

Multi-Dimension Modulation for Image Restoration with Dynamic Controllable Residual Learning

Jingwen He* Chao Dong* Yu Qiao†

ShenZhen Key Lab of Computer Vision and Pattern Recognition, SIAT-SenseTime Joint Lab,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

{jw.he, chao.dong, yu.qiao}@siat.ac.cn

Abstract

Based on the great success of deterministic learning, to interactively control the output effects has attracted increasingly attention in the image restoration field. The goal is to generate continuous restored images by adjusting a controlling coefficient. Existing methods are restricted in realizing smooth transition between two objectives, while the real input images may contain different kinds of degradations. To make a step forward, we present a new problem called multi-dimension (MD) modulation, which aims at modulating output effects across multiple degradation types and levels. Compared with the previous single-dimension (SD) modulation, the MD task has three distinct properties, namely joint modulation, zero starting point and unbalanced learning. These obstacles motivate us to propose the first MD modulation framework – CResMD with newly introduced controllable residual connections. Specifically, we add a controlling variable on the conventional residual connection to allow a weighted summation of input and residual. The exact values of these weights are generated by a condition network. We further propose a new data sampling strategy based on beta distribution to balance different degradation types and levels. With the corrupted image and the degradation information as inputs, the network could output the corresponding restored image. By tweaking the condition vector, users are free to control the output effects in MD space at test time. Extensive experiments demonstrate that the proposed CResMD could achieve excellent performance on both SD and MD modulation tasks. Code is available at <https://github.com/hejingwenhejingwen/CResMD>.

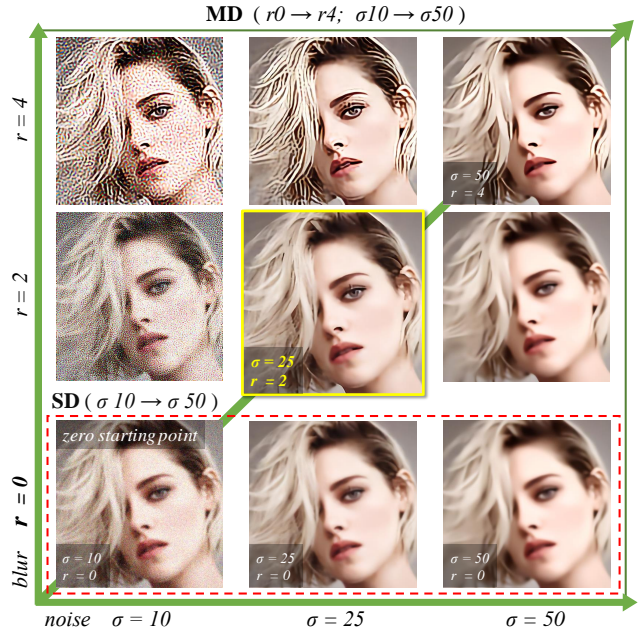


Figure 1. 2D modulation for a corrupted image with blur $r=2$ and noise $\sigma=25$. When the blur level is fixed to $r=0$, we can only modulate the denoising effect ($\sigma=10 \rightarrow \sigma=50$), which is a typical SD modulation. In MD modulation, the users are allowed to modulate both the deblurring and denoising levels.

1. Introduction

Conventional deep learning methods for image restoration (e.g., image denoising, deblurring and super resolution) learn a deterministic mapping from the degraded image space to the natural image space. For a given input, most of these methods can only generate a fixed output with a pre-determined restoration level. In other words, they lack the flexibility to alter the output effects according to different users' flavors. This flexibility is essential in many image processing applications, such as photo editing, where users desire to adjust the restoration level/strength continuously by a sliding bar. To adapt conventional deep models to real

*The first two authors are co-first authors.

†Corresponding author

scenarios, several recent works investigate the use of additional branches to tune imagery effects, such as AdaFM [7], CFSNet [18], Dynamic-Net [16], DNI [19] and Decouple-Learning [5]. The outputs of their networks can be interactively controlled by a single variable at test-time, without retraining on new datasets. They can generate continuous restoration results between the pre-defined start level and end level (e.g., JPEG quality $q40 \rightarrow q10$).

These pioneer modulation works assume that the input image has only a single degradation type, such as noise or blur, thus the modulation lies in one dimension. However, the real-world scenarios are more complicated than the above assumptions. Specifically, real images usually contain multiple types of degradations, e.g., noise, blur, compression, etc [20, 17]. Then the users will need separate buttons to control each of them. The solution is far beyond adding more controllable parameters. As these degradations are coupled together, altering a single degradation will introduce new artifacts that do not belong to the pre-defined degradation types. We denote this problem as multi-dimension (MD) modulation for image restoration. Compared with single-dimension (SD) modulation, MD modulation has the following three major differences/difficulties.

Joint Modulation. MD modulation aims to remove the effects of individual degradations as well as their combinations. Different types of degradations are coherently related. Removing one type of degradation could unavoidably affect the other degradations. It is highly challenging to decouple different degradations and modulate each of them separately. An example is shown in Figure 1. When we fix the blur level and adjust the noise level, the outputs will contain less and less noise but with fixed deblurring effects. All restored images should also be natural-looking and artifacts-free, indicating a high PSNR/SSIM.

