Blindly Assess Image Quality in the Wild Guided by A Self-Adaptive Hyper Network – Supplementary Material

Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, Yanning Zhang
School of Computer Science and Engineering, Northwestern Polytechnical University
https://github.com/SSL92/hyperIQA

Overview

In this supplementary material, we show more implementation details and experimental comparisons with the state-of-the-art methods.

1. More Details about Implementation

In the main paper, we have shown the overall architecture of our proposed network. In this section, we show in Table 1 more detailed descriptions of each operation in our network, including source of input, output parameter setting and convolution kernel size. In Table 1, LDA stands for local distortion aware module output, FC means fully-connection, and GAP is the abbreviation for global average pooling. FC1, FC2, FC3, FC4 refers to four fully connected layers of the target quality prediction network. The parameters delivered to the target quality prediction network are shown in bolder in Table 1, including target network input, fully connected layer weights and biases. Output size is shown in the order of $Channels \times Height \times Width$, note that in our model, the number of channels are set as the multiple of 7 for size match, as our backbone network ResNet50 receives the input image of size 224 $\times$ 224 pixels and extracts deep features with size 7 $\times$ 7.

Module	Operation	Input	Output Size	Ker_size
Multi-scale Content Feature Extraction	Local distortion aware Conv_1	ResNet Stage 1	16 × 1 × 1	1
	Local distortion aware Conv_2	ResNet Stage 2	16 × 1 × 1	1
	Local distortion aware Conv_3	ResNet Stage 3	16 × 1 × 1	1
	Global Average Pooling	ResNet Stage 4	176 × 1 × 1	—
	Concatenation	LDA_1, LDA_2, LDA_3, GAP	**224 × 1 × 1**	—
Hyper Network	Channel-wise Down-sample_1	ResNet Stage 4	1024 × 7 × 7	1
	Channel-wise Down-sample_2	Channel-wise Down-sample_1	512 × 7 × 7	1
	Channel-wise Down-sample_3	Channel-wise Down-sample_2	112 × 7 × 7	1
	FC_Bias_GAP	Channel-wise Down-sample_3	112 × 1 × 1	—
	FC1_Weight_Conv	Channel-wise Down-sample_3	512 × 7 × 7	3
	FC1_Werght_Reshape	FC1_Weight_Conv	**224 × 112**	—
	FC1_Bias_FC	FC_Bias_GAP	**112 × 1 × 1**	—
	FC2_Weight_Conv	Channel-wise Down-sample_3	128 × 7 × 7	3
	FC2_Werght_Reshape	FC2_Weight_Conv	**112 × 56**	—
	FC2_Bias_FC	FC_Bias_GAP	**56 × 1 × 1**	—
	FC3_Weight_Conv	Channel-wise Down-sample_3	32 × 7 × 7	3
	FC3_Werght_Reshape	FC3_Weight_Conv	**56 × 28**	—
	FC3_Bias_FC	FC_Bias_GAP	**28 × 1 × 1**	—
	FC4_Weight_FC	FC_Bias_GAP	**28 × 1 × 1**	—
	FC4_Bias_FC	FC_Bias_GAP	**1 × 1 × 1**	—

Table 1. Implementation details of our proposed network.

2. SRCC Test Results on TID2013

When testing our model on single IQA databases, we didn't present results on the database TID2013 [8] because of limited space. However, our approach actually outperforms the rest state-of-the-art methods on TID2013 and we show the results in Table 2. Both individual distortion types and overall performance are shown in the table.

Methods	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13
BRISQUE [7]	0.711	0.432	0.746	0.252	0.842	0.765	0.662	0.871	0.612	0.764	0.745	0.301	0.748
GM-LOG [9]	0.766	0.560	0.782	0.577	0.900	0.738	**0.832**	**0.896**	0.709	0.844	0.885	0.375	0.718
FRIQUEE [2]	0.730	0.573	0.866	0.345	0.847	0.730	0.764	0.881	0.839	0.813	0.831	0.498	0.660
HOSA [10]	0.833	0.551	0.842	0.468	**0.897**	**0.809**	0.815	0.883	0.854	0.891	**0.919**	0.730	0.710
rankIQA [3]	0.667	0.620	0.821	0.365	0.760	0.736	0.783	0.809	0.767	0.866	0.878	0.704	0.810
DBCNN [12]	**0.790**	**0.700**	0.826	**0.646**	0.879	0.708	0.825	0.859	**0.865**	**0.894**	0.916	**0.772**	0.822
Ours	0.769	0.613	**0.918**	0.448	0.839	0.758	0.828	0.873	0.804	0.860	0.888	0.723	**0.846**

