# PBR-GAN: Imitating Physically-Based Rendering With Generative Adversarial Networks

Ru Li, *Member, IEEE*, Peng Dai, *Student Member, IEEE*, Guanghui Liu, *Senior Member, IEEE*, Shengping Zhang, Bing Zeng, *Fellow, IEEE*, and Shuaicheng Liu, *Member, IEEE*

***Abstract*— We propose a Generative Adversarial Network (GAN)-based architecture for achieving high-quality physically based rendering (PBR). Conventional PBR relies heavily on ray tracing, which is computationally expensive in complicated environments. Some recent deep learning-based methods can improve efficiency but cannot deal with illumination variation well. In this paper, we propose PBR-GAN, an end-to-end GAN-based network that solves these problems while generating natural photo-realistic images. Two encoders (the shading encoder and albedo encoder) and two decoders (the image decoder and light decoder) are introduced to achieve our target. The two encoders and the image decoder constitute the generator that learns the mapping between the generated domain and the real domain. The light decoder produces light maps that pay more attention to the highlight and shadow regions. The discriminator aims to optimize the generator by distinguishing target images from the generated ones. Three novel loss items, concentrating on domain translation, overall shading preservation, and light map estimation, are proposed to optimize the photo-realistic outputs. Furthermore, a real dataset is collected to provide realistic information for training GAN architecture. Extensive experiments indicate that PBR-GAN can preserve the illumination variation and improve the image perceptual quality.**

***Index Terms*— Physically based rendering, generative adversarial network, illumination variation.**

## I. Introduction

PHOTO-REALISTIC computer graphic methods are recently ubiquitous, with applications that include entertainment (movies and video games), and product design. Over the past decades, physically based rendering (PBR) has become widely used, where accurate modeling of the physics of light scattering is important for image synthesis [3]. Photo-realistic rendering aims to describe a 3D scene using an image that is realistic and indistinguishable from a photograph. The core problem of physically based rendering is the global illumination problem. Almost all photo-realistic rendering systems, such as Blender [4], Maya [5], and Mitsuba [1] are based on the ray-tracing algorithm. The introduction of the radiosity algorithm by Goral et al. was the first incentive towards an exact physical approach to the rendering problem [6]. Another milestone was the proposal of the rendering equation, which provides a more general physical framework [7]. Conventional photo-realistic rendering is computationally expensive, and handling complicated indoor scenes may even spend several hours.

The introduction of deep learning (DL) provides valuable inspiration for exploring better photo-realistic rendering techniques [8]. Zhang et al. studied the effects of rendering methods and scene lighting on training for three tasks [9]. Dai et al. proposed two stacked neural networks to predict the shading images and the color images, respectively [2]. Several DL-based methods apply PBR to achieve different tasks, such as intrinsic decomposition [10], material editing [11], and scene projector [12]. Although various techniques are proposed for achieving photo-realistic rendering with learning-based methods, handling illumination variation, especially in the highlight and shadow regions, remains a challenge.

Recently, Neural Radiance Field (NeRF) has been designed to render images of 3D scenes from novel viewpoints [13]. NeRF-based methods [14], [15], [16], [17] imitate the rendering by encoding the color radiance and density of a scene within the weights of a coordinate-based multi-layer perceptron (MLP). Many studies have been carried out to dive deeper into NeRF-based network architecture, including the faster training and inference for NeRF [18], extending NeRF from image to video [19], and handling dynamic scenes [20]. However, NeRF-based methods primarily focus on synthesizing novel viewpoints rather than generating more realistic images. Moreover, these methods require per-scene optimization.

In this paper, we propose PBR-GAN, an end-to-end GAN-based pipeline that achieves high-quality rendering with efficiency. The advantages of our design include two aspects: (1) the rendering procedure is accelerated with the proposed architecture; (2) specific modules are designed to address the challenging illumination variation. Inspired by intrinsic image decomposition [21], [22], we design an inverse composition network that fuses shading and reflectance information into color images. Perfect shading is a crucial characteristic of photo-realistic images, and accurate shading estimation is

Manuscript received 5 April 2023; revised 7 June 2023 and 11 July 2023; accepted 16 July 2023. Date of publication 26 July 2023; date of current version 7 March 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62071097 and Grant 62031009 and in part by the Sichuan Science and Technology Program under Grant 2023NSFSC0458 and Grant 2023NSFSC0462. This article was recommended by Associate Editor Y. J. Jung. *(Corresponding authors: Guanghui Liu; Shuaicheng Liu.)*

Ru Li and Shengping Zhang are with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China.

Peng Dai is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, SAR, China.

Guanghui Liu, Bing Zeng, and Shuaicheng Liu are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: guanghuiliu@uestc.edu.cn; liushuaicheng@uestc.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3298929.

Digital Object Identifier 10.1109/TCSVT.2023.3298929

essential for PBR tasks. We formulate the shading estimation into a convolution neural network (CNN). The shading network takes a sequence of 2D rendering images, such as surface normal, depth, and illumination, as inputs. Additionally, a parallel reflectance extraction network is designed to obtain reflectance features, which are then concatenated to the shading features to generate complete color image information. Compared with previous studies, our method is an end-to-end architecture with parallel shading and reflectance estimation encoders and two decoders concentrated on color image reconstruction and light map prediction. The image decoder outputs the photo-realistic images, while the light decoder generates light maps to represent the illumination variation. The overall pipeline is fluent and physically imitates the inverse procedure of intrinsic decomposition. The light decoder is a pioneering contribution for estimating illumination variation and can be applied to corresponding tasks, such as light estimation and modification. These modules are specifically designed to generate images with improved shading information and to address illumination variation, resulting in enhanced performance.

We adopt the rendered images from Mitsuba [1] as the ground truth to optimize the generated photo-realistic images. Generally, the synthetic data performs well but poorly in certain difficult situations, such as regions lacking illumination and limited rendering time, which easily generates images with strong noise [9]. Moreover, the synthetic data cannot capture all real-world statistics. Therefore, a real dataset containing diverse indoor scenes is further collected to improve the clarity and realism of the generated images. However, there is no one-to-one correspondence between the real and generated images. The GAN architecture is introduced to solve the unpaired problem. The discriminator can discriminate between these two types of images. In addition to the adversarial loss, we introduce three novel loss items to concentrate on different components of illumination variation. First, we introduce the shading loss to preserve the overall shading information of generated images, which constrains the distance between the predicted shading images and the shading ground truth. Second, we proposed the light loss to obtain accurate light maps and a novel mask-based PBR loss, which designs the light map as a mask to constrain the generated color images, directing the network to focus more on the highlight and shadow regions.

Fig. 1 shows the comparisons with the recent CNN-based PBR-Net [2]. PBR-Net is a two-stage composition architecture that aims to obtain the shading image and photo-realistic image separately. On the contrary, the PBR-GAN is an end-to-end architecture specifically designed to generate images with better illumination variation, which is essential for producing high-quality photo-realistic images. The example in Fig. 1 demonstrates that our method better captures the illumination variation compared to PBR-Net. First, our method learns the cast shadow generated by the window frame better. Second, PBR-GAN produces the highlight region (the base of the lamp) more prominently. Finally, the proposed method generates the decoration outside the window more clearly. These improvements are important for photo-realistic generation tasks.

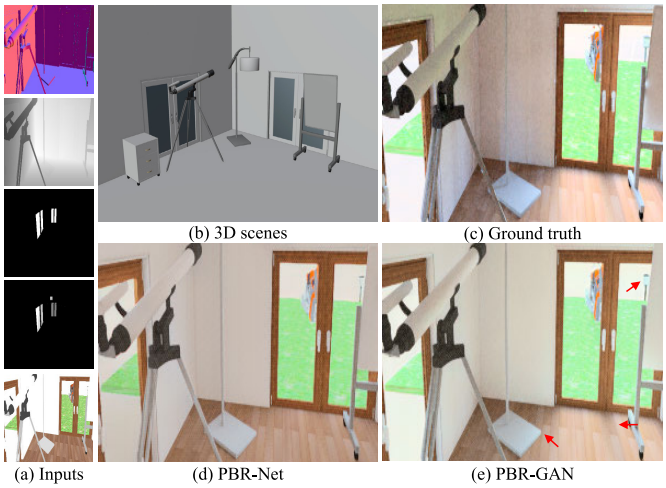



Fig. 1. (a) Input images, including surface normal, depth, panoramic illumination intensity, panoramic illumination distance, and the albedo. (b) 3D scenes. (c) Ground truth rendered by Mitsuba [1]. (d) Result of PBR-Net [2]. (e) Result of the proposed PBR-GAN. Compared with PBR-Net, our method can generate more realistic images with natural illumination.

Overall, the main contributions are:

- We propose an end-to-end GAN-based architecture for generating color outputs from rendering sources. A real dataset is collected to optimize the model to get realistic images that are indistinguishable from real images.
- We design the generator to concentrate on the illumination variation. Specifically, an image decoder is proposed to produce photo-realistic images, and a light decoder is introduced to generate light maps. Three novel losses are applied to estimate the highlight and shadow regions.
- We provide qualitative and quantitative comparisons with several state-of-the-art methods to demonstrate the superiority of our PBR-GAN.

