Supplementary Material

Zehong Zhou, Fei Zhou, and Guoping Qiu

I. Introduction

This is the supplementary material of our manuscript *Blind Image Quality Assessment based on Separate Representations and Adaptive Interaction of Content and Distortion*. More descriptions of the proposed methods and experimental results will be included here. Notice that all the bibliography indexes refer to the ones in the manuscript.

II. More Implementation Details

A. Bilinear Pooling

In the manuscript, we adopt bilinear pooling in the DCI module of the SAWAR to generate interaction representation from the content and distortion representations. Specifically, given two feature maps $\mathbf{X}_1$ and $\mathbf{X}_2$ with the size of $h_1 \times w_1 \times c_1$ and $h_2 \times w_2 \times c_2$ respectively. The bilinear pooling of $\mathbf{X}_1$ and $\mathbf{X}_2$ requires $h_1 \times w_1 = h_2 \times w_2$, and then is formulated as:

$$\mathbf{Y} = \mathbf{X}_1^T \mathbf{X}_2, \tag{1}$$

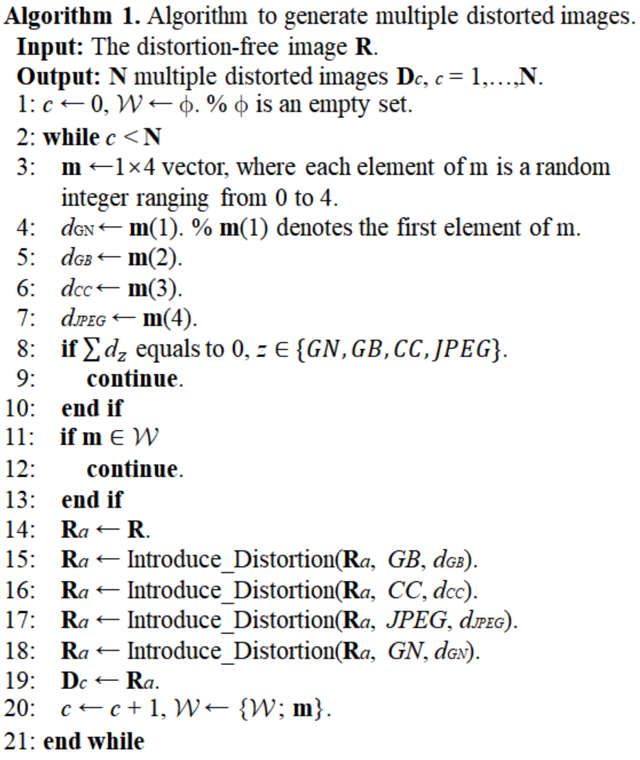
where the bilinear-pooling result $\mathbf{Y}$ is of dimension $c_1 \times c_2$.

As mentioned in [36], the bilinear representation is usually mapped from Riemannian manifold into an Euclidean space using signed square root and L2 normalization:

$$\mathbf{Y}^{'} = \frac{sign(\mathbf{Y}) \odot \sqrt{|\mathbf{Y}|}}{\|sign(\mathbf{Y}) \odot \sqrt{|\mathbf{Y}|}\|_2}, \tag{2}$$

where $\odot$ denotes element-wise multiplication. In our work, the content feature maps $\mathbf{F}_c^{'}$ and the distortion feature maps $\mathbf{F}_d$ serve as the input feature maps $\mathbf{X}_1$ and $\mathbf{X}_2$. However, since the height and width of the elements in $\mathbf{F}_d$ is not consist with $\mathbf{F}_c^{'}$, four elements $\mathbf{F}_d^{j}$ are first resized and concatenated with each other to generate $\mathbf{F}_d^{'}$ with the same size as $\mathbf{F}_c^{'}$. In this way, the elements of $\mathbf{F}_c^{'}$ and $\mathbf{F}_d^{'}$ will correspond to each other in spatial position. When the two types of feature maps are calculated as Eq. 1, the elements of $\mathbf{F}_c^{'}$ and $\mathbf{F}_d^{'}$ in the same spatial position are multiplied and pooled into a mixed feature, which can be regarded as the interaction information.

Further, the channels of $\mathbf{F}_c^{'}$ and $\mathbf{F}_d^{'}$, c_1 and c_2, are adapatively adjusted as 2 and 256 by two 1×1 convolutional layers so that the bilinear-pooling result has the same length as the quality-aware feature $\mathbf{f}_q$, i.e., 512.



