the bottom of each case.

a uniform enhancer and achieving a globally overall average over all regions of the enhanced result that decrease the accuracy of enhancement and can lead to sub-optimal performance.

Our proposed 4D LUT extends the lookup table architecture into a 4-dimensional space and achieves content-dependent image enhancement. As shown in Tab. II and Tab. III, it achieves a result of 24.96dB and 24.61dB PSNR and significantly outperforms the other algorithms for all datasets by a large margin. Specifically, on the MIT-Adobe-5K-UPE [40] and MIT-Adobe-5K-DPE [4] datasets, 4D LUT outperforms 3D LUT [51] by **0.36dB** and **0.28dB**, respectively. Besides, our 4D LUT also has significant superiority in perceptual quality (*i.e.,* LPIPS [52]) and objective quality (*i.e.,* BRISQUE and NIQE). This large margin demonstrates the power of 4D LUT in image enhancement.

To further verify the generalization capabilities of 4D LUT, we evaluate 4D LUT on another larger-scale portrait photo retouching dataset PPR10K [25]. As shown in Tab. IV, due to the well-designed 4D LUT and the content-dependent learning capability, 4D LUT achieves better results in all three experts-retouch results, which outperforms other SOTA methods between **0.22dB** to **0.30dB**. The performances demonstrate that 4D LUT has strong generalization capabilities under different scenarios.

2) Qualitative Comparison: To further compare the visual qualities of different algorithms, we show visual results enhanced by proposed 4D LUT and other SOTA methods on different datasets in Fig. 3 and Fig. 4. For fair comparisons, we either directly take the original enhanced results of the author-released or use author-released models to get results.

It can be observed that 4D LUT has a great improvement in visual quality and evaluation metrics (*i.e.,* PSNR, SSIM, and LPIPS). For example, in the first row in Fig. 4, compared to other methods using a unified enhancer for landscapes and portraits, 4D LUT can simultaneously obtain both blue landscapes and comfortably colored portraits. In the third row

Fig. 4. Visual comparison with state-of-the-arts on PPR10K-a [25] dataset. The quantitative comparison (PSNR↑ and SSIM↑) is shown at the bottom of each case.

in Fig. 4, our 4D LUT can enable a stronger enhancement capability by introducing an additional contextual dimension, which can produce brighter green plants and portraits. As the analysis mentioned above, the results verify that 4D LUT has stronger enhancement capability and can achieve better results, especially for content-rich photos.

3) Complexity Analysis: Model sizes and inference time are usually important in real applications. We follow previous works [9], [28] to report them enhancing an image with 360p by using an RTX 2080Ti GPU. As shown in Tab. IV, compared with other SOTA methods, 4D LUT achieves higher performance while keeping comparable #Param. It should be emphasized that due to the expanded dimensionality, the parameters number of a 4D LUT are larger compared to the 3DLUT [25], [51] (*i.e.,* 216K *vs* 108K). Besides, 4D LUT is slower compared to 3D LUT [51] due to the generation of the context map and quadrilinear interpolation, but it also significantly exceeds the real-time runtime (*i.e.,* 30fps). It should be emphasized that the quadrilinear interpolation of each pixel is independent of the others, and this transformation can be easily parallelized using the GPU, thus not adding much extra time.

C. Ablation Study

In this section, we conduct ablation experiments on model design and loss function on the MIT-Adobe-5K-UPE [40] dataset.

1) Model Design: To demonstrate the effectiveness of each component in 4D LUT, we conduct ablation experiments for each component. The experimental results are shown in Tab. V. The "Base" indicates the result that no context encoder (*i.e.,* the context map is an all-zero image) and no parameter encoder (*i.e.,* directly summing the basis 4D LUTs to fuse). "CE" and "PE" indicate the context encoder and parameter encoder, respectively. The results show that the PSNR has increased by **1.34dB** by joining the CE. It demonstrates that the addition of the context encoder enables the network to learn content-dependent image enhancement, yielding stronger enhancement capabilities. With the addition of PE, PSNR can be increased by **2.01dB**, which verifies that the learnable image-adaptive coefficients can be better fused into context-aware 4D LUT, increasing the enhanced capability of the context-aware 4D LUT. When CE and PE are involved at the same time to boost each other, the performance is improved by **2.32dB**.