## II. Related Works

### A. Photo-Realistic Generation Tasks

Photo-realistic image generation is applied in many research studies [23], such as text-to-image [24] and image super-resolution [25]. Inspired by CNN, Guo et al. applied inverse-rendered photo-realistic face images to achieve face reconstruction [26]. Li et al. introduced InteriorNet to improve large-scale interior scene understanding and mapping [27]. Some methods apply variational autoencoder to obtain photo-realistic images [28], [29]. Liu et al. narrowed down the latent subspace using the conditional sampling mechanism to achieve photo-realistic image super-resolution [28]. Liu et al. proposed a reference-based photo-realistic image super-resolution approach, which transfers the knowledge from the reference to the super-resolved images [29].

The generative adversarial networks provide advantages for generating photo-realistic images with diversity and higher quality. The GAN architecture was proposed by Goodfellow et al. [30], which has achieved remarkable results across various fields [31], [32]. Zhang et al. introduced the StackGAN to generate photo-realistic images according to the text descriptions [24]. Liu et al. proposed Semantic-GAN to transfer the semantic label map into high-resolution

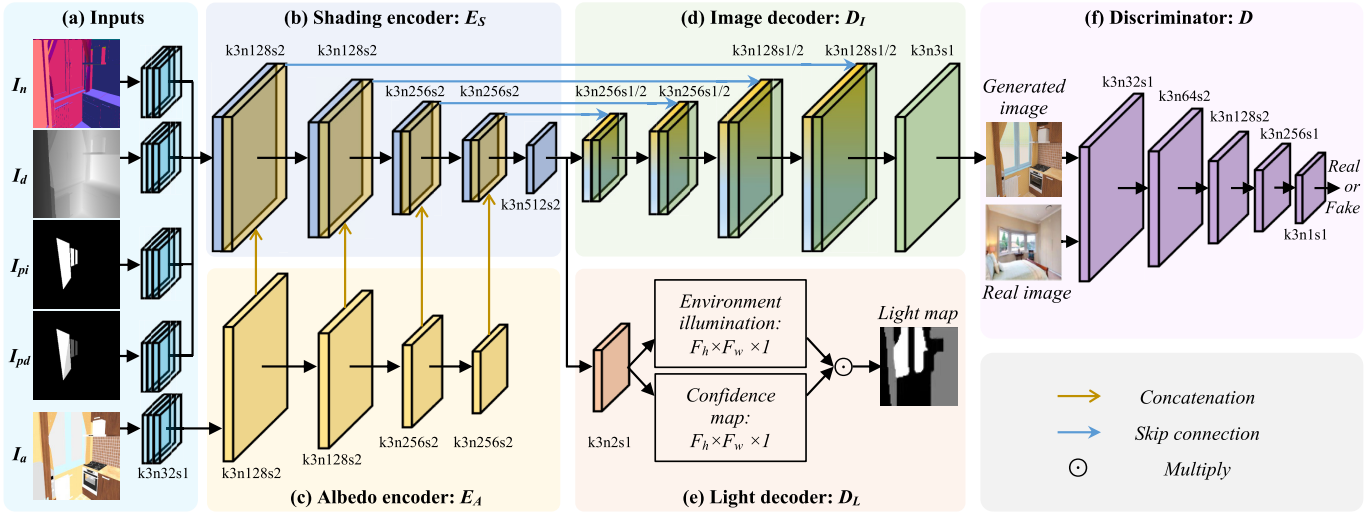



Fig. 2. The pipeline exhibits the architecture of PBR-GAN, which consists of five modules, each playing a different role. **The shading encoder** $E_S$ extracts shading information from the surface normal $I_n$, depth $I_d$, panoramic illumination intensity $I_{pi}$ and panoramic illumination depth $I_{pd}$. **The albedo encoder** $E_A$ obtains the albedo features from the albedo image $I_a$ and concatenates them with the shading features to obtain complete color image information. **The image decoder** $D_I$ recovers the color information to the photo-realistic images using up-convolution layers. **The light decoder** $D_L$ constructs a light map to indicate the illumination variation. The shading encoder $E_S$, the albedo encoder $E_A$, and the image decoder $D_I$ together constitute the generator $G$, which transfers the input images to output images with the desired characteristics. **The discriminator** $D$ distinguishes whether a given image belongs to the synthetic or real dataset. Here, $k$ represents the kernel size, $n$ denotes the number of feature maps, and $s$ represents the stride in each convolutional layer.

images [33]. Some recent GAN-based photo-realistic image generation works are proposed for combining multi-input images [34], [35], [36]. Recently, Mildenhall et al. introduced the differentiable volumetric rendering technique to optimize a neural radiance field and generate novel viewpoints [13]. Subsequently, a series of works recovered the radiance field using deep neural networks [14], [15], [16], [17], which have enabled significant progress toward view synthesis. However, NeRF-based methods require per-scene training and optimization, limiting their generality. As for the PBR task, we design a GAN-based architecture to generate photo-realistic images from multiple rendering sources, which is suitable for any kind of scene and pays more attention to the physical procedure of rendering.

### B. Image Composition and Decomposition

Intrinsic image decomposition has been studied for many years. The classical intrinsic image decomposition approaches apply various priors. For example, the seminal Retinex algorithm assumes that the reflectance reflects the large image gradients, while the shading represents the smaller gradients [37]. Then, the success of deep learning leads to the exploration of high-quality decomposition methods. Li et al. proposed a partial learning method that predict reflectance and shading by combining the ground truth and the sparse annotations from the IIW [38] and SAW [39] datasets. Han et al. proposed to synthesize the training pairs, however, the performance of the synthetic dataset is also not satisfactory due to the inability to capture all the real-world statistics, leading to models trained on synthetic data underperforming on real images [40].

## III. Method

We propose a GAN-based architecture to achieve photo-realistic rendering with the assistance of real natural images. The architecture includes two streams that output different images. The first stream generates photo-realistic images while being constrained by corresponding ground truth and real images. The second stream produces a light map to focus the network's attention on illumination variation. We adopt a set of five 2D images $x = \{I_n, I_d, I_{pi}, I_{pd}, I_a\}$ obtained from rendering sources as our input data, including the surface normal $I_n$, depth $I_d$, panoramic illumination intensity $I_{pi}$, panoramic illumination distance $I_{pd}$ and albedo $I_a$. We utilize Mitsuba [1] to generate the ground truth shading image $I_{gt_s}$ and color image $I_{gt_c}$. However, the ground truth images generated by the rendering software are not realistic enough if the ray-tracing simulation is not accurately modeled, such as imprecise modeling of material. We further gather a collection of real images as the target domain $Y$ to make the results more realistic, denoted as $\{y_j\}_{j=1,\ldots,M} \in Y$. The proposed PBR-GAN is capable of achieving photo-realistic rendering with necessary detail and illumination variation.

### A. Network Architecture

The network architecture is shown in Fig. 2. The shading encoder $E_S$, albedo encoder $E_A$ and image decoder $D_I$ constitute the generator $G$. Table I presents the layer configurations of these modules. The shading information is derived from four inputs: the surface normal $I_n$, depth $I_d$, panoramic illumination intensity $I_{pi}$ and panoramic illumination distance $I_{pd}$, which are extracted by three convolutional blocks and fed into the shading encoder (Fig. 2 (b)). The albedo encoder (Fig. 2 (c)) extracts the albedo features, which are then concatenated with the shading features to obtain complete color image features. The image decoder (Fig. 2 (d)) and the light decoder (Fig. 2 (e)) recover the photo-realistic images and the light maps, respectively. The encoder and decoder are connected using a short-cut connection [42].

TABLE I
LAYER CONFIGURATIONS OF THE PROPOSED ARCHITECTURE

(a) Encoders

| Conv | | | BN | Activation | Max pooling |
|---|---|---|---|---|---|
| Kernel | Stride | Channel | channel | | Stride |
| 3 | 1 | 32 | 32 | LReLU | - |
| 3 | 1 | 32 | 32 | LReLU | - |
| 3 | 1 | 32 | 32 | LReLU | - |
| 3 | 2 | 128 | 128 | LReLU | 2 |
| 3 | 2 | 128 | 128 | LReLU | 2 |
| 3 | 2 | 256 | 256 | LReLU | 2 |
| 3 | 2 | 256 | 256 | LReLU | 2 |
| 3 | 2 | 512 | 512 | LReLU | 2 |

(b) Decoders

| Conv | | | BN | Activation |
|---|---|---|---|---|
| Kernel | Stride | Channel | Channel | |
| 3 | 1/2 | 256 | 256 | LReLU |
| 3 | 1/2 | 256 | 256 | LReLU |
| 3 | 1/2 | 128 | 128 | LReLU |
| 3 | 1/2 | 128 | 128 | LReLU |
| 3 | 1 | 3 | - | - |
| 3 | 1 | 4 | - | Softplus [41] |

(c) Discriminator

| Conv | | | BN | Activation |
|---|---|---|---|---|
| Kernel | Stride | Channel | Channel | |
| 3 | 1 | 32 | - | LReLU |
| 3 | 2 | 64 | - | LReLU |
| 3 | 1 | 64 | 64 | LReLU |
| 3 | 2 | 128 | - | LReLU |
| 3 | 1 | 128 | 128 | LReLU |
| 3 | 1 | 256 | 256 | LReLU |
| 3 | 1 | 1 | - | - |

*1) Encoders:* Table I (a) first presents three identical convolutional blocks for feature extraction of input images (from row 3 to row 5), and then displays the layer configurations of the shading encoder $E_S$ and albedo decoder $E_A$ (from row 6 to row 10). The two encoders have identical architectures, each comprising five convolution blocks to extract the shading and albedo features, respectively.

