GraphIQA: Learning Distortion Graph Representations for Blind Image Quality Assessment

Simeng Sun*, Tao Yu*, Jiahua Xu, Jianxin Lin, Wei Zhou and Zhibo Chen, *Senior Member, IEEE,*

***Abstract*—Learning-based blind image quality assessment (BIQA) methods have recently drawn much attention for their superior performance compared to traditional methods. However, most of them do not effectively leverage the relationship between distortion-related factors, showing limited feature representation capacity. In this paper, we show that human perceptual quality is highly correlated with distortion type and degree, and is biased by image content in most IQA systems. Based on this observation, we propose a Distortion Graph Representation (DGR) learning framework for IQA, called GraphIQA. In GraphIQA, each distortion is represented as a graph i.e. DGR. One can distinguish distortion types by comparing different DGRs, and predict image quality by learning the relationship between different distortion degrees in DGR. Specifically, we develop two sub-networks to learn the DGRs: a) Type Discrimination Network (TDN) that embeds DGR into a compact code for better discriminating distortion types; b) Fuzzy Prediction Network (FPN) that extracts the distributional characteristics of the samples in DGR and predicts fuzzy degrees based on a Gaussian prior. Experiments show that our GraphIQA achieves the state-of-the-art performance on many benchmark datasets of both synthetic and authentic distortion. The code is available at https://github.com/geekyutao/GraphIQA.**

***Index Terms*—image quality assessment, graph representation learning.**

I. Introduction

With the rapid development of social networks, a massive amount of digital images have been produced. They could be distorted in any stage of the whole media technical chain, from acquisition, processing, compression to transmission and consumption. Therefore, a reliable image quality assessment (IQA) metric is critical for measuring multimedia model results and guiding its optimization.

Existing IQA methods can be divided into distortion-specific and general-purpose methods. The former can achieve better performance for specific distortion, but their application scope is limited as the distortion tends to be unknown in real-world applications. Therefore, how to improve the performance of general-purpose methods has become particularly important.

Recently, learning-based BIQA methods have drawn much attention for their superior prediction performance. However, most of them do not effectively leverage the relationship between distortion-related factors, showing limited feature representation capacity. We notice that various distortions have different effects on the perceptual image quality, which may imply why general-purpose methods are often inferior to distortion-specific methods [1], [2]. To further investigate the concrete manifestations of the effects of various distortions, we choose a large-scaled IQA dataset and perform statistical analysis on it, which will be illustrated in detail in Section III-A. We observe that: 1) subjective quality ratings are highly affected by distortion-related factors such as distortion type and degree (which is also expressed as level); 2) the level distribution varies greatly with distortion; 3) subjective quality ratings are also biased by image content, which basically obeys a certain Gaussian distribution given the type and level. Motivated by the observations, we integrate the graph representation learning method to learn the characteristic of distortions, and model the relationship of distortion-related factors. The learned representations with plentiful distortion prior can significantly improve the performance of IQA.

In this paper, we propose GraphIQA to model the distribution of distorted examples for building the distortion graph representation (DGR) for a specific distortion. In each DGR, the nodes represent the feature of samples and the edges illustrate their correlation. The core idea of GraphIQA is

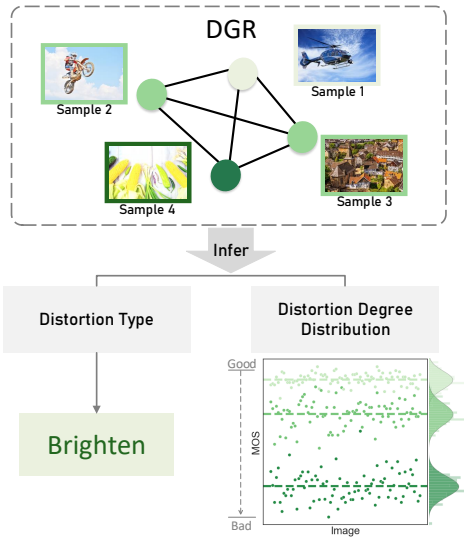


Fig. 1: The core idea of GraphIQA. We develop DGR to represent each distortion. The DGR can be utilized to infer distortion type and degree based on its internal structure. The DGR learning also considers the rating deviation of image content for better prediction. The learned DGRs with plentiful distortion prior can help improve IQA accuracy.

