BokehDiff: Neural Lens Blur with One-Step Diffusion

Chengxuan Zhu[1,2†] Qingnan Fan[2*] Qi Zhang[2] Jinwei Chen[2] Huaqi Zhang[2] Chao Xu[1] Boxin Shi[3,4*]

[1]National Key Lab of General AI, School of Intelligence Science and Technology, Peking University

[2]Vivo Mobile Communication Co., Ltd.

[3]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University

[4]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University

{peterzhu,shiboxin}@pku.edu.cn, qingnanfan@vivo.com

https://github.com/FreeButUselessSoul/bokehdiff

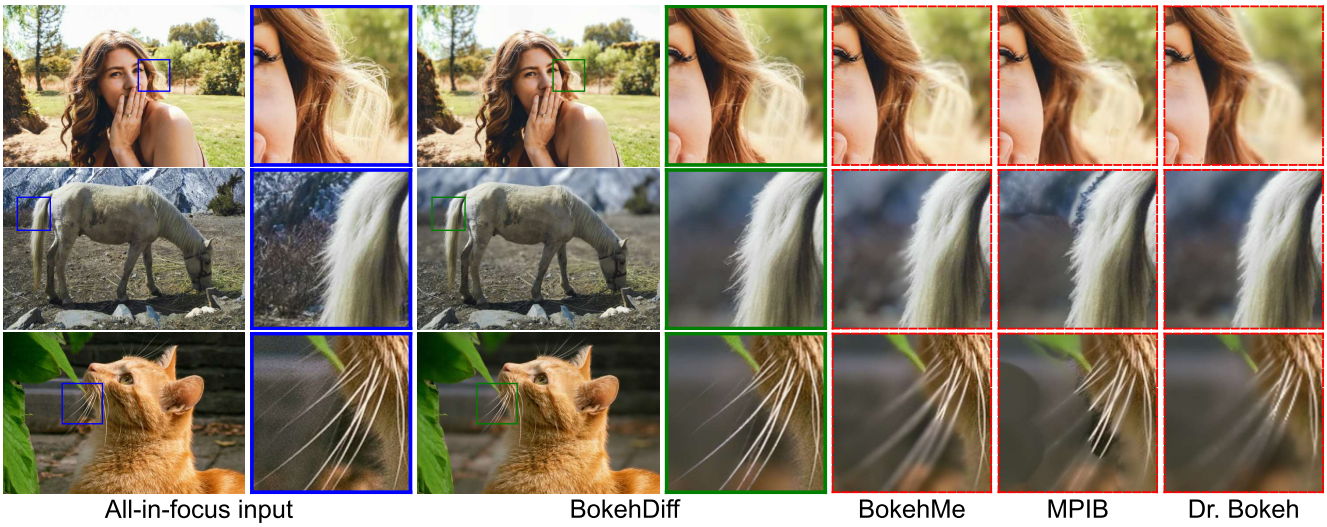

Figure 1. BokehDiff bridges the gap between physics and diffusion priors, and is able to synthesize photorealistic lens blur effects even when inaccurate depth estimation causes previous methods (BokehMe [37], MPIB [38], and Dr. Bokeh [46]) to fail, especially at the depth discontinuities. The examples show previous methods over-blur the horse's tail, the person's hair, and the whiskers of the cat.

Abstract

*We introduce **BokehDiff**, a novel lens blur rendering method that achieves physically accurate and visually appealing outcomes, with the help of generative diffusion prior. Previous methods are bounded by the accuracy of depth estimation, generating artifacts in depth discontinuities. Our method employs a physics-inspired self-attention module that aligns with the image formation process, incorporating depth-dependent circle of confusion constraint and self-occlusion effects. We adapt the diffusion model to the one-step inference scheme without introducing additional noise, and achieve results of high quality and fidelity. To address the lack of scalable paired data, we propose to synthesize photorealistic foregrounds with transparency with diffusion models, balancing authenticity and scene diversity.*

1. Introduction

The bokeh effect is the out-of-focus blurriness observed in photos, physically caused by using a lens with a large aperture, and is often used in portrait photography to emphasize the subject. Due to the cost of large aperture lenses, bokeh rendering has become a hot topic in the computational photography community. Previous works [28, 37, 38, 65] mostly aim to simulate the blurriness accurately with a pixel-level accurate depth estimation. However, since depth prediction tends to fail on edges and intricate details, artifacts can often be observed on structures such as people's hair and animals' fur, as shown in Fig. 1. As state-of-the-art diffusion models (*e.g.*, SDXL [41]) are already capable of generating photorealistic lens blur effects from text instructions [60], *can they be applied to render lens blur effects from a given image?*

The answer is frustrating, primarily due to diffusion models' inherent tendency to alter the content of the input

†Work done during an internship at Vivo.

*Corresponding authors.

image. The problem traces down to the iterative denoising process of diffusion methods, where the input image is injected into the model to guide the denoising process. The original noise introduces much uncertainty and tends to break the original structure of the input image. The denoising process is also too time-consuming to serve as a lens blur rendering tool, making the rich generative priors difficult to exploit. BokehDiff proposes to denoise the input image with only one denoising step, without adding any noise. It simply treats the all-in-focus image as the combination of the image with lens blur and unknown noise that needs to be removed. The noise prediction network is finetuned to learn the noise for transformation, and acquires the image lens blur with only one forward pass. BokehDiff effectively preserves the structures since no noise is added.

Another problem of diffusion models lies in the design of self-attention module. To emphasize more important features, self-attention may discard less important ones, even contradicting the underlying physics. It performs well in tasks like inpainting [1, 2, 27] and image super resolution [9, 32, 51, 61], where adjacent pixels are not influenced by each other. But for the task of lens blur rendering where the blur is aggregated from neighboring pixels, it is difficult for self-attention to control the results, due to the global receptive field and the neglect of unimportant pixels. The proposed BokehDiff, features a *physics-inspired self-attention* (PISA) module that is designed to immitate the physics in the image formation process. For the light sources in an image, the PISA module normalizes their contribution in an energy-conserved way, limits their impact by a physics-based *circle-of-confusion* (CoC) term, and mask the self-occlusion in light propagation.