Algorithm 1. Algorithm to generate multiple distorted images.

```
Input: The distortion-free image R.
Output: N multiple distorted images D_c, c = 1,…,N.
c ← 0, $\mathcal{W}$ ← ϕ. % ϕ is an empty set.
while c < N
    m ←1×4 vector, where each element of m is a random integer ranging from 0 to 4.
    $d_{GN}$ ← m(1). % m(1) denotes the first element of m.
    $d_{GB}$ ← m(2).
    $d_{CC}$ ← m(3).
    $d_{JPEG}$ ← m(4).
    if $\sum d_z$ equals to 0, $z \in \{GN, GB, CC, JPEG\}$.
        continue.
    end if
    if m ∈ $\mathcal{W}$
        continue.
    end if
    $R_a$ ← R.
    $R_a$ ← Introduce_Distortion($R_a$, GB, $d_{GB}$).
    $R_a$ ← Introduce_Distortion($R_a$, CC, $d_{CC}$).
    $R_a$ ← Introduce_Distortion($R_a$, JPEG, $d_{JPEG}$).
    $R_a$ ← Introduce_Distortion($R_a$, GN, $d_{GN}$).
    $D_c$ ← $R_a$.
    c ← c + 1, $\mathcal{W}$ ← {$\mathcal{W}$; m}.
end while
```

B. Network Details

The configurations of each layer and module in the COAE and SAWAR are expressed in the format as follows:

Convolutional: *Conv(kernel_size, output_dim, stride)*
Deconvolutional: *DeConv(kernel_size, output_dim, stride)*
Maxpooling: *MaxPool(kernel_size, stride)*
Global Average Pooling: *GAP(output_size)*
Spatial Pyramid Pooling: *SPP()*
Bilinear Pooling: *BiPool(output_size)*
Fully Connected Layer: *Linear(input_dim, output_dim)*
Residual Block: *ResBlock(kernel_size, output_dim, stride)* (including *Conv(kernel_size, output_dim, stride)*$\times 2$)
SPP Branch: *SPPBranch(dim)* (including *ResBlock(3, dim, 1), Conv(3, dim, 1), MaxPool(2, 2), Conv(1, dim, 1), SPP(), Linear($dim \times 21$, dim)*)
Sub-Modulation Residual Block: *SMResBlock(dim)* (including *Conv(3, dim, 1), Conv(1, dim, 1)*$\times 4$)

(1) CAE: The content encoder consists of 3 convolutional layers, *Conv(7, 64, 1), Conv(4, 128, 2), Conv(4, 256, 2)*, and 4 residual blocks, *ResBlock(3, 256, 1)*$\times 4$. The content decoder consists of 4 residual blocks, *ResBlock(3, 256, 1)*$\times 4$, and 3 deconvolutional layers, *DeConv(4, 128, 2), DeConv(4, 64, 2), DeConv(1, 3, 1)*.

(2) DAE: The distortion encoder consists of 2 convolution layers *Conv(7, 64, 1), Conv(3, 64, 1)*, 1 maxpooling layer *MaxPool(2, 2)*, and 4 SPP Branches *SPPBranch(64)*$\times 4$. The distortion decoder consists of 1 fully conntected layer *Linear(256, 1024)*, 4 sub-modulation residual blocks, *SMResBlock(256)*$\times 4$, and 3 deconvolutional layers, *DeConv(4, 128, 2), DeConv(4, 64, 2), DeConv(1, 3, 1)*.