* Equal contribution.
Simeng Sun, Tao Yu, Jiahua Xu, and Zhibo Chen are with the Department of Electronic Engineer and Information Science, University of Science and Technology of China, Hefei, Anhui, 230026, China (e-mail: smsun20@mail.ustc.edu.cn; yutao666@mail.ustc.edu.cn; xujiahua@mail.ustc.edu.cn; linjx@mail.ustc.edu.cn; weichou@mail.ustc.edu.cn; chenzhibo@ustc.edu.cn). Corresponding Author: Zhibo Chen.

shown in Figure1.This DGR is constructed from two aspects: (a) distinguish the distortion type by contrasting the DGRs of different distortions; (b) predict the most likely distortion level of a distorted image according to the internal topological relationship in each DGR. We design Type Discrimination Network (TDN) and Fuzzy Prediction Network (FPN) to learn the DGRs respectively. In detail, the TDN encodes DGR to a low-dimensional code to distinguish distortion types by aggregating the global information of nodes and the relationship between them. Specifically, it achieve the ability of discriminating distortion type by performing a triplet loss [3] on the extracted code. The FPN extracts the distributional characteristics of the samples in DGR, and predicts fuzzy levels based on a Gaussian prior since the subjective quality ratings are often biased by image content. In the end, as the learned DGRs can model the relationship between perceptual image quality and distortion-related factors, GraphIQA can achieve the state-of-the-art on most typical IQA datasets (*e.g.*, KonIQ-10k [4] or LIVE [5]). Note that the training cost of DGRs is very low as it only requires weak supervision, i.e. distortion type and level. Our contributions can be summarized as follows:

- We investigate the inherent relationship between perceptual image quality and distortion, in which we observe that subjective quality ratings are highly related to distortion-related factors and biased by image content.
- We propose GraphIQA to learn the relationship of distortion-related factors via graph representation learning. We develop Type Discrimination Network (TDN) and Fuzzy Prediction Network (FPN) to learn the proposed
- Experiments show that the learned DGRs help GraphIQA achieve the state-of-the-art performance on various IQA datasets of both synthetic and authentic distortion.

II. Related Work

A. Blind Image Quality Assessment

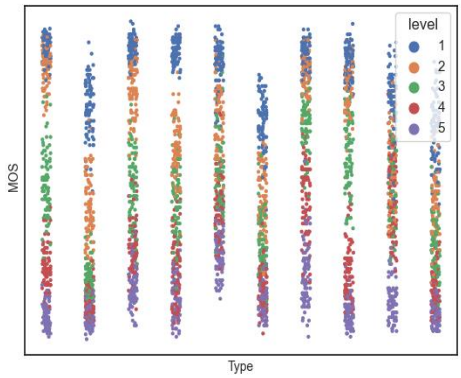
Blind Image Quality Assessment(BIQA) can be categorized into distortion-specific methods [6]–[9] and general-purpose algorithms [10]–[19]. The distortion-specific BIQA methods are favored for their higher accuracy and robustness, when distortion types or distortion process is already known. However, their application scope is limited, as the authentic distortion dataset is mixed with complex distortions and the type of distortion is not clearly specified [20], [21]. Therefore, the research on general-purpose methods has become particularly important and received extensive attention recently. Natural scene statistics (NSS) is one of the powerful tools for general-purpose BIQA, as quality degradations can cause deviation from the originally statistical properties of natural scene images [10]–[12], [22]. For example, Saad *et al.* [10] leverage the statistics of local DCT coefficients as the feature for image quality assessment, while Moorthy *et al.* [23] leverage the feature obtained from the wavelet transform. To simplify the process of feature extraction, Mittal *et al.* [11] propose the method using the NSS in the spatial domain directly. And Zhang *et al.* [12] leverage not only the statistics of the mean subtracted contrast normalized coefficients, but also the statistics of gradients.

Recently, benefit from its ability to efficiently and adaptively extract feature, the deep learning-based general-purpose BIQA methods have drawn considerable attention. Kim *et al.* [24] propose an efficient approach and prove that using backbone pretrained on large classification dataset ImageNet [25] can improve the performance of IQA. Based on this, Talebi *et al.* [26] propose a DCNNs-based model to predict the perceptual distribution of IQA scores instead of the mean value. Similarly, Zeng *et al.* [27] propose the probabilistic quality representation to describe the image subjective score distribution. Noticing that, in synthetic distortion data, effectively utilizing distortion-related information can enhance the representation ability of the IQA model, Kang *et al.* [28] introduce a compact multi-task network into IQA in which type identification task and IQA share all the internal structure. Ma *et al.* [29] introduce a two-step training strategy, which is first training a distortion type identification sub-network, and then adding second sub-network for IQA task. Though multi-task related method has brought progress on IQA, this is difficult to be utilized on authentically distorted dataset, as we cannot obtain accurate ground truth for type identification task. For better performance on both synthetic and authentic distortion, Zhang *et al.* [30] combine two sub-network, with one extracts features to represent synthetic distortion and another extracts semantic features, and then both of them are fused by bilinear pooling to predict the subjective quality score.