Methods	#14	#15	#16	#17	#18	#19	#20	#21	#22	#23	#24	**ALL**
BRISQUE [7]	0.269	0.207	0.219	-0.001	0.003	0.717	0.196	0.609	0.831	0.615	0.807	0.604
GM-LOG [9]	0.173	0.379	0.119	0.155	-0.199	0.738	0.353	0.692	**0.908**	0.570	0.893	0.689
FRIQUEE [2]	0.076	0.032	0.254	0.585	0.589	0.704	0.318	0.641	0.768	0.737	0.891	0.680
HOSA [10]	0.242	0.268	0.211	0.362	0.045	**0.768**	0.622	0.838	0.896	**0.753**	0.909	0.735
rankIQA [3]	**0.512**	**0.622**	0.268	0.613	0.662	0.619	0.644	0.800	0.779	0.629	0.859	0.780
DBCNN [12]	0.270	0.444	-0.009	0.548	0.631	0.711	0.752	0.860	0.833	0.732	0.902	0.816
Ours	0.369	0.428	**0.424**	**0.740**	**0.710**	0.767	**0.786**	**0.879**	0.785	0.739	**0.910**	**0.822**

Table 2. SRCC comparisons on individual distortion types on the TID2013 Database.

As shown in Table 2, out approach outperforms competing algorithms and achieves the highest hit count (8 times) on individual distortion types against the rest approaches (DBCNN [12] ranks in the second place for 6 times). Among all the distortion types, we found our approach significantly outperforms the rest methods on distortion # 16 Mean Shift (intensity shift) and # 17 Contrast Change, this is probably because the change of intensity and contrast disturbs our network for content understanding, and thus extracts intensity and contrast aware features in the target network.

3. gMAD Competition Results with the State-of-the-art Methods

In the main paper, we showed gMAD competing results against two BIQA models, in this section, we present more generalization test results from gMAD [6] competition on Waterloo database for a direct visualization. gMAD efficiently selects image pairs with maximum quality difference predicted by one IQA model to challenge an other model. Specifically, fixing quality level predicted by one IQA method (defender), another IQA method (attacker) calculates and picks an image pair with the max quality difference. The image pair is presented to observer for determining whether the attack is aggressive or not. In Figure 1 - Figure 6, we present gMAD results with six state-of-the-art IQA methods, including two NSS based approaches BRISQUE [7], GM-LOG [9] and four deep learning based approaches deepIQA [1], MEON [5], Yan *et al.*'s method [11] and DBCNN [12]. In the first two columns, our model is fixed as defender, image pairs selected from a bad quality level and a good quality level are presented respectively. In the last two columns, our model attacks other competing methods. Similarly, each column represents image pairs selected from a bad and a good quality level predicted by the defender.

For a good defender, image pairs in the first column are supposed to have both low quality, and possess similar high quality in the second column. While for a good attacker, the top image in the third row ought to exhibit high quality and a low quality image should be picked at the bottom in the fourth row. In gMAD competition results, we found this holds true for most of the image pairs, indicating our model is powerful in defending and holds strong aggressiveness to other models.

Note that when our model plays as attacker, some of the selected best quality images from the low quality level (the top image in the third column) contain either smooth area or flat color, i.e. the "blue sky" image when attack against GM-LOG [9], the "mountain" image when attack against deep-IQA [1], and the "sunset" image when attack against DBCNN [12], while they indeed represent fairly good image quality. This further verifies the effectiveness that since our model learns image semantic before quality prediction, it can not be easily deceived by image contents which are confusing to other IQA models.

Figure 1. gMAD competition results between the proposed method and BRISQUE [7].

