

The Supplementary Material of User-Instructed Disparity-aware Defocus Control

Anonymous Author(s)

Affiliation

Address

email

1 Abstract of Appendix

2 This appendix provides more implementation details (Appendix A), details of curated DP image
3 dataset (Appendix B), adaptation for refocusing functionality (Appendix C), more visualization
4 results and analysis (Appendix D).

5 A More Implementation Details

6 **Network Architecture.** Our method adopts a 4-stage UNet-style architecture for progressive feature
7 refinement. As mathematically formulated in Eq. 9 of main paper, each stage has two cascaded
8 parameter-independent modules: (i) disparity-aware feature convolution, followed by (ii) invertible
9 block $Inv(\cdot)$ and $Inv^{-1}(\cdot)$. For (i), we use the disparity feature $\mathbf{F}_{coc}(i, j)$ to retrieve a kernel $\mathbf{K}_{i,j}$
10 from kernel pool $\{\mathbf{K}_n\}_{n=1}^N$, and then a point-wise convolution [7] with $\mathbf{K}_{i,j}$ or $\mathbf{K}_{i,j}^{-1}$ is employed
11 for feature refinement to obtain \mathbf{U}^l or \mathbf{U}^{l+1} , and the architecture of (ii) is elaborated in Table 1.
12 Notably, we parameterize learnable matrix \mathbf{W} directly in its LU decomposition [3] for efficiency, i.e.,
13 $\mathbf{W} = \mathbf{P}\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))$, where \mathbf{P} is a permutation matrix, \mathbf{L} is a lower triangular matrix with ones
14 on the diagonal, \mathbf{Q} is upper triangular matrix with zeros on the diagonal, and \mathbf{s} is a vector.

Table 1: 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	$\tilde{\mathbf{U}}^l = \mathbf{P}\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))\mathbf{U}^l$	$\tilde{\mathbf{U}}^{l+1} = (\mathbf{Q} + \text{diag}(\mathbf{s}))^{-1}\mathbf{P}^{-1}\mathbf{L}^{-1}\mathbf{U}^{l+1}$	$\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))$ denotes the matrix by LU decomposition [3], 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.

15 As mentioned in Table 6 of main paper, we adopt two variants $v1$ and $v2$ to investigate the impact of
16 different invertible blocks towards visual representation learning. Their architectures are illustrated in
17 Table 2 and Table 3, respectively.

18 **Hyper-parameter Setting.** Our \mathcal{L}_{deb} and \mathcal{L}_{reb} both uses a combination of Multi-Scale Charbonnier
19 loss \mathcal{L}_{char} [8], Multi-Scale Edge loss \mathcal{L}_{edge} [8], and Multi-Scale Frequency loss \mathcal{L}_{freq} [5], i. e.,
20 $\mathcal{L}_{reb} = \mathcal{L}_{deb} = \mathcal{L}_{char} + \lambda_1 \mathcal{L}_{edge} + \lambda_2 \mathcal{L}_{freq}$. We set $\lambda_1 = 5 \times 10^{-2}$ and $\lambda_2 = 1 \times 10^{-2}$. Regarding
21 other two loss supervision λ_{coc} and λ_{grad} , we set $\lambda_{coc} = 0.5$ due to that a excessive large λ_{coc} would

Table 2: The invertible block of variant $v1$.

#	Forward Operation	Backward Operation	Specification
R0	$\mathbf{U}^{l+1} = \mathbf{P}\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))\mathbf{U}^l$	$\mathbf{U}^l = (\mathbf{Q} + \text{diag}(\mathbf{s}))^{-1}\mathbf{L}^{-1}\mathbf{P}^{-1}\mathbf{U}^{l+1}$	$\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))$ denotes the matrix by LU decomposition [3], and \mathbf{P} is a permutation matrix.

Table 3: The invertible block of variant $v2$.