We further explore the visual differences as shown in Fig. 5, context encoder can produce content-dependent image enhancement, while the parameter encoder produces richer color. It demonstrates the superiority of each component

TABLE V

ABLATION STUDY OF EACH COMPONENT IN 4D LUT. CE: CONTEXT ENCODER. PE: PARAMETER ENCODER

Components			PSNR(dB)↑	SSIM↑	LPIPS↓
Base	CE	PE			
✓			22.64	0.895	0.0636
✓	✓		23.30	0.897	0.0491
✓		✓	24.65	0.920	0.0479
✓	✓	✓	**24.96**	**0.924**	**0.0371**

Fig. 5. Ablation study on the context encoder (CE) and parameter encoder (PE). ("4D LUT" can be interpreted as "Base+PE+CE").

TABLE VI

ABLATION STUDY OF LOSS FUNCTION USED IN 4D LUT. CE: CONTEXT ENCODER MODULE. PE: PARAMETER ENCODER MODULE

Loss function			PSNR(dB)↑	SSIM↑	LPIPS↓
L_r	L_s	L_m			
✓			24.74	0.923	0.0425
✓	✓		24.79	0.923	0.0395
✓		✓	24.81	0.924	0.0397
✓	✓	✓	**24.96**	**0.924**	**0.0371**

TABLE VII

RESULTS OF DIFFERENT NUMBER OF BINS N_{bin} IN 4D LUT ON MIT-ADOBE-5K-UPE [40] DATASET

N_{bin}	9	17	33	64
PSNR(dB)↑	24.67	24.79	24.96	25.03
SSIM↑	0.920	0.923	0.924	0.925

of 4D LUT, which can get better performance for image enhancement.

2) Loss Function: To demonstrate the effectiveness of each loss function in 4D LUT, we conduct ablation experiments for them. The experimental results are shown in Tab. VI. The "L_r" indicates the pairwise reconstruction loss. "L_s" and "L_m" indicate the 4D smooth regularization loss and 4D monotonicity regularization loss, respectively. With the addition of L_s, PSNR can be improved from 24.74dB to 24.79dB, which verifies that the 4D smooth regularization loss can ensure a stable transition from the input space (*i.e.,* RGBC) to the obtained color space (*i.e.,* RGB). When L_m is involved, the color transformation can reserve the relative color brightness/saturation and cover the entire RGBC space, and the performance is improved to 24.96dB. It demonstrates the superiority of each loss function of 4D LUT, which can get better performance for image enhancement.

V. DISCUSSIONS

In this section, to further demonstrate the reasonableness of 4D LUT, we first visualize the proposed context map and context-aware 4D LUT. Then we discuss the effect of bins N_{bin} in the 4D LUT and the number of basis 4D LUT N_{lut}. Finally, we discuss the sensitivity of smooth regularization weight α_s and monotonicity regularization weight α_m.

A. Visualization of Context Map

The context map is used to distinguish the high-level semantic differences between different regions. To explore the effectiveness of context map C in 4D LUT, we visualize it as shown in Fig. 6. Among them, we use eight kinds of colors to visualize the different contents in the generated context map from small to large, and it can be seen that the generated context map effectively divides the regions with different high-level semantic differences adaptively. Compared with the results of 3D LUT not involving the context map, the results (indicated by the red box) demonstrate that our context-aware 4D LUT is effective in achieving content-dependent enhancement with better results.

TABLE VIII

RESULTS OF DIFFERENT NUMBER OF BASIS 4D LUT N_{lut} ON MIT-ADOBE-5K-UPE [40] DATASET

N_{lut}	1	2	3	4	5
PSNR(dB)↑	22.92	24.27	24.96	25.11	25.16
SSIM↑	0.894	0.915	0.924	0.925	0.925

B. Visualization of Context-Aware 4D LUT

To better study this content-dependent enhancement property of context-aware 4D LUT, in Fig. 7, we visualize the context-aware 4D LUT for two different images and show their corresponding enhancement results. Besides, to visualize the differences of each of the R, G, and B channels on the context-aware 4D LUT more clearly, we fix the values of the context map to the maximum and minimum, respectively, and then visualize 17 slices (*i.e.,* {1, 3, ..., 31, 33}) of the whole 4D LUT (33 slices in total).