Zero Starting Point. Zero starting point means that the modulation of each degradation type should start from level 0. It is essential in MD modulation, as the input image may only contain a part of degradation types. To an extreme, when the input has no degradations, the algorithm should perform identity mapping. However, this is not as easy as it seems. As a common assumption in SD papers, the base network should have the ability to perform restoration of a start level, so that the tuning block can adapt it to another end level. If the start level is zero, then the base network would have no restoration abilities, let alone adaptation and modulation. It is easy to realize 1 to N, but hard for 0 to 1.

Unbalanced Learning. As there are different degradation types with a large range of degradation levels, the pixel-wise loss (e.g., MSE) will be severely unbalanced for different inputs. For instance, given an input image, the MSE for its blurry version and noisy version could have different orders of magnitude. Furthermore, as the degradation level starts from 0, the MSE can be pretty small

around zero points. When we collect these different kinds of data as a training batch, the updating mechanism will favor the patches with large losses and ignore those with small ones. This phenomenon will result in inferior performance on mild degradations.

To address the aforementioned problems, we propose the first MD modulation framework with dynamic Controllable Residual learning, called CResMD. This is based on a novel use of residual connection. In conventional ResNet [8], the original input and its residual are combined by direct addition. In our settings, we reformulate it as a weighted sum – “ $output = input + residual \times \alpha$ ”, where α is the summation weight. If we add a global residual connection and set $\alpha = 0$, the output will be exactly the input. Then we can realize a special case of “zero starting point” - identity mapping. In addition, we can also add more local residual connections on building blocks. The underlying assumption is that the building blocks have their unique functions. When we enable some blocks and disable the others, the network can deal with different degradations. Therefore, the “joint modulation” can also be achieved by dynamically altering the weights α . We further propose a condition network that accepts the degradation type/level as inputs and generates the weight α for each controllable residual connection. During training, the base network and the condition network are jointly optimized. To further alleviate “unbalanced learning”, we adopt a new data sampling strategy based on beta distribution. The key idea is to sample more mild degradations than severe ones.

To verify the effectiveness of the proposed methods, we conduct extensive experiments on MD modulation for deblurring, denoising and JPEG deblocking. We have also made comparisons with SD methods, such as AdaFM [7], CFSNet [18], and DNI [19]. Experimental results show that the proposed CResMD could realize MD modulation with high accuracy and achieve superior performance to existing approaches with much less (0.16%) additional parameters.

2. Related Work

Image Restoration. Deep learning methods have been widely used in image restoration problems, and most of them focus on a specific restoration task, such as denoising, deblurring, super-resolution and compression artifacts reduction [2, 3, 22, 4, 11, 12]. Here we review some recent works that are designed to handle a wide range of degradation types or levels. Zhang et al. [22] propose DnCNN to deal with different levels of Gaussian noise. Then, Guo et al. [6] attempt to estimate a noise map to improve the denoising performance in real-world applications. Different from these task-specific methods, Yu et al. [20] aim to restore images corrupted with combined distortions with unknown degradation levels by exploiting the effectiveness of reinforcement learning. Later on, they propose a multi-

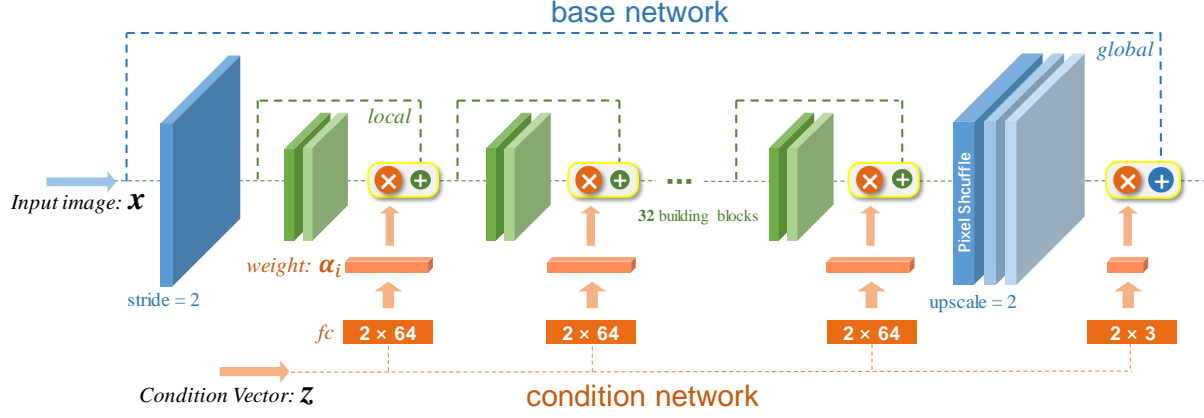


Figure 2. Framework of CResMD, consisting of two branches: base network and condition network. The base network deals with image restoration, while the condition network generates the weights α for the controllable residual connections.

path CNN [21] that can dynamically determine the appropriate route for different image regions. In addition, the work in [17] utilizes the attention mechanism to select the proper operations in different layers based on the input itself. However, these fixed networks cannot be modulated to meet various application requirements.