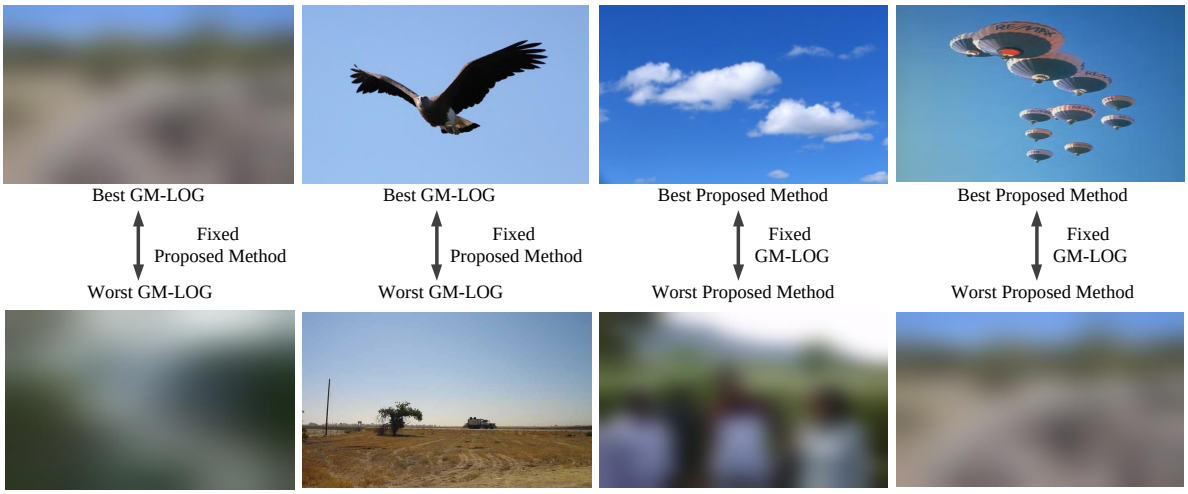


Figure 2. gMAD competition results between the proposed method and GM-LOG [9].

Figure 3. gMAD competition results between the proposed method and deepIQA [1].

Figure 4. gMAD competition results between the proposed method and MEON [5].

Figure 5. gMAD competition results between the proposed method and Yan's method [11].

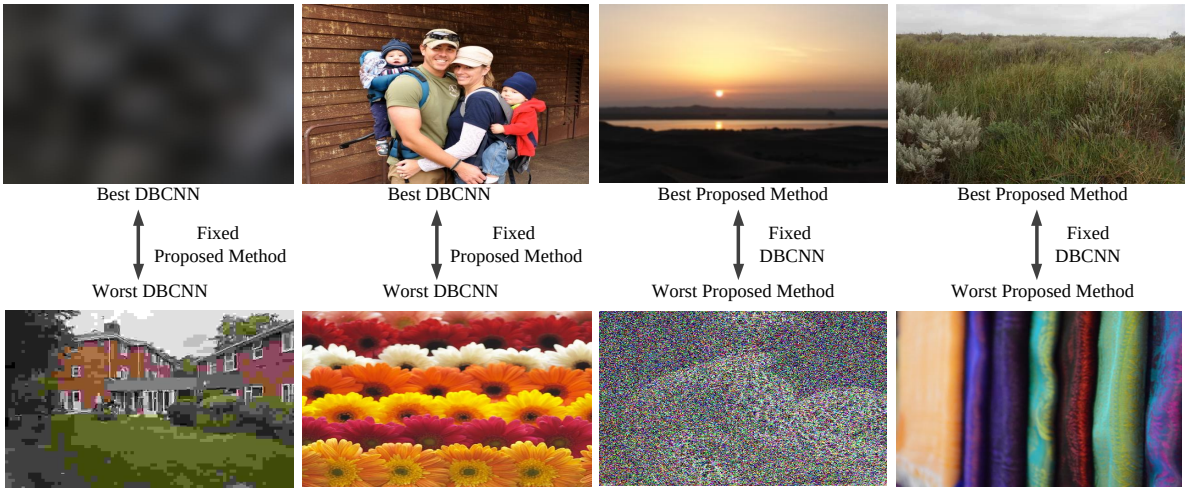


Figure 6. gMAD competition results between the proposed method and DBCNN [12].