As shown in Fig. 7, it is observed that for images containing blue ocean and green grass, our method can adaptively generate different 4D LUT corresponding to different shapes. This demonstrates that 4D LUT has the ability to establish color transformation relationships for images with different contents. Besides, the shapes of visualized LUT corresponding to different C-dimensions in each image also have significant differences. This proves that the 4D LUT has the ability to perform different color transformations according to the different contents in each image, which enables finer control of color transformation and stronger image enhancement. It can be found that 4D LUT can obtain pleasing visual results for different images or different contents in each image.

C. Discussion on Number of Bins N_{bin} in 4D LUT

To explore the influence of the number of bins N_{bin} in 4D LUT on the enhancement effect. As shown in Tab. VII, we divide the 4D LUT into different number of bins (*i.e.,* {9, 17, 33, 64}). As the value of N_{bin} increases from 9 to 33, the PSNR is increased from 24.67dB to 24.96dB. It is because the increase in N_{bin} makes the interpolated elements used for color transformation more accurate, and vice versa. Besides, to prevent the 4D LUT from overfitting the color transformations of the training data and to preserve the generalization of the color transformations, we choose N_{bin} as 33 in our experiments.

D. Discussion on Number of Basis 4D LUTs N_{lut}

To explore the influence of the number of basis 4D LUTs N_{lut} used on the enhancement effect. As shown in Tab. VIII, we use different number of basis 4D LUTs (*i.e.,* {1, 2, 3, 4, 5}) to fuse into a context-aware 4D LUT that described in Sec. III-B.3. The performance is positively correlated with the number of basis 4D LUTs. It demonstrates using multiple basis 4D LUTs improves the expressiveness of color transformations. Besides, it can be seen easily that when the number of base 4D LUTs is increased from 1 to 3, its PSNR is significantly improved from 22.92dB to 24.96dB, while the

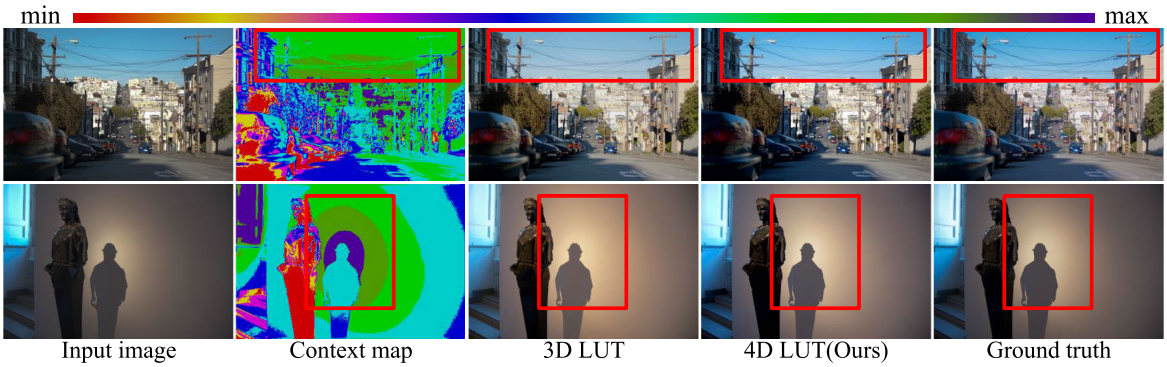


Fig. 6. Visualization of generated context map and enhanced results. The generated context map is visualized in eight kinds of colors from small to large according to the high-level semantic differences.

