User-Instructed Disparity-aware Defocus Control

Yudong Han[1] **Yan Yang**[2] **Hao Yang**[1] **Liyuan Pan**[1,3]✉
[1]Beijing Institute of Technology, Beijing, China
[2]BDSI, Australian National University, Canberra, Australia
[3] Yangtze Delta Region Academy of Beijing Institude of Technology, Jiaxing, China
hanyudong.sdu@gmail.com, yan.yang@anu.edu.au, {hao.yang, liyuan.pan}@bit.edu.cn

Figure 1: An example of refocusing using our method. The input and refocused images are on the left and right. Two regions are highlighted with red and blue boxes, and their zoomed-in views are displayed in the center.

Abstract

In photography, an All-in-Focus (AiF) image may not always effectively convey the creator's intent. Professional photographers manipulate Depth of Field (DoF) to control which regions appear sharp or blurred, achieving compelling artistic effects. For general users, the ability to flexibly adjust DoF enhances creative expression and image quality. In this paper, we propose UiD, a User-Instructed DoF control framework, that allows users to specify refocusing regions using text, box, or point prompts, and our UiD automatically simulates in-focus and out-of-focus (OoF) regions in the given images. However, controlling defocus blur in a single-lens camera remains challenging due to the difficulty in estimating depth-aware aberrations and the suboptimal quality of reconstructed AiF images. To address this, we leverage dual-pixel (DP) sensors, commonly found in DSLR-style and mobile cameras. DP sensors provide a small-baseline stereo pair in a single snapshot, enabling depth-aware aberration estimation. Our approach first establishes an invertible mapping between OoF and AiF images to learn spatially varying defocus kernels and the disparity features. These depth-aware kernels enable bidirectional image transformation—deblurring out-of-focus (OoF) images into all-in-focus (AiF) representations, and conversely reblurring AiF images into OoF outputs—by seamlessly switching between the kernel and its inverse form. For user-guided refocusing, we first generate masks based on user prompts using SAM, which modulates disparity features in closed form, allowing dynamic kernel re-estimation for reblurring. This achieves user-controlled refocusing effects. Extensive experiments on both common datasets and the self-collected dataset demonstrate that UiD offers superior flexibility and quality in DoF manipulation imaging.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Modern cameras utilise sophisticated compound lenses designed to focus light onto the sensor. However, adjusting the aperture has another effect: a larger aperture (lower f-number) introduces depth-dependent blur, rendering only specific regions in focus [20]. This depth of field (DoF) effect is often intentionally used by photographers to direct viewer attention to a subject (e.g., macro photography) or create artistic compositions (e.g., tilt-shift photography) [2]. Unlike professional photographers, who possess the expertise to control DoF effectively, general users often struggle with achieving the desired DoF composition. Therefore, a framework that enables users to flexibly manipulate sharp and blurred regions in their captured images using simple text or point-based inputs to create visually striking effects would be a valuable application.

Given images with undesired DoF regions captured by general users, effectively controlling depth-dependent blur involves several key steps: (i) estimating the all-in-focus (AiF) image; (ii) computing the corresponding depth/defocus map of the undesired image; (iii) generating a mask to designate regions for in-focus and out-of-focus (OoF); and (iv) adjusting the defocus map with the mask for refocusing. Inspired by [52], which employs dual-pixel (DP) sensors [8] to synthetically generate blur on human subjects, we integrate DP sensors into our framework for flexible, user-instructed refocusing.

DP sensors are widely used in DSLR and smartphone cameras [22]. These sensors divide each pixel into two photodiodes, capturing separate left and right views of a scene, effectively forming a stereo system with a tiny baseline [39]. The depth information inherently encoded in the DP pair also contains defocus information [6], which makes DP image pairs highly beneficial for tasks such as defocus deblurring, refocusing, and bokeh effect synthesis [52]. Therefore, in this work, we focus on DP refocusing by incorporating user-specified prompts to identify desired focus regions, thereby enabling our network to effectively address all key steps within a single unified framework.

Different from existing work [52]—which synthesizes DSLR-style images from smartphone-captured data by directly treating the input DP image as the AiF image during refocusing—our task is more challenging. First, to estimate the latent AiF image and the defocus map, we jointly optimize defocus deblurring and reblurring. In contrast to earlier methods [5, 45, 56, 57] that treat deblurring and reblurring as independent processes or rely on ground truth depth maps for reblurring, we propose an invertible deblurring and reblurring framework that learns the robust defocus map and the spatially-varying associated blur kernels to assist the final refocusing step. Next, to facilitate mask generation, we employ flexible user inputs—such as point or box prompts—and models like SAM [43, 62] to generate masks that guide the refocusing process. Finally, to ensure that the generated mask effectively controls the depth-aware blur, we use the blur kernels learned by the invertible deblurring and reblurring framework to produce the final refocused image.

Specifically, we first analyze the relationship between DP disparity, the circle of confusion (CoC, or defocus map), and spatially varying defocus blur, which leads to a closed-form solution for controlling the disparity of a DP pair to achieve refocusing. This insight enables our network to perform controllable refocusing at test time simply by specifying regions of interest—even without explicit training on refocused images or reliance on depth annotations. Building on this understanding, we perform both invertible deblurring and reblurring within a unified framework, enabling the network to learn and regularize the mapping between blurred DP inputs and the sharp image in a self-consistent manner via learned disparity features. These learned disparity features are then converted into CoC features, which are used to construct spatially varying deblurring kernels. To ensure consistency, these kernels are further constrained to perform reblurring when inverted, thereby enabling effective simulation of the DP model. During testing, we first restore an all-in-focus image from the input DP pair and then modulate its disparity using a user-specified mask to generate the target CoC features, guiding the reblurring process for spatially controllable refocusing. In addition to comprehensively evaluating the robustness and flexibility of the proposed method in real-world refocusing scenarios and to enrich community benchmarking, we further construct a dual-pixel real-world evaluation dataset. The detailed evaluations and visualizations will be provided in the supplementary material. Our contributions are as follows:

- A framework that enables general users to flexibly manipulate sharp and blurred regions using simple text or point-based inputs.
- An invertible deblurring and reblurring framework that learns the defocus map and associated blur kernels in close alignment with the mathematical formulation of the DP model.

- Extensive experiments, including a self-collected dual-pixel real-world dataset, on defocus deblurring and image refocusing, demonstrate the superior performance of our method.

2 Related Work

Defocus Deblurring. Defocus blur refers to the blur effect when observed objects fall outside the depth-of-field (DoF) of a camera lens. To recover lost details from such blurred images, one widely-embraced scheme [1, 26, 58] is a two-stage pipeline: (1) estimating an explicit defocus map that accurately reflects the level of blur, followed by (2) performing deblurring guided by this map. Earlier approaches often relied on hardware assistance [10, 12, 17] to obtain a depth map as a proxy for defocus map in deblurring tasks. To improve flexibility, several studies have explored using pre-trained monocular depth estimators to infer depth maps from single images, which are then used as guidance for defocus-aware deblurring [40, 41]. While these methods offer greater accessibility, they often struggle to generalize beyond their training datasets. To enhance depth estimation accuracy for defocus map recovery, other researchers have incorporated additional cues, such as dual-pixel (DP) data [52]. DP images provide defocus-related disparity information that significantly improves the guidance quality during the deblurring process [42]. Wadhwa *et al.* [52] propose the first framework that uses DP sensors to enlarge the blur effect in the background to sythesize the DSLR-style images, and assume the foreground faces are in focus. Different from the settings in [52], in this paper, our UiD leverages the disparity cues encoded in DP image pairs to flexibly control the blur in a scalable and effective manner.

Image Refocus. Achieving accurate refocusing in post-capture images is challenging, as it requires both deblurring sharp regions and synthesizing realistic blur effects that align with the desired DoF [45]. Several studies [16, 36, 54] have explored software- and hardware-based approaches to enhance refocusing performance. Prior work [36, 54] often relies on specialized hardware to capture light field information, enabling post-capture focus control. However, light field cameras typically suffer from limited spatial resolution, making it difficult to capture intricate scene dynamics. An alternative solution [7, 48] is to capture a focus stack and leverage multi-frame information to estimate the desired focus distance. However, capturing long-term focus stacks inevitably leads to scene evolution, restricting refocusing to static environments [5]. Another line of research focuses on single-image refocusing, where the typical framework first deblurs the image to obtain an all-in-focus (AiF) representation, followed by reblurring. A recent representative study, RefocusGAN [45], trains a two-stage conditional GAN [35] for sequential deblurring and reblurring, enabling flexible single-image refocusing using a focus control vector. The most recent work, DC^2 [5], adopts a dual-camera setup, where the captured wide-frame and ultra-wide-frame images are first aligned, and a built-in depth sensor is leveraged to guide the image refocusing process. In this work, we follow the deblur-and-reblur paradigm and develop a disparity-aware invertible framework to overcome the limitations of inaccessible hardware (e.g., built-in depth sensors), the constraints of capturing a focus stack, and the challenge of achieving both defocus deblurring and selective reblurring.