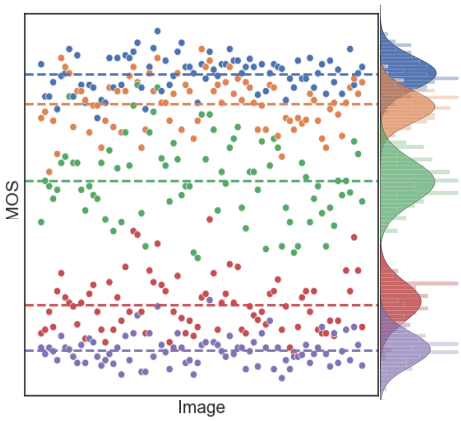
Here, based on above methods, we raise the following questions: 1) is the type identification task the best way to improve representation ability of model? Is there a better choice for sub-task or pre-training strategy? 2) In addition to distortion type there also distortion level and image content that have an impact on perceptual image quality. Can the usage of more distortion related information result in better representation ability of model for IQA? 3) Can the processing of synthetic distortion and authentic distortion be implemented with one module? Therefore, we begin with studying the statistical distribution of IQA dataset and are inspired to integrate graph representation learning method into IQA task, which is proved to efficiently represent the ways how distortion-related factors affect perceptual image quality.

B. Graph Representation Learning

A graph can represent data that are generated from non-Euclidean domains with relationships and inter-dependency between data. The challenge in graph representation learning is finding a way to properly represent, or encode, the graph structure so that it can be easily integrated into the machine learning model. Most of the traditional methods are based on hand-crafted features, such as statistics or kernel functions. Recently, encouraged by the success of CNNs in the computer vision field, a large number of methods that are based on automatically learned low-dimensional embeddings to encode the structure of graphs have been developed. Having the ability of neighborhood aggregation, Graph convolutional



(a)



(b)

Fig. 2: Statistical analysis of Kadid10k database. (a) shows the Type-MOS distribution where different colors denote the distortion level. (b) the detailed distribution of specific distortion type, which obeys the Gaussian-like distribution around their respective mean value.

networks (GCNs) have been successfully applied to plenty of tasks [31], [31]–[34]. Graph attention network (GAT) [35] further integrated masked self-attention mechanism in GCN. Different from the aggregation method of weighted sum in GCN, Hamiton *et al.* [36] proposed GraphSAGE, which introduced an inductive learning mode. By training the model to aggregate neighbor nodes using max-pooling and LSTMs [37], GraphSAGE was extended to inductive learning task, so that it can achieve the generalization for unknown nodes. However, the mentioned methods were based on neighborhood aggregation resulting in the shallow representation of graph, which prevents the model from obtaining adequate global information. Therefore, Hu *et al.* [38] proposed hierarchical graph convolutional network (H-GCN) with graph pooling mechanism to solve the above problem, and showed great improvement. In this paper, from our statistics, it is observed that the effect of distortion on the perceptual image quality is not only manifested in the characteristics of distortion, but also in the distribution of samples at different levels. Therefore, we introduce graph to efficiently represent various distortions, which will be used to improve the performance on IQA task.

III. GraphIQA

A. Motivation

Subjective image quality assessment is commonly obtained by collecting mean opinion scores from many subjects, which is labor-intensive and impractical. Recently, the learning-based methods have drawn much attention as the high efficiency and accuracy. However, most of them pay less attention to the relationship between distortion-related factors, which leads to the limitation of network representation ability for IQA. As widely accepted, human visual system has different sensitivity to different type of distortions [1], [2]. To further investigate the specific manifestation of how perceptual image quality is affected by distortion-related factors, we start from the analysis of IQA datasets. In order to get a more generalized conclusion, our analysis is based on Kadid10k dataset [9], which is a large scale dataset including $10,125$ images with 25 distortion types and 5 distortion levels. As observed from the statistics of Kadid10k that is shown in Figure 2(a), the distortion types are crucial influential factor to the distribution of IQA scores which is consistent with our common knowledge. Meanwhile, the distributions of diverse distortions also have similarity, one of which is shown in Figure 2(b) in detail. IQA scores present a sequential distribution according to different distortion levels, that is, the higher the level (means to be worse distorted) the lower the IQA scores, and for them with the same distortion, the scores tend to cluster together. Furthermore, perceptual image quality is still affected by image content, because under the same type and level, the vibration of scores still exists, and it obeys the Gaussian distribution according to our statistics.

