Contrastive Monotonic Pixel-Level Modulation: Supplementary Material

Kun Lu, Rongpeng Li, and Honggang Zhang

Zhejiang University, Hangzhou, China
{lukun199, lirongpeng, honggangzhang}@zju.edu.cn

1 Ternary Search Implementations

We mention in Section 3.4 of the main paper the strategy of ternary search (TS). Now we detail how it is implemented. Let $f(c)$ be the aesthetic score function where we tacitly assume it is concave, to find the maximum value of $f(c)$, ternary search iteratively updates two boundary points c_{low} and c_{high} with the time complexity of $O(\log N)$, as shown in Algorithm 1. Note that for monotonically increasing aesthetic functions, ternary search also holds as the left boundary c_{low} will always yield a lower score than the right boundary point, thus resulting in a monotonic approximation towards the global maxima.

Algorithm 1 Ternary Search for MonoPix

Input: Concave or monotonically increasing aesthetic function $f(c)$, search range $[c_{left}, c_{right}]$, iteration times N
Output: The best enhance intensity $c_\star$

set boundary point $c_{low} \leftarrow c_{left}$, $c_{high} \leftarrow c_{right}$
for epoch=1:N **do**
  mid point $c_1 \leftarrow c_{low} + (c_{high} - c_{low})/3$
  mid point $c_2 \leftarrow c_{high} - (c_{high} - c_{low})/3$
  if $f(c_1) \leq f(c_2)$ **then**
    $c_{low} \leftarrow c_1$
  else
    $c_{high} \leftarrow c_2$
  end if
end for
$c_\star \leftarrow (c_{low} + c_{high})/2$

2 Network Structure and Training Details

2.1 Structures and Parameters

As shown in Table 1, we use the U-net [11] structure as the backbone for our generator, which consists of four sets of up and down-samplings. Following [5], we replace the original deconvolution with bilinear samplings to better mitigate the checkerboard artifacts. Note that MonoPix has an extra pixel-level control signal, so it only modifies the first convolution layer with marginal extra computations. The discriminator has three stride convolution layers, and is borrowed from CycleGAN [15]. Overall, the model has 8.6M parameters in generator, and 2.8M in discriminator, which is quite efficient compared with StarGANv2 [3]

Table 1: The detailed structure of our generator and discriminator. We represent the parameters of a convolution layer by (input channels, output channels, kernel size, and stride). `C`, `CLN`, `MP`, `UP`, `Cat` denotes convolution, convolution with leaky ReLU followed by a normalization layer, max pooling, bilinear upsampling, and concatenation respectively

Generator	
layer	operations
1	[`CLN`(3+1, 32, 3, 1), `CLN`(32, 32, 3, 1), `MP`(2)]
2-4	[`CLN`(C, $2C$, 3, 1), `CLN`($2C$, $2C$, 3, 1), `MP`(2)] $\times 3$, from $C = 32$ to 128
5	[`CLN`(256, 512, 3, 1), `CLN`(512, 512, 3, 1)]
6-9	[`UP`(2), `C`(C, $C/2$, 3, 1), `Cat`, `CLN`(C, $C/2$, 3, 1), `CLN`($C/2$, $C/2$, 3, 1)] $\times 4$, from $C = 512$ to 64
10	`C`(32, 3, 1, 1)
Discriminator	
1	`CL`[3, 64, 4, 2]
2-3	[`CNL`(C, $2C$, 4, 2)] $\times 2$, from $C = 64$ to 128
4	`CL`(256, 512, 4, 1)
5	`C`(512, 1, 4, 1)

(33.9M and 20.8M), CycleGAN [15] (11.4M and 2.8M), and SAVI2I [9] (6.9M not including attribute encoder, and 26.6M).

By default, we set normalization layers in MonoPix as identity mappings, which can be further adjusted for different tasks. We note that it is important to introduce certain non-linearity before instance normalization, as we have discussed in Section 3.2 of the main paper. For LOL low-light enhancement task, we integrate our contrastive modulation scheme into the authors' code instead.