For learning-based methods, the scarcity of high-quality paired data also poses a problem. Real-world paired data [21, 33] tend to suffer from misalignment caused by motion, lens breathing, or different exposure, with examples shown in the supplementary material. As for synthetic data, applying 3D engines to render bokeh from user-defined assets is constrained by the numbers of available scenes [33, 37], and the CG rendering differs from the reality. Another trend is to perform ray-tracing from several image layers [37–39, 46, 59, 65], but the imperfect matting contents make the final rendered results look fake, especially for intricate structures such as hair and hands. BokehDiff proposes a data synthesis paradigm to synthesize paired and aligned high-quality data for training and evaluation, by exploiting an off-the-shelf text-to-image model to synthesize photorealistic foreground with transparency [62] instead of segmenting the foreground from photos and build a synthetic dataset for training and testing.

We propose the first neural lens blur rendering pipeline based on pretrained diffusion priors, outperforming previous works in error-prone depth discontinuous areas. The contributions are summarized as follows:

- a physics-inspired self-attention module that follows the image formation model, considering the energy conservation laws, circle of confusion, and self-occlusion;
- an efficient one-step inference scheme with diffusion models, exploiting the generative priors;
- a new scalable data synthesis paradigm as well as a curated dataset for bokeh rendering, which solves the dilemma of ground-truth accuracy and scene diversity.

2. Related Works

2.1. Bokeh Rendering

As a common technique in photography and 3D rendering, lens blur is caused by the wide aperture of the camera. Mathematically, it equals the weighted sum of views from the neighborhood of the principle point [25, 42, 54].

However, real-world cases lack the complete 3D model or multiple view input. With the image as the only input, researchers face two main challenges, namely the missing information about the hidden surface and the inaccurate depth estimation. For the first problem, classical rendering extrapolate visible pixels to occluded ones [24–26] or performs inpainting [6, 38, 46, 50] to hallucinate the missing information. Either way, however, requires segmenting the scene into multiple planes, which is error-prone on depth discontinuous regions. Though efforts have been made to make the operation smooth [6, 38] or differentiable [46], they are outperformed by neural rendering methods when handling scenes with complex geometry.

Neural rendering uses a neural network to mimic the image formation model, and is often trained end-to-end on synthetic data with ground truth depth map [13, 20, 28, 33, 37, 43, 52, 59]. As the network learns to add specific amount of blur to the input image, the problem of inaccurate depth estimation constitutes the major bottleneck of performance, as seen in Fig. 1.

In this paper, we endow the diffusion priors to bokeh rendering, and significantly improves photorealism in regions where depth prediction methods fail.

2.2. Image Editing with Diffusion Models

As a powerful tool for image generation, diffusion models [17, 47] have caught much attention in the community, especially about the possibility exploiting the diffusion priors for controllable generation [7, 41, 56, 63] and editing [1, 2, 5, 7, 14, 22, 34, 53]. However, the stochastic nature of adding and removing noise makes it difficult for previous diffusion models to retain the original structure. Some guide the generation with the original latent map [7, 14, 29, 34, 53] or information injection [58, 63], but cannot preserve the pixel-wise structure; Others propose to blend the edited part with the original image [1, 2], but

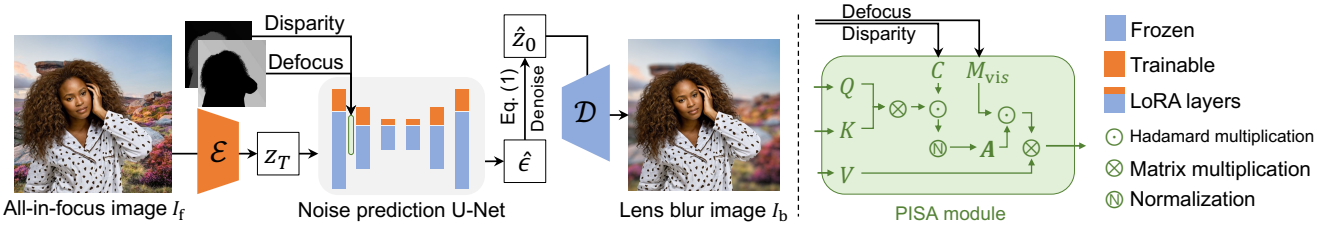

Figure 2. The framework of the proposed method. Given a paired synthetic data with disparity map, we optimize a LoRA of the U-Net and the encoder $\mathcal{E}$, while the decoder $\mathcal{D}$ remains frozen. A tailored PISA module (colored in green) is applied during downsampling, and is detailed in the right column, which is introduced in Sec. 3.2.

is limited to image inpainting task where only a small part should change.

Recently, researchers have found it more structure-preserving if some initial denoising steps, when the coarse structure is hallucinated from noise, are truncated [49]. The diffusion process can even be removed completely [51], with the low-quality image being the input to be denoised. The multi-step is also found redundent as it introduces accumulating error [16].

Taking the idea one step further, we use the all-in-focus input image as input, without adding noise to it, and adopt the efficient inference scheme of one-step denoise for the task of neural lens blur rendering.

3. Method

The task of lens blur rendering takes an all-in-focus image I_f as input, and blurs it with respect to the disparity map d and focus disparity d_f. The goal is to acquire the image with the correct lens blur I_b. Classical rendering methods apply a physics-based image formation model, such as the one illustrated in Fig. 3, while neural rendering methods learn the mapping from I_f to I_b directly. We aim to imitate the image formation model in the diffusion model, and prove that diffusion models can be adapted for the task. We first introduce the one-step diffusion framework in Sec. 3.1, and then detail the PISA module in Sec. 3.2, designed to make the diffusion model aware of the physics-related constraints. The framework of the proposed method is shown in Fig. 2. We then describe the data synthesis paradigm in Sec. 3.3 and the supervision in Sec. 3.4.

3.1. One-Step Diffusion for Bokeh Rendering

To save memory, diffusion models perform on latent space nowadays [17], with a pretrained encoder $\mathcal{E}$ to compress the images into latents and another decoder $\mathcal{D}$ to revert latents back to image space. Given a noisy latent z_t at timestep t, the denoised latent $\hat{z}_0$ is estimated by

$$\hat{z}_0 = \frac{z_t - \beta_t \cdot \epsilon_\theta(z_t; c_\mathrm{txt})}{\alpha_t}, \tag{1}$$

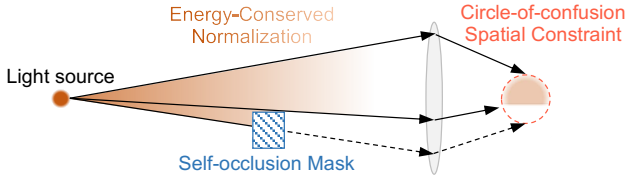

Figure 3. An illustration of the image formation model, and the three physics-related aspects considered in the PISA module.

where c_txt is the encoded text embedding as condition, and ϵ_θ stands for the U-Net predicting the noise. It's worth noting that the nature of Eq. (1) is only a transformation from z_t to $\hat{z}_0$, and the generative priors lie in the seeming omnipotence of ϵ_θ, as it is pre-trained on massive amount of data for noise prediction. In this sense, the pretraining of latent diffusion models is to map the Gaussian distribution into a desired output distribution. To exploit the rich generative priors of diffusion models, we base our generation on finetuning an off-the-shelf SDXL [41] text-to-image model.