Modulation. we briefly review four representative SD methods AdaFM[7], CFSNet[18], Dynamic-Net[16] and DNI[19]. As a common property, all these methods train a couple of networks on two related objectives, and achieve the intermediate results at test time. The main differences lie on the network structure and the modulation strategy. In the first three works, they decompose the model into a main brunch and a tuning brunch. AdaFM adopts feature modulation filters after each convolution layer. CFSNet uses a side-by-side network upon the main brunch and couple their results after each residual block. Dynamic-Net adds modulation blocks directly after some convolution layers. During training, only the tuning brunch is optimized to another objective. Due to this finetuning strategy, the modulation could only happen between two objectives. DNI interpolates all network parameters, thus has the flexibility to do MD modulation. However, the linear interpolation strategy of DNI cannot achieve high accuracy (PSNR/SSIM) for image restoration tasks. In contrast, CResMD adopts the joint training strategy with much fewer additional parameters. It could achieve MD as well as SD modulation.

3. Method

Problem Formulation. We first give the formulation of multi-dimension (MD) modulation. Suppose there are N degradation types $\{D_j\}_{j=1}^N$. For each degradation D_j , there is a degradation range $[0, R_j]$. Our goal is to build a restoration model that accepts the degraded image together with its degradation information as inputs and generates the corresponding restored image. The degradation information (type and level) will act like tool bars, which can

be interactively modulated during testing. We use a two-dimension (2D) example to illustrate the modulation process. As shown in Figure 1, there are two separate bars to control the blur level D_1 and noise level D_2 . The modulation space is a square 2D space, spreading from $[0, 0]$ to $[R_1, R_2]$. We can fix D_1 and change D_2 , then the modulation trajectory is a horizontal line. We can also modulate D_1 and D_2 simultaneously, then the trajectory will become a diagonal line. This is completely different from SD modulation, where the trajectory is in one dimension and unique. If D_1 or D_2 is fixed on level 0 (zero starting point), then 2D degenerates to 1D, indicating that the MD modulation should cover all SD cases. What if the starting point is not zero? For instance, if D_1 can starts from a non-zero blur level, then the model cannot deal with images without blur. In other words, all restored images will contain a certain degree of deblurring effects, sometimes artifacts.

Framework. To achieve MD modulation, we propose a general and effective strategy based on controllable residual connections. The framework is depicted in Figure 1. The framework comprises of two branches - a base network and a condition network. The base network is responsible for image restoration, while the condition network controls the restoration type and level. The base network has a general form with downsampling/upsampling layers at two ends and several building blocks in the middle. The building block can be residual block[8], recurrent block[9], dense block[10], and etc. This structure is widely adopted in advanced image restoration models [11, 12, 23, 22, 6]. The only difference comes from the additional “controllable residual connections”, shown as blue and green dash lines in Figure 1. These residual connections are controlled by the condition network. Take any degradation type/level as input, the condition network will first convert them into a condition vector, then generate the weights α for controllable residual connections. At inference time, we can modulate the degradation level/type - $\{D_i\}_{i=1}^N$, then the model can generate continuous restoration results.

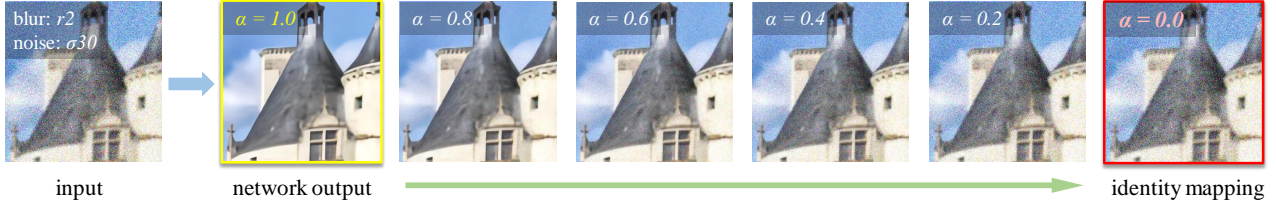


Figure 3. Different levels of restoration effects by setting different weights α on global residual.

Controllable Residual Connection. The proposed controllable residual connection comes from the standard residual connection, thus it is essential to review the general form of residual connection. Denote X and Y as the input and output feature maps. Then the residual connection can be represented as

$$Y = f(X, W_i) + X, \quad (1)$$

Where $f(X, W_i)$ refers to the residual feature maps and $f(\cdot)$ is the mapping function. While in our controllable residual connection, we add a tunable variable α to control the summation weight. The formulation becomes

$$Y = f(X, W_i) \times \alpha + X. \quad (2)$$

Where α has the same dimension as the number of feature maps. This simple change gives residual connection two different properties. First, through tuning the variable α from 0 to 1, the output Y will change continuously from X to $f(X, W_i) + X$. Second, the residual part can be fairly skipped by setting $\alpha = 0$. We can add the following two types of controllable residual connections.

(1) Global connection – X, Y are input/output images. The initial motivation of adding global connection is to handle the extreme case of zero starting point, where all degradation levels are zero. Generally, it is hard for a conventional neural network to perform identity mapping and image restoration simultaneously. However, with the help of global connection, the identity mapping can be easily realized by setting $\alpha = 0$. Furthermore, when we change the values of α , the output will exhibit different levels of restoration effects. This phenomenon is illustrated in Figure 3, where the input image is degraded by noise+blur and the intermediate results are obtained by using different α .