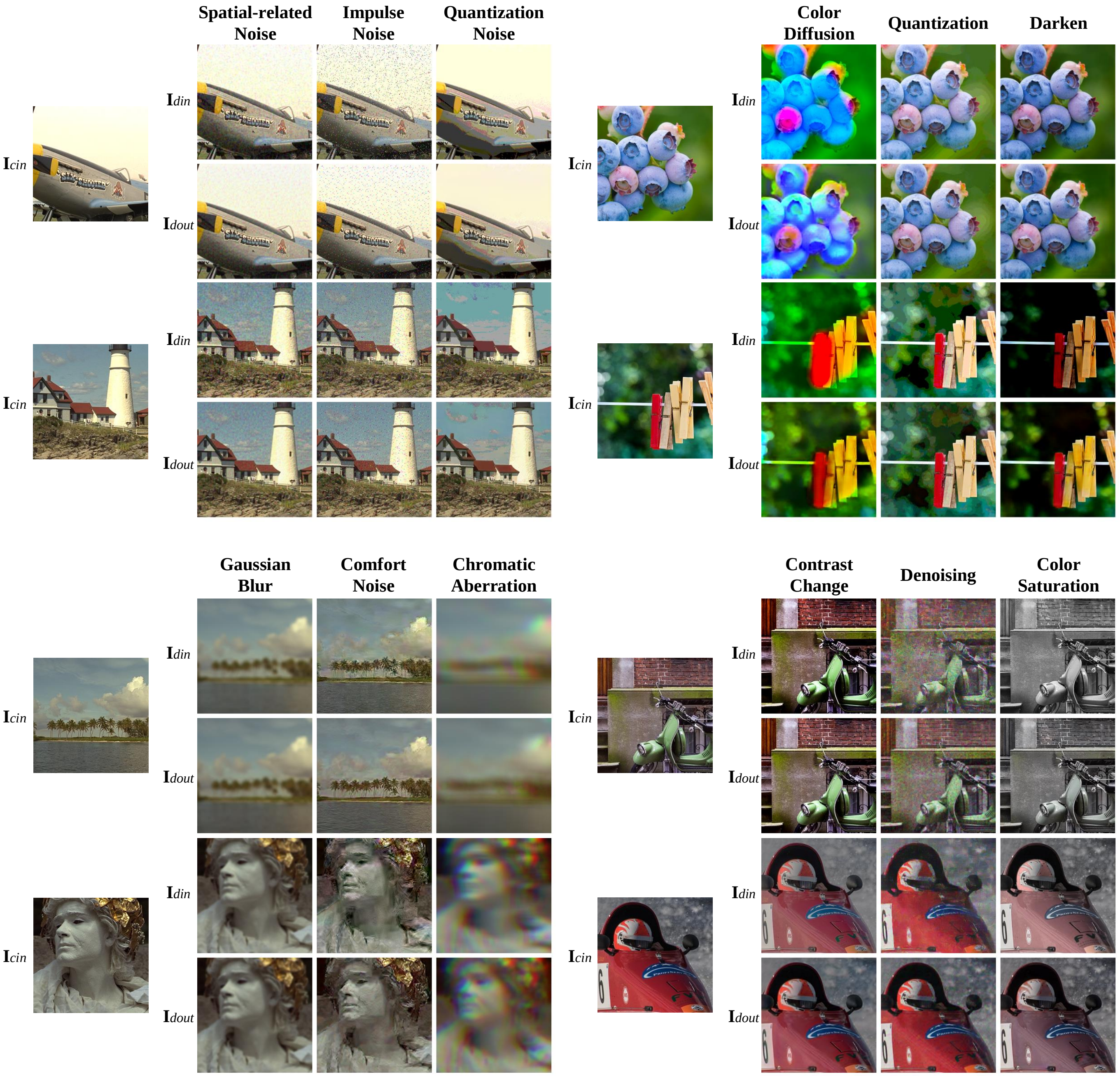


Fig. 1. More examples of the reconstruction of distorted images in COAE. $\mathbf{I}_{cin}$ denotes the input distortion-free image of content autoencoder, while $\mathbf{I}_{din}$ and $\mathbf{I}_{dout}$ respectively denote the input distorted image and output reconstruction image of distortion autoencoder.

(3) SAWAR: The SAWAR consists of the content encoder, the distortion encoder, 1 global average pooling layer *GAP(*1×1*)*, 2 convolutional layers *Conv(1, 2, 1), Conv(1, 256, 1)*, 1 bilinear pooling layer *BiPool(*256×1*)*, and 3 fully connected layers *Linear(512, 256), Linear(256, 256), Linear(256, 1)*.

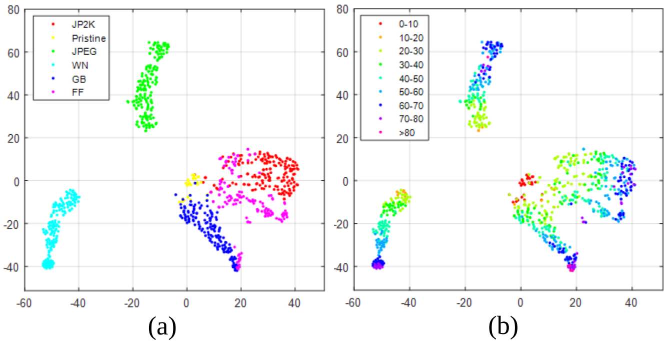
C. The Procedure of Generating Multiply Distorted Images

In Section VI-G of the manuscript, an ablation study is conducted, where multiply distorted images are needed . Here we detail the procedure of generating multiply distorted images. Specifically, four types of distortion which might appear in the stage of image acquisition, transmission, and display, are employed to generate the multiply distorted images, and they are Gaussian blur, contrast change, JPEG compression, and Gaussian noise. The procedure to generate multiply distorted images can be seen in Algorithm 1. The levels for the above distortions are randomly selected from $\{0, 1, 2, 3, 4\}$. The order of introducing distortions and the setting of distortion parameters of different levels follow the suggestions in [71].