Here, we integrate the graph representation learning method to learn the representation of distortions, named as distortion graph representations (DGRs), as it can simultaneously represent characteristic of each distortion and its internal structure. Correspondingly, we learn DGRs from two aspects, and that will be described in details in next sub-sections.

B. Distortion Graph Representation

We build DGR as shown in Figure 3, whose nodes represent samples, while edges indicate the relationships between each of them. The DGR of distortion k is formulated as $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$, in which the $\mathcal{V}_k$ denotes the set of nodes and the $\mathcal{E}_k$ denotes the set of edges to describe the relationship between nodes. Specifically, the input batch with N samples from the same distortion type is first fed into a CNN backbone such as ResNet50 [39] to obtain the feature set $\mathcal{F}_k = \{f_i | i = 1, 2, ..., N\}$ where $f_i \in \mathbb{R}^C$ and C is the feature dimension. The extracted feature f_i from each sample is used as the initialization of the node in $\mathcal{V}_k$, while the similarity between each node serves as the initialization of the edge in $\mathcal{E}_k$ which is commonly expressed as a 2D adjacency matrix $A_k \in \mathbb{R}^{N \times N}$. However, considering that the relationship between the samples is non-trivial, we expand the 2D adjacency matrix to a 3D adjacency matrix $A_k \in \mathbb{R}^{N \times N \times C_E}$ where the representation of each edge is a vector with dimension size C_E instead of a scalar. To build the DGRs, we design two learnable modules: Node Builder and Edge Builder respectively.

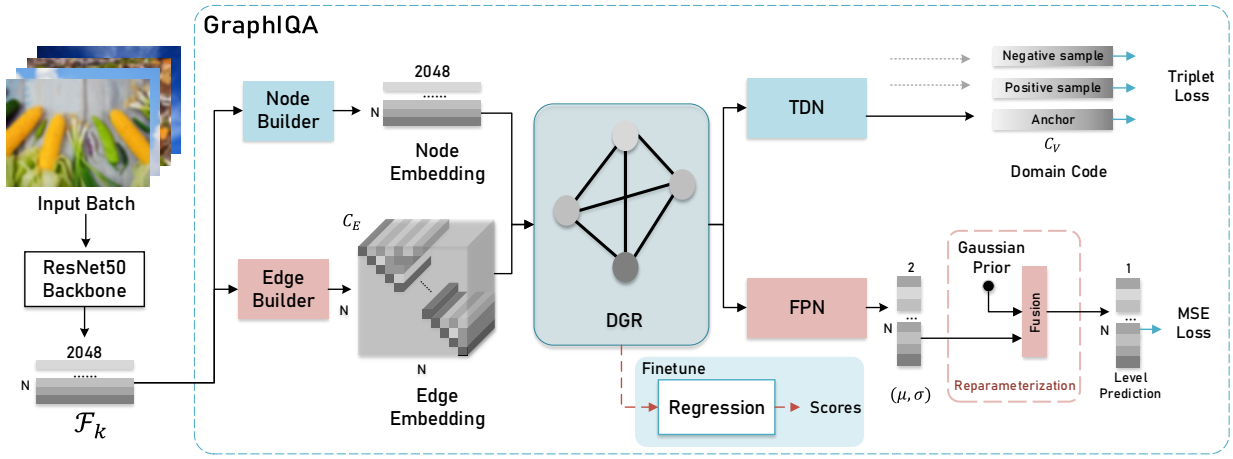


Fig. 3: The illustration of proposed GraphIQA. We first train the networks to learn Distortion Graph Representations (DGRs). The learned DGRs are utilized to improve IQA performance by finetuning the regression network on a target dataset. Note that GraphIQA requires no MOS or DMOS supervision during the first stage.