While diffusion models are originally trained to remove noise, it is recently found that diffusion models can also be trained to invert other imposed degradation such as blurring, masking, or downsampling [3]. This motivates us to take one step further and ponder the possibility of implicitly defining the image quality as the amount of lens blur effect, and learn the transformation from I_f to I_b with physically correct lens blur effects.

As found by previous works, diffusion models tend to perform better in the timesteps they are trained on [15], implying the possibility of a one-step inference diffusion model [36, 51], which is trained on that particular timestep. In this paper, as the all-in-focus image is already close to the target domain, we fix the timestep as $T = 499$, and fintunes the LoRA [18] of the U-Net and the encoder $\mathcal{E}$, to fit the altered latent distribution.

3.2. Physics-Inspired Self-Attention Module

The Achilles's heel of applying the noise prediction network for neural lens rendering lies in the self-attention module, because it is ignorant of the 3D formulation of lens blur. We design the PISA module that follows the three physics-related aspects as illustrated in Fig. 3.

Energy-Conserved Normalization. In the vanilla self-

attention formulation, the output equals the product of the value vectors and the normalized similarity between query vectors and key vectors, namely

$$\text{Attn}(Q,K,V) = \mathbf{A}^{(\text{K})}V, \text{where } \mathbf{A}^{(\text{K})}_{qk} = \frac{\exp(A_{qk})}{\sum_s \exp(A_{qs})}. \quad (2)$$

Here Q, K, and V represent the query, key and value matrix, with each row representing a point in the latent map. $A = d_{\text{key}}^{-\frac{1}{2}} QK^\top$ is the similarity matrix, and d_{key} is the size of the key matrix. For convenience of notation, we use the subscript q and k to refer to the row of query point and the column of key point. d is the number of pixels, placed in the denominator for numerical stability. As previous works suggest, the normalized similarity $\mathbf{A}^{(\text{K})}$ between Q and K contains the structural information [19], while V possess the appearance information in the context of vision tasks. Thus, the result is a structurally weighted sum of appearance. Note that the normalization operation, Softmax$(\cdot)$ is applied on the channel of key, which guarantees that each row in the output is a normalized linear combination of the rows in V, with the weights summed up as 1. Since the latent pixels corresponds to the image pixels spatially, and V_k stands for the appearance feature at pixel k, the contribution of pixel k towards pixel q in the attention output can be measured by $\mathbf{A}_{qk}$. In most cases, the formulation enables neural networks to focus on important appearance features, without the concern that the rows in V can contribute very differently to the output.

We first redesign the normalization scheme, so that the energy of light does not increase or diminish as it spreads to neighboring pixels. As self-attention is originally designed to emphasize important features while discarding trivial ones, the total contribution of any given row V_k towards the output matrix varies drastically. Based on the physical inspiration, we propose to modify the softmax operation to normalize on query dimension, simply by

$$\mathbf{A}^{(\text{Q})}_{qk} = \frac{\exp(A_{qk})}{\sum_s \exp(A_{sk})}, \quad (3)$$

in which the energy conservative law holds, and the total contribution from any row in V to the output matrix is 1, with $\sum_i \mathbf{A}^{(\text{Q})}_{ik} = 1$.

Circle-of-Confusion Spatial Constraint. For a light source k that is off the focal plane, the CoC is formed on the camera sensor. Its radius $r_c(k)$ describes the extent of blurriness, and is proportionate to the disparity difference between the point and the focal plane [25, 26], given by

$$r_c(k) = |d_f - \text{dis}(P_k)| \cdot A, \quad (4)$$

where A is the camera parameters of the aperture diameter, and d_f is the disparity of the focal plane. k is any point light source in the context of self-attention, while P_k is the pixel location of point light source k. Let dis(P_k) denote the disparity (the reciprocal of depth) of k, shortened as d_k for convenience. For a practical application as lens blur rendering, the lens can be assumed as thin lens model [24, 25, 46], and thus Eq. (4) holds. In practice, $r_c(k)$ marks the theoretical limit of how far k can influence, by casting a cone of light through space. Without the spatial constraint, every feature can have an unlimited global effect, making it difficult for the network to neglect irrelevant pixels.

To consider the spatial constraint into the self-attention design, we propose to mask it at the softmax module. In this way, the conservation of energy still holds inside the circle-of-confusion, while the impact from outside is discarded by design, formulated as

$$\mathbf{A}^{(\text{QC})}_{qk} = \frac{\exp(A_{qk}) \odot C_{qk}}{\sum_s \exp(A_{sk}) \odot C_{qk}}, \quad (5)$$

and the mask C_{qk} is computed via

$$C_{qk} = \text{Soft}[r_c(k) - c_i \cdot \|P_q - P_q\|_2]. \quad (6)$$

For easier optimization, we apply a differentiable soft edge function Soft$(\cdot)$, which becomes sharper as the training goes, following previous works [46]. The detailed implementation is given in the supplementary material.

Self-Occlusion Mask. So far, the attention module has been modified to focus on the neighborhood with a given radius calculated from per-pixel disparity. We then consider the self-occlusion, caused by other pixels blocking the light propagation in 3D space. Different from previous methods that builds upon multi-plane images [38, 46], we calculate the pixel-wise occlusion map with sampling.