III. More Experiments and Experimental Results

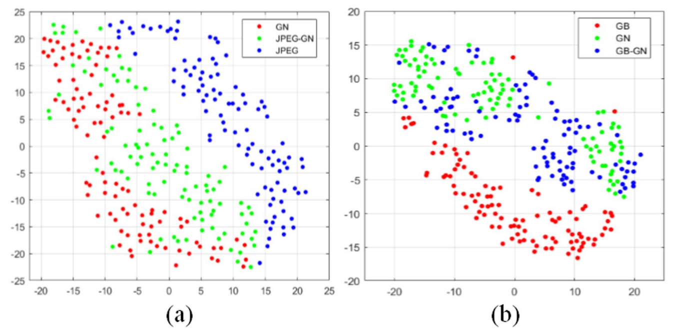
A. Reconstruction Results

A satisfactory reconstruction of distorted images by the DAE implies the distortion encoder has succeed in accom-



(a) (b)

Fig. 2. t-SNE results of the distortion-aware features. (a) Scatter points colorized according to the distortion types of images. (b) Scatter points colorized according to the DMOS of images. The notations are: FF - Fast Fading, GB - Gaussian Blur, WN - White Noise.



(a) (b)

Fig. 3. t-SNE results of the distortion-aware features of single and multiply distorted images. All the scatter points are colorized according to the distortion type. The notations are: GN - Gaussian noise, JPEG - JPEG compression, GB - Gaussian blur.

TABLE I
COMPLETELY BLIND IQA TESTING ON LIVE

LIVE	JP2K	JPEG	GB	WN	FF	ALL
SRCC	0.881	0.885	0.972	0.911	0.840	0.866
PLCC	0.876	0.865	0.973	0.923	0.862	0.851

plishing its duty, i.e., encoding the distortion information. Here we show some examples of the reconstruction results, as shown in Fig. 1. These results show that the COAE can be easily and fully trained, and the distortion decoder can well reconstruct the distorted images. Thus, we can conclude that sufficient distortion information has been captured by the distortion encoder.

B. More Analysis on Distortion Representation

In the manuscript, we have analysed the distortion representation on TID2013. As mentioned in [4], extracting distortion representation via an AE is very challenging, we would like to further analyse the distortion representation in the supplementary materials. Here, the t-SNE is used to visualize the distortion feature vectors. Specifically, distortion feature vectors $\mathbf{f}_d$ are extracted from the distorted images and the reference images in LIVE dataset, and then t-SNE algorithm is utilized to map the feature vectors onto a 2D visualization space. Fig. 2(a) shows the features for different distortions. It is seen that almost all the images are grouped together according to their distortion types. Some images of Fast Fading are grouped with images of Gaussian Blur which is not surprising because they are visually similar. Fig. 2(b) shows the features according to the images' DMOS. It is interesting to observe that pristine images are grouped together in the region near the origin. Besides, images with lower DMOS (higher quality) are closer to this area, whereas those with higher DMOS (lower quality) are further away from the origin.

From the above analysis we can conclude that the distortion feature vectors extracted by the novel COAE framework can effectively represent not only the characteristics of distortions but also the visual quality of the images. To further confirm this, we simply adopt a completely blind quality predictor based on the Euclidean distance of the distortion feature vectors and test it on LIVE [43]. The predicted score $d(z)$ of the z^{th} distorted image is formulated as:

$$d(z) = \|\mathbf{f}_d(z) - \mathbf{f}_p\|_2 \, , \tag{3}$$

where $\mathbf{f}_d(z)$ denotes the distortion feature vector of the z^{th} distorted image, and $\mathbf{f}_p$ is the average of the distortion feature vectors of the 29 reference images in LIVE. The SRCC and PLCC results between the predicted scores and their corresponding MOS are shown in Table I. It is seen that scores of such a simple predictor have very high correlation with the MOS demonstrating distortion feature vectors are highly related to image qualities, even though the features are never guided by any subjective annotation.

In this work, the COAE is trained with large-scale single synthetic distorted images, yet shows competitive performance on the unseen distortions, such as realistic distortions. To explore the representation ability of the distortion-aware feature when training only on the single synthetic distorted images, we conduct another t-SNE visualization experiment. Specifically, two sets of images are collected. In the first set, there are 300 distorted images with the Gaussian noise (GN) distortion, the JPEG compression distortion, and a multiple distortion included both of them (JPEG-GN). In the second set, there are also 300 distorted images with the Gaussian blur (GB) distortion, the Gaussian noise (GN) distortion, and a multiple distortion included both of them (GB-GN). The t-SNE visualization results are shown in Fig. 3, where the scatter points are colorized according to their distortion types. From Fig. 3(a) we can see that the scatter points with two single distortions, GN and JPEG, well distinguish from each other. For the multiply distortion JPEG-GN, the scatter points distribute between the GN points and the JPEG points. It means the COAE regards the JPEG-GN feature as a combine of JPEG feature and GN feature. Thus, the proposed method is able to work well on multiply distorted images. It is worthwhile to notice that, in Fig. 3(a), we do not re-train the COAE on multiply distorted images. From the results in Fig. 3(b), we can draw a similar conclusion. The results in Fig. 3 show that although only a variety of single distorted data are used for training at present, it has shown generalization ability with great potential to handle mixed distortions.