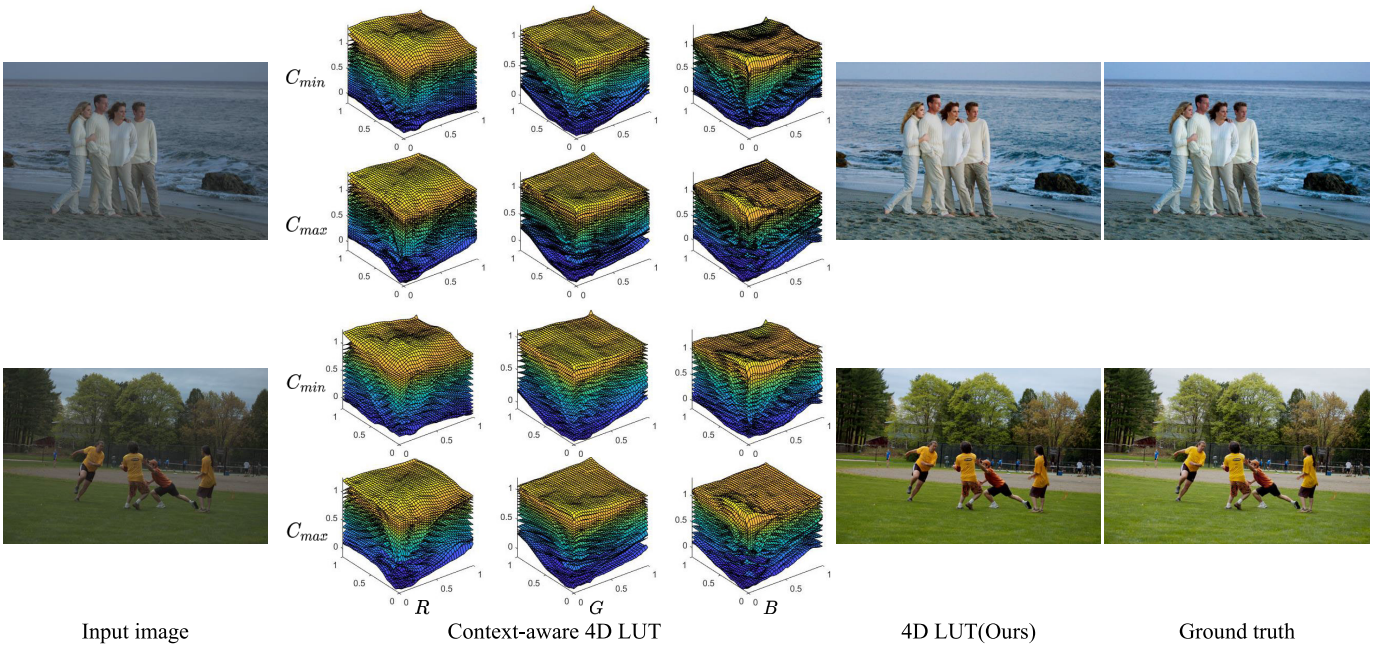


Fig. 7. Visualization of context-aware 4D LUT and enhanced results. The context-aware 4D LUT is visualized by dividing it into R, G, and B channels when the content values are minimum and maximum.

improvement becomes smaller when the basis 4DLUTs is further increased. Therefore, by the trade-off between the number of model parameters and performance, we experimentally set N_{lut} to 3.

E. Discussion on Smooth Regularization Weight α_s

As described in Sec. III-C.1 above, we use a 4D smooth regularization to ensure the stability of the locally color transformation. Therefore, we perform experiments by setting the smooth regularization weight α_s distributed in $\{0, e^{-5}, e^{-4}, e^{-3}, e^{-2}, e^{-1}\}$ to select the appropriate values in Eqn. 17. As shown in Fig. 8, it can be seen that when α_s is too large, excessive smoothing makes 4D LUT missing a detailed description of the color transformations, while reducing performance. On the contrary, when α_s is too small, insufficient smoothing makes the network lack the ability to generalize the color transformations. We set α_s as 0.0001 in the final experimentally.