3 User-Instructed Disparity-aware Defocus Control

Motivation. In the DP image pair, two images $\mathbf{B}_l$ and $\mathbf{B}_r$ are generated by light rays from the real scene passing through the left and right sub-apertures into two juxtaposed half pixels. The disparity $\mathbf{D}$ for a DP pair $(\mathbf{B}_l, \mathbf{B}_r)$ at location (i, j) is approximately,

$$\mathbf{D}(i,j) = f \times B/\mathbf{Z}(i,j) + \text{const} \approx f \times B/\mathbf{Z}(i,j)\ , \tag{1}$$

where f is the focal distance, B denotes the shift between two sub-apertures, and $\mathbf{Z}(i,j)$ denotes the scene depth. DP disparity is strongly correlated with the degree of defocus blur [42]. Specifically, the blur radius at location (i, j) is given by,

$$\mathbf{C}(i,j) = \mathcal{J}(f) \cdot \mathbf{D}(i,j)\ , \tag{2}$$

where $\mathcal{J}(f)$ is a function of the focus distance f, controlling the strength of the blur. $\mathbf{C}$ is known as CoC. When refocusing to objects indicated by a binary mask $\mathbf{M}$, its blur radius follows a linear relation with respect to the disparity offset from the target focal disparity. The refocused blur radius at (i, j) becomes,

$$\mathbf{C}(i,j) = \mathcal{J}(f) \cdot (\mathbf{D}(i,j) - d_{trg})\ , \tag{3}$$

where the target focal disparity d_{trg} is computed as the average disparity within the masked region:

$$d_{trg} = \frac{1}{|\Omega|} \sum_{(i',j') \in \Omega \,:\, \mathbf{M}(i',j')=1} \mathbf{D}(i',j') \,, \tag{4}$$

with $|\Omega|$ denoting the number of pixels where $\mathbf{M}(i',j') = 1$. This formulation enables spatially adaptive refocusing to user-specified regions of interest defined by the mask.

Overview. Our key idea is to learn a disparity-aware feature space that adheres to Eq. 2, enabling a blur-aware representation suitable for both deblurring and refocusing. Leveraging standard DP defocus-deblurring datasets containing DP pairs $(\mathbf{B}_l, \mathbf{B}_r)$ and their corresponding sharp images $\mathbf{I}$, we train an invertible network that jointly models deblurring and reblurring (i.e., red, orange, and green arrow in Figure 2). During training, a CoC-based feature $\mathbf{F}_{coc}$ is computed in the feature space by combining a disparity feature and a blur strength feature according to Eq. 2. This feature is used to generate spatially varying deblurring kernels $\mathbf{K}$ that restore the input DP pair $(\mathbf{B}_l, \mathbf{B}_r)$ into a sharp image $\hat{\mathbf{I}}$. Reblurring is then performed by inverting these kernels, using $\mathbf{K}^{-1}$ to degrade the sharp image $\hat{\mathbf{I}}$ back into a blurred DP pair. At test time, we first apply the deblurring process to $(\mathbf{B}_l, \mathbf{B}_r)$ to recover the sharp image $\hat{\mathbf{I}}$. To perform refocusing, the disparity feature obtained during deblurring is modulated according to Eq. 4, resulting in a target CoC feature $\mathbf{F}^{trg}_{coc}$, computed using the user-provided mask $\mathbf{M}^{trg}$ and blur level as in Eq. 2. The mask can be derived from point or box prompts by leveraging the SAM-like model [43, 62]. This target feature guides the reblurring process, transforming the sharp image $\hat{\mathbf{I}}$ into a refocused DP image. An overview of the network architecture is shown in Figure 2, where deblurring, reblurring, and refocusing operations are carried out in a shared feature space learned by an encoder $\mathcal{F}_e(\cdot,\cdot)$ and a decoder $\mathcal{F}_d(\cdot)$.

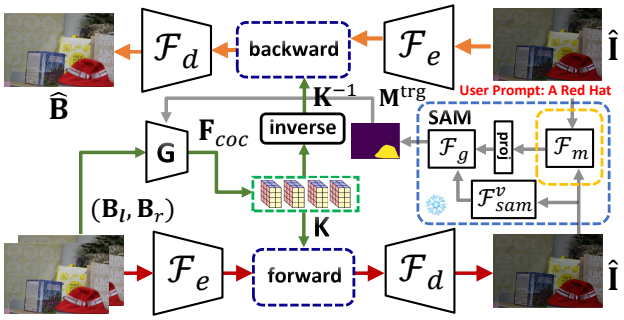

Figure 2: Illustration of overall framework. The arrow with red color and green color denotes the deblurring pathway, and the arrow with orange color and green color denotes the reblurring pathway. These two pathways are collaboratively trained with disparity-aware shared kernel learning (i.e., green arrow). The arrow with gray color indicates the user-instructed mask generation, which is only activated during inference time. The dotted box with yellow color denotes the early fusion paradigm. $\mathcal{F}_m$ can be multimodal encoder or MLLM [29–31], *proj* is the projector, and $\mathcal{F}^v_{sam}$ denotes the visual encoder of SAM.

3.1 Deblurring

Feature Encoding. We restore the DP pair $(\mathbf{B}_l, \mathbf{B}_r)$ in feature space by first embedding them into a base feature representation $\mathbf{F}_B$ using an encoder $\mathcal{F}_e(\cdot,\cdot)$. In parallel, a disparity-related feature $\mathbf{F}_G$ is extracted using a stereo matching network $\mathbf{G}(\cdot,\cdot)$ applied to the DP pair. Within $\mathbf{G}(\cdot,\cdot)$, we first use dual visual encoder for feature extraction, forming $\mathbf{F}_{l,b}$ and $\mathbf{F}_{r,b}$. For each pixel in $\mathbf{F}_{l,b}$, we search for its counterpart $\mathbf{F}_{r,b}$ within scope d horizontally, where d is defined as the disparity index. We calculate all the possible disparity using widely-used matching costs following [50], where the left and right features concatenated to form the cost volume,

$$\mathbf{C}_d(x,y) = [\mathbf{F}_{l,b}(x,y), \mathbf{F}_{r,b}(x-d,y)] \,, \tag{5}$$

where x and y are the pixel coordinates. Then, all the obtained disparity feature $\mathbf{C}_d$ are concatenated, forming $\mathcal{C} = [\mathbf{C}_1, \mathbf{C}_d, ..., \mathbf{C}_D]$, where D denotes the maximum disparity scope. Afterwards, two cascaded 3D convolutional networks are sequentially employed for aggregating the geometry information stored in the cost volumes $\mathcal{C}$, learning the approximate disparity clues in each location. Finally, a lightweight regression block is used on the top of network to compute the expected disparity feature $\mathbf{F}_G$.

While disparity or depth is fundamental to refocus control, directly manipulating the feature space spanned by $\mathbf{F}_G$ based on these geometric cues is infeasible. In other words, the lack of an interpretable

correspondence between feature dimensions and defocus blur prevents meaningful, controllable editing. Therefore, we thus further normalize the DP disparity feature and scale it as follow,

$$\mathbf{F}_D = 2 \cdot \mathrm{Tanh}\left(\mathrm{Norm}(\mathbf{F}_G)/\rho\right) - 1 \,, \tag{6}$$

where $\mathrm{Tanh}(\cdot)$ denotes the hyperbolic tangent function, $\mathrm{Norm}(\cdot)$ is the ℓ_2 normalization along with channel dimension, and ρ is a temperature parameter that sharp the distribution of the output. By this way, we condense the depth information conveyed in $\mathbf{F}_G$ into a one-channel disparity map $\mathbf{F}_D$, which serves as a direct interface for refocus control. Recognizing that the DP pair also encodes focal length–dependent blur information, we extract this information from feature $\mathbf{F}_G$ using an encoder $\mathcal{J}(\cdot)$. Following Eq. 2, the initial CoC-related feature can be rewritten as,

$$\mathbf{F}_{init} = \mathcal{J}(\mathbf{F}_G) \cdot \mathbf{F}_D \,. \tag{7}$$

Next, we apply a decoder–encoder structure to refine $\mathbf{F}_{init}$. The refined feature is projected into RGB space via a projection head $\mathcal{P}_{rgb}(\cdot)$, and projected back to the feature space through a second projection head $\mathcal{P}_{feat}(\cdot)$, with a gating residual connection [47] for numerical stability,

$$\mathbf{F}_{coc} = \mathbf{R} \odot \mathbf{F}_G + \mathcal{P}_{feat}\left(\mathcal{P}_{rgb}\left(\mathbf{F}_{init}\right)\right) \,, \tag{8}$$

where $\mathbf{R}$ denotes the learnable gate (see supplementary material for details). This decoder–encoder structure enables auxiliary supervision either through ground-truth CoC annotations or via self-supervised CoC estimation applied to the RGB-projected output $\mathcal{P}_{rgb}\left(\mathbf{F}_{init}\right)$.