In practice, if a light source s is visible to P_q on the camera sensor, for any sampling point with disparity $\tilde{d}$ that lies between the light source s and P_q, it should not be blocked by the scene. Through the collinear relationship, the pixel location of the sampling point $\tilde{P}$ can be computed as

$$\tilde{P} = \frac{(1-\tilde{d})d_s}{(1-d_s)\tilde{d}}(P_s - P_q) + P_q. \quad (7)$$

Assuming a simple geometry of the scene, any sampling point should be closer to the camera so as to be not occluded. Therefore the visibility mask M_{vis} is given by

$$M_{\text{vis}} = \bigwedge_{\tilde{d}\in(d_s,1)} \left[\text{dis}\left(\frac{(1-\tilde{d})d_s}{(1-d_s)\tilde{d}}(P_s - P_q) + P_q \right) < \tilde{d} \right]. \quad (8)$$

For a more accurate rendering of the light source's impact, we super-sample k in the ϵ_s neighborhood of point light source s, and the PISA module is formulated as

$$\text{Attn}(Q,K,V)_{qk} = (\mathbf{A}^{(\text{QC})}_{qk} \odot \mathbb{E}_{s\sim\mathcal{N}(P_k,\epsilon_s)}[M_{\text{vis}}])V \quad (9)$$

3.3. Data Synthesis Pipeline

To learn the mapping of $I_{\text{f}} \rightarrow I_{\text{b}}$, high-quality paired data is needed for fine-tuning the noise prediction network. As depicted in Fig. 4, the data synthesis pipeline follows previous works [11, 37, 38] by using a ray tracing pipeline to synthesize images with various defocus amount and focus distances, given multiple layers of all-in-focus images. The bottleneck of previous works lies in the fact that high-quality foreground images are hard to acquire. Bounded by the accuracy of object segmentation, separating foreground objects from photos creates fake-looking photos [59], especially on regions like hair and fur. On the other hand, photos with green screen background and 3D models are not suitable for synthesizing data at a large scale.

We propose to synthesize photorealistic foreground with a state-of-the-art diffusion model [62], alleviating the dilemma of scalability and data quality. Samples of the synthetic dataset are demonstrated in the supplementary material. As shown in Fig. 4, we use real-world photos captured with a small aperture for background, and overlap it to synthetic foreground with transparency. By randomly placing the locations and facing angles of the layers, while controlling the focus on the average disparity of background or foreground, we are able to generate photorealistic synthetic data with a simplified ray tracer, with known disparity and focus distance, following the practice of previous methods [37, 40]. Note that the skewed facing angles makes it possible to learn the progressive blurring caused by a continuously changing disparity map. In this way, the model can learn to render the scene faithfully to the disparity map, instead of to semantic information only.

3.4. Supervision

Previous latent diffusion methods usually calculate the loss function in latent space, but as this paper aims for detail reconstruction, the loss are calculated in pixel space. For a robust reconstruction of the shape, we first calculate the

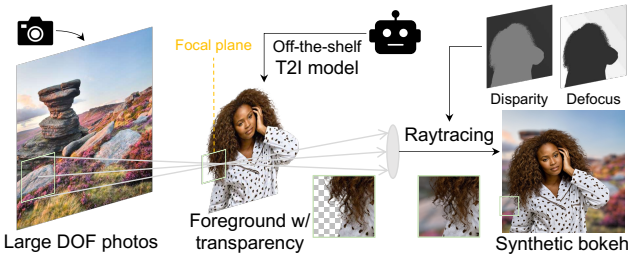

Figure 4. The data synthesis pipeline. A pretrained text-to-image model is applied to generate foreground with transparency [62], and the large depth-of-field background is selected from real-world images. With the layers randomly placed with various facing angles and various depths, a classical ray-tracing method is applied to render the image with lens blur.

Mean Square Error (MSE) $\mathcal{L}_{\text{MSE}}$ between the predicted image $\hat{I}$ and the ground truth I_{b}. But as MSE is insensitive to blurriness, relying on MSE can lead to the trivial solution of returning the all-in-focus input, or the other extreme of over-blurring. Therefore we consider the following loss functions which should be more sensitive to the lens blur:
(i) Perceptual loss $\mathcal{L}_{\text{VGG}}$. We apply the LPIPS loss which computes the distance between the image features extracted by a pretrained VGG network [64].
(ii) Multi-scale edge loss. As a strong visual clue, an obvious edge often indicates the image being in focus or not. To overcome the shortcomings of MSE, which can lead to blurry results, we follow previous works [8, 35, 44] and design the loss to focus more on the edges before and after the lens blur effect is applied, given by

$$\mathcal{L}_{\text{edge}} = \sum_{l=1}^{3} \frac{1}{l^2} \left\| (\nabla_l \hat{I} - \nabla_l I_{\text{b}}) \odot \max_{I \in \{I_{\text{b}}, I_{\text{f}}\}} |\nabla_l I| \right\|_1, \quad (10)$$

where ∇_l is the extended Sobel operator pair with the kernel size of l, in both horizontal and vertical directions. The term $\max_{I \in \{I_{\text{b}}, I_{\text{f}}\}} |\nabla I|$ basically neglects smooth regions, which is already a easy target to be optimized with $\mathcal{L}_{\text{MSE}}$.
(iii) Adversarial loss $\mathcal{L}_{\text{adv}}$. It employs a discriminator network D with a pretrained ConvNext [30] backbone to distinguish real images with lens blur I_b and generated images $\hat{I}$. The loss for the discriminator is given by

$$\mathcal{L}_{\text{D}} = \mathbb{E}_I[\log D(I_b)] + \mathbb{E}_{\hat{I}}[\log(1 - D(\hat{I}))], \quad (11)$$

while $\mathcal{L}_{\text{adv}} = -\mathbb{E}_{\hat{I}}[\log D(\hat{I})]$ is used for finetuning diffusion model. In all, the finetuning loss is given by

$$\mathcal{L} = \lambda_{\text{MSE}}\mathcal{L}_{\text{MSE}} + \lambda_{\text{VGG}}\mathcal{L}_{\text{VGG}} + \lambda_{\text{edge}}\mathcal{L}_{\text{edge}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}. \quad (12)$$

4. Experiments

4.1. Experimental Settings

Baselines. We select the following open-source state-of-the-art methods for baselines: DeepLens [28], an early end-to-end neural rendering method trained on synthetic data; MPIB [38], a physics-based method that considers the scene in layers, which inpaints on each layer and then blends the multi-layer by classical rendering; BokehMe [37], a hybrid rendering method that applies neural rendering in error-prone depth discontinuous regions, complementing the rest with a more controllable classical renderer; Dr.Bokeh [46], a hybrid rendering method that uses neural network for salient object segmentation and inpainting, and blends the layers differentiably.

We finetune the off-the-shelf BokehMe [37] model with the same synthetic data, input (disparity map and all-in-focus image), and loss terms as BokehDiff, to further validate the effectiveness of the model design, in addition to a Restormer model [61] trained from scratch.

Table 1. Quantitative comparison on the exposure-aligned EBB Val294 [21] dataset (left), and the user study results (right). The ratings are for *(i)* accuracy, *(ii)* authenticity, and *(iii)* preference. ↑ (↓) indicates larger (smaller) values are better, and **bold** font indicates the best results. ★ denotes that the method is trained or finetuned on the same dataset as BokehDiff.