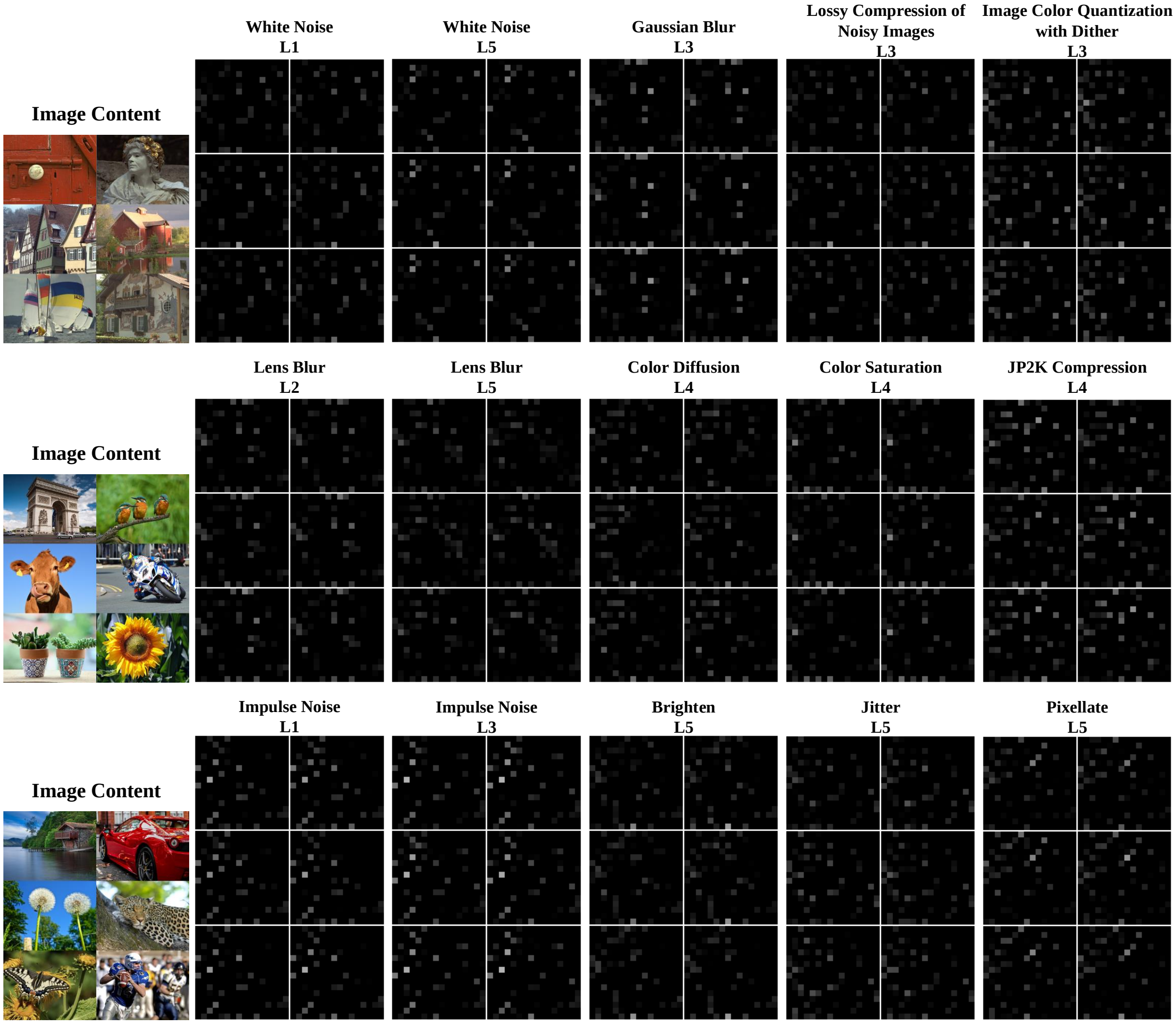


Fig. 4. More examples of the reshaped maps of the extracted distortion-aware features on TID2013 and KADID-10K.

Besides, Fig. 4 and Fig. 5 shows more examples of the distorted images and their reshaped maps of the distortion-aware features. In Fig. 4, we can see the synthetic distortions that never appear in the COAE training images, e.g., jitter, can still be well represented and distinguished from other distortions. In Fig. 5 we can see that although the images in the same group are with different contents, their feature maps are quite similar since they are affected by similar distortion. We attribute this to the large-scale training of the COAE with distorted images under diverse degradations, which brings strong generalization ability to the distortion representation.