Feature Restoration. By maintaining a set of N learnable blur kernels $\{\mathbf{K}_n\}_{n=1}^{N}$, we use the CoC-related feature $\mathbf{F}_{coc}$ to query spatially-varying deblurring kernels from the set, and apply the queried kernels to restore the feature $\mathbf{F}_B$. Note that we keep $\mathbf{F}_{coc}$ and $\mathbf{F}_B$ as the same spatial size. Specifically, for each pixel (i, j), a spatially-varying kernel $\mathbf{K}_{i,j}$ is constructed as a weighted sum of the kernel set,

$$\mathbf{K}_{i,j} = \sum_{n=1}^{N} \mathbf{a}_{i,j}(n) \cdot \mathbf{K}_n \,, \quad \mathbf{a}_{i,j} = \mathrm{Softmax}\left(\mathcal{P}_k\left(Avg(\{\mathbf{K}_n\}_{n=1}^{N}) \odot \mathcal{P}_{coc}(\mathbf{F}_{coc}(i,j))\right)\right) \,, \tag{9}$$

where $\mathbf{a}_{i,j}$ is a soft attention vector over the N kernels. The function $Avg(\cdot)$ computes the average of each kernel $\mathbf{K}_n$ (resulting in a scalar per kernel), forming a N-dim vector representation. This vector is element-wise multiplied $\odot$ with the CoC-projected feature $\mathcal{P}_{coc}(\mathbf{F}_{coc}(i,j))$, and passed through a linear layer $\mathcal{P}_k(\cdot)$ followed by a Softmax function to produce the attention weights. The learnable projections $\mathcal{P}_k(\cdot)$ and $\mathcal{P}_{coc}(\cdot)$ are both implemented as linear layers.

We then convolve $\mathbf{F}_B$ with queried kernels for each pixel (i, j), and refine the feature by using forward mapping of invertible block $Inv(\cdot)$, to get restored feature $\hat{\mathbf{F}}_I$,

$$\hat{\mathbf{F}}_I(i,j) = Inv(\sum_{\delta i, \delta j} \mathbf{F}_B(i+\delta i, j+\delta j) \cdot \mathbf{K}_{i,j}(\delta i, \delta j)) \,, \tag{10}$$

where $(\delta i, \delta j)$ is offset of the receptive field of the spatial kernel $\mathbf{K}_{i,j}$. Notably, the special design of architecture theoretically guarantees that Jacobian determinant $\left|\frac{\partial \hat{\mathbf{F}}_I}{\partial \mathbf{F}_B}\right|$ is non-zero, ensuring that invertibility of overall network, irrespective of the specific form of the components within $Inv(\cdot)$. The details can be referred to supplementary materials.

Feature Decoding. We decode the $\hat{\mathbf{F}}_I(i,j)$ to RGB space for calculating an all-in-focus restoration with a decoder $\mathcal{F}_d(\cdot)$ by $\hat{\mathbf{I}} = \mathcal{F}_d(\hat{\mathbf{F}}_I)$.

3.2 Reblurring

Our reblurring process is analogous to the deblurring procedure, but it employs the inverse of the queried kernels. For an all-in-focus image $\mathbf{I}$, we first obtain feature $\mathbf{F}_I$ by $\mathbf{F}_I = \mathcal{F}_e(\mathbf{F}_I, \mathbf{F}_I)$. Then, we reverse the computation performed by the invertible block $Inv(\cdot)$ by applying its corresponding inverse operation $Inv'(\cdot)$. Simultaneously, for each pixel (i, j), we calculate the inverse of the deblurring kernel $\mathbf{K}_{i,j}$ with the Fourier transform $\mathcal{F}(\cdot)$,

$$\mathbf{K}^{-1} = \mathcal{F}^{-1}(\frac{1}{\mathcal{F}(\mathbf{K})}), \tag{11}$$

outputting an inverse kernel $\mathbf{K}_{i,j}^{-1}$ for reblurring, where $\mathcal{F}^{-1}(\cdot)$ denotes the inverse Fourier transform. The reblurred feature is calculated as,

$$\hat{\mathbf{F}}_B(i,j) = \sum_{\delta i, \delta j} Inv'(\mathbf{F}_I)(i+\delta i, j+\delta j) \cdot \mathbf{K}_{i,j}^{-1}(\delta i, \delta j) \ . \tag{12}$$

The resulting reblurred feature map $\hat{\mathbf{F}}_B$ is then decoded into an image $\hat{\mathbf{B}}$ using a decoder $\mathcal{F}_d(\cdot)$. The goal is to synthesize the center view of the DP pair $\mathbf{B} = \frac{1}{2}\mathbf{B}_l + \frac{1}{2}\mathbf{B}_r$.

3.3 Refocusing

During testing, to perform refocusing on a DP pair $(\mathbf{B}_l, \mathbf{B}_r)$ using a user-provided focus mask $\mathbf{M}^{trg}$, we first restore an all-in-focus image $\hat{\mathbf{I}}$ following the forward mapping of our invertible network (Eq. 10). We then reblur the image by modulating the original disparity feature $\mathbf{F}_D$ by

$$\mathbf{F}_D^{trg}(i,j) = \mathbf{F}_D(i,j) - \frac{1}{|\Omega|} \sum_{(i',j') \in \Omega \,:\, \mathbf{M}^{trg}(i',j')=1} \mathbf{F}_D(i',j') \ , \tag{13}$$

This operation shifts the disparity feature such that the selected region appears in focus, effectively implementing a disparity-aware refocusing strategy, as outlined in Eq. 4. We then calculate an updated CoC feature $\mathbf{F}_{coc}^{trg}$ based on Eq. 8, and repeat the kernel querying process defined in Eq. 9. These kernels are applied in the reblurring process according to Eq. 12, ultimately producing the refocused feature to decode the target refocused image, denoted as $\mathbf{B}^{trg}$.

Training-free Instructed SAM. Given a user instruction $\mathcal{T}$ specifying the region of interest, the instructed Segment Anything Model (SAM) generates a target mask $\mathbf{M}^{trg}$ semantically aligned with the prompt. A key challenge lies in obtaining a high-quality segmentation mask to effectively guide the refocusing process. While end-to-end joint training of the instructed SAM with our deblur-and-reblur framework is a natural solution, it suffers from two critical drawbacks: (1) the segmentation mask, when suboptimal in early training stages, dynamically affects the refocusing network and may produce erroneous gradients, impeding optimal convergence; and (2) fine-tuning SAM from scratch with deblur-and-reblur framework entails significant computational overhead, making the overall process resource-intensive. To circumvent these issues, we first forgo the joint training tactics. Instead, we first training our deblur-and-reblur network using a mask-free manner, which means that it does not require masks during training. Mask are only provided at test time, where the user inputs a binary mask to specify the region of interest for refocusing. Therefore, the degradation on mask would not affect the dynamics and robustness of our model training. Given the inaccessibility of clear reference images during testing, we adopt an early fusion strategy to deeply exploit the rich semantic cues conveyed by both the user prompt and the input image, enabling more precise mask generation. This approach demonstrates superior performance compared to the conventional late fusion paradigm [62]. Specifically, it takes the restored image $\hat{\mathbf{I}}$ that generated via forward mapping in Eq. 10, prompt $\mathcal{T}$, and a random initialized $[cls]$ query as input, and using multimodal transformer-based encoder $\mathcal{F}_m(\cdot)$ to enhance the interaction between modalities,

$$\mathbf{q}_{cls} = \mathcal{F}_m(\hat{\mathbf{I}}, \mathcal{T}, \mathbf{q}_{cls}), \tag{14}$$

the output $[cls]$ query $\mathbf{q}_{cls}$ encodes the compact multimodal clues, and then fed into the mask decoder together with the SAM-extracted image features $\mathbf{F}_{sam}$, enabling accurate spatial alignment towards robust mask generation,

$$\mathbf{M}_{trg} = \mathcal{F}_g(\mathbf{q}_{cls}, \mathbf{F}_{sam}). \tag{15}$$

Furthermore, we localize the region of interest using structured, template-based prompts that combine number, object color, category, and spatial position, such as "a brown dog on the chair", to enhance spatial grounding precision. For those still imperfect masks, we further apply post-processing techniques such as dilation or erosion operations for refinement. The detailed implementation of $\mathcal{F}_m(\cdot)$ and $\mathcal{F}_g(\cdot)$ can be referred in supplementary material.

3.4 Loss

Our network is trained with a combination of objectives to supervise deblurring, reblurring, and CoC estimation. Specifically, we use a deblurring loss $\mathcal{L}_{deb}(\mathbf{I}, \hat{\mathbf{I}})$ and a reblurring loss $\mathcal{L}_{reb}(\mathbf{B}, \hat{\mathbf{B}})$ based

on multi-scale setting following [57], a MSE-based CoC regression loss $\mathcal{L}_{coc}(\mathbf{C}, \mathcal{P}_{rgb}(\mathbf{F}_{int}))$, and a MSE-based CoC gradient loss $\mathcal{L}^{\nabla}_{coc}(\mathbf{C}, \mathcal{P}_{rgb}(\mathbf{F}_{int}))$ to enhance edge precision. The total loss is,

$$\mathcal{L}_{total} = \mathcal{L}_{deb}(\mathbf{I}, \hat{\mathbf{I}}) + \mathcal{L}_{reb}(\mathbf{B}, \hat{\mathbf{B}}) + \lambda_{coc} \cdot \mathcal{L}_{coc}(\mathbf{C}, \mathcal{P}_{rgb}(\mathbf{F}_{init})) + \lambda_{grad} \cdot \mathcal{L}^{\nabla}_{coc}(\mathbf{C}, \mathcal{P}_{rgb}(\mathbf{F}_{init})) \,, \tag{16}$$

where each λ controls the relative weight of the corresponding loss term.