Dataset	EBB Val294 [21] (real)				BLB Level 5 [37] (synthetic)			Real (user study)		
Method	PSNR↑	SSIM↑	DISTS↓	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	*(i)*↑	*(ii)*↑	*(iii)*↑
DeepLens [28]	22.703	0.7623	0.1483	0.4191	20.301	0.6901	0.2976	1.55	1.68	1.96
MPIB [38]	23.334	0.7920	0.1581	0.4031	28.162	0.8997	0.2561	1.83	1.89	2.04
BokehMe [37]	24.014	0.8134	0.1460	0.3921	**38.802**	**0.9870**	0.1404	3.81	3.93	4.03
Dr.Bokeh [46]	23.479	0.8221	0.1225	0.3771	22.650	0.7452	0.4539	3.41	3.38	3.64
Restormer★ [61]	23.960	0.7961	0.1297	0.3778	16.781	0.6866	0.7802	2.85	3.02	2.92
BokehMe★ [37]	23.753	0.7919	0.1437	0.3967	30.044	0.9409	0.1660	3.67	3.80	3.48
BokehDiff	**24.652**	**0.8357**	**0.1155**	**0.3737**	36.798	0.9814	**0.0888**	**4.42**	**4.37**	**4.56**

Datasets. Quantitative experiments are conducted on the real-world EBB Val294 [33, 59] dataset, and the synthetic datasets of BLB (Level 5) [37] and SYNBOKEH300 (synthesized as described in Sec. 3.3). As EBB Val294 dataset contains slight misalignments, we align the global mean value of the input image to the ground truth bokeh image. Please refer to the supplementary material for examples and more descriptions about the quantitative datasets.

For qualitative comparison and user study, the input images are gathered from the Unsplash dataset [10], the Easy Portrait dataset [23], and some photos taken by the authors in the wild with an aperture of f/22. The disparity maps are estimated by Depth Anything V2 [55, 57], and are shared across all the methods for a fair comparison.

Metrics. Following previous works, we report Peak Signal-to-Noise Ratio (PSNR) that focuses on pixel-wise accurate estimation, and Structural Similarity (SSIM) that measures structural similarity to the ground truth. As pointed by previous works [46, 64], PSNR is not sensitive to blurring. To complement the insufficient metrics, we additionally include LPIPS [64] and DISTS [12] for perceptual similarity, which mimics the response of human vision.

Implementation Details. The backbone model is a pre-trained SDXL model [45], and only the LoRA [18] of the downsampling layers in ϵ_θ and the middle block and output layers of $\mathcal{E}$ are trained, and the rest of the diffusion network is fixed. The AdamW [31] optimizer is used, with a cosine annealing learning rate scheduler, starting from 10^{-4}. The finetuning takes about 12 hours on a single NVIDIA L40s GPU, with a batch size of 2. The rank of LoRA module is set at 8 empirically. For hyper-parameter settings, we have $\lambda_{\text{MSE}} = 1$, $\lambda_{\text{VGG}} = 5$, $\lambda_{\text{adv}} = 0.5$, and $\lambda_{\text{edge}} = 1$.

4.2. Results and Comparisons

Quantitative Comparisons. Though the EBB Val294 dataset [21] involves aberration, camera motion, and other uncontrollable factors, BokehDiff still surpasses all previous baselines, as shown in the left columns in Tab. 1. For a more informed comparison, the comparison on the *original* (not exposure-aligned) EBB Val294 dataset [21] is attached in the supplementary material.

For the BLB dataset, the multi-layer based methods (MPIB [38] and Dr. Bokeh [37]) fail due to the complex scene layout, while the learning based DeepLens [28] and Restormer [61] also fails due to the insufficient knowledge of the underlying physics, as shown in the middle columns of Tab. 1. Both BokehMe [37] and BokehDiff have a decent performance, while the blur-sensitive LPIPS [64] indicates that BokehDiff renders more realistic bokeh pattern.

To measure the robustness to depth prediction error, we follow BokehMe [37] and conduct a test on SYNBOKEH300 dataset by eroding and dilating the disparity map. Shown in Fig. 7, BokehDiff constantly outperforms BokehMe [37] and Dr. Bokeh [46], with a less performance drop as the degeneration level raises, and the narrower quartiles further shows the stability of BokehDiff.

User Study. We conduct a user study, in which 50 volunteers with at least 1 year of photography experience are involved. Participants are shown with the all-in-focus image and the rendered results, and are asked to rate the results from 1 to 5. For each case, participants are randomly asked to focus on one of the following aspects: *(i)* accuracy, *e.g.*, the edge should be the same blurry as the surface on which it is located; *(ii)* authenticity, *e.g.*, the blurriness should change gradually with respect to the distance from focal plane; or simply *(iii)* preference as users. The results are listed on the rightmost columns in Tab. 1.

Qualitative Comparisons. According to Tab. 1, we only show the methods with superior quantitative performance here. In Fig. 5, three exemplar cases are shown, with more shown in the supplementary material. BokehDiff manages to maintain the intricate hair and fur details of the focused foreground in every example, even when the erroneous depth estimation erodes or dilates the defocus map. The transition from the focal plane to blurriness is smooth, as shown from the grass in the first column and the car roof in the second column. It can also blur the foreground off focus, such as the hands of the teenager in the first example.

As for the baselines, BokehMe [37] has the second best quality, by being loyal to the defocus map. Thus it also fails when depth estimation is inaccurate, especially in intricate depth discontinuities. In the zoomed patches of Fig. 5, it