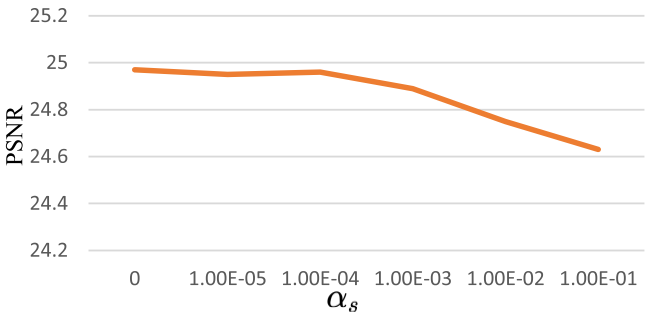


Fig. 8. Sensitivity of smooth regularization weight α_s.

F. Discussion on Monotonicity Regularization Weight α_m

To explore the influence of the monotonicity regularization weight on the color enhancement effect in Sec. III-C.2. We perform experiments by setting the smooth regularization weight α_m distributed in $\{0, 0.1, 1, 10, 100\}$ in Eqn. 17. The experimental results are shown in Fig. 9. Compared with

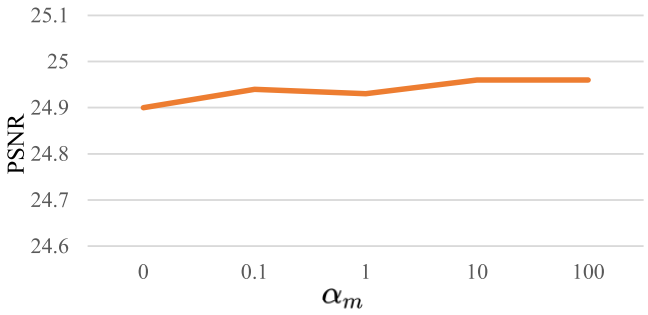


Fig. 9. Sensitivity of monotonicity regularization weight α_m.

TABLE IX
QUANTITATIVE COMPARISON (PSNR↑ AND SSIM↑) ON LOW-CONTRAST HAZY SOTS [22] INDOOR DATASET

Method	#Param	PSNR↑	SSIM↑
KDDN [12]	5.99M	34.72	0.9845
FFA-Net [36]	4.68M	36.39	0.9886
AECR-Net [44]	2.61M	37.17	0.9901
FSDGN [48]	2.73M	38.63	0.9903
4D LUT(Ours)	924.4K	37.02	0.9898

not adding the monotonicity regularization, the monotonicity regularization of the 4D LUT can preserve the relative color brightness, and make the color transformation cover the whole RGBC space. The impact of the larger monotonicity regularization weight α_m is minor, and we experimentally set α_m to 10 finally.

G. Discussion on low-contrast hazy images

To further explore the potential of our 4D LUTs on low-contrast hazy images, we evaluate the proposed method on RESIDE [22] dataset as shown in Tab. IX and Fig. 10. The subset Indoor Training Set (ITS) of RESIDE as our training set, which contains a total of 13,990 hazy indoor images generated from 1,399 clear images. The subset Synthetic Objective Testing Set (SOTS) of RESIDE as our testing set, which consists of 500 indoor hazy images. As shown in Tab. IX, 4D LUT also has a promising performance in low-contrast hazy images using fewer parameters. This is due to the ability of 4D LUT to learn low-contrast color space mapping to high-contrast color space. However, as shown in Fig. 10, with the increase of haze, the 4D LUT cannot distinguish well between regions with different intensities of haze, which is due to the fact that they have the same context map. Potentially, a haze encoder can be developed to replace the context encoder to learn to distinguish between regions with different haze intensities. If doing so, regions with various haze intensities can be differentially enhanced, thus effectively eliminating various intensities of haze.

VI. LIMITATION

In our method, the context map is obtained from the raw image and is used to distinguish the different contents. However, the annotation of each image in the dataset is obtained by the retouching expert independently, so when the same contents (*e.g.,* the grass in Fig. 11) are retouched to different colors

Fig. 10. Visualization of 4D LUT on low-contrast hazy SOTS [22] dataset.