4 Experiment

Dataset. For defocus deblurring task, we evaluate our method on widely-used DPD-blur [2] dataset and recent DP5K [27] dataset. For refocusing task, we use three datasets for evaluation, DP5K dataset, DPD-disp dataset, and our self-collected DP dataset. Our model is trained using the training splits of DPD-blur and DP5K. Refer to our supplementary material for detailed dataset configurations and evaluation protocols.

Evaluation Metrics. We evaluate the model with using standard metrics, i.e., peak signal-to noise ratio (PSNR) [19], structural similarity (SSIM) [55], relative error (RMSE rel) [37], mean absolute error (MAE), and learned perceptual image patch similarity (LPIPS) [61].

Implementation Detail. For training, we use the AdamW optimizer [24] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 3×10^{-4}, and a weight decay of 10^{-6}. A cosine annealing learning rate [32] scheduler with warmup is employed, where the cycle steps, warmup steps, and minimum learning rate are set to 200, 100, and 6×10^{-5}, respectively. For the DPD-blur dataset, the model is trained for 40k iterations with a batch size of 4. For the DP5K dataset, we train the model for 64k iterations with a batch size of 6. We elaborate our network architecture and more hyper-parameter setting in supplementary material.

4.1 Experimental Results

As mentioned Sec. 3, image refocusing task is decomposed into two key stages: (1) defocus deblurring, and then (2) refocusing. In this section, we conduct extensive experiments to evaluate the effectiveness of each part in two stages. First, we perform quantitative and qualitative evaluations of both deblurring and refocusing, comparing their performance with several strong baselines. Then, we provide a detailed component-wise analysis of our refocusing framework.

4.1.1 Quantitative and Qualitative Results on Image Refocus

Setting. Image refocus takes a user-specified mask and an all-in-focus image as input, the goal is to generate a new image where the masked region remains sharp while other areas exhibit controlled defocus blur. Therefore, obtaining an all-in-focus image from blurred image is the first step to achieve refocusing, and its restored quality greatly determine refocus performance.

Table 1: Deblurring quantitative comparisons on DPD-blur dataset [2] with several SOTA baselines.

Method	Computional Cost		Quantitative Metrics			
	Params (M)	Flops (G)	PSNR↑	SSIM↑	$\text{MAE}_{(10^{-1})}$ ↓	$\text{MSE\_rel}_{(10^{-1})}$ ↓
single-image defocus deblurring						
EBDB	-	-	23.45	0.683	0.49	0.67
DMENet	26.71	4787	23.41	0.714	0.51	0.67
RDPD	24.28	901	25.39	0.772	0.40	0.53
IFAN	10.48	794	25.99	0.804	0.37	0.50
BAMBNet	4.50	1804	26.40	0.821	0.36	0.47
DeepRFT	9.60	3682	25.71	0.801	0.37	0.51
Restormer	26.13	4458	26.66	**0.833**	0.35	0.46
dual-pixel defocus deblurring						
DPDNet	31.03	3150	25.13	0.786	0.41	0.55
DDDNet	6.40	1661	25.36	0.768	0.41	0.54
K3DN	5.00	1033	26.84	0.829	0.35	**0.46**
Ours	**3.12**	**495**	**26.89**	0.829	**0.33**	0.47

Figure 3: Qualitative comparison on the DP5K-test dataset [3]. We present the blurry image in the first column, and boxed regions with red color are zoomed results.

Table 2: Comparison of refocusing performance on DP5K-test. † denotes the model version adapted for refocusing

Methods	PSNR↑	SSIM↑	$\text{MAE}_{(10^{-1})}\downarrow$	LPIPS↓
Omni-Kernel†	25.32	0.816	0.47	23.5
K3DN†	30.76	0.943	0.30	14.4
Ours	31.21	0.953	0.33	13.8

Table 3: Comparison of refocusing performance on DPD-disp for OOD evaluation. † denotes the model version adapted for refocusing.

Methods	PSNR↑	SSIM↑	$\text{MAE}_{(10^{-1})}\downarrow$	LPIPS↓
Omni-Kernel†	18.75	0.692	0.79	31.4
K3DN†	21.72	0.765	0.67	29.7
Ours	21.93	0.772	0.56	25.1

Methods. To evaluate the deblurring ability of our framework, we compared with two primary categories of defocus deblurring approaches. (1) Single-image defocus deblurring: KPAC [49], RDPD [4], DRBNet [44], DeepRFT [34], IFAN [26], RAMBNet [28], and Restormer [59]. (2) Dual-pixel defocus deblurring: DPDNet [2], DDDNet [38], K3DN [57]. For refocusing, BokehMe [40] and Dual-Camera [5] are most closely related to our approach. However, they neither release training code nor pretrained models. Thus, we manually modify two SOTA methods on deblurring task, K3DN and Omni-Kernel [11], to enable their refocusing functionality. Refer to our supplementary materials for details.

Evaluation. The deblurring comparisons with state-of-the-art methods are given in Table 1 and Table 5. The result reveals that we achieve superior performance, while having smallest computation cost (495 GFlops). Figure 3 showcases the deblurring results of two cases against several state-of-the-art baselines. Our method is able to restore severely blurred regions, especially textual region, compared to existing models. For refocusing, we report the qualitative comparison in Table 2 and

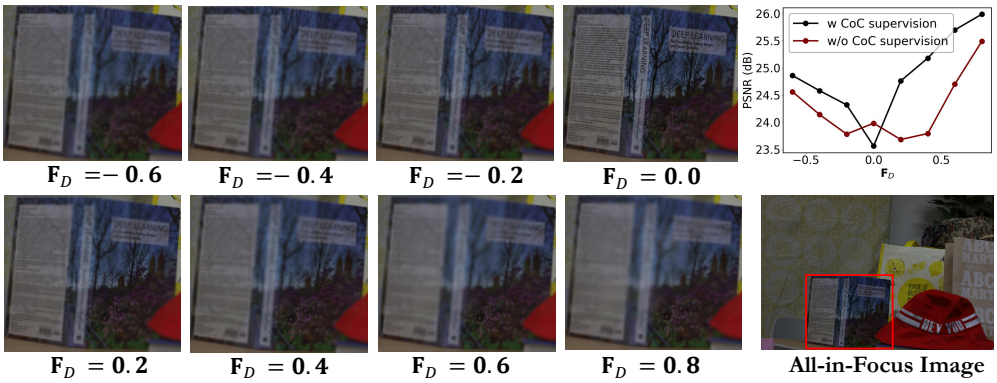

Figure 4: Visualization of blur effect by setting $\{\mathbf{F}_D(i,j)|(i,j) \in \Omega_{book}\} = 0$ when performing reblurring using Eq. 12. We take the all-in-focus and a user-specified mask (a mask referring to *a unfolding book* in this case) as input, and observe the blur variation with different $\mathbf{F}_D$. The top-right figure shows the overall PSNR variation trend with different $\mathbf{F}_D$.

Table 4: The effect of joint deblurring-reblurring training on performance.

Model Variants	PSNR↑	SSIM↑	MAE$_{(10^{-1})}$ ↓
Deblurring Performance			
w $\mathcal{L}_{reb}$ + $\mathcal{L}_{deb}$	26.89	0.829	0.35
w $\mathcal{L}_{deb}$	26.80	0.823	0.35
Reblurring Performance			
w $\mathcal{L}_{reb}$ + $\mathcal{L}_{deb}$	29.18	0.875	0.22
w $\mathcal{L}_{reb}$	29.06	0.863	0.28

Table 5: Comparison of deblurring performance on DP5K-test. The best is bold with black.

Methods	PSNR↑	SSIM↑	RMSE_rel$_{(10^{-1})}$ ↓	MAE$_{(10^{-1})}$ ↓
DDDNet	25.59	0.777	0.525	0.58
IFAN	28.23	0.875	0.387	0.44
DeepRFT	30.24	0.915	0.307	0.37
BAMBNet	30.54	0.917	0.297	0.34
Restormer	30.71	0.922	0.291	0.33
K3DN	**30.82**	0.923	0.291	0.32
Ours	30.72	**0.926**	**0.280**	**0.30**

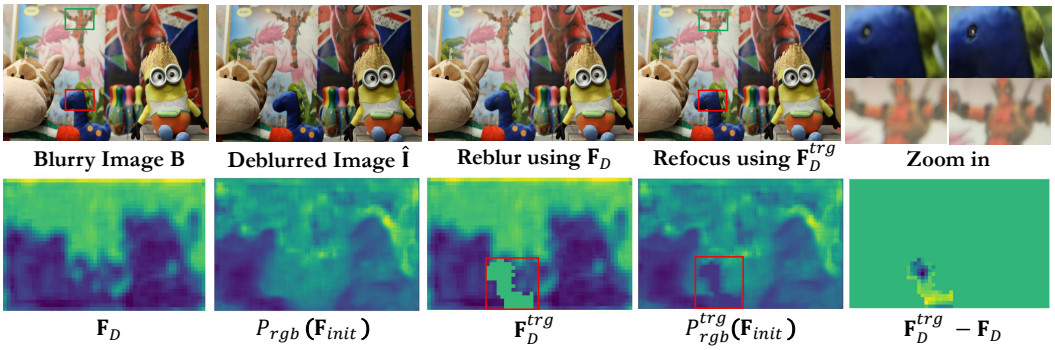

Figure 5: Visualization on immediate process in refocusing. Notably, $\mathbf{F}_D$ and $\mathbf{F}_D^{trg}$ denotes the disparity map used in reblurring and refocusing, and $P_{rgb}(\mathbf{F}_{init})$ and $P_{rgb}^{trg}(\mathbf{F}_{init})$ are predicted CoC map from their corresponding disparity map (Eq. 8), which both exhibits satisfactory consistency.