*2) Decoders:* Table I (b) provides the detailed configuration of the image decoder $D_I$ (from row 3 to row 7) and the light decoder $D_L$ (row 8). The image decoder, which is responsible for recovering features to generate images, comprises four transposed convolutional blocks and a convolutional layer. For the light decoder, inspired by [43], we utilize a convolutional block to generate two maps: a gray environment illumination and a confidence map. The dimensions of the gray environment illumination and confidence map are also $F_w \times F_h \times 1$, where $F_w$ and $F_h$ are the width and height of the features generated by the light decoder. We perform element-wise multiplication of the gray environment illumination and confidence map to obtain the final light map.

*3) Discriminator:* Table I (c) exhibits the architecture of the discriminator $D$. The discriminator is complementary to the generator, which is composed of several convolutional blocks. Such a simple discriminator uses fewer parameters and can work on images of arbitrary sizes.

### B. Loss Function

The complete objective function includes four components: (1) the PBR loss $L_{\text{PBR}}$, which encourages the image decoder to recover desired photo-realistic images; (2) the adversarial loss $L_{\text{GAN}}$, which incorporates the real domain images; (3) the shading loss $L_{shading}$, which preserves the overall shading information of the generated images; and (4) the light loss $L_{light}$, which guides the light decoder to generate accurate light maps. For the sake of brevity, we merge the shading encoder $E_S$ and albedo encoder $E_A$ as the image encoder $E_I$. The complete objective function is formulated as follows:

$$\begin{aligned} L(E_I, D_I, D_L, D) = {} & w_{\text{PBR}} L_{\text{PBR}}(E_I, D_I) \\ & + w_{\text{GAN}} L_{\text{GAN}}(E_I, D_I, D) \\ & + w_{shading} L_{shading}(E_I, D_I) \\ & + w_{light} L_{light}(E_I, D_L), \end{aligned} \tag{1}$$

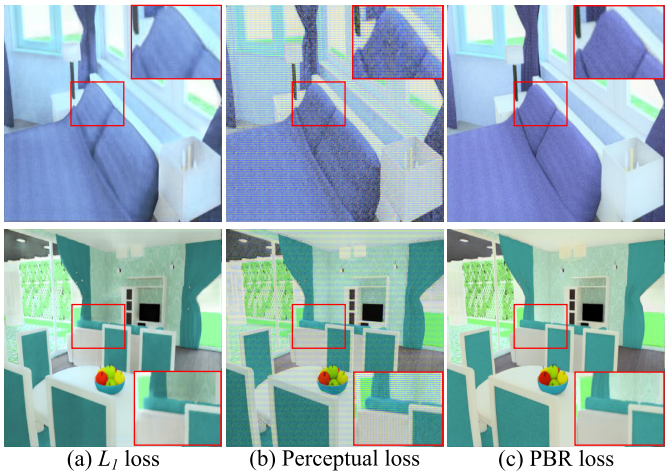
where the weights determine the relative importance of each loss. $w_{\text{PBR}}$ and $w_{\text{GAN}}$ significantly influence the balance between the rendering ground truth and collected real images. When $w_{\text{GAN}}$ is set to 0, the real dataset is ineffective and the network is solely constrained by the ground truth. Empirically, we assign a higher value to $w_{\text{GAN}}$ to encourage the generated images to capture more realistic information. We set $w_{\text{PBR}} = 1$ and $w_{\text{GAN}} = 1.5$ in our implementation. The weight of the shading loss $w_{shading}$ is useful for generating results with more reasonable illumination variation. $w_{shading}$ is set to 1 in our implementation. The weight of the light loss $w_{light}$ controls the accuracy of the light map. Since obtaining matching light ground truth is challenging, we apply the panoramic illumination intensity $I_{pi}$ to guide the light decoder and set $w_{light}$ to 1 at the initial stage to make the light decoder to be converged. Subsequently, as the training stabilizes and the illumination variation information is adequately learned, $w_{\text{light}}$ is gradually reduced to minimize the impact of the panoramic illumination intensity image. $w_{light}$ is gradually decreased to reduce the impact of the panoramic illumination intensity image after training is stable and the illumination variation information is properly learned.

*1) PBR Loss:* We modify the standard perceptual loss [44] to create the PBR loss, which constrains the generated images and the ground truth. The PBR loss is defined as follows:

$$\begin{aligned} & L_{\text{PBR}}(E_I, D_I) \\ & \quad = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\sum_{l}^{N} \lambda_l ||V_l(D_I(E_I(x))) - V_l(I_{gt_c})||_1], \end{aligned} \tag{2}$$

where $V$ represents the VGG network, $l \in \{1, \ldots, N\}$ denotes the layers in the VGG19 network, and $\lambda_l$ are the hyper-parameters to balance the contributions of each layer. In our implementation, we apply the features from the first 3 layers to calculate the perceptual loss, and the hyper-parameters $\lambda_l$ are set to 1.5, 1.5, and 1 to ensure a balanced contribution.

Fig. 3 displays different results by using different loss functions as the PBR loss. The $L_1$ loss is first applied as the PBR loss to minimize the discrepancy between the generated images and the ground truth. As shown in Fig. 3 (a), the generated images appear blurry due to the extraction of low-level features by the $L_1$ loss function. On the contrary, the standard perceptual loss utilizes 5 convolutional blocks of the VGG19 network [45] pre-trained on ImageNet [46] to calculate loss values, making it sensitive to high-level abstractions. The results of conventional perceptual loss on our task are shown in Fig. 3 (b). The obvious patterns in Fig. 3 (b) are inevitable due to the high-level feature calculation. To strike a balance between the blurry caused by $L_1$ loss and patterns generated by perceptual loss, we use the features of the first 3 convolutional



(a) $L_1$ loss (b) Perceptual loss (c) PBR loss

Fig. 3. (a) When we select $L_1$ loss as PBR loss, the results are blurry. (b) The standard perceptual loss produces unavoidable patterns. (c) When we apply the modified perceptual loss as PBR loss, the results are clean and sharp.

(a) Generated images (b) Gray estimated shading images (c) Gray shading ground truth images

Fig. 4. Some results correspond to the shading loss. (a) Generated images. (b) The estimated shading images. (c) The shading ground truth images.

blocks of the VGG19 network to create our PBR loss. The results of our new PBR loss are shown in Fig. 3 (c), which can effectively avoid blur and decrease the patterns.

*2) Adversarial Loss:* The adversarial loss balances the generator and the discriminator. In our task, adversarial loss drives $G$ (the image encoder $E_I$ and the image decoder $D_I$) to generate outputs that closely resemble real images. We gather a real dataset with diverse indoor scenes as the target domain to constrain the generated images similar to real images. The examples of the real dataset are shown in Section IV-A. The adversarial loss is defined as follows:

$$L_{\text{GAN}}(E_I, D_I, D) = \mathbb{E}_{y\sim p_{\text{data}}(y)}\big[\log D(y)\big] + \mathbb{E}_{x\sim p_{\text{data}}(x)}\big[\log(1 - D(D_I(E_I(x)))\big]. \tag{3}$$

*3) Shading Loss:* The shading loss is designed to maintain the overall shading information of generated color images. We first obtain the output shading $O_s$ according to the output color image and the input albedo image by $O_s = D_I(E_I(x))/I_a$, and then minimize the difference between the output shading $O_s$ and the shading ground truth $I_{gt_s}$ using $L_2$ loss. To avoid the differences caused by the scenes outside the windows and the doors, we calculate the difference in shading information in the form of gray images. $L_{shading}$ is defined as follows:

$$L_{shading}(E_I, D_I) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[||Gray(O_s) - Gray(I_{gt_s})||_2], \tag{4}$$

where $Gray$ is the gray images. Fig. 4 shows some shading results. If the network performs well and generates realistic images, the gray estimated shading images (Fig. 4 (b)) should closely resemble the gray ground truth images (Fig. 4 (c)).

*4) Light Loss:* The PBR loss preserves the content information, the adversarial loss ensures that the generated images resemble the real domain images, and the shading loss maintains the overall shading information. They are inadequate for desired photo-realistic transformation, particularly in regions with illumination variation. Reasonable highlight and shadow reconstruction is essential for obtaining more realistic images. We design a light loss $L_{light}$ to detect the illumination condition of the generated images.