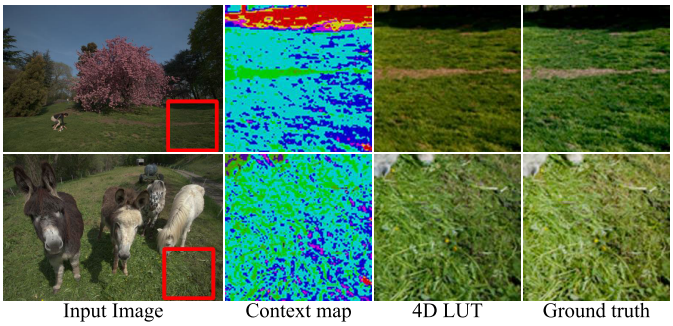


Fig. 11. Failure cases when the same content is retouched with different colors.

in different images, incorrect color mapping is generated. This uncertainty introduced by human factors remains a challenge for the LUT-based enhancement algorithm.

VII. CONCLUSION

In this paper, we extend the lookup table architecture into a 4-dimensional space and propose a novel learnable context-aware 4-dimensional lookup table (4D LUT). It includes four closely-related components. 1) A context encoder is used to generate the content-dependent context map. 2) A parameter encoder for generating image-adaptive coefficients. 3) A 4D LUTs fusion module integrates the coefficients and learnable basis 4D LUTs into a content-aware 4D LUT. 4) A quadrilinear interpolation module output the enhanced image. This design introduces the image content and enables finer control of color transformations for pixels in each image, resulting in content-dependent image enhancement via learning image contents adaptively. Experimental results show significantly superior between the proposed 4D LUT and existing SOTA models. In the future, we will focus on extending our 4D LUT in more low-level vision tasks through more explorations.

REFERENCES

[1] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand, "Fast local Laplacian filters: Theory and applications," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 1–14, Sep. 2014.

[2] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," in *Proc. CVPR*, Jun. 2011, pp. 97–104.

[3] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. ICCV*, Oct. 2017, pp. 1511–1520.

[4] Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, "Deep photo enhancer: Unpaired learning for image enhancement from photographs with GANs," in *Proc. CVPR*, Jun. 2018, pp. 6306–6314.

[5] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Aug. 2017.

[6] I. Goodfellow et al., "Generative adversarial nets," in *Proc. NeurIPS*, vol. 27, 2014, pp. 1–9.

[7] C. Guo et al., "Zero-reference deep curve estimation for low-light image enhancement," in *Proc. CVPR*, Jun. 2020, pp. 1780–1789.

[8] X. Guo, Y. Li, and H. Ling, "LIME: Low-light image enhancement via illumination map estimation," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 982–993, Feb. 2017.

[9] J. He, Y. Liu, Y. Qiao, and C. Dong, "Conditional sequential modulation for efficient global image retouching," in *Proc. ECCV*. Glasgow, U.K.: Springer, Aug. 2020, pp. 679–695.

[10] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.

[12] M. Hong, Y. Xie, C. Li, and Y. Qu, "Distilling image dehazing with heterogeneous task imitation," in *Proc. CVPR*, Jun. 2020, pp. 3462–3471.

[13] Y. Hu, H. He, C. Xu, B. Wang, and S. Lin, "Exposure: A white-box photo post-processing framework," *ACM Trans. Graph.*, vol. 37, no. 2, pp. 1–17, Apr. 2018.

[14] A. Ignatov, N. Kobyshev, R. Timofte, and K. Vanhoey, "DSLR-quality photos on mobile devices with deep convolutional networks," in *Proc. ICCV*, Oct. 2017, pp. 3277–3285.

[15] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. CVPR*, Jul. 2017, pp. 1125–1134.

[16] H.-U. Kim, Y. J. Koh, and C.-S. Kim, "Global and local enhancement networks for paired and unpaired image enhancement," in *Proc. ECCV*. Glasgow, U.K.: Springer, Aug. 2020, pp. 339–354.