Table 3. We also show two cases from DPD-disp dataset to evaluate the out-of-distribution (OOD) generalization capability of our model, as shown in Figure 6.

4.1.2 Component-wise Analysis

Effect of Joint Deblurring and Reblurring Training. Joint deblurring-reblurring training enables robust blur kernel estimation and exhibits more flexible refocusing. As can be seen in the Table 4, model with both $\mathcal{L}_{reb}$ and $\mathcal{L}_{deb}$ supervision achieves substantial advantages compared to that only with $\mathcal{L}_{reb}$ or $\mathcal{L}_{deb}$. This observation indicates that one process not only does not disturb another branch but also effectively help to optimize the learned blur kernel, thereby enhancing the representation capability learned by our network. Using the shared kernel in deblurring process, user-specified refocusing process can be simply achieved by using the with inverse kernel obtained in Eq. 11, which exhibits satisfactory flexibility and viability.

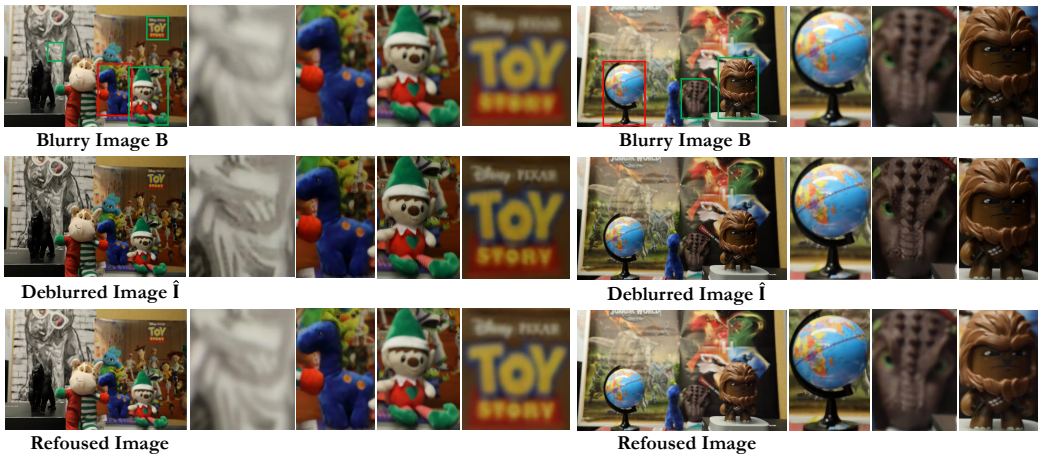

Figure 6: Visualization of refocusing on DPD-disp dataset. The bounding box is highlighted for clear observation, and region with red box refers to the user-specified region. The images in the first row refers to blurred image $\mathbf{B}$ (unified by $\mathbf{B} = \frac{1}{2}(\mathbf{B}_l + \mathbf{B}_r)$), images in the second row are deblurred image recovered by our model (Eq. 10), and the last row showcases the refocused images.

Visualization of Kernel Selection Behavior. We provide the visualization map of retrieved kernel index in kernel pool of different pixel locations to reveal deeper understanding. We first talk from the learned disparity feature $\mathbf{F}_G$, because the retrieved kernel intrinsically rely on the disparity feature. As shown in the second row in Figure 7, we use K-means clustering [23] to group the similar disparity feature with different cluster numbers, $c = 6, 10$ and then visualize the clustered group index. We can observe that our learned $\mathbf{F}_G$ conforms with the actual depth or blur cues in $\mathbf{B}$. Done well with $\mathbf{F}_G$, we verify the reliability of using $\mathbf{F}_G$ for kernel selection. As shown in *kernel selection* of Figure 7, the distribution of retrieved kernel index via Eq. 9 tends to approximate the blur distribution of the original blurred image. This observation further demonstrate the feasibility of our design: when perform refocusing, specifying the mask of focus would specify the disparity thereby further determining the kernel selection.

Necessity of Disparity Alignment towards Refocusing Based on Eq. 2, we conclude that CoC map and disparity map provide the approximate indication clues for blur kernel estimation. Particularly, a pixel (i, j) with $\mathbf{C}(i, j) = \mathbf{D}(i, j) = 0$ indicates to be well-focused. In practice, we empirically observe that training network only with disparity feature $\mathbf{F}_{init}$ as blur kernel guidance enables faster convergence while achieving better deblurring and reblurring performance than that only using CoC feature. However, indirect supervision of $\mathbf{F}_{init}$ through joint training on deblurring and reblurring tasks may not adequately capture explicit blur cues. Ideally, regions with $\mathbf{F}_D = 0$ should exhibit perfect focus. As illustrated in Figure 4, in the absence of CoC map supervision, the disparity value corresponds to minimum PSNR happen to shift. With the aid of CoC supervision, the book is observed to be visually well-focused exactly. $\{\mathbf{F}_D(i, j)|(i, j) \in \Omega_{book}\} = 0$ corresponds to minimum PSNR 23.6dB compared with ground-truth blurred image. It indicates that an accurate correspondence between disparity and blur kernel or defocus blur can be explicitly calibrated with the supervision of CoC map $\mathbf{C}$. Moreover, we visualize the correspondence between $\mathbf{F}_D$ and predicted CoC map, known as $\mathcal{P}_{rgb}(\mathbf{F}_{init})$, in Figure 5.

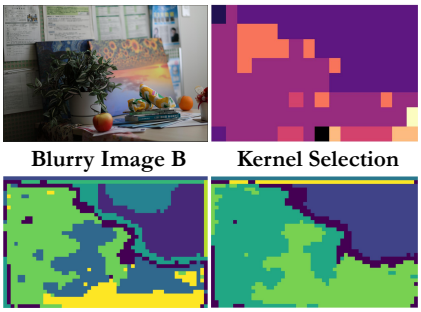

Figure 7: Illustration of kernel selection and clustered disparity feature $\mathbf{F}_G$.

Ablation of Components in Invertible Mapping. To investigate the impact of different invertible component designs towards the robustness of the visual representation, we tailor several variants. As shown in Table 6, model $v1$ denotes adopting the 1×1 convolution for channel shuffle [13], model $v2$ adopts the residual invertible network similar to [18], and model $v3$ incorporates affine transformation operations for more stable invertibility. Among them, model $v3$ achieves the best performance. It demonstrates that robust invertible mapping enables more effective kernel learning for refocusing. More details of these variants are provided in the supplementary material.

Table 6: The ablation of joint training on both deblurring and reblurring performance.

Variants	PSNR↑	SSIM↑	MAE$_{(10^{-1})}$ ↓
$v1$ [13]	25.56	0.795	0.52
$v2$ [13]	26.21	0.810	0.35
$v3$ [18]	26.89	0.829	0.31

5 Conclusion and Future Work

Without bells and whistles, we re-frame the task of image refocus as two interconnected subtasks—joint reblurring and deblurring using the shared kernel, and the disparity feature extracted from dual-pixel (DP) image pairs is leveraged as an indicator for blur kernel estimation. Specifically, our work experimentally explores several key aspects: (1) We investigate the potential benefits of using DP image pairs to achieve effective defocus control. (2) We mark the first attempt to demonstrate the feasibility of using joint deblurring and reblurring as a proxy of image refocusing, which is a non-trivial endeavor. (3) We analyze the behavior of kernel selection and test its sensitivity to adjustment of disparity. (4) We incorporate an early-fusion SAM into our deblur-and-reblur framework to accurately identify the region of interest, enabling robust mask generation and flexible defocus control. In future work, we aim to adapt our method to the diffusion framework to further explore its practical applications and commercial potential.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under grant No. 62302045, and the Beijing Institute of Technology Special-Zone.