2.2 Training Details

We use the official training/testing split for all datasets. The detailed implementations are described as follows:

Yosemite Summer-Winter Translation. The training set contains 1231 summer photos and 962 winter photos. Following the author's implementation, we resize these images to 256, and implement random horizontal flip as the augmentation. We set λ_{df} as 0.25, and the margin ϵ as 0.5. The whole model is trained with a batch size of 8 (4×2 since we generate two $\boldsymbol{c}$ for each patch) for 300 epochs, where the learning rate starts to drop linearly from epoch 200.

AFHQ Cat-Dog Translation. We train MonoPix with 5153 cat images and 4739 dog images. λ_{df} is set as 0.5. The rest settings are exactly the same as in Yosemite summer-winter translation, except for the training epochs which is set to be 200. The learning rate drops from epoch 100.

LOL Low-Light Enhancement. We follow EnlightenGAN [5] to use the 485 LOL dark images as the source domain, and 1016 normal images as the target domain. Note that this task is concentrated on enhancing dark images only, so we train a unidirectional generator and discriminator and do not use the domain fidelity loss. The margin ϵ is set to be 0.33. For MonoPix, we train with a batch size of 32 (16×2) while for EnlightenGAN we train with batch size of 16 (so that the total iterations are the same). Both are trained for 200 epochs.

Table 2: More experimental results. Variant (a) is the model adopted in the main paper

Variant	Description	Summer → Winter			Winter → Summer		
		AL↑/Rg↑	RL↑/Sm↓	ACC↑/FID↓	AL↑/Rg↑	RL↑/Sm↓	ACC↑/FID↓
(a)	MonoPix	0.939/0.154	0.940/0.015	0.950/38.1	0.960/0.145	0.964/0.019	0.949/47.3
(b)	$w/o\ \mathcal{L}_{df}$	0.927/0.091	0.914/0.008	0.866/41.3	0.963/0.097	0.964/0.010	0.847/50.3
(c)	$\mathcal{L}_{df} \rightarrow L1$	0.896/0.111	0.916/0.010	0.893/39.8	0.909/0.128	0.930/0.016	0.887/47.7
(d)	Add IN	0.975/0.148	0.972/0.014	0.917/38.6	0.981/0.120	0.972/0.008	0.860/47.8
(e)	Add BN	0.861/0.086	0.918/0.012	0.830/41.1	0.862/0.081	0.852/0.012	0.824/50.7

SIDD Noise Generation. We follow the setting in DANet [13] to process the dataset. Differently, we train MonoPix in an unsupervised and unidirectional manner, with a batch size of 16 (8×2) and patch size 128 for 200 epochs.

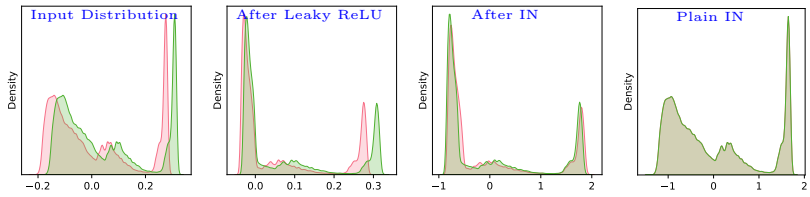
3 More Experimental Analysis

3.1 Domain Fidelity vs. Identity

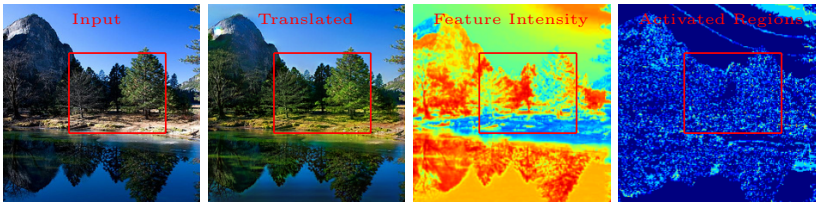
In Section 3.3 and Section 5.1 of the main paper, we point out that merely using monotonicity loss can lead to a biased control, and present the domain fidelity loss $\mathcal{L}_{df}$. Here we provide more analysis on this setting and compare it with another naive alternative i.e., using a weak identity loss. As reported in Table 2 (c), we first observe that adding an identity *L1* loss on variant (b) does bring a wider control since it encourages the model to better preserve the fidelity of inputs, which otherwise is not modeled when merely using monotonicity constraint. However as a side effect of identity mapping, the GAN loss is thereby sacrificed and causes certain degradation on the control linearity. Further, through the comparison with our full model (a), using an identity loss also produces an inferior result, which on the other hand reveals the benefits of domain fidelity constraint, and supports our claims in the main paper.