a) Node Builder (NB): We believe that features extracted from samples of the same distortion tend to have a similar distortion-related part. Therefore, it is expected that the representation of each node will contain more distortion-related information so that it can be further used as evidence to distinguish distortions. Therefore, we use learnable network NB, composed of fully connected layers, to optimize node embedding, which is formulated as

$$V_k = \{v_{k,i} | v_{k,i} = F_{NB}(f_{k,i}; \theta), v_{k,i} \in \mathbb{R}^C, i = 1, 2, ..., N\}, \tag{1}$$

where $v_{k,i}$ denotes the node embedding of i-th sample, and θ denotes the network parameters of NB.

b) Edge Builder (EB): The 3D adjacency matrix should realize the representation of the contrast relationship among the nodes in each DGR, thereby establishing the internal structure of the DGR. Therefore, we take the edge vectors $a^0_{k,i,i} \in \mathbb{R}^C$ as initial edge embedding E^0_k, which is the result of dot multiplication between each node embedding. The edge embedding is further optimized to represent internal structure by graph convolution network (GCN) [40]. In detail, given the edge embedding E^0_k and $A_{E_k} \in \mathbb{R}^{N^2 \times N^2}$ as input, the process of computation of each layer for GCN with L layers can be formulated as:

$$E^{l+1}_k = \sigma(\hat{A}_{E_k} E^l_k W_l), \tag{2}$$

where

$$D = \text{diag}(\sum_{p=1}^{N^2} (A_{E_k} + I)_p), \tag{3}$$

$$\hat{A}_{E_k} = D A_E D. \tag{4}$$

The initialization of edge embedding E^0_k serves as the input of the first layer of edge builder, and E^l_k denotes the output of l-th GCN layer. W_l is the trainable parameter of GCN l-th layer, and σ denotes the no-linear activation function which is ReLU in this paper. In the end, the optimized edge embedding of DGR is defined as:

$$E_k = \{a_{k,i,j} | a_{k,i,j} \in \mathbb{R}^{C_E}, i, j = 1, 2, ..., N\}, \tag{5}$$

where the C_E is the dimension of edge embedding, which is set much smaller than C to reduce computational complexity.

C. Domain Graph Optimization

As shown in Figure 3, to equip the DGRs with the ability of both representing each distortion and the relationship between distortion levels, GraphIQA learns DGRs from the following two aspects. a) To learn the unique representation of a certain distortion type that can be distinguished from other types, we design the TDN; b) To learn the distributional characteristic of distortion levels and content-biased distribution model, we design the FPN.

a) Type Discrimination Network (TDN): TDN is used to obtain the typical compact representation of each DGR, which helps to distinguish it from the others. Specifically, we design a GCN to aggregate global information from node embedding and relationship from edge embedding. The process is formulated as follow:

$$V^{l+1}_k = \sigma(\hat{A}_{V_k} V^l_k W_l), \tag{6}$$

in which the V_k is the node embedding and the $A_{V_k} \in \mathbb{R}^{N \times N}$ is the adjacency matrix of nodes, which is calculated by transforming edge embedding E_k through the average pooling across channels. $\hat{A}_{V_k}$ is defined similar to Equation (4). The output of the TDN will be a vector with dimension C_V, named as code y_{code}. Then triple loss $\mathcal{L}_{dist}$ [3] is utilized to aggregate the anchor DGR and the DGR of the same distortion, while separating it from the DGR of the other distortion,

$$\begin{aligned} \mathcal{L}_{dist} = max(d(y^{Anchor}_{code}, y^{+}_{code}) \\ - d(y^{Anchor}_{code}, y^{-}_{code}) + margin), 0), \end{aligned} \tag{7}$$

where d denotes L2 distance, and y^{+}_{code} is the DGR representing the same type with anchor DGR while y^{-}_{code} is the

DGR representing the different type. Note that applying cross-entropy loss of classification can also distinguish different distortion as [30]. However, triplet loss can maintain a more subtle difference between distortion types. Besides, the triplet only defines the positive label and negative label, which can avoid the increase in computational complexity caused by the last fully connected layer.