References

[1] Abdullah Abuolaim, Mahmoud Afifi, and Michael S Brown. Improving single-image defocus deblurring: How dual-pixel images help through multi-task learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1231–1239, 2022. 3

[2] Abdullah Abuolaim and Michael S. Brown. Defocus deblurring using dual-pixel data. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part X*, volume 12355 of *Lecture Notes in Computer Science*, pages 111–126. Springer, 2020. 2, 7, 8

[3] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *European Conference on Computer Vision*, pages 111–126. Springer, 2020. 8

[4] Abdullah Abuolaim, Mauricio Delbracio, Damien Kelly, Michael S. Brown, and Peyman Milanfar. Learning to reduce defocus blur by realistically modeling dual-pixel data. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2269–2278. IEEE, 2021. 8

[5] Hadi Alzayer, Abdullah Abuolaim, Leung Chun Chan, Yang Yang, Ying Chen Lou, Jia-Bin Huang, and Abhishek Kar. Dc^2: Dual-camera defocus control by learning to refocus. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 21488–21497. IEEE, 2023. 2, 3, 8

[6] Jonathan T. Barron, Andrew Adams, Yichang Shih, and Carlos Hernández. Fast bilateral-space stereo for synthetic defocus. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4466–4474. IEEE Computer Society, 2015. 2

[7] Shiveta Bhat and Deepika Koundal. Multi-focus image fusion techniques: a survey. *Artif. Intell. Rev.*, 54(8):5735–5787, 2021. 3

[8] Myungsub Choi, Hana Lee, and Hyong-Euk Lee. Exploring positional characteristics of dual-pixel data for camera autofocus. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 13112–13122. IEEE, 2023. 2

[9] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics, 2020. 21

[10] Robert Correll. *Digital SLR photography all-in-one for dummies*. John Wiley & Sons, 2020. 3

[11] Yuning Cui, Wenqi Ren, and Alois Knoll. Omni-kernel network for image restoration. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 1426–1434. AAAI Press, 2024. 8, 22

[12] Harold Davis. *Creative Composition: Digital Photography Tips and Techniques*. John Wiley and Sons, 2011. 3

[13] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 10

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 21

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 24

[16] Kashish Galhotra, Azhar Ashraf, and Hiral Singh. Image refocusing–using unsupervised learning and depth estimation. In *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, pages 1–6. IEEE, 2024. 3

[17] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T Barron. Learning single camera depth estimation using dual-pixels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7628–7637, 2019. 3

[18] Aidan N. Gomez, Mengye Ren, Raquel Urtasun, and Roger B. Grosse. The reversible residual network: Backpropagation without storing activations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2214–2224, 2017. 10

[19] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing*. Prentice Hall, 2002. 7

[20] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[21] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 23

[22] Jinbeum Jang, Yoonjong Yoo, Jongheon Kim, and Joonki Paik. Sensor-based auto-focusing system using multi-scale feature extraction and phase correlation matching. *Sensors*, 15(3):5747–5762, 2015. 2

[23] Xin Jin and Jiawei Han. *K-Means Clustering*, pages 563–564. Springer US, Boston, MA, 2010. 10

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 7

[25] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10236–10245, 2018. 21, 22

[26] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2034–2042, 2021. 3, 8

[27] Feiran Li, Heng Guo, Hiroaki Santo, Fumio Okura, and Yasuyuki Matsushita. Learning to synthesize photorealistic dual-pixel images from RGBD frames. In *IEEE International Conference on Computational Photography, ICCP 2023, Madison, WI, USA, July 28-30, 2023*, pages 1–11. IEEE, 2023. 7

[28] Pengwei Liang, Junjun Jiang, Xianming Liu, and Jiayi Ma. Bambnet: A blur-aware multi-branch network for dual-pixel defocus deblurring. *IEEE/CAA Journal of Automatica Sinica*, 9(5):878–892, 2022. 8

[29] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 4

[30] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 4

[32] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017. 7

[33] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*, pages 1150–1157. IEEE Computer Society, 1999. 22

[34] Xintian Mao, Yiming Liu, Wei Shen, Qingli Li, and Yan Wang. Deep residual fourier transformation for single image deblurring. *CoRR*, abs/2111.11745, 2021. 8, 21

[35] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 3

[36] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenopic camera. *Technical Report CTSR 2005-02*, CTSR, 01 2005. 3

[37] Liyuan Pan, Shah Chowdhury, Richard Hartley, Miaomiao Liu, Hongguang Zhang, and Hongdong Li. Dual pixel exploration: Simultaneous depth estimation and image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2021. 7, 8

[38] Liyuan Pan, Shah Chowdhury, Richard Hartley, Miaomiao Liu, Hongguang Zhang, and Hongdong Li. Dual pixel exploration: Simultaneous depth estimation and image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4340–4349. Computer Vision Foundation / IEEE, 2021. 8

[39] Liyuan Pan, Yuchao Dai, Miaomiao Liu, Fatih Porikli, and Quan Pan. Joint stereo video deblurring, scene flow estimation and moving object segmentation. *IEEE Trans. Image Process.*, 29:1748–1761, 2020. 2

[40] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. Bokehme: When neural rendering meets classical rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16262–16271. IEEE, 2022. 3, 8

[41] Dominique Piché-Meunier, Yannick Hold-Geoffroy, Jianming Zhang, and Jean-François Lalonde. Lens parameter estimation for realistic depth of field modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 499–508, 2023. 3

[42] Abhijith Punnappurath, Abdullah Abuolaim, Mahmoud Afifi, and Michael S Brown. Modeling defocus-disparity in dual-pixel sensors. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020. 3

[43] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 2, 4

[44] Lingyan Ruan, Bin Chen, Jizhou Li, and Miu-Ling Lam. Learning to deblur using light field generated and real defocus images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16283–16292. IEEE, 2022. 8

[45] Parikshit Sakurikar, Ishit Mehta, Vineeth N. Balasubramanian, and P. J. Narayanan. Refocusgan: Scene refocusing using a single image. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, volume 11208 of *Lecture Notes in Computer Science*, pages 519–535. Springer, 2018. 2, 3

[46] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4937–4946. Computer Vision Foundation / IEEE, 2020. 22

[47] Pedro H. P. Savarese. Learning identity mappings with residual gates. *CoRR*, abs/1611.01260, 2016. 5

[48] Mashhour Solh. Real-time focal stack compositing for handheld mobile cameras. In Charles A. Bouman and Ken D. Sauer, editors, *Computational Imaging XII, part of the IS&T-SPIE Electronic Imaging Symposium, San Francisco, California, USA, February 2, 2014, Proceedings*, volume 9020 of *SPIE Proceedings*, page 90200Z. IS&T/SPIE, 2014. 3

[49] Hyeongseok Son, Junyong Lee, Sunghyun Cho, and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2622–2630. IEEE, 2021. 8

[50] Mahmoud Tahmasebi, Saif Huq, Kevin Meehan, and Marion McAfee. Dcvsmnet: Double cost volume stereo matching network. *Neurocomputing*, 618:129002, 2025. 4

[51] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 05 2019. 21

[52] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Trans. Graph.*, 37(4):64, 2018. 2, 3

[53] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *CoRR*, abs/2208.10442, 2022. 21

[54] Yingqian Wang, Jungang Yang, Yulan Guo, Chao Xiao, and Wei An. Selective light field refocusing for camera arrays using bokeh rendering and superresolution. *IEEE Signal Processing Letters*, 26:204–208, 2019. 3

[55] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 7

[56] Shumian Xin, Neal Wadhwa, Tianfan Xue, Jonathan T Barron, Pratul P Srinivasan, Jiawen Chen, Ioannis Gkioulekas, and Rahul Garg. Defocus map estimation and deblurring from a single dual-pixel image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2228–2238, 2021. 2

[57] Yan Yang, Liyuan Pan, Liu Liu, and Miaomiao Liu. K3DN: disparity-aware kernel estimation for dual-pixel defocus deblurring. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 13263–13272. IEEE, 2023. 2, 7, 8

[58] Yan Yang, Liyuan Pan, Liu Liu, and Miaomiao Liu. K3dn: Disparity-aware kernel estimation for dual-pixel defocus deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13263–13272, June 2023. 3, 8

[59] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5718–5729. IEEE, 2022. 8

[60] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022. 8

[61] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 7

[62] Yuxuan Zhang, Tianheng Cheng, Rui Hu, Lei Liu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. Evf-sam: Early vision-language fusion for text-prompted segment anything model. 2024. 2, 4, 6, 21

[63] Yi Zhang, Qixue Yang, Damon M. Chandler, and Xuanqin Mou. Reference-based multi-stage progressive restoration for multi-degraded images. *IEEE Trans. Image Process.*, 33:4982–4997, 2024. 21

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper has a clear abstract and an introduction section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations in the future work section included in the supplementary material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This work does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We state all experimental details in Experiment section in the main paper and Supplementary Material A. We state which datasets we used and provide references.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: If the paper is accepted, the code and data will be released.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We state all experimental details in Experiment section in the main paper and Supplementary Material A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Abstract of Appendix

This appendix provides more implementation details (Appendix A), details of curated DP image dataset (Appendix B), adaptation for refocusing functionality (Appendix C), comparison with existing methods (Appendix D), and more visualization results and analysis (Appendix E).

A More Implementation Details

Network Architecture. Our method adopts a 4-stage UNet-style architecture for progressive feature refinement. As mathematically formulated in Eq. 9 and Eq. 10 of main paper, each stage has two cascaded parameter-independent modules: (i) disparity-aware feature convolution, followed by (ii) invertible block $Inv(\cdot)$ and $Inv^{-1}(\cdot)$. For (i), we use the disparity feature $\mathbf{F}_{coc}(i,j)$ to retrieve a kernel $\mathbf{K}_{i,j}$ from kernel pool $\{\mathbf{K}_n\}_{n=1}^{N}$, and then a point-wise convolution [51] with $\mathbf{K}_{i,j}$ or $\mathbf{K}_{i,j}^{-1}$ is employed for feature refinement to obtain $\mathbf{U}^l$ or $\mathbf{U}^{l+1}$, and the architecture of (ii) is elaborated in Table 7. Notably, we parameterize learnable matrix $\mathbf{W}$ directly in its LU decomposition [25] for efficiency, i.e., $\mathbf{W} = \mathbf{PL}(\mathbf{Q} + \text{diag}(\mathbf{s}))$, where $\mathbf{P}$ is a permutation matrix, $\mathbf{L}$ is a lower triangular matrix with ones on the diagonal, $\mathbf{Q}$ is upper triangular matrix with zeros on the diagonal, and $\mathbf{s}$ is a vector.