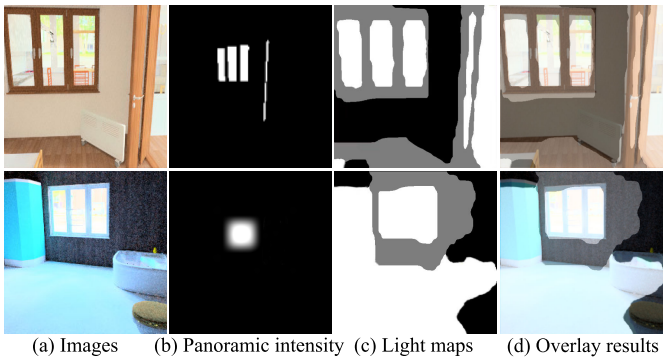
The light map is defined as a gray image, obtained by multiplying the environment illumination and the confidence map. The confidence weights reflect the values of a patch for inferring the illumination variation, which is integrated into a confidence-weighted pooling [43]. The light decoder can learn from the PBR dataset about which local areas in an image are informative for highlight and shadow generation. This technology draws inspiration from color constancy algorithms [43], where they apply the environment illumination to estimate the special colors and the confidence map for inferring the global color constancy. We modify it to adapt to the PBR task to detect the illumination condition in the images, which is helpful for detecting the light source and light irradiation areas. We introduce the panoramic illumination intensity $I_{pi}$ to guide the light decoder to generate light maps that can locate the light source better. The light loss minimizes the difference between the light map $D_L(E_I(x))$ and the panoramic illumination intensity $I_{pi}$, which is formulated as follows:

$$L_{light(E_I, D_L)} = \mathbb{E}_{x\sim p_{\text{data}}(x)}[||D_L(E_I(x)) - I_{pi}||_2]. \tag{5}$$

Training the light decoder without any guidance poses a significant challenge. The panoramic illumination intensity $I_{pi}$ appears similar to a binary image and only indicates the location of the light source. The light map may represent the panoramic illumination rather than the viewpoint illumination if the light decoder is still constrained by $I_{pi}$. Therefore, we provide a large decay factor to reduce the influence of $I_{pi}$ as the training epochs progress. After that, leveraging the light decoder inspired by color constancy algorithms, the network can learn from the PBR dataset about which local areas in an image provide informative cues for highlight and shadow generation. The variation trend of $w_{light}$ is defined as:

$$w_{light} = w_{light} \times 0.5^{\lfloor N_e/2 \rfloor}, \tag{6}$$

where $N_e$ is the number of the training epochs. With a decay factor of 0.5, the weight will rapidly decrease towards 0 as



(a) Images (b) Panoramic intensity (c) Light maps (d) Overlay results

Fig. 5. Some examples of light maps. The overlay results indicate that the light detection is consistent with the images and almost accurate.

the training epoch progresses and the light maps will not be affected by $I_{pi}$ as its influence approaches 0 before epoch 10.

We show the light outputs in Fig. 5, among which the overlay results (Fig. 5 (d)) indicate that the light detection is consistent with the images and almost accurate. Generally, the white regions in the light maps (Fig. 5 (c)) represent the light sources and the light irradiation areas that easily produce the highlight and shadow effects. Theoretically, if an object is covered by light (located in the white regions), its shadow is also located in the white regions. Alternatively, in some cases, the shadow may locate in the black regions while the object is situated in the white regions (the cushion in the bottom case in Fig. 5). The gray color tends to represent regions with a uniform texture. The light map has the potential to convey more semantic value and encompass scene areas at an object level. Note that, light detection differs from conventional semantic segmentation. The segmentation generates accurate segmentation results based on scene semantics, while our light detection learns the illumination variation over the scene.

*5) Mask-Based PBR Loss:* In order to make the network concentrate more on highlight and shadow regions, the light map is further designed as a mask to constrain the PBR loss. We first define $O_l$ as the output of the light network, which means that $O_l = D_L(E_I(x))$. The light map $O_l$ is then normalized to $O_{lm}$ with pixel values belonging to 1, 2, and 3, as shown in the following equation:

$$O_{lm}(i) = \begin{cases} 3 & O_l(i) \in white\ regions \\ 2 & O_l(i) \in black\ regions \\ 1 & O_l(i) \in gray\ regions, \end{cases} \tag{7}$$

where $i$ represents the pixel in the light map. The normalized light map is designed to have large values in white regions (highlight regions) and relatively large values in black regions (shadow regions). The mask-based PBR loss is designed to impose more penalties on these regions if the image decoder cannot recover them correctly, as defined in Eq. 8:

$$L_{\text{PBR}_m}(E_I, D_I) = E_{x\sim p_{\text{data}}(x)}[\sum_{l}^{N} \lambda_l || \quad O'_{lm} \cdot V_l(D_I(E_I(x))) - O'_{lm} \cdot V_l(I_{gt_c})||_1], \tag{8}$$

where the light mask $O_{lm}$ is down-sampled and its channels are duplicated to match the VGG feature dimensions. The adjusted light mask is denoted as $O'_{lm}$.

## IV. Experiments

We compare our method with several representative works, including the rendering software: OpenGL [47] and Mitsuba [1], and some learning-based methods: a typical paired GAN-based image translation method pix2pix [48], a popular unpaired GAN-based image translation method CycleGAN [49], an edge-preserving method CartoonGAN [50], a popular diverse translation model StarGAN [51], a semantic contrastive learning-based method Hneg-SRC [52] and the recent physically-based rendering method PBR-Net [2]. Further, we conduct the comparisons with NeRF [13] in terms of detail and illumination preservation.

### A. Datasets and Implementation Details

*1) Data Collection:* Two kinds of datasets are collected for our task, among which the first one contains the PBR images and the second one is composed of diverse realistic photos. Some examples of these two datasets are shown in Fig. 6.

*a) PBR dataset:* SUNCG dataset [53] is applied as the rendering source because it provides various indoor scenes with realistic furniture layouts. The camera viewpoints are sampled according to [9]. Our PBR datasets include five input images: the surface normal $I_n$, depth $I_d$, panoramic illumination intensity $I_{pi}$, panoramic illumination distance $I_{pd}$ and albedo $I_a$, and two ground truth images: shading ground truth $I_{gt_s}$ and color ground truth $I_{gt_c}$. $I_n$, $I_d$, $I_a$ and $I_{gt_c}$ can be directly rendered from Mitsuba [1]. The shading ground truth $I_{gt_s}$ is generated by removing the texture from the virtual scene and re-rendering the PBR image. Refer to [2], we produce the panoramic illumination intensity map ($I_{pi}$) and the panoramic illumination distance map ($I_{pd}$) to indicate the light source information. For the training dataset, 20,000 and 1,000 groups of images are randomly selected. For the test dataset, 1,000 groups of images were chosen.

*b) Real dataset:* We gather real images from the Internet to increase the diversity of indoor scenes. We first gather the images from popular search engines 'Google' [54] and 'Baidu' [55] using the following keywords: indoor decoration, bedroom, washroom, parlor, kitchen, and so on to include a wide range of indoor situations. In order to enhance the diversity of the real dataset, we further use these keywords to select videos from the 'YouTube' [56] website and extract frames from the collected videos. Then, similar images are discarded using LPIPS similarity [57]. The first two steps are automatically performed, which may cost several minutes. After that, in order to obtain the final real dataset, we manually filter the collected images by preserving images with similar furniture layouts to PBR images and deleting unreasonable images. The last step takes up to two hours. In summary, the process of collecting the real dataset is efficient and advisable for incorporating real-world information into the pipeline. Increasing the number of real images can enhance the performance of the model, but finding the optimal balance between effectiveness and efficiency is crucial. We empirically find that a number between 1,000 and 2,000 is the best choice for the real dataset because this range provides a diverse set of images that aligns well with the capabilities of our