(2) Local connection – X, Y are input/output feature maps. If the imagery effects can be affected by a simple variable, we can also control the feature maps to achieve more complicated transformation. A reasonable idea is to add local residual connection on each function unit, which is responsible for specific degradation. By disabling/suppressing some function units, we can deal with different degradations. However, it is almost impossible to decouple these degradations and define a clear function for each block. Thus we roughly group some basic building

blocks and add controllable residual connections. The minimum function unit consists of a single building block. Experiments in Figure 8 show that more local residual connections achieve better performance at the cost of more controlling variables. More analysis can be found in Section 4.4.

Condition Network. We further propose a condition network that accepts the degradation type/level as inputs and generates the weight α for each controllable residual connection. As each degradation has its own range, we should first encode the degradation information into a condition vector. Specifically, we linearly map each degradation level to a value within the range $[0, 1]$, and concatenate those values to a vector z . Then the condition vector is passed through the condition network, which can be a stack of fully-connected layers.

Data Sampling Strategy. Data sampling is an important issue for MD modulation. As the training images contain various degradation types/levels, the training loss will be severely biased. If we uniformly sample these data, then the optimization will easily ignore the patches with small MSE values, and the performance of mild degradations cannot be guaranteed. To alleviate the unbalanced learning problem, we sample the degradation levels for each degradation type based on the beta distribution:

$$g(z, \alpha, \beta) = \frac{1}{B(\alpha, \beta)} z^{\alpha-1} (1-z)^{\beta-1}. \quad (3)$$

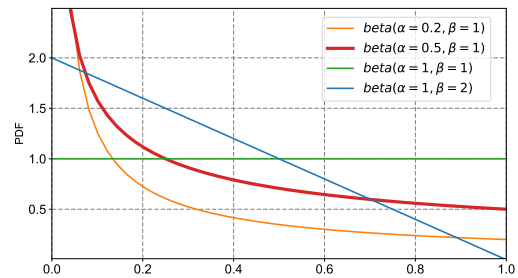


Figure 4. Beta distribution.

As shown in Figure 4, a larger value of α is associated with a steeper curve, indicating that the sampled degradation levels are inclined to the mild degradations. In our experiments, α and β are set to 0.5 and 1, respectively. We have also compared the results of different sampling curves in Section 4.4.

Training and Testing. The training strategy of the base network and the condition network is straightforward. Each input image is associated with an encoded condition vector based on the degradation type and level. The model takes both the corrupted image and the condition vector as inputs, while the original clean image is the corresponding ground truth. The joint training based on L1 loss will enable different restoration effects under different condition vectors.

In the testing stage, the users are allowed to modify the elements of the condition vector to obtain various restoration effects. For example, given a corrupted image with blur level $r = 2$ (range $[0, 4]$) and noise level $\sigma = 30$ (range $[0, 50]$), we recommend the users to first clean up the noise by gradually changing the condition vector from $[0, 0]$ to $[0, 0.6]$. Then the blurry effects can be easily eliminated by altering the first element, from $[0, 0.6]$ to $[0.5, 0.6]$. The best choice of the condition vector will be around $[0.5, 0.6]$.

4. Experiments

4.1. Implementation Details

We first describe the network architectures. For the base network, we adopt the standard residual block as the building block, which consists of two convolution layers and a Relu activation layer. There are 32 building blocks, of which the convolution layers have 64 filters with kernel size 3×3 . The first convolution layer downsamples features to half size with a stride of 2. The last upsampling module uses a pixel-shuffle [15] layer followed by two convolution layers. Note that the first and last convolution layers are not followed by Relu activation. We add a local controllable residual connection on each building block. For the condition network, we use a single fully-connected layer to output a 64-dimension vector α for each local controllable residual connection. In total, there are 32 layers for 32 local connections and 1 layer for the global connection.

To ease the burden of evaluation, we conduct most experiments and ablation studies on 2D modulation. To demonstrate the generalization ability, we conduct an additional experiment on 3D modulation at last. In 2D experiments, we adopt two widely-used degradation types Gaussian blur and Gaussian noise. JPEG compression is further added in the 3D experiment.

The training dataset is DIV2K[1], and the test datasets are CBSD68 [13] and LIVE1 [14]. The training images are cropped into 64×64 sub-images. To generate corrupted input images, we employ mixed distortions on the training data. In particular, blur, noise and JPEG are sequentially added to the training images with random levels. For Gaussian blur, the range of kernel width is set to $r \in [0, 4]$, and the kernel size is fixed to 21×21 . The covariance range of Gaussian noise is $\sigma \in [0, 50]$, and the quality range of JPEG compression is $q \in [100, 10]$. We sample the degra-

dations with stride of 0.1, 1, and 2 for blur, noise, and JPEG compression, respectively.

These training images are further divided into two groups, one with individual degradations and the other with degradation combinations. To augment the training data, we perform horizontal flipping and 90-degree rotations. To obtain more images with mild degradations, we force the sampling to obey beta distribution, where α and β are set to 0.5 and 1, respectively. The mini-batch size is set to 16. The L1 loss is adopted as the loss function. During the training process, the learning rate is initialized as $1 \times 10^{-4}/5 \times 10^{-4}$ for single/multiple degradations, and is decayed by a factor of 2 after 2×10^5 iterations. All experiments run 1×10^6 iterations. We use PyTorch framework and train all models on NVIDIA 1080Ti GPUs.