b) Fuzzy Predictor Network (FPN): We design an FPN to achieve the uncertain prediction of levels. Even if samples share the same distortion type and level, the perceptual image quality of them still is different due to their different content. According to Figure 2(b), we assume that the vibrations near the mean scores of samples from the same level, obey the Gaussian distribution. Then we achieve the prediction by randomly sampling from the Gaussian prior distribution $\mathcal{N}(\mu, \sigma^2)$. As this process is not differentiable, the reparametrization trick [41] is used to ensure end-to-end training of the network. In detail, the ϵ is sampled from Normal distribution $\mathcal{N}(0, 1)$, and then mapped to an arbitrary Gaussian distribution according to the generated hyper-parameters:

$$y_i = \mu_i + \sigma_i \epsilon, \epsilon \in \mathcal{N}(0, 1), \tag{8}$$

where μ and σ are the predicted mean and scale generated by the hyper predictor, as is shown in Figure 3. Because the prediction of distortion level is not only achieved by analyzing the distortion-related representation of nodes, but also the comparison between nodes to estimate the level of distortion, both the node embedding and edge embedding are needed. Specifically, node embedding V_k is fed into FPN directly, while edge embedding E_k is averaged across the rows, to obtain the relationship between the current node and all the other nodes, as $E'_k = [\sum_j a_{k,i,j}]/N$. Then mean square error (MSE) loss function is utilized when training the hyper predictor:

$$\mathcal{L}_{level} = \sum\nolimits_i |y_i - y'_i|^2, \tag{9}$$

where y'_i denotes the target level. The entire model will be trained end-to-end to minimize the combination of above loss functions, which are weighted by hyper-parameter λ:

$$\mathcal{L} = \mathcal{L}_{dist} + \lambda \mathcal{L}_{level}. \tag{10}$$

D. Finetune and Inference

Benefiting from the improved representation ability of DGRs, GraphIQA shows the potential for better fulfillment of IQA tasks. Specifically, when finetuning on the target dataset, both the node embedding and edge embedding are used to regress the IQA scores. The edge embedding, which is averaged across the rows and then concatenated with node embedding, is fed into the regression module. The regression module is a small and simple network with two fully connected layers. When using DGRs for IQA on authentically distorted datasets, the prediction of authentic distortion datasets is achieved based on the Gaussian prior distribution so that it can better handle the unknown distortion type. Then the entire model is finetuned to minimize the MSE between ground truth

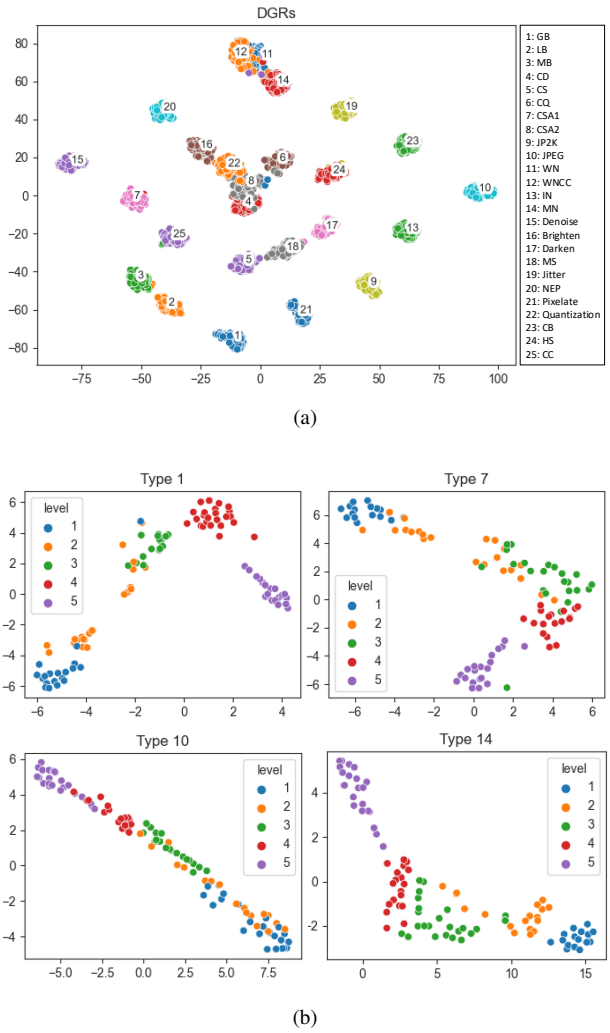


Fig. 4: The visualization of DGRs is shown in (a), and the visualization of the internal structure of DGR of 4 distortion type is shown in (b). More results are shown in our supplementary material.

(which is MOS/DMOS) and predicted scores, which is defined as:

$$\mathcal{L}_{scores} = 1/N_f \sum_{i=1}^{N_f} |c_i - c'_i|^2, \tag{11}$$

where the N_f denotes the mini-batch size. It is worth noting that as it already has the ability to infer the DGR, GraphIQA can support any size of input batch during inference stage.