3.2 Impact of Different Normalization Methods

We provide a theoretical analysis in Section 3.2 on the different normalization methods. As part of the model, we complete our paper with a more detailed analysis in Table 2 (d-e). These models trained with additional normalization layers, generally, exhibit better smoothness yet with slightly inferior but acceptable translation quality compared with variant (a). Typically we find batch normalization (BN) leads to certain degradation on the linearity while IN on the contrary brings a boost. This is probably because IN better helps to model the style transition property in this task, while the inner contrastivity and dependency on training batch size narrow the improvements of BN. Overall, the choice of normalization layers serves as an option when implementing MonoPix, which brings certain differences but does not impede the monotonic modulation process.



(a) Adding a non-linear activation solves the IN dilemma. Red and green shadows denote the distribution of two feature maps obtained with varied modulation intensity



(b) Visualization on the activated domain-specific features. The last but one picture evaluates the intensity of feature maps, while the rightmost picture provides visualization by locating where a non-linear transformation takes place

Fig. 1: Illustration of the instance normalization-compatible modulation in MonoPix, and the corresponding visualizations

3.3 Visualization on the Activated Features

In the main paper, we propose to inject the contrastive controlling signal via a non-linear activation before adopting instance normalization (IN). Here we show in detail how this solution works in Figure 1 (a), and the by-product of a visualization of activated regions in Figure 1 (b). In the first sub-figure, we provide the numerical distribution of two feature maps obtained from different enhance levels of a same image. As can be found that a single non-linearity, combined with the convolution kernel, injects the intensity signal by "pushing" the feature distribution across non-linear activation regions and prevents a homogeneous representation after IN. Based on this property, we can further visualize the model's attention by examining where a non-linear transformation takes place (in leaky ReLU (LReLU) for example, we focus on the pixel entries that change their signs when the modulation intensity changes, as the non-linearity in LReLU is at zero), as we show in the rightmost picture in Figure 1 (b). It provides a more precise visualization than simply relying on the absolute value of feature maps, whose result is provided in the last but one picture.

3.4 Intensity of Interest

As we mentioned in Section 4.3 of the main paper, MonoPix exhibits a considerably better absolute linearity which shows the translation is loyal and monotonic,

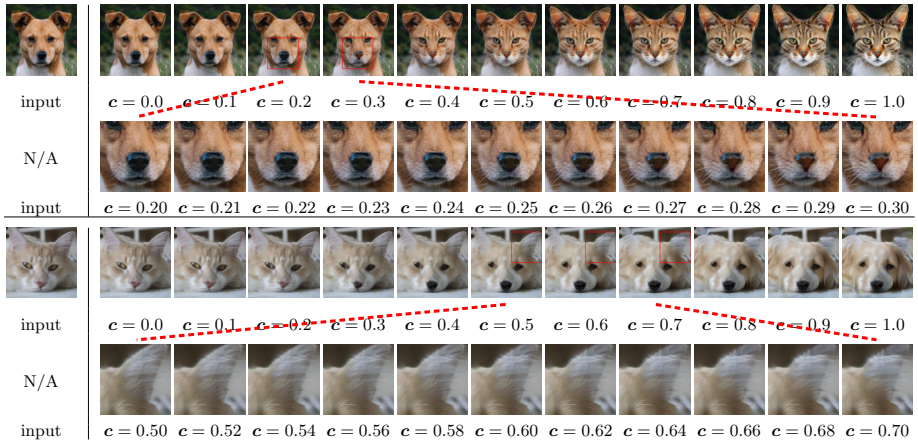


Fig. 2: Illustration of Intensity of interest (IOI), where $\boldsymbol{c}$ denotes the translation intensity. A fine-grained continuous translation can be obtained on intensities of interest selectively, for example to observe how whiskers and cats' eyes grow (the second row) and how a cat's ear evolves to that of a dog (the last row)