4.2. Complexity Analysis

The proposed CResMD is extremely light-weight, contributing to less than 4.2k parameters. As the additional parameters come from the condition network, the number of introduced parameters in 2D modulation is calculated as $32 \times 2 \times 64 + 2 \times 3 = 4102$. Note that the base network contains 32 building blocks with parameters around 2.5M, CResMD only comprises 0.16% of entire model. In contrast, the tuning blocks in AdaFM and CFSNet account for 4% and 110% of the total parameters of the base network, respectively. Another appealing property is that the computation cost of condition network is a constant, as there are no spatial or convolution operations. In other words, the computation burden is nearly negligible for a large input image.

4.3. Performance Evaluation

To evaluate the modulation performance, we follow AdaFM [7] and use PSNR distance. Specifically, if we want to evaluate the performance on D_1, D_2 , then we train a baseline model using the architecture of the base network purely on D_1, D_2 . With the ground truth images, we can calculate PSNR of CResMD and the baseline model respectively. Their PSNR distance is used as the evaluation metric.

2D modulation. First, we evaluate the 2D modulation performance of the proposed method. The quantitative results¹ of different degradations on CBSD68 dataset are provided in Table 1. We can observe different trends for different degradation types. For two degradations, the PSNR distances are all below 0.2 dB, indicating a high modulation accuracy. For one degradation, where there are zero starting points, the performance will slightly decrease. Furthermore, blur generally leads to higher PSNR distances than noise. The largest PSNR distance appears in $r1$, which is a starting point as well as a mild degradation. Nevertheless, its absolute PSNR value is more than 38 dB, thus the

¹Results on more datasets can be found in supplementary file.

Table 1. 2D experiments. The PSNR distances within 0.2 dB are shown in bold.

<i>one degradation</i>							<i>two degradations</i>								
blur r	1	2	4	0	0	0	1	1	1	2	2	2	4	4	4
noise σ	0	0	0	15	30	50	15	30	50	15	30	50	15	30	50
condition vector \mathbf{z}	[0.25,0]	[0.5,0]	[1,0]	[0,0.3]	[0,0.6]	[0,1]	[0.25,0.3]	[0.25,0.6]	[0.25,1]	[0.5,0.3]	[0.5,0.6]	[0.5,1]	[1,0.3]	[1,0.6]	[1,1]
CBSD68 [13] single	39.07	30.24	26.91	34.12	30.56	28.21	29.11	27.38	26.07	26.30	25.35	24.55	24.08	23.53	23.03
CResMD	38.38	30.09	26.53	33.97	30.43	28.06	29.00	27.27	25.96	26.24	25.29	24.48	24.03	23.46	22.95
PSNR distance	0.69	0.15	0.38	0.15	0.13	0.15	0.11	0.11	0.11	0.06	0.06	0.07	0.05	0.07	0.08

<i>one degradation</i>							<i>two degradations</i>					<i>three degradations</i>		
blur r	1	4	0	0	0	0	1	4	1	4	0	0	1	4
noise σ	0	0	15	50	0	0	15	50	0	0	15	50	15	50
JPEG q	∞	∞	∞	∞	80	10	∞	∞	80	10	80	10	80	10
condition vector \mathbf{z}	[0.25,0,0]	[1,0,0]	[0,0.3,0]	[0,1,0]	[0,0.24]	[0,0,1]	[0.25,0.3,0]	[1,1,0]	[0.25,0,0.24]	[1,0,1]	[0,0.3,0.24]	[0,1,1]	[0.25,0.3,0.24]	[1,1,1]
CBSD68 single	39.07	26.91	34.12	28.21	36.22	27.63	29.11	23.03	31.30	23.25	32.71	26.21	28.65	22.61
CResMD	38.20	26.43	33.92	28.01	35.93	27.37	28.97	22.93	30.96	22.99	32.58	26.00	28.55	22.49
PSNR distance	0.87	0.48	0.20	0.20	0.29	0.26	0.14	0.10	0.34	0.26	0.13	0.21	0.10	0.12

Table 2. 3D experiments. The PSNR distances within 0.2 dB are shown in bold.

restoration quality is still acceptable. We further show qualitative results in Figure 9, where all images exhibit smooth transition effects.

Comparison with SD methods. As the state-of-the-art methods are all proposed for SD modulation, we can only compare with them on single degradation types. We want to show that even trained for MD modulation, CResMD can still achieve excellent performance on all SD tasks. Specifically, we compare with DNI, AdaFM, and CFSNet on deblurring, starting from r_2 to r_4 . Deblurring is harder than denoising, thus could show more apparent differences. To re-implement their models, we first train a base network on the start level r_2 . Then we finetune (1) the whole network in DNI, (2) the AdaFM layers with kernel size 5×5 in AdaFM, (3) the parallel tuning blocks and coefficient network in CFSNet, to the end level r_4 . To obtain the deblurring results between r_2 and r_4 , we interpolate the networks of two ends with stride 0.01. For CResMD, we directly use the deblurring results in the 2D experiments (Table 1). From Figure 5, we observe that our method significantly outperforms the others in almost all intermediate points. In particular, the SD methods tend to yield high PSNR distances (> 1.0 dB) on tasks $r_{2.2} \sim r_{3.0}$. It is not surprising that they perform perfect at two ends as they are trained and finetuned on these points. This trend also holds for denoising, but with much smaller distances. All these results demonstrate the effectiveness of the proposed method in SD modulation. Results of deblurring $r_1 \rightarrow r_4$, denoising $\sigma_5 \rightarrow \sigma_{50}$ and $\sigma_{15} \rightarrow \sigma_{50}$ can be found in the supplementary file.