Towards $\mathcal{F}_m(\cdot)$ and $\mathcal{F}_g(\cdot)$ in instructed SAM, we primarily adopt BEIT-3 [53] as $\mathcal{F}_m(\cdot)$, which implements a multi-way Transformer architecture. The input text is tokenized using the XLM-RobertaTokenizer [9]. Within each encoder block, image and text tokens are fused through cross-modal attention mechanisms and subsequently processed by separate Feed-Forward Networks (FFNs). Following the ViT paradigm [14], we extract the compact multimodal representation by taking the output of the $[cls]$ token. Specifically, the original prompt encoder generates sparse embeddings of shape $B \times P \times D$, where B denotes the batch size, P is the number of input points or boxes, and D represents the embedding dimension. Following [62], we replace these spatial prompts with the projected multimodal embeddings from the multimodal encoder $\mathcal{F}_m(\cdot)$. If there are point or box inputs, these are concatenated with an empty sparse embedding tensor and perform alignment with the image feature extracted from $\mathcal{F}_{sam}^{v}(\cdot)$, subsequently fed into the mask decoder.

Table 7: Illustration of invertible block operations $Inv(\cdot)$ and $Inv^{-1}(\cdot)$ in our method. In the forward mapping, the input and output to each block are denoted as $\mathbf{U}^l$ and $\mathbf{U}^{l+1}$. During the backward mapping, only *Plus* ($+$) and *Multiply* ($\odot$) needs to be inverted. ϕ_1, ϕ_2, ϕ_3 and ϕ_4 do not need to be inverted, which can be any neural networks.

#	Forward Operation $Inv(\cdot)$	Backward Operation $Inv^{-1}(\cdot)$	Specification
R0	$\tilde{\mathbf{U}}^l = \mathbf{PL}(\mathbf{Q} + \text{diag}(\mathbf{s}))\mathbf{U}^l$	$\tilde{\mathbf{U}}^{l+1} = (\mathbf{Q} + \text{diag}(\mathbf{s}))^{-1}\mathbf{P}^{-1}\mathbf{L}^{-1}\mathbf{U}^{l+1}$	$\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))$ denotes the matrix by LU decomposition [25], and $\mathbf{P}$ is a permutation matrix.
R1	$\mathbf{U}_a^l, \mathbf{U}_b^l = \text{Split}(\tilde{\mathbf{U}}^l)$	$\mathbf{U}_a^{l+1}, \mathbf{U}_b^{l+1} = \text{Split}(\tilde{\mathbf{U}}^{l+1})$	$\text{Split}(\cdot)$ denotes the channel-wise split.
R2	$\mathbf{U}_a^{l+1} = \mathbf{U}_a^l \odot \exp(\phi_1(\mathbf{U}_b^l)) + \phi_2(\mathbf{U}_b^l)$	$\mathbf{U}_b^l = (\mathbf{U}_b^{l+1} - \phi_4(\mathbf{U}_a^{l+1}))/\exp(\phi_3(\mathbf{U}_a^{l+1}))$	ϕ_1, ϕ_2, ϕ_3 and ϕ_4 can be any neural networks. $\odot$ is the multiply operation.
R3	$\mathbf{U}_b^{l+1} = \mathbf{U}_b^l \odot \exp(\phi_3(\mathbf{U}_a^{l+1})) + \phi_4(\mathbf{U}_a^{l+1})$	$\mathbf{U}_a^l = (\mathbf{U}_a^{l+1} - \phi_2(\mathbf{U}_b^l))/\exp(\phi_1(\mathbf{U}_b^l))$	
R4	$\mathbf{U}^{l+1} = \text{Concat}(\mathbf{U}_a^{l+1}, \mathbf{U}_b^{l+1})$	$\mathbf{U}^l = \text{Concat}(\mathbf{U}_a^l, \mathbf{U}_b^l)$	$\text{Concat}(\cdot)$ is the channel-wise concatenation.

As mentioned in Table 6 of main paper, we adopt two variants $v1$ and $v2$ to investigate the impact of different invertible blocks towards visual representation learning. Their architectures are illustrated in Table 8 and Table 9, respectively.

Table 8: The invertible block of variant $v1$.

#	Forward Operation	Backward Operation	Specification
R0	$\mathbf{U}^{l+1} = \mathbf{PL}(\mathbf{Q} + \text{diag}(\mathbf{s}))\mathbf{U}^l$	$\mathbf{U}^l = (\mathbf{Q} + \text{diag}(\mathbf{s}))^{-1}\mathbf{L}^{-1}\mathbf{P}^{-1}\mathbf{U}^{l+1}$	$\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))$ denotes the matrix by LU decomposition [25], and $\mathbf{P}$ is a permutation matrix.

Hyper-parameter Setting. Our $\mathcal{L}_{deb}$ and $\mathcal{L}_{reb}$ both uses a combination of Multi-Scale Charbonnier loss $\mathcal{L}_{char}$ [63], Multi-Scale Edge loss $\mathcal{L}_{edge}$ [63], and Multi-Scale Frequency loss $\mathcal{L}_{freq}$ [34], i. e., $\mathcal{L}_{reb} = \mathcal{L}_{deb} = \mathcal{L}_{char} + \lambda_1\mathcal{L}_{edge} + \lambda_2\mathcal{L}_{freq}$. We set $\lambda_1 = 5 \times 10^{-2}$ and $\lambda_2 = 1 \times 10^{-2}$. Regarding

Table 9: The invertible block of variant $v2$.

#	Forward Operation	Backward Operation	Specification
R0	$\tilde{\mathbf{U}}^l = \mathbf{PL}(\mathbf{Q} + \text{diag}(\mathbf{s}))\mathbf{U}^l$	$\tilde{\mathbf{U}}^{l+1} = (\mathbf{Q} + \text{diag}(\mathbf{s}))^{-1}\mathbf{L}^{-1}\mathbf{P}^{-1}\mathbf{U}^{l+1}$	$\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))$ denotes the matrix by LU decomposition [25], and $\mathbf{P}$ is a permutation matrix.
R1	$\mathbf{U}_a^l, \mathbf{U}_b^l = \text{Split}(\tilde{\mathbf{U}}^l)$	$\mathbf{U}_a^{l+1}, \mathbf{U}_b^{l+1} = \text{Split}(\tilde{\mathbf{U}}^{l+1})$	$\text{Split}(\cdot)$ denotes the channel-wise split.
R2	$\mathbf{U}_a^{l+1} = \mathbf{U}_a^l + \phi_1(\mathbf{U}_b^l)$	$\mathbf{U}_b^l = \mathbf{U}_b^{l+1} - \phi_3(\mathbf{U}_a^{l+1})$	ϕ_1, ϕ_2, ϕ_3 and ϕ_4 can be any neural networks.
R3	$\mathbf{U}_b^{l+1} = \mathbf{U}_b^l + \phi_2(\mathbf{U}_a^{l+1})$	$\mathbf{U}_a^l = \mathbf{U}_a^{l+1} - \phi_4(\mathbf{U}_b^l)$	
R4	$\mathbf{U}^{l+1} = \text{Concat}(\mathbf{U}_a^{l+1}, \mathbf{U}_b^{l+1})$	$\mathbf{U}^l = \text{Concat}(\mathbf{U}_a^l, \mathbf{U}_b^l)$	$\text{Concat}(\cdot)$ is the channel-wise concatenation.

other two loss supervision λ_{coc} and λ_{grad}, we set $\lambda_{coc} = 0.5$ due to that a excessive large λ_{coc} would overwhelm the useful cues in $\mathbf{F}_{init}$ learned from reblurring and deblurring task, and $\lambda_{grad} = 0.5$ to reserve the high-frequency information and sharp the edge in restored image.

Design of Gate Vector R in Eq. 7. A computed in Eq. 8 of main paper,

$$\mathbf{F}_{coc} = \mathbf{R} \odot \mathbf{F}_G + \mathcal{P}_{feat}\left(\mathcal{P}_{rgb}\left(\mathbf{F}_{init}\right)\right) \ , \tag{17}$$

where $\mathbf{R}$ serves as the gating vector to balance the learning between the vanilla disparity feature $\mathbf{F}_G$ and CoC-aligned feature $\mathcal{P}_{rgb}(\mathbf{F}_{init})$. Specifically, it is formulated as,

$$\mathbf{R} = \text{Tanh}(\mathcal{P}_G(\mathbf{F}_G) + \mathcal{P}_C(\mathcal{P}_{feat}(\mathcal{P}_{rgb}(\mathbf{F}_{init})))), \tag{18}$$

where $\mathcal{P}_G(\cdot)$ and $\mathcal{P}_C(\cdot)$ are two simple linear neural layers. We empirically observe that using the gate $\mathbf{R}$ could strengthen the gradient of $\mathbf{F}_{init}$, and adaptively balance the contribution of $\mathbf{F}_{init}$ and $\mathbf{F}_G$ to feature vector $\mathbf{F}_{coc}$.