but the adjacent modifications are not perfectly and evenly distributed. We note that this is not a practical problem since we can, actually, provide a finer and monotonic modulation among "intensity of interest" instead of relying on uniformly sampled intensities. In real scenarios, it is also expected in most cases that a user modulates in a selective and attentive manner (such as employing ternary search), where a high absolute linearity may count more than relative linearity (a low absolute linearity incurs risk of non-unimodal aesthetic curve, while a low relative linearity will not impede ternary search but degrades the complexity). In Figure 2, we take the growth of whiskers and cats' eyes, and evolution of dog ears as examples to better demonstrate this point. A finer and detailed modulation is provided next to the overall modulations.

3.5 Generalization on Latent Representation

Besides the main results, it is critical to examine the generalization ability of a learned generator. In previous works, it is evidenced that a well-learned generator retains the robustness and generates meaningful results by style interpolation [6], feeding unseen conditional inputs [10], or out-of-the-bound inference [2]. In the main paper, we have provided a detailed explanation on how a global intensity training enables pixel-level manipulation in the inference stage, which can be viewed as a spatial-level generalization. Here we provide more examples on the out-of-the-bound (OOB) manipulation, which is only mentioned in Section 4.4 (in the main paper) but not specifically presented. As can be found in Figure 3, MonoPix is compatible with a slightly OOB inference and generates natural-looking and acceptable results. Specifically, modulation with a negative

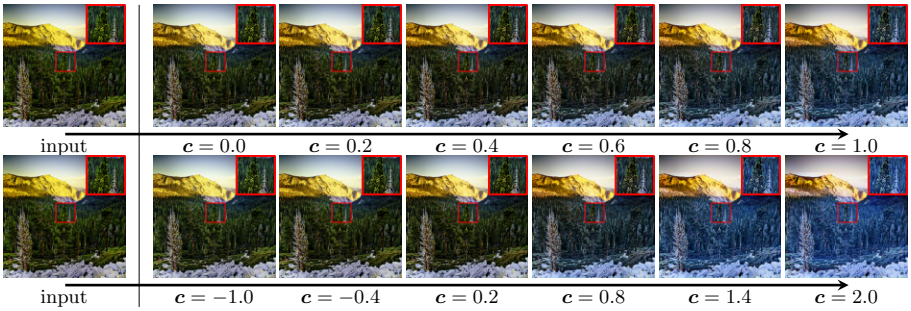


Fig. 3: Out-of-the-bound (OOB) generalization. MonoPix retains its robustness and generates realistic domain-specific images in a wider control range

modulation intensity can usually yield a better fidelity while those with higher intensities more resemble the target domain. This evidence suggests that the derived model does not merely memorize the input intensities, but learns to represent them in a linear and smooth latent space.

4 More Qualitative Evaluations

4.1 Image Quality over the Modulation Trajectory

One major difference between MonoPix and prevalent generative models is that, MonoPix does not assume $\boldsymbol{c} = 1$ is the optimal translation intensity, instead it learns through relative "reward" from domain discriminator. As a result, the optimal translation intensities vary for different input samples and are determined (hopefully) by ternary search. An example can be found in Figure 4. Though we are able to collect the highest accuracy (ACC) across the whole generation trajectory, calculating FID on a specific, and fixed intensity for MonoPix can lead to an underestimation. In Table 3, we provide an example of evaluating FID on the last translation intensity $\boldsymbol{c} = 1$, from which we can observe a relatively poor performance than on the whole trajectory. Nevertheless, we notice that evaluating FID on the whole, evenly-spaced intensities may not provide an ideal metric, as it discourages the model from exploring mixed-style intermediate samples, which can be identified from a poor relative linearity.