4.4. Ablation Study

Effectiveness of Controllable Residual Connection. First, we want to know that whether the performance comes from the joint training strategy or the proposed controllable

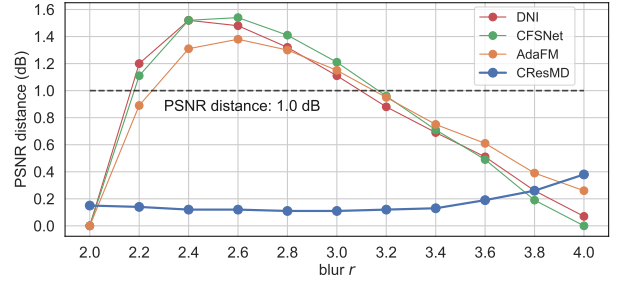


Figure 5. Comparison with SD methods on CBSD68 data set.

residual connection. To answer this question, we remove the additional condition network as well as the controllable residual connections, and concatenate the condition vector directly with the input image. As an alternatively controlling manner, we can use Adaptive feature modification (AdaFM [7]) layer to modulate the features within each residual block. Different from conventional AdaFM, we use a small condition network to generate the parameters of the AdaFM layers and adopt joint training strategy as CResMD. We evaluate the above three different conditional methods in 2D modulation. The comparisons of PSNR distances on CBSD68 dataset are shown in Figure 6. Obviously, direct concatenation has the worst performance especially on individual degradations: > 2.0 dB for blur r_1 and > 0.8 dB for blur r_4 . This indicates that an appropriate conditional strategy is required for effective modulation. Compare with AdaFM and CResMD, we can find that our CResMD is able to achieve almost half the PSNR distances of AdaFM on most degradations, even with much fewer additional parameters. In conclusion, the proposed CResMD explores a more efficient way to utilize the condition information compared with AdaFM and the concatenating strategy.

Effectiveness of Global Connection. The global connection is initially designed to handle the problem of zero

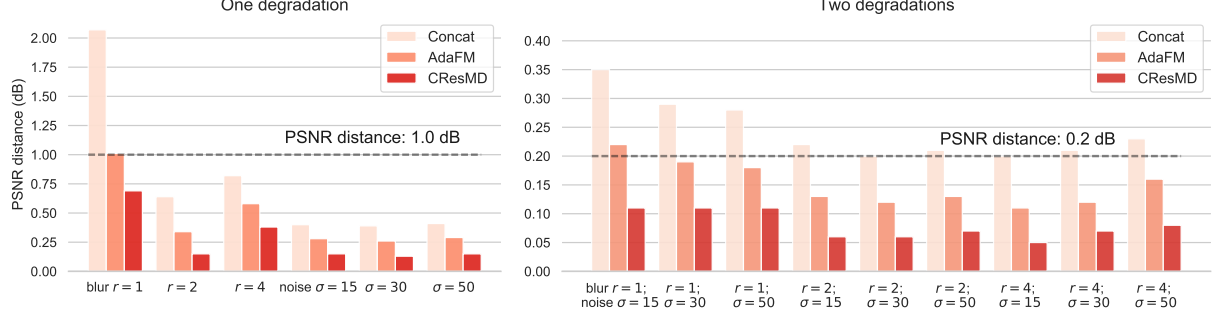


Figure 6. Performance of three different conditional strategies: concatenating, AdaFM and CResMD. The results are evaluated on CBSD68.

	blur r	1	2	4	0	0	0	1	2	4	total
	noise σ	0	0	0	5	30	50	5	30	50	
$\alpha = 1.0, \beta = 1.0$		38.66	30.01	26.26	39.90	30.63	28.24	31.78	24.97	22.58	
$\alpha = 0.5, \beta = 1.0$ (CResMD)		38.85	30.03	26.14	39.99	30.65	28.25	31.86	24.98	22.56	
		+0.19	+0.02	-0.12	+0.09	+0.02	+0.01	+0.08	+0.01	-0.02	+0.28
$\alpha = 0.2, \beta = 1.0$		38.94	29.98	26.07	39.97	30.55	28.10	31.68	24.85	22.39	
		+0.28	-0.03	-0.19	+0.07	-0.08	-0.14	-0.10	-0.12	-0.19	-0.50
$\alpha = 1.0, \beta = 2.0$		38.93	30.08	25.80	40.00	30.66	28.24	31.90	24.99	22.50	
		+0.27	+0.07	-0.46	+0.10	+0.03	+0.00	+0.12	+0.02	-0.08	+0.07