C. More Cross-dataset Evaluations

To further analyze the generalization ability of the proposed SAWAR, more cross-dataset evaluations are conducted. All the compared methods, BRISQUE [4], NFERM [8], WaDIQaM [15], DBCNN [31], P2P-BM [47], UNIQUE [21], and VCRNet [38], are trained on one specific dataset and tested on different datasets. Six IQA datasets mentioned in the manuscript, LIVE [41], CSIQ [43], TID2013 [44], KADID-10K [45], LIVEC [42], and KonIQ-10K [46], are utilized as benchmark datasets. The SRCC results are listed in Table II. As we can see, SAWAR achieves 11 of 14 best and second-best results on the synthetic and authentic datasets compared with others. In 5 of 6 cases when conducting cross-dataset evaluation between synthetic and authentic datasets, the SAWAR is not the best one. The main reason is that the self-adaptive weighting feature, i.e., the interaction representation f_w, learned from an authentic dataset would be not appropriate to balance the content and the distortions in a synthetic dataset, and vice versa (See Section IV-E in the manuscript). Thus, to learn a more comprehensive interaction representation, the SAWAR should be trained on synthetic and authentic datasets simultaneously. This study has been conducted in the Section IV-F of the manuscript, which has demonstrated the remarkable superiority of the SAWAR on both synthetic and authentic datasets, in comparison with state-of-the-art methods.

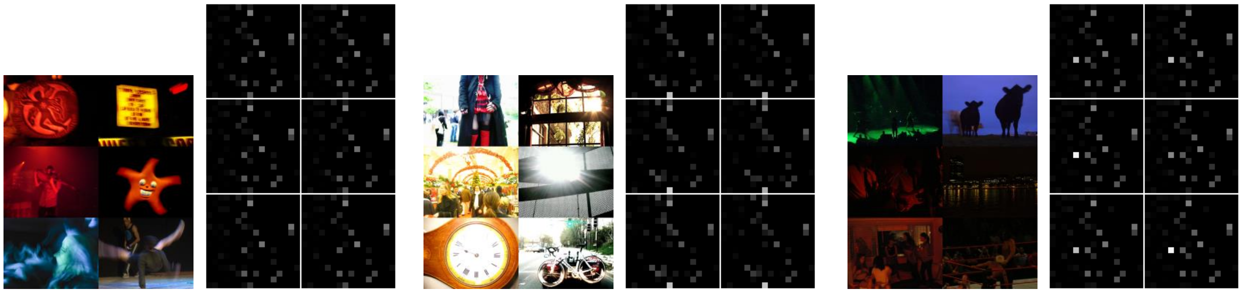

(a) Motion Blur (b) Over-exposure (c) Under-exposure

Fig. 5. More examples of the reshaped maps of the extracted distortion-aware features on KonIQ-10K.

TABLE II
SRCC RESULTS ON CROSS-DATASET EVALUATIONS

Training	LIVE			CSIQ			LIVEC
Testing	CSIQ	TID2013	LIVEC	LIVE	TID2013	LIVEC	KonIQ-10K
BRISQUE [4]	0.513	0.443	0.337	0.776	0.459	0.131	0.587
NFERM [8]	0.595	0.389	0.307	0.794	0.351	0.174	0.579
WaDIQaM [15]	0.704	0.462	0.271	0.859	0.407	0.301	0.622
DBCNN [31]	0.758	0.524	0.567	0.877	0.540	0.452	0.707
P2P-BM [47]	0.776	0.539	0.510	0.916	0.555	0.417	0.696
UNIQUE [21]	0.758	0.563	0.534	0.921	0.566	0.403	0.691
VCRNet [38]	0.768	0.502	**0.615**	0.886	0.542	**0.463**	0.670
SAWAR (ours)	**0.797**	**0.581**	0.573	**0.948**	**0.571**	0.452	**0.723**
Training	TID2013			LIVEC			KonIQ-10K
Testing	LIVE	CSIQ	LIVEC	LIVE	CSIQ	TID2013	LIVEC
BRISQUE [4]	0.814	0.612	0.254	0.440	0.241	0.280	0.525
NFERM [8]	0.634	0.551	0.158	0.510	0.242	0.270	0.493
WaDIQaM [15]	0.792	0.690	0.225	0.492	0.390	0.182	0.646
DBCNN [31]	0.891	0.807	**0.457**	**0.746**	**0.697**	0.424	**0.749**
P2P-BM [47]	0.859	0.706	0.318	0.494	0.503	0.326	0.719
UNIQUE [21]	0.837	0.725	0.329	0.560	0.518	0.308	0.666
VCRNet [38]	0.822	0.721	0.307	**0.746**	0.566	0.416	0.616
SAWAR (ours)	**0.942**	**0.813**	0.411	0.712	0.656	**0.452**	0.739