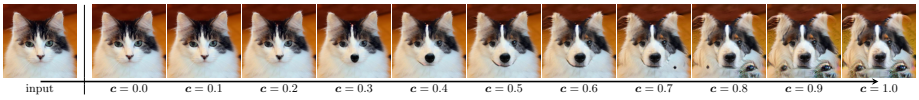


Fig. 4: Discrepancy between single-image and overall quality. MonoPix does not assume that images with the highest translation intensities are best candidates

Table 3: More FID comparisons. We report FID with the highest translation intensity (last modulation result) and across the whole trajectory (overall). Translation from Summer to Winter is denoted as S2W, and Cat to Dog as C2D

	last				overall			
Directions	S2W	W2S	C2D	D2C	S2W	W2S	C2D	D2C
CycleGAN-DNI [15]	60.2	66.0	-	-	35.0	42.8	-	-
StarGANv2-Rdm [3]	71.7	68.9	53.9	17.7	46.1	50.1	33.9	25.7
StarGANv2-Cent [3]	78.8	67.7	115	69.4	49.3	52.8	50.2	38.0
Liu *et al.* -Rdm [7]	-	-	72.8	23.1	-	-	37.1	27.5
SAVI2I-Rdm [9]	70.7	74.1	44.1	16.6	44.5	47.6	28.6	24.2
SAVI2I-Cent [9]	75.9	70.1	158	63.7	47.5	49.2	53.6	31.4
MonoPix	63.5	74.7	87.4	27.1	38.1	47.3	41.1	14.4

4.2 Subjective Consistency Analysis

We further carry out subjective consistency analysis to both check the tenability of our metrics and provide complementary evaluations. The results are listed in Table 4. We first notice that the *linearity*, *fidelity*, and *smoothness* metrics are fairly consistent with quantitative scores in the main paper. However, the quality of translation (i.e., *success* and *quality*) differs in Yosemite and AFHQ datasets. The primary reason behind this lies in the fact that MonoPix is now integrated with a CycleGAN-style translation framework, where the shape-related attributes are not particularly taken into consideration (compared with StarGANv2). Consequently, MonoPix shows a better fidelity and low-level style manipulation capability, but may not behave well in multi-attribute modulations. We also notice that the performance degradation is also related to the property we mention in Section 4.1 (in the supplementary material. Also in Section 4.3 of the main paper), where in some cases MonoPix provides unrealistic images near $\boldsymbol{c} = 1$. Though evaluating FID (typically on the whole trajectory) does not particularly reveal this issue, human critics are more sensitive to the final translated results.

Table 4: User study on MonoPix versus SAVI2I and StarGANv2. We show user preferences in percentages

	Yosemite		AFHQ	
Queries	vs. SAVI2I	vs. StarGANv2	vs. SAVI2I	vs. StarGANv2
Linearity	43.3	48.0	41.3	70.2
Fidelity	84.0	84.7	81.9	81.0
Smoothness	60.0	56.8	78.4	87.9
Success	63.2	60.4	27.4	30.4
Quality	65.9	67.4	26.3	37.9