4. More Visualization Results on Generated Weights

In the main paper, we only visualize one group of generated target network weights due to limited page. In this section, we exhibit more examples to show that our proposed network successfully separate the task of content understanding with quality prediction and thus conduct quality assessment in a self-adaptive manner. In each figure, we plot four subfigures to show generated weights of four fully connected layers. Still, PCA is used to reduce the dimension of generated weights and we plot them in a 3D space. As shown in Figure 7 to 10, same patterns can be found all across the sub-layers of target network, as images with the same content, despite of different quality level, generate similar weights, indicating same quality prediction rules are utilized for same objects.

References

[1] Sebastian Bosse, Dominique Maniry, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Transactions on Image Processing*, 27(1):206–219, 2017. 2, 3

[2] Deepti Ghadiyaram and Alan C Bovik. Perceptual quality prediction on authentically distorted images using a bag of features approach. *Journal of vision*, 17(1):32–32, 2017. 2

[3] Xialei Liu, Joost van de Weijer, and Andrew D Bagdanov. Rankiqa: Learning from rankings for no-reference image quality assessment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1040–1049, 2017. 2

[4] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016.

[5] Kede Ma, Wentao Liu, Kai Zhang, Zhengfang Duanmu, Zhou Wang, and Wangmeng Zuo. End-to-end blind image quality assessment using deep neural networks. *IEEE Transactions on Image Processing*, 27(3):1202–1213, 2017. 2, 4

[6] Kede Ma, Qingbo Wu, Zhou Wang, Zhengfang Duanmu, Hongwei Yong, Hongliang Li, and Lei Zhang. Group mad competition-a new methodology to compare objective image quality models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1664–1673, 2016. 2

[7] A Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 21(12):4695, 2012. 2, 3

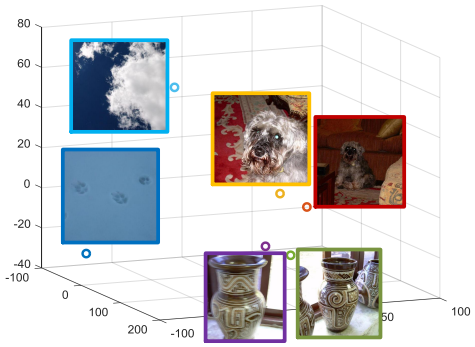
[8] Nikolay Ponomarenko, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Lina Jin, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, and Federica Battisti. Color image database tid2013: Peculiarities and preliminary results. In *European Workshop on Visual Information Processing*, 2013. 2

[9] Xue Wufeng, Mou Xuanqin, Zhang Lei, Alan C Bovik, and Feng Xiangchu. Blind image quality assessment using joint statistics of gradient magnitude and laplacian features. *IEEE Trans Image Process*, 23(11):4850–4862, 2014. 2, 3

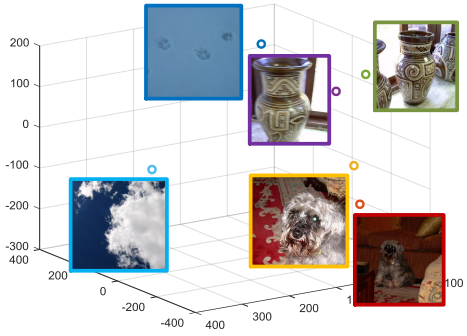
[10] Jingtao Xu. Blind image quality assessment based on high order statistics aggregation. *IEEE Transactions on Image Processing*, 25(9):4444–4457, 2016. 2

[11] Qingsen Yan, Dong Gong, and Yanning Zhang. Two-stream convolutional networks for blind image quality assessment. *IEEE Transactions on Image Processing*, 28(5):2200–2211, 2018. 2, 4

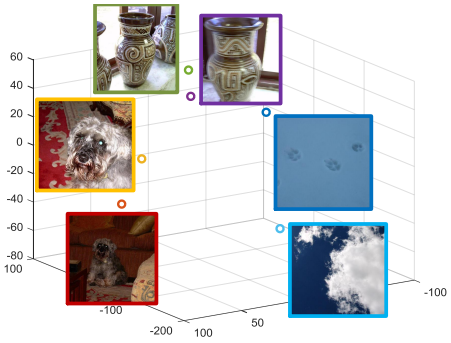
[12] Weixia Zhang, Kede Ma, Jia Yan, Dexiang Deng, and Zhou Wang. Blind image quality assessment using a deep bilinear convolutional neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 2, 4