TABLE III
IMPACTS OF DIFFERENT LOSS FUNCTIONS

	CSIQ		LIVEC	
Loss Function	SRCC	PLCC	SRCC	PLCC
MSE	0.944	0.952	0.836	0.860
MSE+PLCC	0.946	0.957	0.851	**0.872**
MSE+SRCC	0.944	0.953	**0.853**	0.867
MSE+SRCC+PLCC	**0.952**	**0.960**	**0.853**	0.871

TABLE IV
IMPACTS OF DIFFERENT MINI-BATCH SIZE

	CSIQ		LIVEC	
Mini-batch Size	SRCC	PLCC	SRCC	PLCC
4	0.936	0.951	0.831	0.849
8	0.948	0.957	0.852	0.869
16	**0.952**	0.959	**0.853**	**0.871**
32	0.951	**0.960**	0.851	**0.872**
64	**0.952**	**0.960**	**0.853**	**0.871**

D. More Ablation Studies

When training the SAWAR, the MSE loss, PLCC loss, and SRCC loss are employed. Here we conduct an ablation study on the loss function in Eq. (6). Several combinations of the MSE term, the PLCC term, and the SRCC term are tested, and the results are shown in Table Table III. As can be seen, the combination of the three terms achieves the best result. Thus, we include all the three terms when training the SAWAR.

Since the PLCC Loss and the SRCC Loss employed in the manuscript are the list-wise loss function, which might be affected by the size of the mini-batch. Therefore, a total of five different sizes of mini-batch employed in the SAWAR are tested and the results are shown in Table IV. It can be seen that when the mini-batch size is too small (such as 4), the two list-wise losses bring slight changes on the performance. When the mini-batch size becomes larger, the performance improves and tends to be stable.

E. gMAD Competition

To further demonstrate the performance of SAWAR, we conduct the group MAximum Differentiation (gMAD) competition game [70] on the Waterloo Exploration Database [55]. The gMAD competition is a methodology for IQA model comparison on large-scale unannotated datasets, whose main idea is to disprove the quality predictor instead of proving it.