#	Forward Operation	Backward Operation	Specification
R0	$\tilde{\mathbf{U}}^l = \mathbf{P}\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))\mathbf{U}^l$	$\tilde{\mathbf{U}}^{l+1} = (\mathbf{Q} + \text{diag}(\mathbf{s}))^{-1}\mathbf{L}^{-1}\mathbf{P}^{-1}\mathbf{U}^{l+1}$	$\mathbf{L}(\mathbf{Q} + \text{diag}(\mathbf{s}))$ denotes the matrix by LU decomposition [3], 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.

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 \mathbf{R} in Eq. 7. A computed in Eq. 7 of main paper,

$$\mathbf{F}_{coc} = \mathbf{R} \odot \mathbf{F}_G + \mathcal{P}_{feat}(\mathcal{P}_{rgb}(\mathbf{F}_{int})) , \quad (1)$$

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,

$$\mathbf{R} = \text{Tanh}(\mathcal{P}_G(\mathbf{F}_G) + \mathcal{P}_C(\mathcal{P}_{feat}(\mathcal{P}_{rgb}(\mathbf{F}_{init})))), \quad (2)$$

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, for refocusing evaluation. The DP image is captured by Canon EOS 5D MarkIV¹. 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 Professional² 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 1, 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} . 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 [4, 6] 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 [1], for refocusing adaption. For K3DN (architecturally similar to our approach), we replace our 4-level

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

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

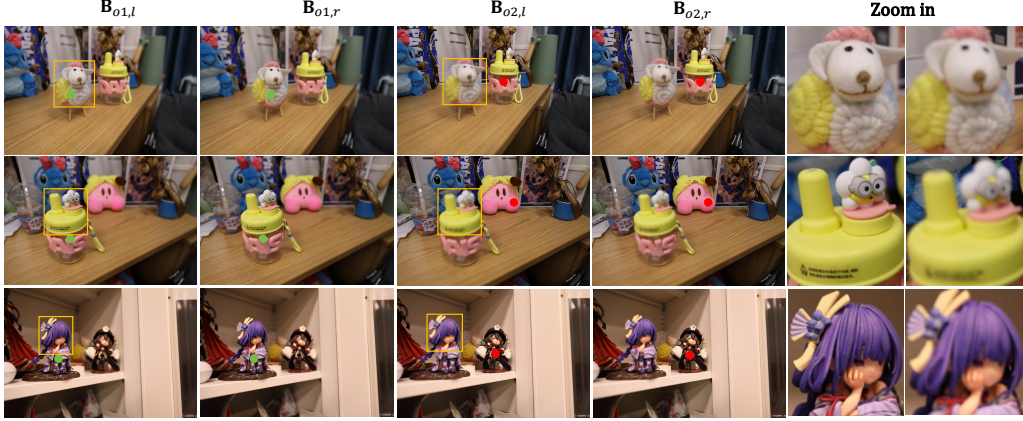


Figure 1: Three examples of our collected DP image pairs. The red and green circles indicate the focal points, while the orange bounding box highlight the zoomed-in regions.

50 UNet with K3DN’s backbone while preserving only R0 (Table 1) to maintain essential reversibility.
 51 For Omni-Kernel that has more complex architecture, we simply concatenate disparity feature and
 52 DP image features in a channel-wise manner, and take them as input for target image approximation.

53 D More Visualization Results and Analysis

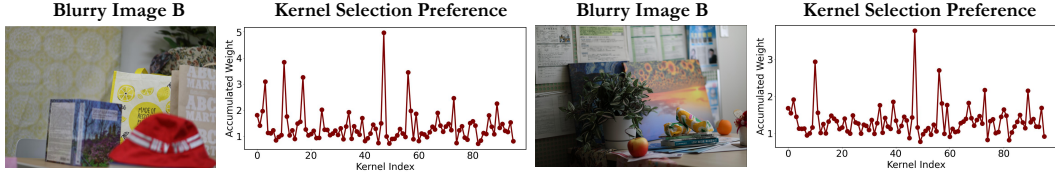


Figure 2: Kernel selection preference when performing refocusing from all-in-focus image. *Accumulated Weight* refers to assigned weight score of each blur kernel K_n accumulated across all the layer in invertible network.