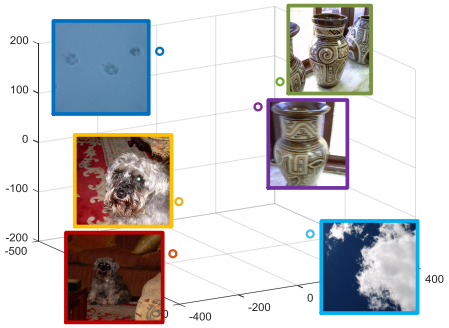
(a) Target network: layer 1



(b) Target network: layer 2

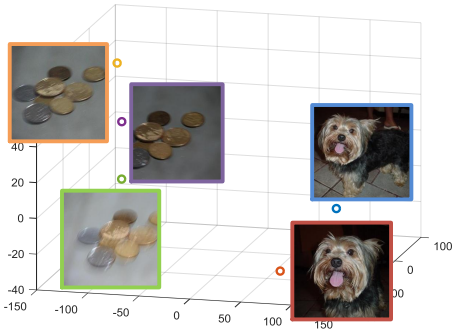


(c) Target network: layer 3

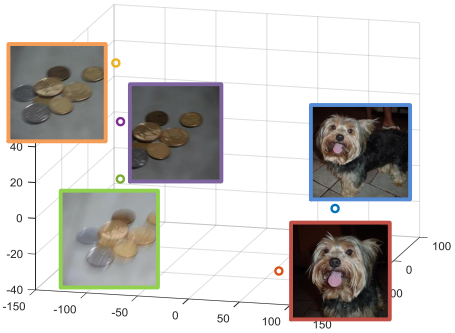


(d) Target network: layer 4

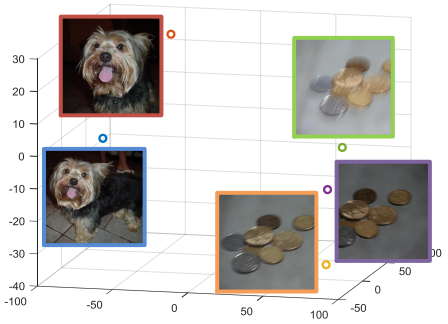
Figure 7. Generated weights of pictures vase, dog, footprint and sky.