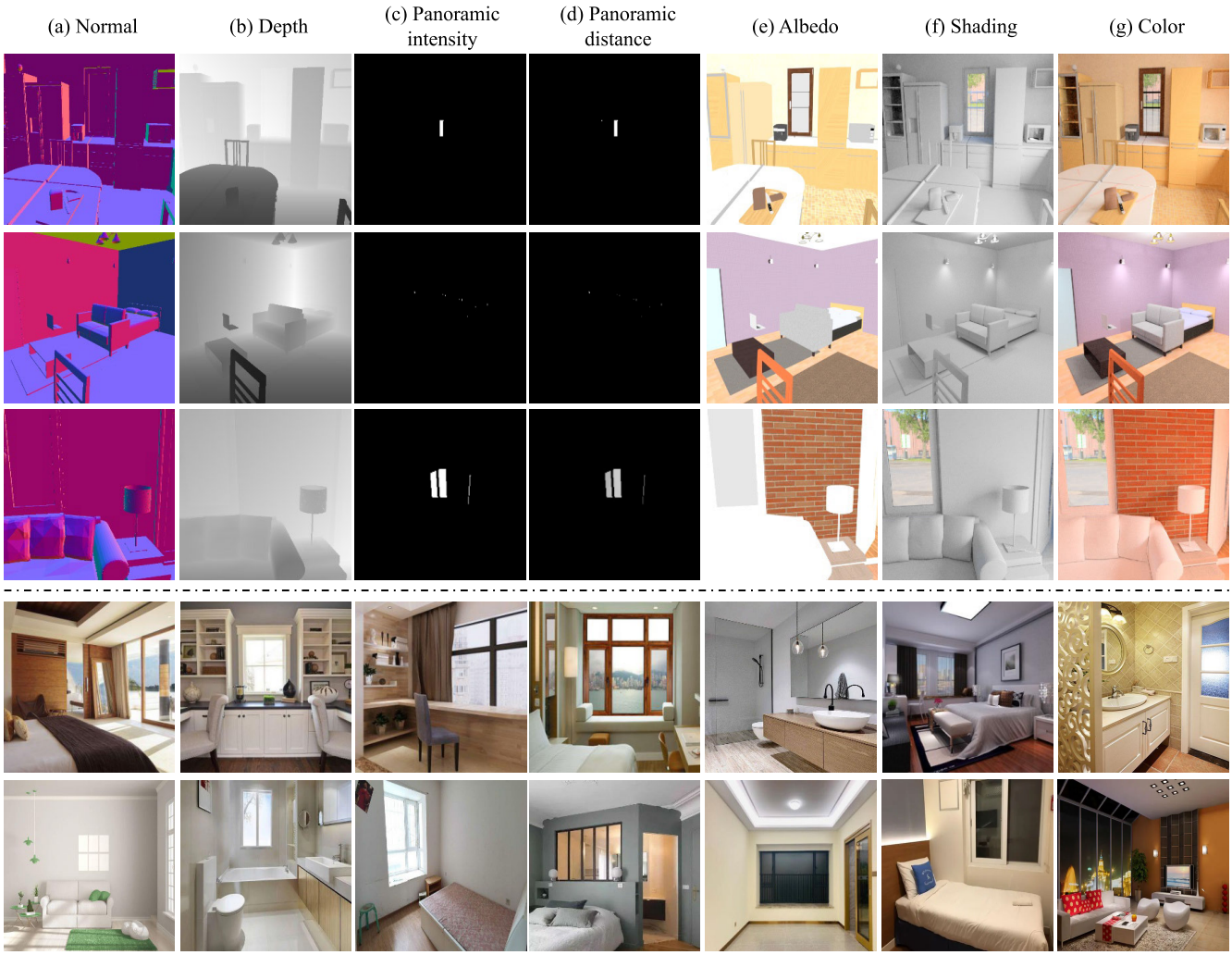



Fig. 6. Some examples of the PBR dataset and the real dataset. The top three rows show three cases of the PBR dataset and each case includes five input images (surface normal, depth, panoramic illumination intensity, panoramic illumination distance and albedo) and two ground truth images (the shading ground truth and the color ground truth). The last two rows show some examples of the collected real dataset with diverse indoor scenes.

TABLE II

QUANTITATIVE COMPARISONS WITH OTHER METHODS IN TERMS OF SSIM, LPIPS AND FID. THESE METRICS SHOW THAT OUR RESULTS ARE CLOSEST TO THE GROUND TRUTH AND REAL DATASET. RED INDICATES THE BEST PERFORMANCE AND BLUE REFERS TO THE SECOND BEST RESULT. THE PERCENTAGE IN THE BRACKET INDICATES THE IMPROVEMENT OVER THE PBR-NET. †: THE PBR-GAN IS TRAINED ON THE REAL DATASET WITH 1,082 IMAGES. ‡: THE PBR-GAN IS TRAINED ON THE REAL DATASET WITH 2,000 IMAGES

| | pix2pix [48] | CycleGAN [49] | CartoonGAN [50] | StarGAN [51] | Hneg-SRC [52] | PBR-Net [2] | Improved PBR-Net | PBR-GAN† | PBR-GAN‡ |
|---|---|---|---|---|---|---|---|---|---|
| SSIM ↑ | 0.638(-14.02%) | 0.613(-17.39%) | 0.624(-15.90%) | 0.681(-8.22%) | 0.702(-5.39%) | 0.742(+0.00%) | 0.749(+0.94%) | 0.802(+8.09%) | 0.804(+8.36%) |
| LPIPS ↓ | 0.067(-39.58%) | 0.079(-64.58%) | 0.071(-47.92%) | 0.059(-22.92%) | 0.056(-16.67%) | 0.048(+0.00%) | 0.048(+0.00%) | 0.045(+6.25%) | 0.044(+8.33%) |
| FID ↓ | 244(-27.08%) | 271(-41.15%) | 255(-32.81%) | 218(-13.54%) | 206(-7.29%) | 192(+0.00%) | 184(+4.17%) | 171(+10.94%) | 168(+12.50%) |

architecture. The ablation studies of the LPIPS threshold and the number of real images are conducted in Section IV-G.4.

*2) Training Details:* We implement PBR-GAN in PyTorch and the model is trained on four NVIDIA RTX 2080Ti GPUs with 30 epochs. The network is iterated using the Adam optimizer with a learning rate of $2.0\times10^{-4}$ for both the generator and discriminator. On average, the training process takes approximately 20 hours.

### *B. Quantitative Comparisons*

We choose three representative evaluation metrics for the quantitative comparisons, including the SSIM [58], learned perceptual image patch similarity (LPIPS) [57] and the Fréchet Inception distance (FID) [59], where the former two metrics evaluate the similarity of generated images and rendering ground truth, and the latter one measures the distance between the generated and the real dataset. Table II lists the SSIM, LPIPS, and FID scores of the aforementioned learning-based methods and PBR-GAN. The PBR datasets are utilized for training pix2pix, CycleGAN, CartoonGAN, and Hneg-SRC with 200 epochs, StarGAN for 200,000 iterations, and PBR-Net for 30 epochs. We first use three identical convolution blocks (described in Section III-A) to extract the features of five images (normal, depth, panoramic intensity, panoramic

TABLE III

AVERAGE COMPUTATIONAL TIMES OF PBR-GAN AND THE COMPARISON METHODS WHEN GENERATING IMAGES WITH SIZE 360 × 480. WE REPORT TWO VERSIONS OF PBR-NET AND PBR-GAN, WHERE THE NUMBERS OUTSIDE THE BRACKETS REPRESENT THE INFERENCE TIME, AND THE NUMBERS IN THE BRACKETS ARE THE TOTAL RUNNING TIME, INCLUDING THE PRE-COMPUTATION AND THE INFERENCE

| Platform | CPU (i7, 4 cores) | | | | GPU (NVIDIA RTX 2080Ti) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | OpenGL | Mitsuba | PBR-Net | PBR-GAN | pix2pix | CycleGAN | CartoonGAN | StarGAN | Hneg-SRC | PBR-Net | NeRF | PBR-GAN |
| Time ($s$) | 0.041 | 180.373 | 10.846 (15.346) | 11.201 (15.701) | 0.009 | 0.030 | 0.017 | 0.009 | 0.017 | 0.012 | 2.714 | 0.014 |

distance, and albedo) and then concatenate the features as inputs for GAN-based methods. Moreover, we apply the adversarial loss to improve the original PBR-Net, named improved PBR-Net, to demonstrate that the advantages of our method stems not only from the real dataset but also from the designed generator. We present the results of two versions of PBR-GAN: PBR-GAN$^{\dagger}$ trained on the real dataset consisting of 1,082 images, and PBR-GAN$^{\ddagger}$ trained on the real dataset comprising 2,000 images. Compared to the other six methods, the PBR-GAN achieves the best scores on average, showing a significant improvement over other GAN-based methods and a considerable improvement over PBR-Net and improved PBR-Net. While the improved PBR-Net has slightly higher scores than the original PBR-Net, the most significant improvement is observed in the FID score because the introduction of adversarial loss helps the network generate results more closely resembling real images. However, the overall performance of the improved PBR-Net is still inferior to the PBR-GAN.

### C. Qualitative Comparisons

We present the qualitative comparisons of PBR-GAN and the aforementioned learning-based methods in Fig. 7. pix2pix [48] is designed for image-to-image translation, learning the transformation between input and corresponding ground truth, while other GAN-based methods [49], [50], [51], [52] focus on translation between two domains. Results in Fig. 7 clearly demonstrate that the GAN-based methods generate desired structural information sometimes but produce unavoidable artifacts. PBR-Net is recently proposed for imitating physically based rendering with CNN, which recovers the indoor scene more reasonably but fails to detect the illumination variation. In the first case in Fig. 7, the light source locates on the right side, and our method generates the highlight in the cupboard (blue arrow) and shadow after the coffee machine (red arrow) more obviously. In the second case, the light source locates in the front of the scene, and the proposed method learns the shadow generated by the book and the cabinet more clearly (red arrows). In the third scene, there is a window located on the left of the scene, therefore producing the cast shadow (red arrow) in the wall, which is learned by our PBR-GAN merely. Moreover, only our method can produce the cast shadow generated by the window in the fourth scene (red arrows). In the last example, the PBR-Net generates images with non-uniform color on the white wall and the lamp, while the PBR-GAN produces results with fewer artifacts and uniform color.

In order to exhibit the improvements compared to PBR-Net, we provide additional comparison results with PBR-Net [2] in Fig. 8. In the first case in Fig. 8, the proposed method generates the highlight near the window (blue arrows) and learns the correct illumination variation on the sofa, while PBR-Net produces an inverted highlight and shadow (red arrows). The phenomenon also appears in the second case, which is possibly due to the incorrect prediction of the light source. In the last three scenes, our method learns the cast shadow more obviously (green arrows). Moreover, in the fourth scene, our method can recover the details of the decoration outside the window (blue boxes). By preserving detailed information and simulating natural illumination variation, our method produces results with realistic characteristics and higher quality. These improvements are important for photo-realistic generation tasks.

### D. Comparisons With NeRF

We also conduct the comparison with recent popular NeRF [13] in Fig. 9. We capture dozens of images of one scene to train a forward-facing NeRF and set the test pose to generate novel viewpoints for comparison. While NeRF is capable of reconstructing low-frequency geometry, it falls short in generating high-quality fine details. Furthermore, the rendering procedure employed by neural radiance fields involves sampling a scene with a single ray per pixel and producing results that are blurred (green boxes) or aliased (red boxes and arrows) when training or testing images observe scene content at different resolutions.