54 **Kernel Setting.** We visualize the kernel selection prefer-
 55 ence of two cases in Figure 2 for deeper behavior revela-
 56 tion. It can be observed that most kernels has a smaller
 57 accumulated weight while only a small part of kernels are
 58 activated for blur indication. In these two cases, the com-
 59 monly used kernels lies in index {56, 57, 65, 73, 86}, and
 60 kernels with index {3, 17, 22, 80} are sample-specific. We
 61 also employ the hard selection of blur kernel, which can be
 62 differentially achieved by Gumbel-Softmax [2]. However,
 63 we empirically observe that hard selection achieves the
 64 suboptimal deblurring and reblurring performance. We posit that this issue arises because hard
 65 selection induces an initial bias toward specific kernels during the early stages of training, which
 66 subsequently hinders the optimization process for kernel selection. Additionally, we further investi-
 67 gate how different number of pre-defined kernels affect model performance. As shown in Table 4,
 68 Both insufficient and excessive numbers of kernels impair model performance. Fewer kernels cannot
 69 adequately cover the necessary blur range across the dataset, while too many introduce excessive
 70 non-trainable parameters that compromise model robustness.

71 **More Refocusing Results on Self-Collected DP image**
 72 **pair.** We give the quantitative results on our self-collected
 73 dataset in Table 5, our method achieves the consistent
 74 superior results across all the metrics. As shown in Fig-
 75 ure 3, we take the first case as the example, the sheep is
 76 focused initially, and our goal is to transfer the focal point

Table 4: Performance with different number of pre-defined kernel $\{K_n\}_{n=1}^N$.

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

Table 5: Comparison of refocusing performance on self-collected dataset.

Variants	PSNR \uparrow	SSIM \uparrow	MAE(10^{-1}) \downarrow	LPIPS \downarrow
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

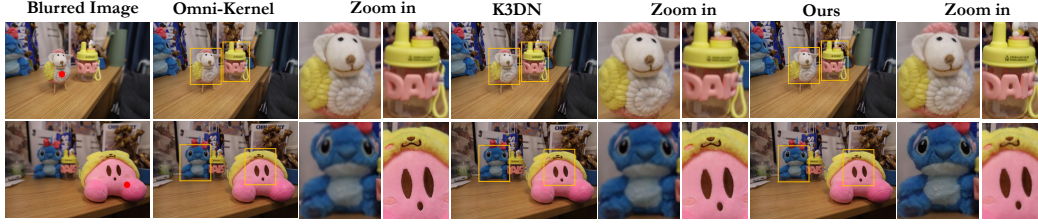


Figure 3: Visualization comparison of refocusing on self-collected DP image pairs. The red point indicates the focal object, and the yellow bounding box highlights the zoomed-in regions.

77 from sheep to its right bottle. We compare our method
78 with two SOTA baselines K3DN and Omni-Kernel, the
79 comparison results show that our refocusing result is more
80 scene-realistic.

81 References

- 82 [1] Yuning Cui, Wenqi Ren, and Alois Knoll. Omni-kernel network for image restoration. In
83 Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI
84 Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applica-
85 tions of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in
86 Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 1426–1434.
87 AAAI Press, 2024. 2
- 88 [2] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In
89 *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April
90 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 3
- 91 [3] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 con-
92 volutions. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò
93 Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Sys-
94 tems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018,
95 December 3-8, 2018, Montréal, Canada*, pages 10236–10245, 2018. 1, 2
- 96 [4] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the
97 International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*,
98 pages 1150–1157. IEEE Computer Society, 1999. 2
- 99 [5] Xintian Mao, Yiming Liu, Wei Shen, Qingli Li, and Yan Wang. Deep residual fourier transforma-
100 tion for single image deblurring. *CoRR*, abs/2111.11745, 2021. 1
- 101 [6] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue:
102 Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on
103 Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*,
104 pages 4937–4946. Computer Vision Foundation / IEEE, 2020. 2
- 105 [7] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural
106 networks, 05 2019. 1
- 107 [8] Yi Zhang, Qixue Yang, Damon M. Chandler, and Xuanqin Mou. Reference-based multi-stage
108 progressive restoration for multi-degraded images. *IEEE Trans. Image Process.*, 33:4982–4997,
109 2024. 1