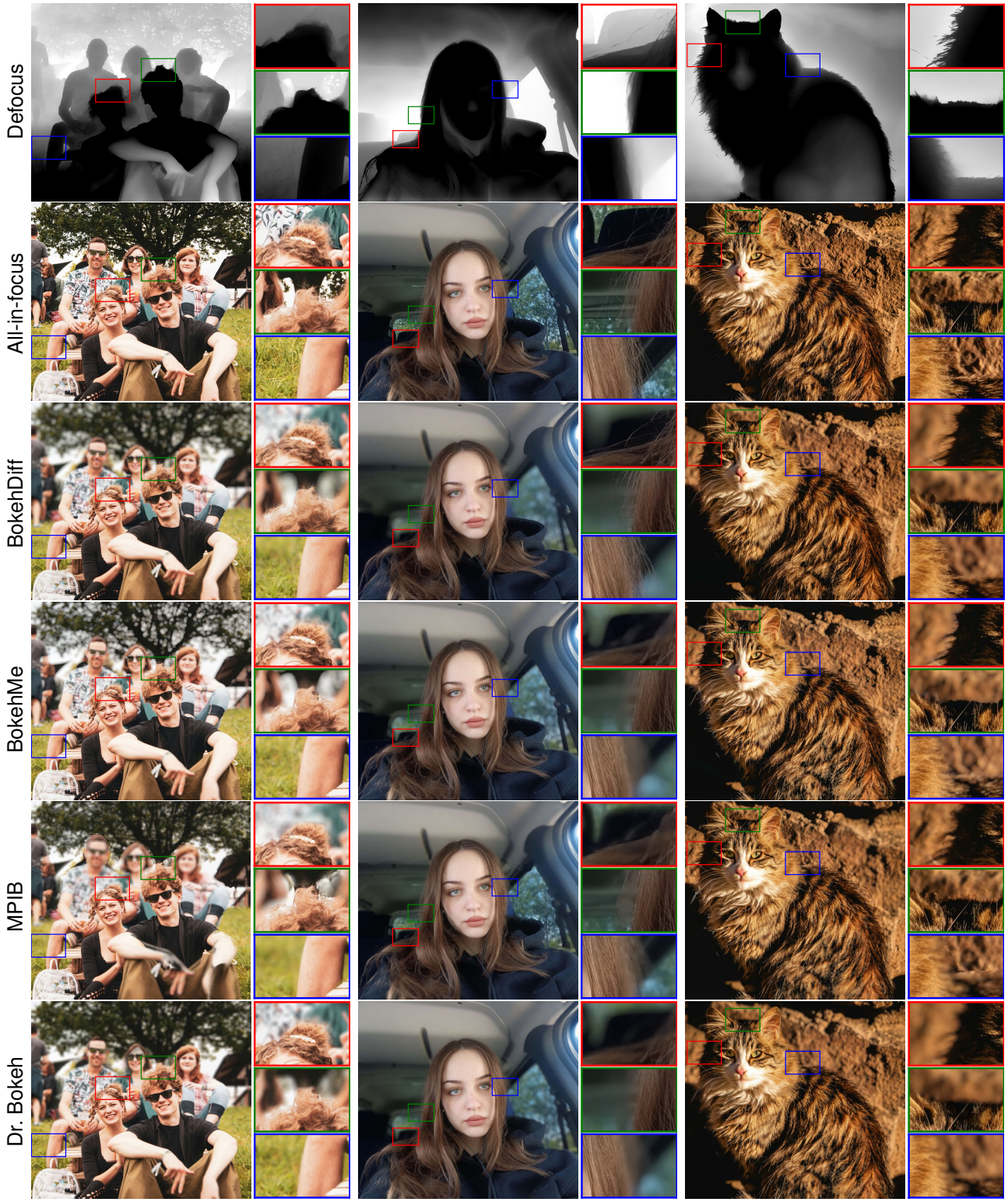

Figure 5. The qualitative comparisons of BokehDiff with BokehMe [37], MPIB [38], and Dr. Bokeh [46]. Calculated from disparity, the defocus map is shared across the methods to be compared, and three patches are zoomed in for closer observation in each scene. Whiter region in the defocus map indicates more lens blur should be added, but is prone to error caused by depth estimation.

Figure 6. A synthetic focal stack of BokehDiff, given an all-in-focus image selected from the Unsplash [10] dataset.

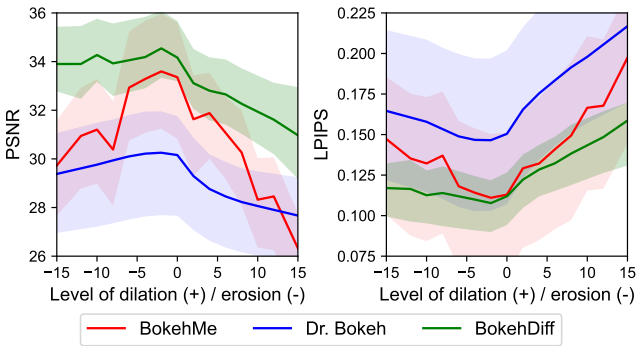

Figure 7. PSNR and LPIPS performance drop with respect to the erosion or dilation to the disparity map, on SYNBOKEH300. The semi-transparent area around each line is bounded by the quartiles.

over-blurs the hair and the cat's whiskers that should be focused. MPIB [38] fails to piece together layers where the complex scene cannot be easily separated into layers, such as the obvious artifacts around the teenager's arm and the man in the back rank in the first example. Though it sometimes renders the hair streaks right, it cannot generate progressive blur as the more focused background in the second and third examples show. Dr. Bokeh [46] does well in cases with a clean separation between foreground and background, but is also limited to the accuracy of the depth estimation, and the number of layers in question. It shows dark tints on the woman in the back rank, and also fails due to the inaccurate depth estimation.

In addition, we adjust the disparity of focal plane d_f, and show a focal stack in Fig. 6, verifying the ability to focus on any designated depth of BokehDiff.

4.3. Ablation Study

The results for the ablation study are listed in Tab. 2 We first ablate the supervisions for the one-step diffusion scheme by removing $\mathcal{L}_{\text{adv}}$, $\mathcal{L}_{\text{VGG}}$, and $\mathcal{L}_{\text{edge}}$. With the same training iterations, these settings achieve an inferior performance, especially when removing the multi-scale edge loss and the perceptual loss. We then consider the PISA module, namely the energy-conserved normalization (Eq. (3)), the circle-of-confusion constraint (Eq. (5)), and the self occlusion (Eq. (9)). The complete model excels in LPIPS and visual effects (shown in the supplementary material), validating the design of the PISA module. A fixed encoder

Table 2. The ablation study conducted on the exposure-aligned EBB Val294 [21] dataset. The setting of "SoftmaxQ", "CoC", and "occlusion" are short for the energy-conserved normalization, circle of confusion constraint, and self-occlusion respectively.

Setting	PSNR↑	SSIM↑	LPIPS↓
w/o $\mathcal{L}_{\text{adv}}$	24.623	0.8322	0.3768
w/o $\mathcal{L}_{\text{VGG}}$	24.285	0.8196	0.4218
w/o $\mathcal{L}_{\text{edge}}$	24.628	0.8346	0.3785
fixed $\mathcal{E}$	24.266	0.8286	0.3811
w/o CoC	22.217	0.6881	0.4280
w/o SoftmaxQ	24.468	0.8325	0.3800
w/o occlusion	24.399	0.8291	0.3808
$T = 249$	24.646	0.8335	0.3781
$T = 749$	24.481	0.8319	0.3838
Complete model	**24.652**	**0.8357**	**0.3737**

slightly decreases the performance, as the backbone needs to modify the latent more in this setting. Different timestep configurations are also tested, and the results indicate similar performance with $T = 249$ or $T = 749$. But in practice, extremely large or low timestep can easily lead to gradient explosion, and 499 is the choice of balance.