[17] H.-U. Kim, Y. J. Koh, and C.-S. Kim, "PieNet: Personalized image enhancement network," in *Proc. ECCV*. Glasgow, U.K.: Springer, Aug. 2020, pp. 374–390.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[19] D. Kneubuehler, S. Gu, L. Van Gool, and R. Timofte, "Flexible example-based image enhancement with task adaptive global feature self-guided network," in *Proc. ECCV*. Glasgow, U.K.: Springer, Aug. 2020, pp. 343–358.

[20] S. Kosugi and T. Yamasaki, "Unpaired image enhancement featuring reinforcement-learning-controlled image editing software," in *Proc. AAAI*, vol. 34, 2020, pp. 11296–11303.

[21] E. H. Land, "The retinex theory of color vision," *Sci. Amer.*, vol. 237, no. 6, pp. 108–129, Dec. 1977.

[22] B. Li et al., "Benchmarking single-image dehazing and beyond," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 492–505, Jan. 2019.

[23] C. Li et al., "Low-light image and video enhancement using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9396–9416, Dec. 2022.

[24] M. Li, J. Liu, W. Yang, X. Sun, and Z. Guo, "Structure-revealing low-light image enhancement via robust retinex model," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2828–2841, Jun. 2018.

[25] J. Liang, H. Zeng, M. Cui, X. Xie, and L. Zhang, "PPR10K: A large-scale portrait photo retouching dataset with human-region mask and group-level consistency," in *Proc. CVPR*, Jun. 2021, pp. 653–661.

[26] Z. Liang, J. Cai, Z. Cao, and L. Zhang, "CameraNet: A two-stage framework for effective camera ISP learning," *IEEE Trans. Image Process.*, vol. 30, pp. 2248–2262, 2021.

[27] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. NeurIPS*, vol. 30, 2017, pp. 1–9.

[28] Y. Liu et al., "Very lightweight photo retouching network with conditional sequential modulation," *IEEE Trans. Multimedia*, early access, Jun. 2, 2022, doi: 10.1109/TMM.2022.3179904.

[29] S.-P. Lu, B. Ceulemans, A. Munteanu, and P. Schelkens, "Spatio-temporally consistent color and structure optimization for multiview video color correction," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 577–590, May 2015.

[30] L. Ma, T. Ma, R. Liu, X. Fan, and Z. Luo, "Toward fast, flexible, and robust low-light image enhancement," in *Proc. CVPR*, Jun. 2022, pp. 5637–5646.

[31] S. Moran, P. Marza, S. McDonagh, S. Parisot, and G. Slabaugh, "DeepLPF: Deep local parametric filters for image enhancement," in *Proc. CVPR*, Jun. 2020, pp. 12826–12835.

[32] S. Moran, S. McDonagh, and G. Slabaugh, "CURL: Neural curve layers for global image enhancement," in *Proc. ICPR*, Jan. 2021, pp. 9796–9803.

[33] Z. Pan, F. Yuan, J. Lei, W. Li, N. Ling, and S. Kwong, "MIEGAN: Mobile image enhancement via a multi-module cascade neural network," *IEEE Trans. Multimedia*, vol. 24, pp. 519–533, 2022.

[34] J. Park, J.-Y. Lee, D. Yoo, and I. S. Kweon, "Distort-and-recover: Color enhancement using deep reinforcement learning," in *Proc. CVPR*, Jun. 2018, pp. 5928–5936.

[35] Y. Qi et al., "A comprehensive overview of image enhancement techniques," *Arch. Comput. Methods Eng.*, vol. 29, pp. 583–607, Apr. 2021.

[36] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia, "FFA-Net: Feature fusion attention network for single image dehazing," in *Proc. AAAI*, vol. 34, 2020, pp. 11908–11915.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Munich, Germany: Springer, Oct. 2015, pp. 234–241.

[38] E. Schwartz, R. Giryes, and A. M. Bronstein, "DeepISP: Toward learning an end-to-end image processing pipeline," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 912–923, Feb. 2019.

[39] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. ICCV*, 1998, pp. 839–846.