IV. Experiments

A. Experiments Setting

a) Dataset.: During pretraining, we use Kadid-10k [9] which is a large synthetic distorted database that contains

TABLE I: Evaluation on clustering performance of DGR on Kadid10K dataset.

Type	Homo.	Comp.	V-m.	Type	Homo.	Comp.	V-m.
GB	0.815	0.735	0.773	MN	0.689	0.639	0.663
LB	0.850	0.772	0.809	Denoise	0.803	0.797	0.800
MB	0.664	0.765	0.711	Brighten	0.573	0.511	0.524
CD	0.424	0.449	0.437	Darken	0.401	0.408	0.404
CS	0.271	0.350	0.306	MS	0.213	0.265	0.236
CQ	0.557	0.583	0.570	Jitter	0.706	0.635	0.669
CSA1	0.554	0.571	0.562	NEP	0.193	0.216	0.204
CSA2	0.353	0.373	0.362	Pixelate	0.778	0.709	0.742
JP2K	0.549	0.614	0.580	Quan.	0.373	0.385	0.379
JPEG	0.717	0.659	0.687	CB	0.177	0.329	0.230
WN	0.780	0.706	0.741	HS	0.519	0.534	0.526
WNCC	0.662	0.794	0.772	CC	0.300	0.427	0.452
IN	0.772	0.712	0.741				

81 images with 25 distortion types[1] and 5 distortion levels. For target datasets, we choose two datasets with authentic distortion (KonIQ-10k [4] and LIVE Challenge (LIVEC) [20]) and two datasets with synthetic distortion (LIVE [5] and CSIQ [42]). KonIQ-10k consists of 10073 images which are selected from the large public multimedia database YFCC100m [43]. Those samples try to cover a wide and uniform quality distortion. LIVEC contains 1162 images taken from different photographers with various cameras. LIVE contains 779 images with 5 distortion types and CSIQ contains 866 images with 6 distortion types. When finetuning, for authentic distorted datasets, we randomly split them into a training set and a test set according to the ratio of $8:2$. For synthetic distorted datasets, we randomly split the source images according to the same ratio to avoid content overlapping. All the results are finetuned and tested on datasets with 10 times randomly splitting, and the average results are reported.

b) Evaluation Metrics.: We mainly adopt two commonly used metrics, which are Spearman's rank order correlation coefficient (SRCC) and Pearson's linear correlation coefficient (PLCC) to measure the prediction monotonicity and prediction accuracy. Both of them range from 0 to 1 and a higher value indicates better performance.

c) Implementation Details.: We implement our model by PyTorch, and both training and testing are conducted on the NVIDIA 1080Ti GPUs. For data augmentation, when pretraining the GraphIQA model, we randomly sample from each distortion type and randomly crop them into 224×224 patches for 25 times, as there tends to be local distortion in the training database. The hyper-parameter λ for loss function is set as 0.25. The margin of the triplet loss function is set to 0.1. We use Adam [44] optimizer to pretrain our representation model for 350000 steps with mini-batch size of 32. Learning rate is set to 1×10^{-5}. The dimension size of node embedding C_V is set to 256, and the size of edge embedding C_E is set to 64. During finetuning, the input samples are randomly cropped into 224×224 at 10 times (some large images are resized to proper size firstly and randomly cropped to 224×224). We use Adam [44] optimizer to finetune on IQA task for 30 epochs with the mini-batch size of 32. The learning rate for finetuning is set to 5×10^{-6}. During the testing stage, all the testing images are randomly cropped to 10 224×224 patches, and their corresponding prediction scores are averaged to get the final quality scores. More details about the network architectures and hyper-parameter ablation studies can be found in our supplementary material.

[1]GB: Gaussian blur; LB: Lens blur; MB: Motion blur; CD: Color diffusion; CS: Color shift; CQ: Color quantization; CSA1: Color saturation 1; CSA2: Color saturation 2; WN: White noise; WNCC: White noise in color component; IN: Impulse noise; MN: Multiplicative noise; MS: Mean shift; NEP: Non-eccentricity patch; Quan.: Quantization; CB: Color block; HS: High sharpen; CC: Contrast change

B. DGR Performance Evaluation

We evaluate the effectiveness of the proposed DGR from two aspects.