5. Conclusions

The paper proposes BokehDiff, a diffusion framework with only one inference step that achieves outstanding quality compared with previous methods, especially in regions where depth prediction fails. The diffusion priors, combined with the PISA module which is specifically designed for physics constraint, shed light on a new possibility for neural lens blur rendering and physic-based deep learning. Quantitative comparisons, visual results, and a user study all validate that BokehDiff is able to synthesize photorealistic lens blur, and robust against error in depth estimation.

Limitations. Though the finetuned diffusion network keeps the majority of the structures from the all-in-focus image, the decoder of the VAE still cause inevitable changes to less noticeable structures. The issues can be addressed by changing the diffusion backbone [4, 48] with less information compression and better detail preservation.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant No. 62136001, 62088102, and 62276007. PKU-affiliated authors would like to thank openbayes.com for providing computing resource. We thank Zhifeng Wang, Zhihao Yang, and Yichen Sheng for providing access and advice for the EBB! [21] dataset. As an important source for lens blur effects rendering demonstration, we appreciate the photographers with Unsplash, the Unsplash developers, and Victor Ballesteros for granting the access to the Unsplash dataset [10]. Also thanks to Jiangang Wang and other colleagues for the discussions during Chengxuan Zhu's internship at Vivo.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proc. of Computer Vision and Pattern Recognition*, 2022. 2

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics*, 42(4), 2023. 2

[3] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. In *Proc. of Advances in Neural Information Processing Systems*, 2023. 3

[4] Black Forest Labs. Flux.1-schnell, 2024. https://huggingface.co/black-forest-labs/FLUX.1-schnell, Last accessed on 2024-10-31. 8

[5] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proc. of Computer Vision and Pattern Recognition*, 2023. 2

[6] Benjamin Busam, Matthieu Hog, Steven McDonagh, and Gregory Slabaugh. SteReFo: Efficient image refocusing with stereo vision. In *Proc. of International Conference on Computer Vision Workshops*, 2019. 2

[7] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. MasaCtrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proc. of International Conference on Computer Vision*, 2023. 2

[8] Gang Chen, Guipeng Zhang, Zhenguo Yang, and Wenyin Liu. Multi-scale patch-gan with edge detection for image inpainting. *Applied Intelligence*, 53(4), 2023. 5

[9] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proc. of Computer Vision and Pattern Recognition*, 2023. 2

[10] Luke Chesser, Timothy Carbone, and Ali Zahid. Unsplash full dataset 1.2.2, 2020. unsplash.com/data, Last accessed on 2024-10-31. 6, 8, 9

[11] Marcos V. Conde, Manuel Kolmet, Tim Seizinger, Tom E. Bishop, Radu Timofte, Xiangyu Kong, Dafeng Zhang, Jinlong Wu, Fan Wang, Juewen Peng, Zhiyu Pan, Chengxin Liu, Xianrui Luo, Huiqiang Sun, Liao Shen, Zhiguo Cao, Ke Xian, Chaowei Liu, Zigeng Chen, Xingyi Yang, Songhua Liu, Yongcheng Jing, Michael Bi Mi, Xinchao Wang, Zhihao Yang, Wenyi Lian, Siyuan Lai, Haichuan Zhang, Trung Hoang, Amirsaeed Yazdani, Vishal Monga, Ziwei Luo, Fredrik K. Gustafsson, Zheng Zhao, Jens Sjölund, Thomas B. Schön, Yuxuan Zhao, Baoliang Chen, Yiqing Xu, and JiXiang Niu. Lens-to-lens bokeh effect transformation. ntire 2023 challenge report. In *Proc. of Computer Vision and Pattern Recognition Workshops*, 2023. 5

[12] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5), 2020. 6

[13] Saikat Dutta, Sourya Dipta Das, Nisarg A Shah, and Anil Kumar Tiwari. Stacked deep multi-scale hierarchical network for fast bokeh effect rendering from a single image. In *Proc. of Computer Vision and Pattern Recognition Workshops*, 2021. 2

[14] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *Proc. of Advances in Neural Information Processing Systems*, 2023. 2

[15] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proc. of International Conference on Computer Vision*, 2023. 3

[16] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Liu, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024. 3

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. of Advances in Neural Information Processing Systems*, 2020. 2, 3

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3, 6

[19] Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian. Global self-attention as a replacement for graph convolution. In *Conference on Knowledge Discovery and Data Mining*, 2022. 4

[20] Andrey Ignatov, Jagruti Patel, and Radu Timofte. Rendering natural camera bokeh effect with deep learning. In *Proc. of Computer Vision and Pattern Recognition Workshops*, 2020. 2

[21] Andrey Ignatov, Radu Timofte, Ming Qian, Congyu Qiao, Jiamin Lin, Zhenyu Guo, Chenghua Li, Cong Leng, Jian Cheng, Juewen Peng, et al. Aim 2020 challenge on rendering realistic bokeh. In *Proc. of European Conference on Computer Vision Workshops*, 2020. 2, 6, 8, 9

[22] Zeyinzi Jiang, Chaojie Mao, Yulin Pan, Zhen Han, and Jingfeng Zhang. SCEdit: Efficient and controllable image diffusion generation via skip connection editing. In *Proc. of Computer Vision and Pattern Recognition*, 2024. 2

[23] Alexander Kapitanov, Karina Kvanchiani, and Kirillova

Sofia. EasyPortrait - face parsing and portrait segmentation dataset. *arXiv preprint arXiv:2304.13509*, 2023. 6

[24] Martin Kraus and Magnus Strengert. Depth-of-field rendering by pyramidal image processing. In *Computer Graphics Forum*, 2007. 2, 4

[25] Sungkil Lee, Gerard Jounghyun Kim, and Seungmoon Choi. Real-time depth-of-field rendering using point splatting on per-pixel layers. In *Computer Graphics Forum*, 2008. 2, 4

[26] Sungkil Lee, Elmar Eisemann, and Hans-Peter Seidel. Real-time lens blur effects and focus control. *ACM Transactions on Graphics*, 29(4), 2010. 2, 4

[27] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. MAT: Mask-aware transformer for large hole image inpainting. In *Proc. of Computer Vision and Pattern Recognition*, 2022. 2