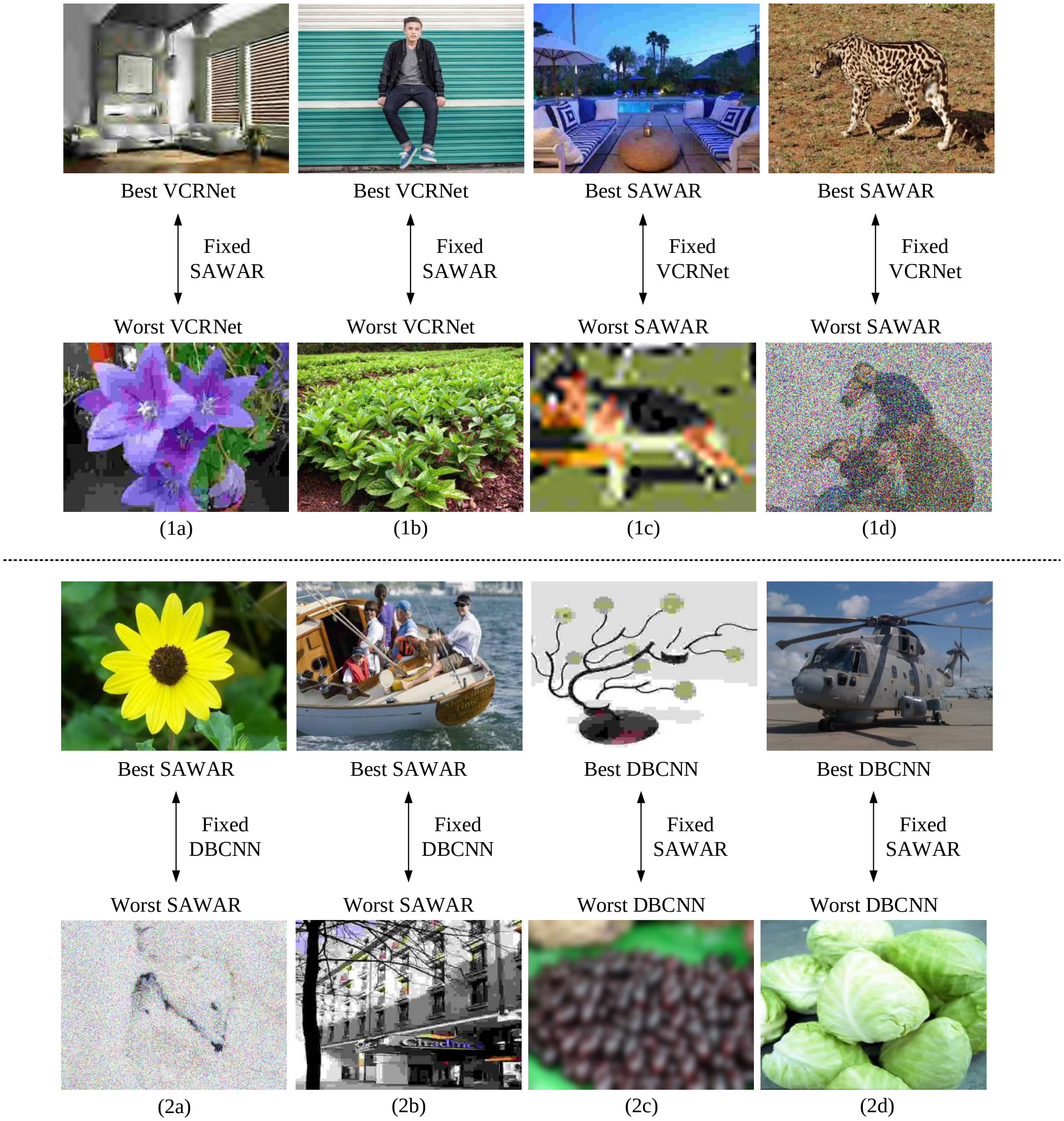


Fig. 6. The upper part is the gMAD competition between SAWAR and VCRNet. (1a) Fixed SAWAR at the low-quality level. (1b) Fixed SAWAR at the high-quality level. (1c) Fixed VCRNet at the low-quality level. (1d) Fixed VCRNet at the high-quality level. The lower part is the gMAD competition between SAWAR and DBCNN. (2a) Fixed DBCNN at the low-quality level. (2b) Fixed DBCNN at the high-quality level. (2c) Fixed SAWAR at the low-quality level. (2d) Fixed SAWAR at the high-quality level.

The harder it is to disprove a model, the better its performance will be. There are two models acting as two roles in the gMAD competition: the defender and the attacker. A group of images with similar quality is first collected by the defender according to the defender's predicted scores, and then the attacker seeks a pair of images with the best quality and the worst quality in the group according to the attacker's predicted scores. If the qualities of the two images are easy to distinguish, the attacker successfully disprove the defender. Otherwise, the attacker fails. After that, the defender and the attacker switch their role and repeat the above competition.

Two state-of-the-art BIQA methods, the DBCNN and VCRNet, are compared with the SAWAR. We first compete SAWAR with VCRNet trained on the entire KADID10K, and the results are shown in the upper part of Fig. 6. In Fig. 6 (1a)-(1b), SAWAR acts as the defender while VCRNet is the attacker. As can be seen, both picked images have similar visual quality, either in the low-quality group or the high-quality group, which indicates that VCRNet fails to disprove SAWAR. In Fig. 6 (1c)-(1d), the roles of SAWAR and VCRNet are exchanged. It is obvious that SAWAR defeats VCRNet by finding inconsistent quality of the two images in both (1c) and (1d). The second competition is conducted between DBCNN

TABLE V
MODEL SIZE AND RUNNING TIME OF DIFFERENT METHODS

Methods	Model Size (Mb)	Running Time (s/image)
WaDIQaM [15]	5.24	0.0467
DIQA [26]	0.41	0.0143
DBCNN [31]	15.31	0.0390
P2P-BM [47]	11.70	0.0190
MetaIQA [22]	13.24	0.0246
UNIQUE [21]	22.06	0.0420
VCRNet [38]	16.66	0.0687
SAWAR (ours)	6.91	0.0352

and SAWAR, as shown in the lower part of Fig. 6. In Fig. 6 (2a)-(2b), DBCNN acts as the defender while SAWAR is the attacker. The quality of the two images in Fig. 6 (2a) and (2b) is distinguished, thus SAWAR wins. In Fig. 6 (2c)-(2d), when SAWAR is the defender and DBCNN is the attacker, DBCNN fails to disprove SAWAR since the two pairs of images in Fig. 6 (2c) and (2d) both have similar visual qualities. Therefore, we can conclude that the SAWAR achieves remarkable superiority to the most competitive methods (DBCNN and VCRNet) in terms of gMAD competition.

F. Model Size and Running Time

The model size and running time of the SAWAR and compared methods are listed in Table V. All the models are tested on a machine with processor Intel Xeon(R) CPU E5-1630 v4 @ 3.70GHz x8 and GPU GeForce GTX 1080Ti. Test images are from TID2013. It can be seen that the proposed SAWAR is with similar level in the model size and prediction time, compared with most deep learning methods.