### E. Comparisons With the Rendering Software Results

To demonstrate the effectiveness of the proposed architecture, we compare the color images generated by our network with results produced by the software, including OpenGL [47] and Mitsuba [1]. Fig. 10 shows the comparison results. OpenGL [47] cannot reconstruct the color and illumination variation of the scenes (Fig. 10 (a)), while the results of Mitsuba [1] are noisy (Fig. 10 (b)). Mitsuba can produce clear images with realistic color and reasonable illumination variation sometimes. However, the ideal rendering process requires a significant amount of time. The generated images of Mitsuba tend to be noisy when suffering from complicated scenes and insufficient rendering time. In contrast, the proposed PBR-GAN can generate satisfactory results (Fig. 10 (c)) while significantly reducing the processing time.

### F. Computational Times

The computational times for generating images with size 360×480 on the test set are reported in Table III. We first show the computational times on an i7, 4 cores CPU. For OpenGL


Fig. 7. Comparisons with pix2pix [48], CycleGAN [49], CartoonGAN [50], StarGAN [51], Hneg-SRC [52] and PBR-Net [2]. PBR-GAN focuses on the highlight (blue arrow) and shadow (red arrows) regions, and gets results with fewer artifacts (left case) and a uniform color (right case). Our method is more effective for handling the illumination variation and produces more realistic results. These improvements are important for photo-realistic rendering tasks.

and Mitsuba, we report the average rendering time from a 3D scene to a viewpoint. OpenGL renders scenes rapidly and spends about $0.041s$, however, it cannot reconstruct the color and illumination variation. The Mitsuba can render images with more realistic details, but it takes a long time. The PBR-Net and PBR-GAN both cost approximately $11s$ to combine five inputs into the color output. The number in corresponding brackets are the total times, which first


Fig. 8. More comparison results with PBR-Net. The proposed PBR-GAN generates the highlight (blue arrows) and cast the shadow (green arrows) more obviously. Moreover, PBR-GAN learns correct illumination variation in the sofa, while the PBR-Net produces inverse highlight and shadow (red arrows).

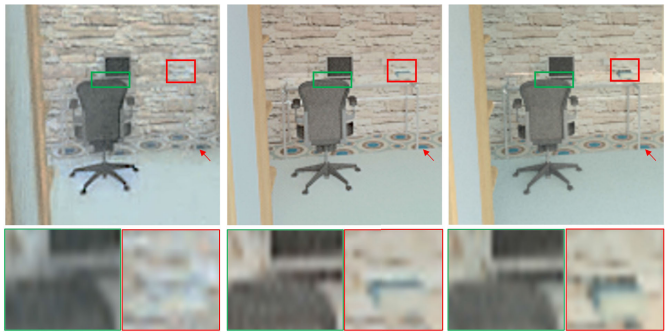


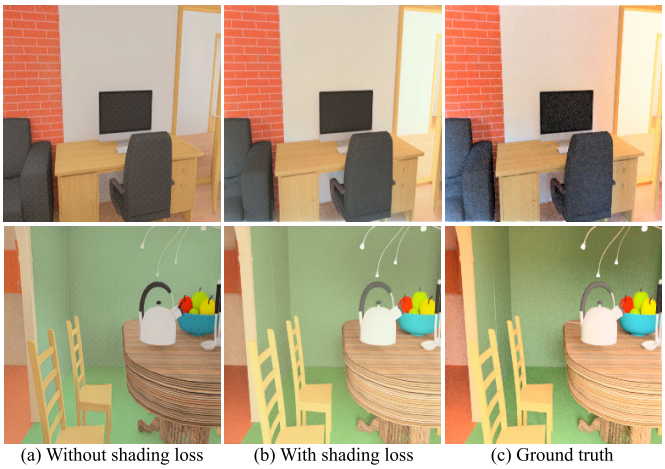

Fig. 9. Comparison results with NeRF [13]. (a) The new viewpoint of NeRF. (b) Color image generated by PBR-GAN. (c) Ground truth image.


Fig. 10. Comparison results with the rendering software. (a) Color image generated by the OpenGL [47]. (b) Color image rendered by the Mitsuba [1]. (c) Color images generated by the proposed method.

generate the five inputs from a viewpoint (the pre-computation process) and then combine them into the color image (the inference process). The pre-computation process takes approximately 4.5$s$. The generation of normal, depth, and albedo images spends 2$s$, while the generation of panoramic intensity and panoramic distance involves a simple equirectangular reprojection of a unit sphere to a 2D regular image, costing about 2.5$s$. The speed of PBR-GAN with pre-computation is still much faster than Mitsuba. We then exhibit the inference time of different learning-based methods on an NVIDIA RTX 2080Ti GPU. We use the same inputs for PBR-GAN and the learning-based comparison methods (except for NeRF). As for the comparison methods, we first apply three identical convolution blocks to extract the features of five inputs and then concatenate these features as their inputs. After training converged models of the comparison methods, the right part of Table III reports the average inference times on the test set when combining five images with size $360 \times 480$ to generate the photo-realistic images. pix2pix [48] and StarGAN [51] are the fastest algorithms due to their simple network architecture. NeRF [13] is much slower than other methods due to the volume rendering process. The proposed method is a bit slower than PBR-Net [2]. It is recommended to invest some additional computational time in incorporating a useful encoder and adding a light decoder to achieve better performance.

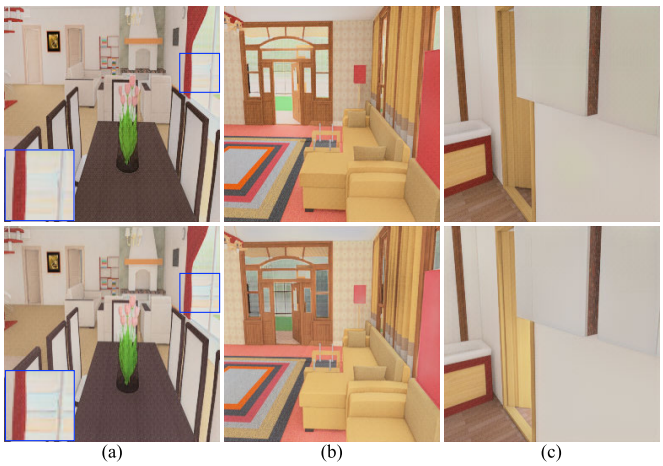
TABLE IV
ABLATION EXPERIMENTS OF DIFFERENT COMPONENTS. RED INDICATES THE BEST PERFORMANCE

| $L_{\text{PBR}}$ | $L_{\text{GAN}}$ | $L_{shading}$ | $D_L$ | $O_{lm}$ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|---|---|---|
| ✓ | | | | | 0.744(-7.23%) | 0.048(-6.67%) | 194(-13.45%) |
| ✓ | | ✓ | ✓ | ✓ | 0.771(-3.87%) | 0.047(-4.44%) | 185(-8.19%) |
| ✓ | ✓ | | ✓ | ✓ | 0.789(-1.62%) | 0.046(-2.22%) | 179(-4.68%) |
| ✓ | ✓ | ✓ | | | 0.765(-4.61%) | 0.047(-4.44%) | 189(-10.53%) |
| ✓ | ✓ | ✓ | ✓ | | 0.768(-4.24%) | 0.047(-4.44%) | 186(-8.77%) |
| ✓ | ✓ | ✓ | ✓ | ✓ | 0.802(+0.00%) | 0.045(+0.00%) | 171(+0.00%) |



(a) Without shading loss (b) With shading loss (c) Ground truth

Fig. 11. Ablation study of the shading loss $L_{shading}$. (a) Results without the shading loss. (b) Results with the shading loss. (c) Ground truth images.



(a) (b) (c)

Fig. 12. Ablation study of the adversarial loss $L_{\text{GAN}}$. The top row and the bottom row show the results when we remove and retain the $L_{\text{GAN}}$, respectively.

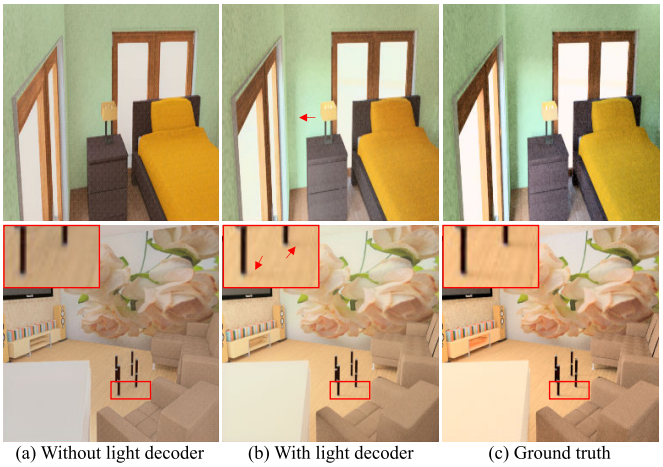
### G. Ablation Studies

We conduct several ablation studies of different components to understand how these main modules work. Fig. 12, Fig. 11, Fig. 13 and Table IV display the ablation results.