[28] Wang Lijun, Shen Xiaohui, Zhang Jianming, Wang Oliver, Lin Zhe, Hsieh Chih-Yao, Kong Sarah, and Lu Huchuan. DeepLens: Shallow depth of field from a single image. *ACM Transactions on Graphics*, 37(6), 2018. 1, 2, 5, 6

[29] Kuan Heng Lin, Sicheng Mo, Ben Klingher, Fangzhou Mu, and Bolei Zhou. Ctrl-X: Controlling structure and appearance for text-to-image generation without guidance. *arXiv preprint arXiv:2406.07540*, 2024. 2

[30] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proc. of Computer Vision and Pattern Recognition*, 2022. 5

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. of International Conference on Learning Representations*, 2019. 6

[32] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution. In *Proc. of Computer Vision and Pattern Recognition*, 2022. 2

[33] David Mandl, Shohei Mori, Peter Mohr, Yifan Peng, Tobias Langlotz, Dieter Schmalstieg, and Denis Kalkofen. Neural bokeh: Learning lens blur for computational videography and out-of-focus mixed reality. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2024. 2, 6

[34] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2

[35] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *Proc. of International Conference on Computer Vision Workshops*, 2019. 5

[36] Gaurav Parmar, Taesung Park, Srinivasa Narasimhan, and Jun-Yan Zhu. One-step image translation with text-to-image models. *arXiv preprint arXiv:2403.12036*, 2024. 3

[37] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. BokehMe: When neural rendering meets classical rendering. In *Proc. of Computer Vision and Pattern Recognition*, 2022. 1, 2, 5, 6, 7

[38] Juewen Peng, Jianming Zhang, Xianrui Luo, Hao Lu, Ke Xian, and Zhiguo Cao. MPIB: An MPI-based bokeh rendering framework for realistic partial occlusion effects. In *Proc. of European Conference on Computer Vision*, 2022. 1, 2, 4, 5, 6, 7, 8

[39] Juewen Peng, Zhiyu Pan, Chengxin Liu, Xianrui Luo, Huiqiang Sun, Liao Shen, Ke Xian, and Zhiguo Cao. Selective bokeh effect transformation. In *Proc. of Computer Vision and Pattern Recognition Workshops*, 2023. 2

[40] Juewen Peng, Zhiguo Cao, Xianrui Luo, Ke Xian, Wenfeng Tang, Jianming Zhang, and Guosheng Lin. BokehMe++: Harmonious fusion of classical and neural rendering for versatile bokeh creation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 5

[41] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1, 2, 3

[42] Michael Potmesil and Indranil Chakravarty. A lens and aperture camera model for synthetic image generation. *Proc. of ACM SIGGRAPH*, 15(3), 1981. 2

[43] Ming Qian, Congyu Qiao, Jiamin Lin, Zhenyu Guo, Chenghua Li, Cong Leng, and Jian Cheng. BGGAN: Bokeh-glass generative adversarial network for rendering realistic bokeh. In *Proc. of European Conference on Computer Vision Workshops*, 2020. 2

[44] George Seif and Dimitrios Androutsos. Edge-based loss function for single image super-resolution. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, 2018. 5

[45] SG_161222. RealVisXL 5.0, 2024. https://huggingface.co/SG161222/RealVisXL_V5.0, Last accessed on 2024-10-31. 6

[46] Yichen Sheng, Zixun Yu, Lu Ling, Zhiwen Cao, Xuaner Zhang, Xin Lu, Ke Xian, Haiting Lin, and Bedrich Benes. Dr. Bokeh: Differentiable occlusion-aware bokeh rendering. In *Proc. of Computer Vision and Pattern Recognition*, 2024. 1, 2, 4, 5, 6, 7, 8

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[48] Stability AI. Stable diffusion 3, 2024. https://huggingface.co/stabilityai/stable-diffusion-3-medium, Last accessed on 2024-10-31. 8

[49] L Sun, R Wu, Z Zhang, H Yong, and L Zhang. Improving the stability of diffusion models for content consistent super-resolution. *arXiv preprint arXiv:2401.00877*, 2024. 3

[50] Karthik Vaidyanathan, Jacob Munkberg, Petrik Clarberg, and Marco Salvi. Layered light field reconstruction for defocus blur. *ACM Transactions on Graphics*, 34(2), 2015. 2

[51] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *arXiv preprint arXiv:2406.08177*, 2024. 2, 3

[52] Lei Xiao, Anton Kaplanyan, Alexander Fix, Matthew Chapman, and Douglas Lanman. Deepfocus: learned image synthesis for computational displays. *ACM Transactions on Graphics*, 37(6), 2018. 2

[53] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *Proc. of Computer Vision and Pattern Recognition*, 2024. 2

[54] Ling-Qi Yan, Soham Uday Mehta, Ravi Ramamoorthi, and Fredo Durand. Fast 4d sheared filtering for interactive rendering of distribution effects. *ACM Transactions on Graphics*, 35(1), 2015. 2

[55] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proc. of Computer Vision and Pattern Recognition*, 2024. 6

[56] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proc. of Computer Vision and Pattern Recognition*, 2024. 2

[57] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 6

[58] Tao Yang, Rongyuan Wu, Peiran Ren, Xuansong Xie, and Lei Zhang. Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization. *arXiv preprint arXiv:2308.14469*, 2023. 2

[59] Zhihao Yang, Wenyi Lian, and Siyuan Lai. BokehOrNot: Transforming bokeh effect with image transformer and lens metadata embedding. In *Proc. of Computer Vision and Pattern Recognition*, 2023. 2, 5, 6

[60] Yu Yuan, Xijun Wang, Yichen Sheng, Prateek Chennuri, Xingguang Zhang, and Stanley Chan. Generative photography: Scene-consistent camera control for realistic text-to-image synthesis. In *Proc. of Computer Vision and Pattern Recognition*, 2025. 1

[61] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proc. of Computer Vision and Pattern Recognition*, 2022. 2, 5, 6

[62] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *arXiv preprint arXiv:2402.17113*, 2024. 2, 5

[63] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proc. of International Conference on Computer Vision*, 2023. 2

[64] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of Computer Vision and Pattern Recognition*, 2018. 5, 6

[65] Xuaner Zhang, Kevin Matzen, Vivien Nguyen, Dillon Yao, You Zhang, and Ren Ng. Synthetic defocus and look-ahead autofocus for casual videography. *ACM Transactions on Graphics*, 38(4), 2019. 1, 2