B Details of Curated DP image Dataset

In this paper, We present a real-world DP dataset consisting of 10 high-quality pairs with a resolution of 6720 $\times$ 4480 pixels, for refocusing evaluation. The DP image is captured by Canon EOS 5D MarkIV [1]. The camera sensor features two independent photodiodes embedded within each pixel, enabling phase-detection autofocus for rapid focusing. During image capture, the left and right sub-pixels combine their outputs to generate the final view. After collection, we use software Digital Photo Professional [2] to split the DP view from the captured center one. In our dataset, the image pairs are captured using aperture settings corresponding to $f/2.8$, which results in the greatest DoF and thus most defocus blur. As shown in Figure 8, we select two objects o_1 and o_2 as focal point to form a image pair $(\mathbf{B}_{o1}, \mathbf{B}_{o2})$, and each of pair $\mathbf{B}_{\cdot}$ contains a dual-pixel image pair $(\mathbf{B}_{\cdot,l}, \mathbf{B}_{\cdot,r})$. When performing evaluation, we take one dual-pixel image pair as the input, and regard another image (center view) as the target image.

Notably, considering that when two objects are far apart, focusing on different objects may introduce noticeable misalignment of the captured scene. To address this issue, we adopt the following strategies during the captured process: (1) Restrict the distance between the two focus targets within a certain range to allow only minimal and acceptable shifts. (2) Manually crop and align the two images when slight misalignment still occur. (3) First adopting classical image matching techniques [33, 46] for pixel matching, and then compute corresponding quantitative metrics. All the datasets will be released.

C Adaptation of Refocusing Functionality

We manually modify two SOTA methods on deblurring task, K3DN and Omni-Kernel [11], for refocusing adaption. For K3DN (architecturally similar to our approach), we replace our 4-level UNet with K3DN's backbone while preserving only R0 (Table 7) to maintain essential reversibility. For Omni-Kernel that has more complex architecture, we simply concatenate disparity feature and DP image features in a channel-wise manner, and take them as input for target image approximation.

[1] https://www.canon.co.uk/cameras/eos-5d-mark-iv/

[2] https://app.ssw.imaging-saas.canon/app/zh/dpp.html?region=6

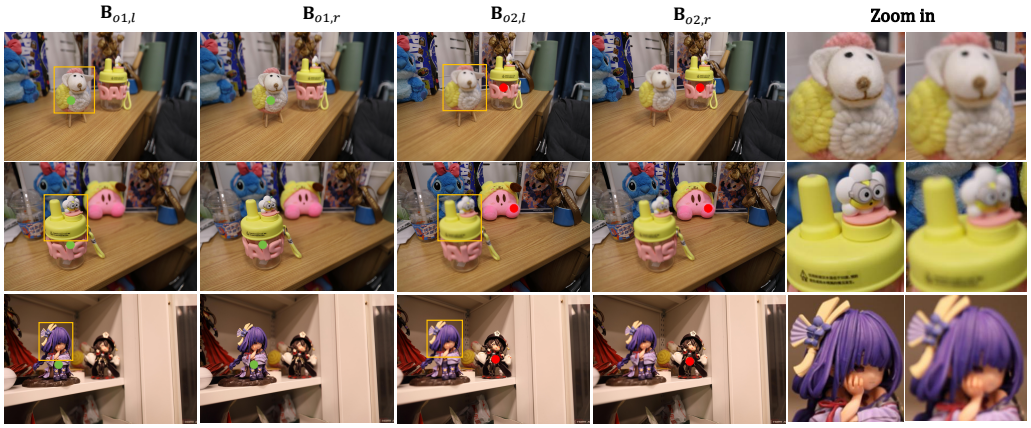

Figure 8: Three examples of our collected DP image pairs. The red and green circles indicate the focal points, while the orange bounding box highlight the zoomed-in regions.

D Comparison with Existing methods

Compared to existing deblur-and-reblur frameworks (e.g., RefocusGAN), our key improvements are:

- **Reversible Block with Shared Kernel:** We establish an invertible connection between deblurring and reblurring with shared kernel learning, which strictly follows the mathematical formulation in Eq. 11 while achieving better deblurring and reblurring performance.
- **Unified Network Architecture:** We reuse one network parameter to achieve refocusing task (deblur-and-reblur), significantly reducing the total number of parameters and improving training efficiency.

To highlight our contribution, we explain the necessity of invertibility in what follows. The network's invertibility ensures that reblurring and deblurring can be jointly trained while sharing the blur kernel, better establishing one-to-one correspondence between disparity feature and blur kernel. Intuitively, an effective disparity-aware blur kernel should be capable of both restoring an image (deblurring) and blurring it (reblurring). This paves a crucial path for the subsequent refocusing, enabling accurate kernel retrieval for defocus control by modifying the disparity (refer to Figure 7 for illustration). When we empirically adopt a traditional encoder-decoder architecture, the correspondence between the kernel and disparity becomes difficult to learn effectively, leading to unintended control effects.

E More Visualization Results and Analysis

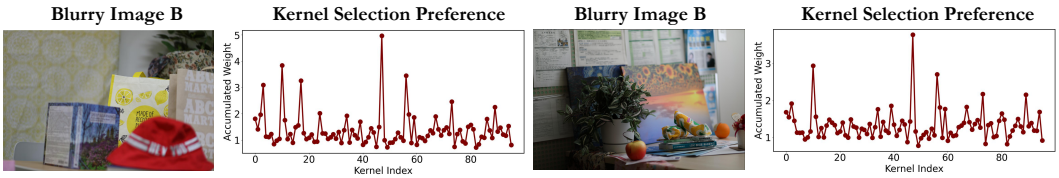

Figure 9: Kernel selection preference when performing refocusing from all-in-focus image. *Accumulated Weight* refers to assigned weight score of each blur kernel $\mathbf{K}_n$ accumulated across al the layer in invertible network.

Kernel Setting. We visualize the kernel selection preference of two cases in Figure 9 for deeper behavior revelation. It can be observed that most kernels has a smaller accumulated weight while only a small part of kernels are activated for blur indication. In these two cases, the commonly used kernels lies in index {56, 57, 65, 73, 86}, and kernels with index {3, 17, 22, 80} are sample-specific. We also employ the hard selection of blur kernel, which can be differentiably achieved by Gumbel-Softmax [21]. However, we empirically observe that hard selection achieves the suboptimal deblurring and reblurring performance. We posit that this issue arises because hard selection induces

Table 10: Performance with different number of pre-defined kernel $\{\mathbf{K}_n\}_{n=1}^{N}$.

N	32	48	96	128
Deblurring Performance				
PSNR↑	25.92	26.14	**26.89**	26.82
Reblurring Performance				
PSNR↑	28.32	28.46	**29.18**	29.18

Table 11: Comparison of refocusing performance on self-collected dataset.

Variants	PSNR↑	SSIM↑	MAE$_{(10^{-1})}$ ↓	LPIPS↓
Omni-Kernel	19.50	0.690	0.56	32.94
K3DN	19.52	0.698	0.52	31.91
Ours	19.68	0.707	0.52	30.87

an initial bias toward specific kernels during the early stages of training, which subsequently hinders the optimization process for kernel selection. Additionally, we further investigate how different number of pre-defined kernels affect model performance. As shown in Table 10, Both insufficient and excessive numbers of kernels impair model performance. Fewer kernels cannot adequately cover the necessary blur range across the dataset, while too many introduce excessive non-trainable parameters that compromise model robustness.

Figure 10: Visualization comparison of refocusing on self-collected DP image pairs. The red point indicates the focal object, and the yellow bounding box highlights the zoomed-in regions.

More Refocusing Results on Self-Collected DP image pair. We give the quantitative results on our self-collected dataset in Table 11, our method achieves the consistent superior results across all the metrics. As shown in Figure 10, we take the first case as the example, the sheep is focused initially, and our goal is to transfer the focal point from sheep to its right bottle. We compare our method with two SOTA baselines K3DN and Omni-Kernel, the comparison results show that our refocusing result is more scene-realistic.

Evaluation on Mask Quality. To evaluate the mask quality from our adopted SAM, we manually annotate two objects for each image of our collected images as the groundtruth. We compare the average mIoU metrics [15] on different version. From Table 12, we observe that early fusion achieves the clear-cut performance compared with late fusion. Furthermore, our structured template-based prompt achieves better segmentation result.

Table 12: Comparison of segmentation performance with different prompts. † and $*$ denote the early-fusion and late-fusion version, respectively. Ours denotes the template-based prompt.

3-Click†	Vanilla Prompt†	Ours*	Ours†
93.6	87.5	81.6	91.8

Performance Variation with Different Blur Degree. Besides, we also exploit instructed SAM segmentation results towards the image inputs with different degree of defocus blur and our restored image $\hat{\mathbf{I}}$. We sample ten groups of sample from DP5K-test and each of which has 5 cases with different f-number, 1.8, 2.8, 4.0, and 5.6. We use the SAM with early fusion to generate the mask, and manually annotate the region with interest as the ground-truth. As can be seen from Table 13, we empirically observe that our used SAM could functions well when handling the image with large defocus blur (i.e., f-1.8), and using our restored image as the SAM input could achieves the best result. Beyond that, with the blur degree decreasing, the segmentation result becomes saturated.

Table 13: Segmentation performance with different degrees of defocus blur.

f-1.8	f-2.8	f-4.0	f-5.6	$\hat{\mathbf{I}}$
85.8	90.3	90.8	91.1	91.8