5 More Visual Results

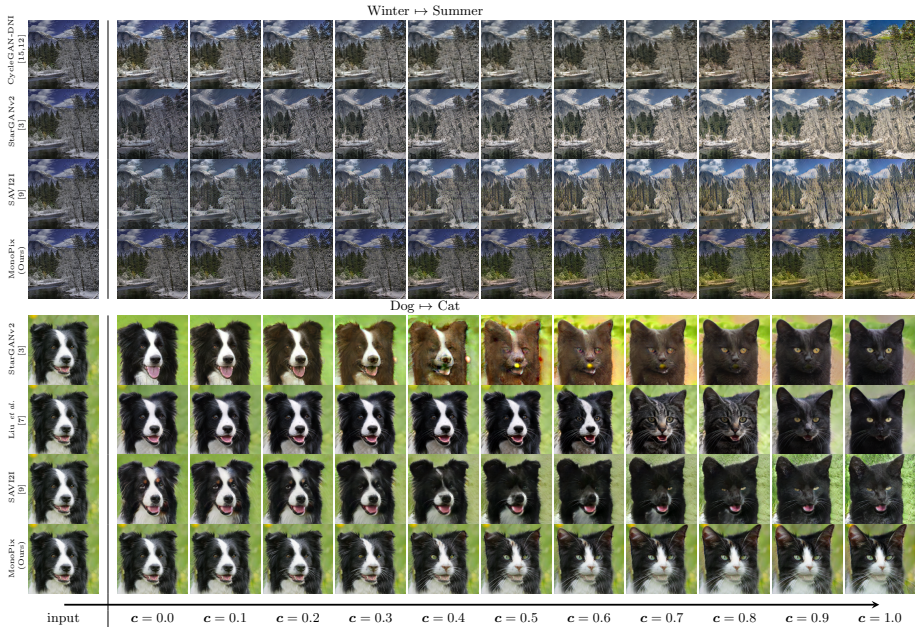


Fig. 5: More qualitative resutls on Yosemite winter to summer and AFHQ dog to cat translation

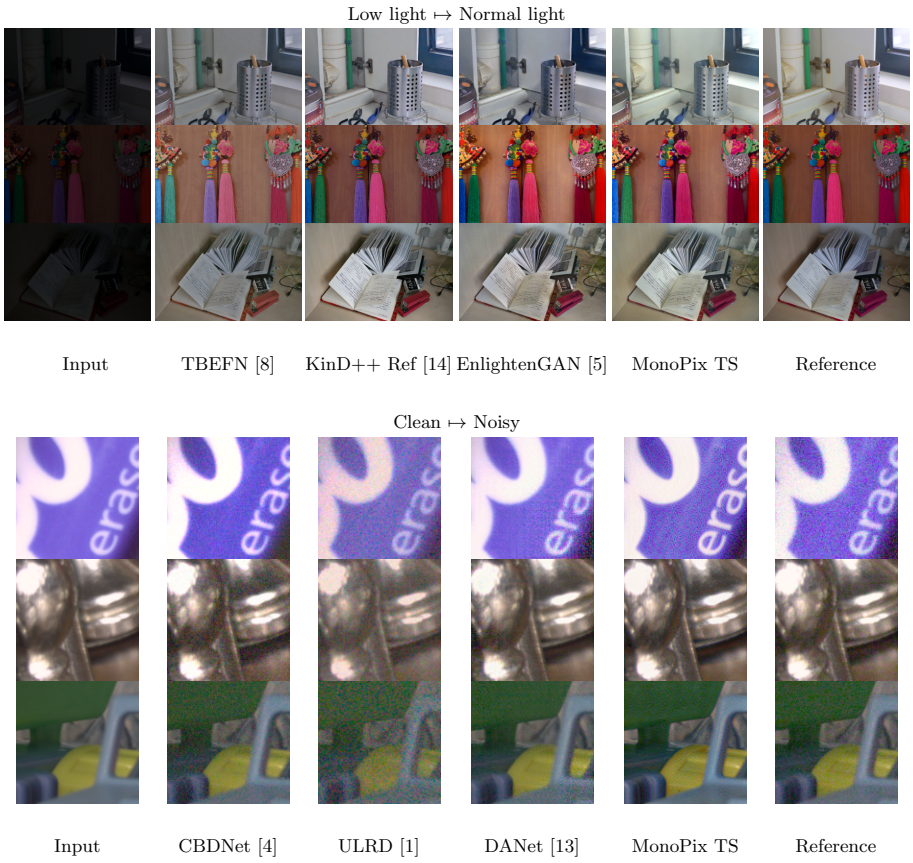


Fig. 6: Visual comparisons on the LOL low-light enhancement task and SIDD natural noise generation. In MonoPix, "TS" denotes "ternary search"