*1) Ablation Study of the Adversarial Loss:* We first perform some experiments to illustrate the importance of adversarial loss. As shown in the third row in Table IV, removing the adversarial loss $L_{\text{GAN}}$ apparently degrades the results, which is consistent with the qualitative result in Fig. 12. The top row of Fig. 12 shows the results when we remove the real dataset and the adversarial loss, while the bottom row displays the results with the adversarial loss included. The real dataset includes diverse indoor scenes, which makes the network sensitive to real distribution, and three cases in Fig. 12 indicate the importance of the collected dataset and the adversarial loss. Our method with the real dataset can preserve the details better in Fig. 12 (a) because the collected real images include semantic information outside the windows and the doors, while the synthetic dataset loses these details. Moreover, the images of the synthetic dataset are almost daytime scenes, whereas the real dataset includes diverse scenes, ranging from daytime to nighttime, which helps the network imitate the night scenes better (Fig. 12 (b)). Results in Fig. 12 (c) demonstrate the architecture can learn that the light often comes from the doors or the windows with the help of real data.

*2) Ablation Study of the Shading Loss:* The shading loss controls the overall illumination condition of the generated images. As shown in Fig. 11, with the shading loss, the top case can learn that the light comes from the right side, and the bottom case is influenced by the front light source. The overall illumination condition of the generated image is more similar to the ground truth with the help of the shading loss. The quantitative values in the fourth row in Table IV also demonstrate the importance of the shading loss. The generated results with better overall illumination exhibit greater similarity to photo-realistic images.

*3) Ablation Study of the Light Decoder:* The light loss is complementary to the shading loss, which concentrates more on the special regions and generates better highlight and shadow effects. We perform two ablation experiments to demonstrate the importance of the light decoder $D_L$ and light mask $O_{lm}$. The light decoder is first removed, which means the light loss $L_{light}$ and the light map $O_{lm}$ are invalid, and the PBR loss $L_{\text{PBR}}$ is adopted as the conventional perceptual loss. The quantitative results when removing the light decoder in Table IV (the fifth row) are worse than the complete PBR-GAN. Fig. 13 (a) displays the qualitative results without the light decoder, which are apparently inferior to the results with the light decoder in Fig. 13 (b). The results with the light decoder have similar illumination to the ground truth. Moreover, the light maps generated by the light decoder help to recover the shadows generated by the window and black table legs (red arrows). The light decoder focuses on the illumination variation and therefore obtains more realistic results. The light mask $O_{lm}$ generated by the light map is then evaluated. The light decoder $D_L$ and the light loss $L_{light}$ work normally. However, the generated light maps do not provide information



(a) Without light decoder (b) With light decoder (c) Ground truth

Fig. 13. Ablation study of the light decoder $D_L$. (a) Results without the light decoder. (b) Results with the light decoder. (c) Ground truth images.

TABLE V
ABLATION STUDY ON THE LPIPS THRESHOLD OF THE REAL DATASET

| $\text{lpips}_t$ | The number of images | SSIM ↑ | FID ↓ | Iteration time ($s$) |
|---|---|---|---|---|
| None | 1945 | 0.802 | 169 | 0.5716 |
| 0.1 | 1082 | 0.802 | 171 | 0.5023 |
| 0.5 | 616 | 0.755 | 180 | 0.4801 |

to get the light masks and the PBR loss is a conventional perceptual loss. The quantitative results in the sixth row of Table IV also indicate a performance decrease compared to the complete PBR-GAN. Note that, the performance without light mask $O_{lm}$ (the sixth row) is similar to the results without light decoder $D_L$ (the fifth row) because training the light decoder without light map information provided to the loss cannot improve the results significantly.

*4) Ablation Study of the LPIPS Threshold and the Number of Real Images:* After collecting the real dataset, LPIPS similarity [57] is applied to identify and remove similar images. We empirically select suitable thresholds of LPIPS ($\text{lpips}_t$) to filter the collected real images. We conduct the ablation study and report the SSIM and FID scores when selecting different thresholds and filtering different numbers of images in Table V. The second row in Table V indicates one collected dataset without any post-processing. The third row shows the results when setting the threshold $\text{lpips}_t$ to 0.1, which may filter the similar images and reserve as many as possible images with diversity. The last row exhibits the results when setting the threshold to 0.5. The running time of each iteration increases and the performance decreases with the larger threshold. When setting the threshold $\text{lpips}_t$ to 0.1, the model can achieve the balance of effectiveness and efficiency.

We then optimize the search keywords and add some adjectives, such as nighttime bedroom and vintage parlor, to expand the search scope. In total, we collect 11,035 images from the Internet. After applying two filtering operations, including the LPIPS similarity ($\text{lpips}_t$=0.1) and the manual selection, the final real dataset with 6,000 images is obtained. To explore the impact of the number of real images, we randomly select subsets of images and evaluate the FID score and iteration time for each subset. Figure 14 illustrates the corresponding

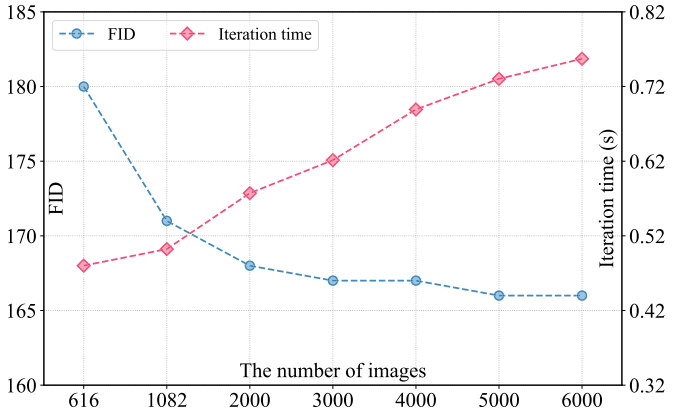
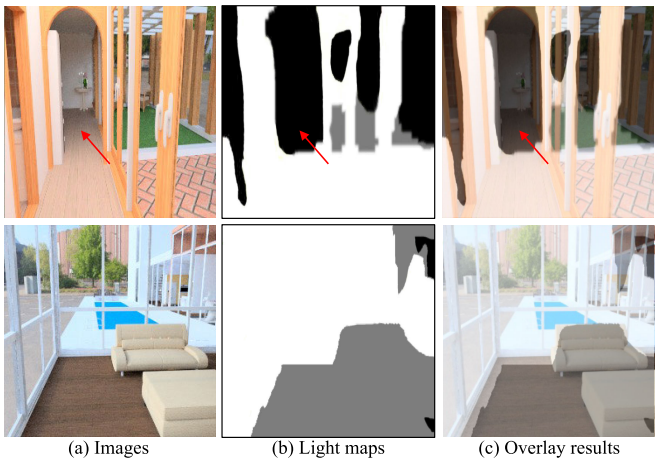



Fig. 14. The variation trend of FID score and iteration time when selecting different numbers of real images.



(a) Images (b) Light maps (c) Overlay results

Fig. 15. The generated light maps of outdoor scenes, which may fail to capture accurate illumination variations due to the discrepancies in lighting conditions between indoor and outdoor environments.

variation trend. It is evident that as the number of real images increases, the iteration time also increases, while the FID score decreases. However, the tendency of FID score from 2,000 to 6,000 is gentle. As for PBR task, increasing the number of images will lead to model improvement but the number between 1,000 and 2,000 is the best choice to strike the balance between effectiveness and efficiency because this range is diverse enough for the current architecture. Sometimes, a relatively small dataset can yield satisfactory performance while reducing training time. For example, the target datasets in CycleGAN include approximately 200-400 images (Van Gogh style: 400 images, Summer style: 309 images, Winter style: 208 images), and these styles can be learned reasonably.

### H. Discussion

While our method improves upon other methods and achieves an average improvement rate of 8%, it is limited by the scenario and may not perform well on outdoor scenes. The predicted indoor light map with three colors indicates the light source and the light irradiation areas, shading, and uniform texture regions, respectively. However, the outdoor light map deviates from the rules and cannot concentrate on the illumination variation. The significant discrepancies in lighting conditions between indoor and outdoor environments pose

challenges. As shown in the first case in Fig. 15, the white color dominates all the outdoor regions, while the illumination variation locates in indoor regions (red arrow). Focusing on the illumination variation is beneficial for generating photo-realistic outputs. Therefore, the illumination variation and cast shadow may easily be ignored and produce non-photo-realistic results. We consider this as a limitation and a future work. We plan to improve the light decoder to adapt to all scenarios, thus broadening its range of applications.

## V. Conclusion

We have proposed PBR-GAN, an end-to-end GAN-based method to achieve high-quality physically based rendering effectively. The proposed method first applies the generative adversarial network to speed up the photo-realistic rendering by simulating the majority of expensive components efficiently and then designs specific modules to deal with the illumination variation. The architecture includes two encoders and two decoders, among which two encoders combine the shading and reflectance information from the rendering sources and the two decoders recover the photo-realistic images and the light maps, respectively. In addition to the necessary conventional adversarial loss, we introduce the shading loss to preserve the shading information, the light loss to obtain accurate light maps, and the novel mask-based PBR loss that utilizes the light map as a mask to constrain the generated color images, enabling the network to focus more on the highlight and shadow regions. Comprehensive experiments have demonstrated the effectiveness of the PBR-GAN.