Table 3. Performance under different sampling curves evaluated on LIVE1 [14]

	blur r	0	0	0.5	1	0.5	0.5	1
	noise σ	0	5	0	0	5	15	5
CBSD68 w/o		71.39	40.21	52.70	38.04	37.80	32.31	31.48
w	$+\infty$	40.33	53.17	38.38	37.92	32.44	31.63	
gain	$+\infty$	0.12	0.47	0.34	0.12	0.13	0.15	
LIVE1 w/o		64.17	39.79	51.22	38.38	37.71	32.51	31.65
w	$+\infty$	39.99	52.21	38.85	37.89	32.69	31.86	
gain	$+\infty$	0.20	0.99	0.47	0.18	0.18	0.21	

Table 4. The effectiveness of global connection.

starting point. In general, it is hard for a conventional network to deal with both identity mapping and image restoration at the same time. With the proposed controllable global connection, we can ideally turn off the residual branch by setting $\alpha = 0$. To evaluate its effectiveness, we conduct a straightforward comparison experiment by just removing the global connection. This new model is trained under the same setting as CResMD. As for testing, we only select those mild degradations, such as blur $r < 1$ and noise $\sigma < 15$. It is clear that the model with controllable global connection could achieve better performance on all mild degradations as we can see from Table 4.

Effectiveness of Local Connection. Here we test the influence of the number of local connections. In particular, we group some basic building blocks as a function unit and add controllable residual connection. All the building blocks are divided into 1, 2, 4, 8, 16 and 32 groups (the details are illustrated in Figure 7). They are evaluated in 2D modulation on CBSD68 dataset. The results are depicted in Figure 8. Obviously, more groups or local connections

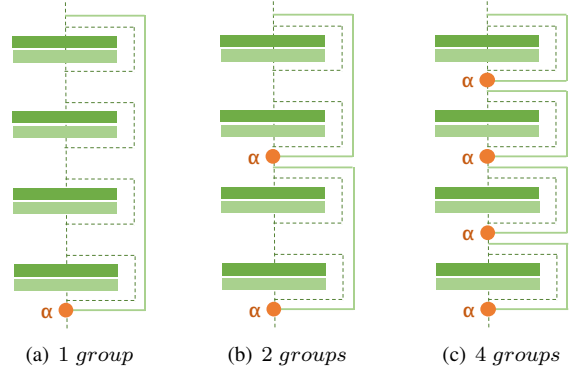


Figure 7. Different options of local connections.

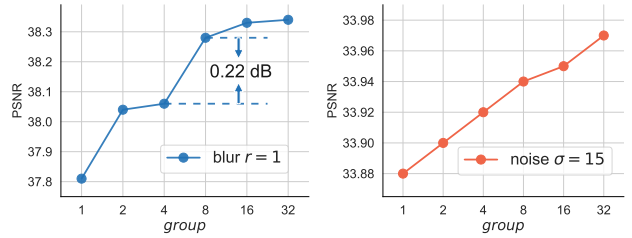
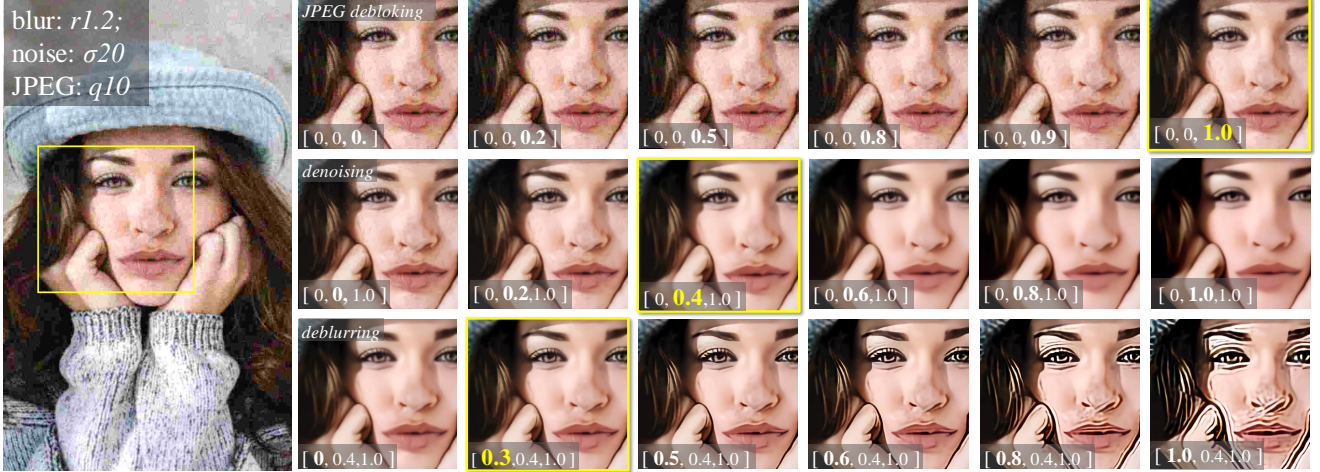


Figure 8. Performance under different local connections evaluated by CBSD68 dataset.

could lead to better performance. Particularly, we also observe a sharp leap (0.22dB) in deblurring $r = 1$ (from 4 to 8 local connections), indicating that at least 8 local connections are required. In contrast, results on denoising tasks are less significant, where the PSNR distance between 1 and 32 local connections is less than 0.1dB in denoising $\sigma = 15$.



(a) 2D modulation.



(b) 3D modulation.