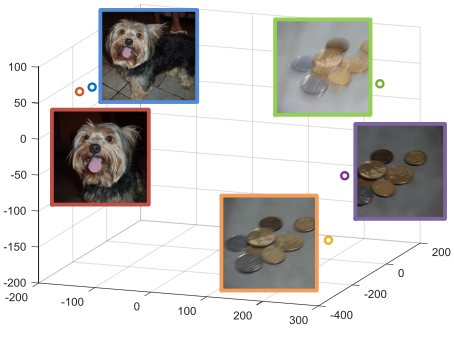
(a) Target network: layer 1



(b) Target network: layer 2

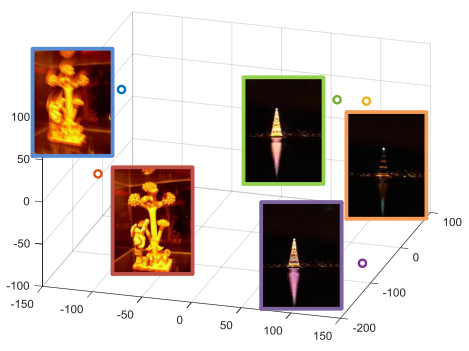


(c) Target network: layer 3

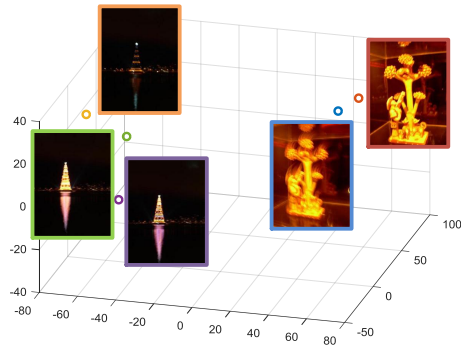


(d) Target network: layer 4

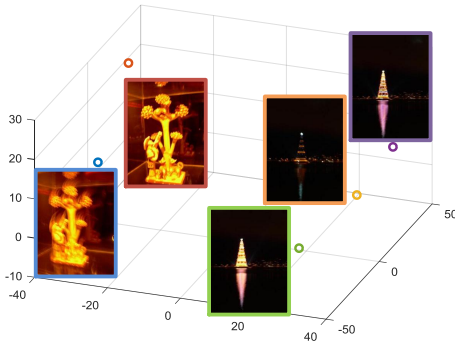
Figure 8. Generated weights of pictures dog and coins.



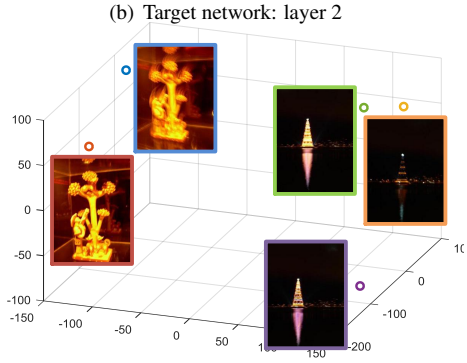
(a) Target network: layer 1



(b) Target network: layer 2

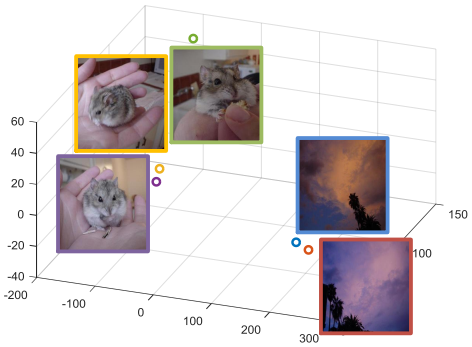


(c) Target network: layer 3

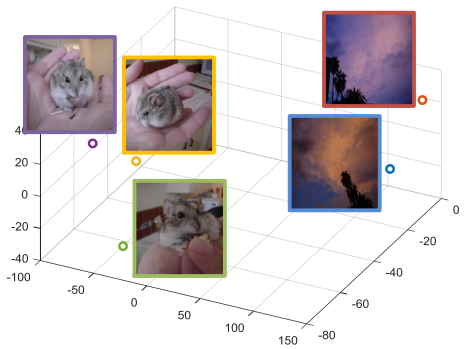


(d) Target network: layer 4

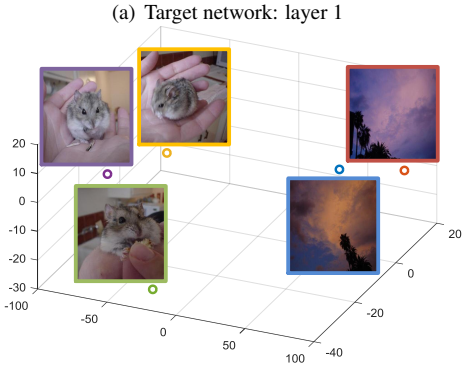
Figure 9. Generated weights of pictures sculpture and tower.



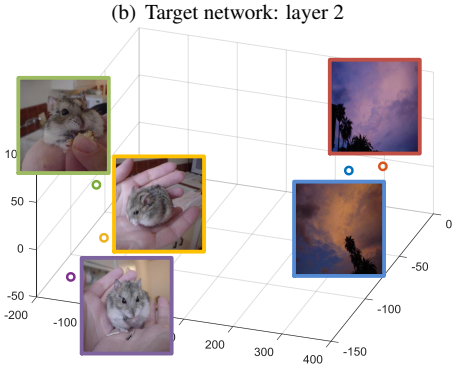
(a) Target network: layer 1



(b) Target network: layer 2



(c) Target network: layer 3



(d) Target network: layer 4

Figure 10. Generated weights of pictures rat and sky.