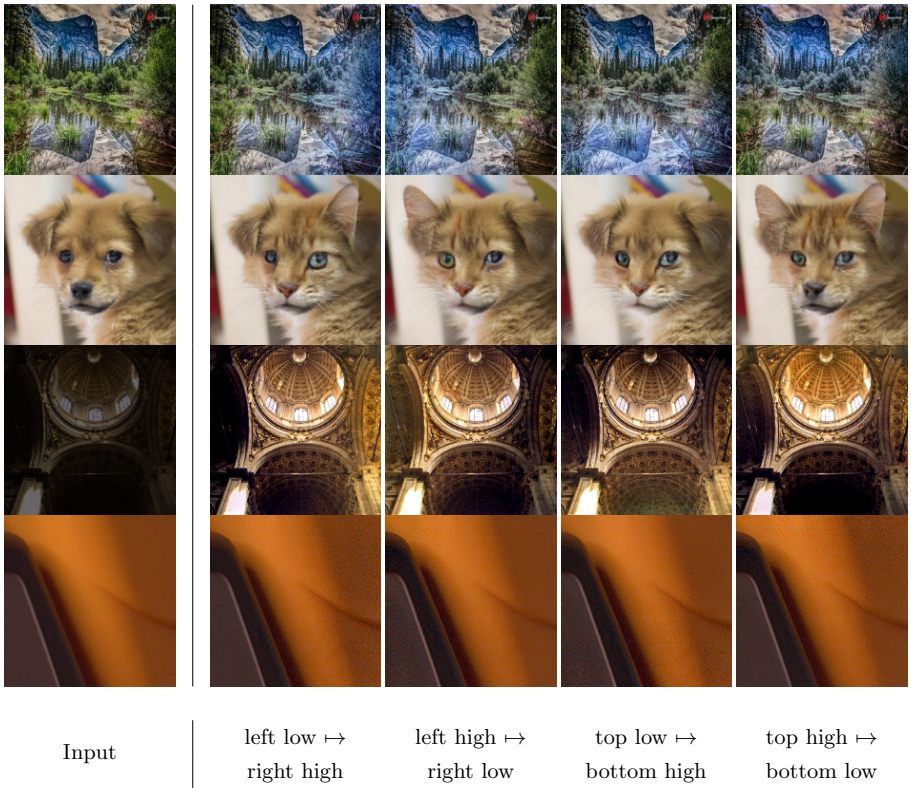


Fig. 7: More results on pixel-level spatial control. From left to right, we show the input image, continuous pixel-level modulation with intensities changing from left low to right high, left high to right low, top low to bottom high, top high to bottom low respectively

References

1. Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J.T.: Unprocessing images for learned raw denoising. In: CVPR. pp. 11036–11045 (2019)
2. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In: NeurIPS. vol. 29 (2016)
3. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: StarGAN v2: Diverse image synthesis for multiple domains. In: CVPR. pp. 8188–8197 (2020)
4. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: CVPR. pp. 1712–1722 (2019)
5. Jiang, Y., Gong, X., Liu, D., Cheng, Y., Fang, C., Shen, X., Yang, J., Zhou, P., Wang, Z.: EnlightenGAN: Deep light enhancement without paired supervision. TIP **30**, 2340–2349 (2021)
6. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: ECCV. pp. 35–51 (2018)
7. Liu, Y., Sangineto, E., Chen, Y., Bao, L., Zhang, H., Sebe, N., Lepri, B., Wang, W., De Nadai, M.: Smoothing the disentangled latent style space for unsupervised image-to-image translation. In: CVPR. pp. 10785–10794 (2021)
8. Lu, K., Zhang, L.: TBEFN: A two-branch exposure-fusion network for low-light image enhancement. TMM **23**, 4093–4105 (2020)
9. Mao, Q., Tseng, H.Y., Lee, H.Y., Huang, J.B., Ma, S., Yang, M.H.: Continuous and diverse image-to-image translation via signed attribute vectors. IJCV **130**, 517–549 (2022)
10. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: ICML. pp. 1060–1069 (2016)
11. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241 (2015)
12. Wang, X., Yu, K., Dong, C., Tang, X., Loy, C.C.: Deep network interpolation for continuous imagery effect transition. In: CVPR. pp. 1692–1701 (2019)
13. Yue, Z., Zhao, Q., Zhang, L., Meng, D.: Dual adversarial network: Toward real-world noise removal and noise generation. In: ECCV. pp. 41–58 (2020)
14. Zhang, Y., Guo, X., Ma, J., Liu, W., Zhang, J.: Beyond brightening low-light images. IJCV **129**(4), 1013–1037 (2021)
15. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV. pp. 2223–2232 (2017)