Figure 9. Qualitative results of MD modulation. In each row, we only change one factor with other factors fixed. We arrive at the best choice in the yellow box. **Better view in zoom and color.**

Effectiveness of Data Sampling. After analysis of the proposed network structures, we then investigate different data sampling strategies. As mentioned in Section 3.4, appropriate data sampling strategies could help alleviate the unbalanced learning problem. To validate this comment, we conduct a set of controlled experiments with different sampling curves, which can be generated using different parameters of beta distribution in Function 3. To be specific, the most commonly used strategy is uniform sampling, corresponding to the green horizontal line in Figure 4. We can generate this curve by setting β and α to 1. Similarly, we can further set α, β to be (0.5, 1.0), (0.2, 1.0) and (1.0, 2.0) to generate linear and non-linear curves, shown in Figure 4. Then we train four CResMDs on different training datasets with the above sampling strategies. Results are shown in Table 3, where we use uniform sampling ($\alpha = 1, \beta = 1$) as our baseline and calculate the PSNR distances with other strategies. Obviously, when we sample more data on mild degradations, the performance will significantly improve. Furthermore, the PSNR increases on some degradation levels generally comes at the cost of the decrease on the others. For instance, in deblurring $r = 1, \alpha = 1.0, \beta = 2.0$ and $\alpha = 0.2, \beta = 1.0$ reach the highest performance, but also

get severe degradation in $r4$. As a better trade-off, we select the setting $\alpha = 0.5, \beta = 1.0$ for our CResMD, which stably improves most degradation levels.

Generalization to 3D modulation. In the above experiments, we mainly use 2D modulation for illustration. Our method can be easily extended to higher dimension cases. Here we show a 3D modulation example with three degradation types: blur, noise and JPEG compression. Note that in JPEG compression, the zero starting point is not quality 100 but quality ∞ , thus we extend the JPEG range as $\{\infty, [100, 10]\}$. In 2D modulation, there is only one degradation combination noise+blur. However, in 3D, the number increases to 4, including noise+blur, noise+JPEG, blur+JPEG and noise+blur+JPEG. Then the difficulty also improves dramatically. Nevertheless, our method can handle this situation by simply setting the dimension of the condition vector to 3. All the other network structures and training strategy remain the same. From the results shown in Table 2, we can observe that most PSNR distances are below 0.3 dB, indicating a good modulation accuracy. Compared with 2D modulation, the performance on single degradations decreases a little bit, which mainly due to the insufficient training data. We also show some qualitative results in

Figure 9, where we modulate one factor and fix the others.

5. Conclusion

In this work, we first present the multi-dimension modulation problem for image restoration, and propose an efficient framework based on dynamic controllable residual learning. With a light-weight structure, the proposed CResMD partially addresses the three difficult problems in MD modulation. Although CResMD could realize modulation across multiple domains, the performance can be further improved. The controlling method can be more accurate and diverse. We encourage future research on better solutions.

References

- [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. [5](#)
- [2] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 576–584, 2015. [2](#)
- [3] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, Feb 2016. [2](#)
- [4] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016. [2](#)
- [5] Qingnan Fan, Dongdong Chen, Lu Yuan, Gang Hua, Nenghai Yu, and Baoquan Chen. Decouple learning for parameterized image operators. 2018. [2](#)
- [6] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#), [3](#)
- [7] Jingwen He, Chao Dong, and Yu Qiao. Modulating image restoration with continual levels via adaptive feature modification layers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#), [3](#), [5](#), [6](#)
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. [2](#), [3](#)
- [9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. [3](#)
- [10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. [3](#)
- [11] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. [2](#), [3](#)
- [12] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. *CoRR*, abs/1707.02921, 2017. [2](#), [3](#)
- [13] S. Roth and M. J. Black. Fields of experts: a framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 2, pages 860–867 vol. 2, June 2005. [5](#), [6](#)
- [14] H. R. Sheikh, M. F. Sabir, and A. C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451, Nov 2006. [5](#), [7](#)
- [15] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. [5](#)
- [16] Alon Shoshan, Roey Mechrez, and Lihi Zelnik-Manor. Dynamic-net: Tuning the objective without re-training for synthesis tasks. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. [2](#), [3](#)
- [17] Masanori Suganuma, Xing Liu, and Takayuki Okatani. Attention-based adaptive selection of operations for image restoration in the presence of unknown combined distortions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#), [3](#)
- [18] Wei Wang, Ruiming Guo, Yapeng Tian, and Wenming Yang. Cfsnet: Toward a controllable feature space for image restoration. *CoRR*, abs/1904.00634, 2019. [2](#), [3](#)
- [19] Xintao Wang, Ke Yu, Chao Dong, Xiaoou Tang, and Chen Change Loy. Deep network interpolation for continuous imagery effect transition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1701, 2019. [2](#), [3](#)
- [20] Ke Yu, Chao Dong, Liang Lin, and Chen Change Loy. Crafting a toolchain for image restoration by deep reinforcement learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2443–2452, 2018. [2](#)
- [21] Ke Yu, Xintao Wang, Chao Dong, Xiaoou Tang, and Chen Change Loy. Path-restore: Learning network path selection for image restoration. *arXiv preprint arXiv:1904.10343*, 2019. [3](#)
- [22] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. [2](#), [3](#)
- [23] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [3](#)