## References


[1] *Mitsuba*. Accessed: 2010. [Online]. Available: http://www.mitsuba-renderer.org/

[2] P. Dai, Z. Li, Y. Zhang, S. Liu, and B. Zeng, "PBR-Net: Imitating physically based rendering using deep neural network," *IEEE Trans. Image Process.*, vol. 29, pp. 5980–5992, 2020.

[3] M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*. San Mateo, CA, USA: Morgan Kaufmann, 2016.

[4] *Blender*. Accessed: 1994. [Online]. Available: http://www.blender.org/

[5] *Maya*. Accessed: 1998. [Online]. Available: https://www.autodesk.com.sg/products/maya/

[6] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, "Modeling the interaction of light between diffuse surfaces," *ACM SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 213–222, Jul. 1984.

[7] J. T. Kajiya, "The rendering equation," in *Proc. 13th Annu. Conf. Comput. Graph. Interact. Techn.*, Aug. 1986, pp. 143–150.

[8] L. Lin, J. Zhu, and Y. Zhang, "Multiview textured mesh recovery by differentiable rendering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 4, pp. 1684–1696, Apr. 2023.

[9] Y. Zhang et al., "Physically-based rendering for indoor scene understanding using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5057–5065.

[10] Z. Li and N. Snavely, "CGIntrinsics: Better intrinsic image decomposition through physically-based rendering," in *Proc. ECCV*, Sep. 2018, pp. 371–387.

[11] G. Liu, D. Ceylan, E. Yumer, J. Yang, and J.-M. Lien, "Material editing using a physically based rendering network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2280–2288.

[12] Y. Gao, X. Wang, Y. Li, L. Zhou, Q. Shi, and Z. Li, "Modeling method of a ladar scene projector based on physically based rendering technology," *Appl. Opt.*, vol. 57, no. 28, pp. 8303–8313, 2018.

[13] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. ECCV*, Aug. 2020, pp. 405–421.

[14] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5835–5844.

[15] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded anti-aliased neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5470–5479.

[16] V. Lazova, V. Guzov, K. Olszewski, S. Tulyakov, and G. Pons-Moll, "Control-NeRF: Editable feature volumes for scene rendering and manipulation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4329–4339.

[17] I. Hwang, J. Kim, and Y. M. Kim, "Ev-NeRF: Event based neural radiance field," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 837–847.

[18] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12872–12881.

[19] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6494–6504.

[20] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-NeRF: Neural radiance fields for dynamic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10313–10322.

[21] Q. Chen and V. Koltun, "A simple model for intrinsic image decomposition with depth cues," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 241–248.

[22] F. Zhang, X. Jiang, Z. Xia, M. Gabbouj, J. Peng, and X. Feng, "Non-local color compensation network for intrinsic image decomposition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 132–145, Jan. 2023.

[23] A. Akram and N. Khan, "SARGAN: Spatial attention-based residuals for facial expression manipulation," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 10, 2023, doi: 10.1109/TCSVT.2023.3255243.

[24] H. Zhang et al., "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5908–5916.

[25] R. Chen and Y. Zhang, "Learning dynamic generative attention for single image super-resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 12, pp. 8368–8382, Dec. 2022.

[26] Y. Guo, j. zhang, J. Cai, B. Jiang, and J. Zheng, "CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1294–1307, Jun. 2019.

[27] W. Li et al., "InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," 2018, *arXiv:1809.00716*.

[28] Z.-S. Liu, W.-C. Siu, and Y.-L. Chan, "Photo-realistic image super-resolution via variational autoencoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1351–1365, Apr. 2021.

[29] Z.-S. Liu, W.-C. Siu, and L.-W. Wang, "Variational AutoEncoder for reference based image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 516–525.

[30] I. Goodfellow et al., "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[31] R. Li et al., "SDP-GAN: Saliency detail preservation generative adversarial networks for high perceptual quality style transfer," *IEEE Trans. Image Process.*, vol. 30, pp. 374–385, 2021.

[32] R. Li, S. Liu, G. Wang, G. Liu, and B. Zeng, "JigsawGAN: Auxiliary learning for solving jigsaw puzzles with generative adversarial networks," *IEEE Trans. Image Process.*, vol. 31, pp. 513–524, 2022.

[33] J. Liu, Y. Zou, and D. Yang, "SemanticGAN: Generative adversarial networks for semantic image to photo-realistic image translation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2528–2532.

[34] D. Joo, D. Kim, and J. Kim, "Generating a fusion image: One's identity and another's shape," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1635–1643.

[35] Y. Niu, J. Wu, W. Liu, W. Guo, and R. W. H. Lau, "HDR-GAN: HDR image reconstruction from multi-exposed LDR images with large motions," *IEEE Trans. Image Process.*, vol. 30, pp. 3885–3896, 2021.

[36] R. Li et al., "UPHDR-GAN: Generative adversarial network for high dynamic range imaging with unpaired data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7532–7546, Nov. 2022.

[37] E. H. Land and J. J. McCann, "Lightness and retinex theory," *J. Opt. Soc. Amer.*, vol. 61, no. 1, pp. 1–11, 1971.

[38] S. Bell, K. Bala, and N. Snavely, "Intrinsic images in the wild," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–12, Jul. 2014.

[39] B. Kovacs, S. Bell, N. Snavely, and K. Bala, "Shading annotations in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 850–859.

[40] G. Han, X. Xie, J. Lai, and W.-S. Zheng, "Learning an intrinsic image decomposer using synthesized RGB-D dataset," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 753–757, Jun. 2018.

[41] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, "Incorporating second-order functional knowledge for better option pricing," in *Proc. NIPS*, 2001, pp. 472–478.

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

[43] Y. Hu, B. Wang, and S. Lin, "FC$^4$: Fully convolutional color constancy with confidence-weighted pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4085–4094.

[44] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. ECCV*, Oct. 2016, pp. 694–711.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[46] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[47] *OpenGL*. Accessed: 1997. [Online]. Available: https://www.opengl.org/

[48] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[49] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

[50] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "CartoonGAN: Generative adversarial networks for photo cartoonization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9465–9474.

[51] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.

[52] C. Jung, G. Kwon, and J. C. Ye, "Exploring patch-wise semantic relation for contrastive learning in image-to-image translation tasks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18239–18248.

[53] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 190–198.

[54] *Google*. Accessed: 2008. [Online]. Available: https://www.google.com/

[55] *Baidu*. Accessed: 2011. [Online]. Available: https://www.baidu.com/

[56] *YouTube*. Accessed: 2005. [Online]. Available: https://www.youtube.com/

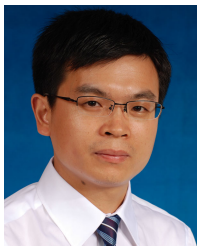
[57] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[58] 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.

[59] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. NIPS*, 2017, pp. 6626–6637.

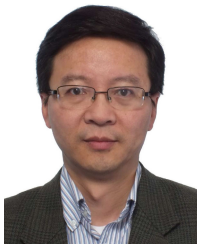

**Ru Li** (Member, IEEE) received the B.E. degree in electronic information engineering from the China University of Petroleum, Qingdao, China, in 2016, and the Ph.D. degree from the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2022. She was a Visiting Student Researcher with the University of Oxford. She is currently a Lecturer with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai. Her research interests include image processing and computer vision.

**Peng Dai** (Student Member, IEEE) received the B.E. and M.Sc. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree with the University of Hong Kong. His research interests include computer vision and computer graphics.

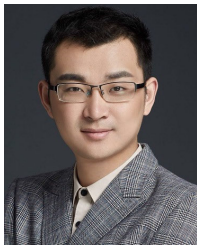


**Guanghui Liu** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2002 and 2005, respectively. In 2005, he joined Samsung Electronics, Seoul, South Korea, as a Senior Engineer. In 2009, he became an Associate Professor with the School of Electronics Engineering, UESTC, where he has been a Full Professor since 2014 and is currently with the School of Information and Communication Engineering. His general research interests include multimedia, remote sensing, and wireless communication.

**Shengping Zhang** received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013. He was a Post-Doctoral Research Associate with Brown University and Hong Kong Baptist University and a Visiting Student Researcher with the University of California at Berkeley. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai. His research interests include deep learning and its applications in computer vision.



**Bing Zeng** (Fellow, IEEE) received the B.E. and M.Sc. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the Ph.D. degree in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 1991. From September 1991 to July 1992, he was a Post-Doctoral Fellow with the University of Toronto. From August 1992 to January 1993, he was a Researcher with Concordia University. Then, he joined The Hong Kong University of Science and Technology (HKUST). He returned to UESTC in Summer 2013, through China 1000-Talent-Scheme. At UESTC, he leads the Institute of Image Processing to work on image and video processing, 3D and multiview video technology, and visual big data.



**Shuaicheng Liu** (Member, IEEE) received the B.E. degree from Sichuan University, Chengdu, China, in 2008, and the M.Sc. and Ph.D. degrees from the National University of Singapore, Singapore, in 2010 and 2014, respectively. In 2014, he joined the University of Electronic Science and Technology of China, Chengdu, where he is currently a Professor with the School of Information and Communication Engineering, Institute of Image Processing. His research interests include computer vision and computer graphics.