[40] R. Wang, Q. Zhang, C. Fu, X. Shen, W.-S. Zheng, and J. Jia, "Underexposed photo enhancement using deep illumination estimation," in *Proc. CVPR*, Jun. 2019, pp. 6849–6857.

[41] T. Wang et al., "Real-time image enhancer via learnable spatial-aware 3D lookup tables," in *Proc. ICCV*, Oct. 2021, pp. 2471–2480.

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[43] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep retinex decomposition for low-light enhancement," in *Proc. BMVC*, 2018, pp. 1–11.

[44] H. Wu et al., "Contrastive learning for compact single image dehazing," in *Proc. CVPR*, Jun. 2021, pp. 10551–10560.

[45] H. Xu, G. Zhai, X. Wu, and X. Yang, "Generalized equalization model for image enhancement," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 68–82, Jan. 2014.

[46] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu, "Automatic photo adjustment using deep neural networks," *ACM Trans. Graph.*, vol. 35, no. 2, pp. 1–15, May 2016.

[47] C. Yang, M. Jin, X. Jia, Y. Xu, and Y. Chen, "AdaInt: Learning adaptive intervals for 3D lookup tables on real-time image enhancement," in *Proc. CVPR*, Jun. 2022, pp. 17522–17531.

[48] H. Yu, N. Zheng, M. Zhou, J. Huang, Z. Xiao, and F. Zhao, "Frequency and spatial dual guidance for image dehazing," in *Proc. ECCV*. Cham, Switzerland: Springer, 2022, pp. 181–198.

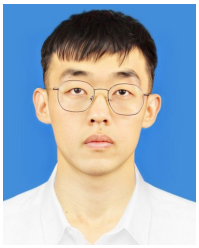
[49] R. Yu, W. Liu, Y. Zhang, Z. Qu, D. Zhao, and B. Zhang, "DeepExposure: Learning to expose photos with asynchronously reinforced adversarial learning," in *Proc. NeurIPS*, vol. 31, 2018, pp. 1–11.

[50] S. W. Zamir et al., "Learning enriched features for real image restoration and enhancement," in *Proc. ECCV*. Glasgow, U.K.: Springer, Aug. 2020, pp. 492–511.

[51] H. Zeng, J. Cai, L. Li, Z. Cao, and L. Zhang, "Learning image-adaptive 3D lookup tables for high performance photo enhancement in real-time," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2058–2073, Apr. 2022.

[52] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. CVPR*, Jun. 2018, pp. 586–595.

[53] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. ICCV*, Oct. 2017, pp. 2223–2232.



Chengxu Liu received the B.E. degree from Xi'an Jiaotong University, Xi'an, China, in 2019, where he is currently pursuing the Ph.D. degree with the SMILES Laboratory. He was an Intern with the Multimedia Search and Mining Group, Microsoft Research Asia, from April 2021 to May 2022. His current research interests include fine-grained image classification, object detection, video super-resolution, video frame interpolation, and image enhancement.

Huan Yang (Associate Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Shanghai Jiao Tong University, China, in 2014 and 2019, respectively. He is currently a Senior Researcher with Microsoft Research Asia, Beijing, China. His current research interests include computer vision, image processing, real-time video processing, and image photography.

Jianlong Fu received the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Science, in 2015. He is a Senior Research Manager with the Multimedia Search and Ming Group, Microsoft Research Asia, Beijing, China. He has shipped core technologies to Microsoft products, including Windows Photo, Bing Image Search, XiaoIce Chatbot, and Microsoft Flower. His current research interests include computer vision and multimedia content analysis, especially on fine-grained image recognition, vision and language, and personal photo experience of browsing, searching, sharing, and storytelling.

Xueming Qian (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, in 2008.

From 2011 to 2014, he was an Associate Professor with Xi'an Jiaotong University, where he is currently a Full Professor and the Director of the SMILES Laboratory. He was a Visiting Scholar with Microsoft Research Asia, Beijing, China, from 2010 to 2011. His current research interests include social media big data mining and search.

Prof. Qian was a recipient of the Microsoft Fellowship in 2006 and the Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province in 2010 and 2011, respectively.