a) Visualization and Clustering Evaluation.: We visualize distribution of learned DGRs using *t-SNE* [47], as are shown in Figure 4(a) and 4(b) respectively. We randomly sample 100 times from each distortion type in Kadid10k dataset to get the dimension-reduced embedding. We can see that the DGRs are well-clustering representation according to their corresponding distortion types on the whole, except for some distortion types with similar characteristics, such as Type 11 white noise and 12 white noise in color component. We also visualize the internal distribution of each DGR, as shown in Figure 4(b). From them, we can observe that the node embeddings are not only clustering well according to the distortion level, but also show a regular pattern according to the order of levels. Table I provides the clustering performance according to levels of each distortion types, which is measured by homogeneity (means all of the observations with the same class label are in the same cluster), completeness (means all members of the same class are in the same cluster) and V-measure (the combination of both homogeneity and completeness). Considering that the prediction of levels is sampled from Gaussian prior distribution to model the influence by image content, the high accuracy is not our main concern. However, in Table I, most of the results are higher than 60%, which further proves the powerful representation ability of DGR on building the relationship between level and content.

b) Leave-One Evaluation on Kadid10k Dataset.: To further validate the contribution of DGRs to IQA task, we test the IQA performance for each distortion type. We compare our method with two CNN based BIQA methods by using the Leave-One-Distortion-Out cross validation. In detail, there is one distortion type left for testing and all of the others are used as training set. All of the results are obtained by using the source code provided by their authors under the same training-testing strategy. With all the best results are highlighted in bold, we can seen from the Table II that GraphIQA scheme can already achieve competing performance without finetuning on the target dataset. After finetuning on the training set, the performance can get even better that we reach the best on most of the distortion types (18 out of 25).

C. Comparison with the State-of-the-arts

We compare our GraphIQA with the state-of-the-art (SOTA) BIQA methods including hand-craft feature based methods [11], [12], [17], deep learning based synthetic IQA

TABLE II: SRCC comparison in cross distortion type on Kadid10k dataset. All the best results are highlighted in bold.

Dist. type	GB	LB	MB	CD	CS	CQ	CSA1	CSA2	JP2K	JPEG	WN	WNCC
WaDIQaM [45]	0.879	0.730	0.730	0.833	0.421	0.806	0.148	0.836	0.539	0.530	0.897	0.925
MetaIQA [46]	0.946	0.917	0.926	0.892	**0.785**	0.717	0.304	**0.931**	**0.945**	0.912	0.905	0.930
Ours (w/o)*	0.925	0.875	0.915	0.811	0.725	0.642	**0.501**	0.618	0.941	0.822	0.817	0.875
Ours	**0.958**	**0.938**	**0.951**	**0.926**	0.738	**0.873**	0.462	0.929	0.938	**0.944**	**0.916**	**0.955**

Dist. type	IN	MN	Denoise	Brighten	Darken	MS	Jitter	NEP	Pixelate	Quan.	CB	HS	CC
WaDIQaM [45]	0.814	0.884	0.765	0.685	0.272	0.348	0.778	0.348	0.700	0.735	0.160	0.558	0.421
MetaIQA [46]	**0.867**	0.925	0.899	0.783	0.622	0.556	0.928	0.418	0.809	**0.877**	0.513	0.437	0.438
Ours (w/o)*	0.811	0.911	0.858	0.412	0.707	0.071	**0.949**	0.541	0.800	0.639	0.334	0.749	0.049
Ours	0.845	**0.951**	**0.922**	**0.889**	**0.806**	**0.745**	0.943	**0.677**	**0.873**	0.867	**0.626**	**0.904**	**0.825**

* denotes the performance of the proposed GraphIQA without being finetuned on target datasets.

TABLE III: Comparison with SOTA methods.

	SRCC			
	KonIQ	LIVEC	LIVE	CSIQ
BRISQUE [11]	0.665	0.608	0.939	0.746
ILNIQE [12]	0.507	0.432	0.902	0.806
HOSA [17]	0.671	0.640	0.946	0.741
BIECON [48]	0.618	0.595	0.961	0.815
WaDIQaM [45]	0.797	0.671	0.954	**0.955**
SFA [49]	0.856	0.812	0.883	0.796
PQR [27]	0.880	0.857	0.965	0.873
HyperIQA [50]	0.905	0.856	0.962	0.920
CNNIQA++ [28]	-	-	0.965	0.892
MEON [51]	-	-	0.951	0.852
DBCNN [30]	0.872	0.852	0.967	0.946
Ours	**0.907**	**0.863**	**0.976**	0.943
	PLCC			
	KonIQ	LIVEC	LIVE	CSIQ
BRISQUE [11]	0.681	0.629	0.935	0.829
ILNIQE [12]	0.523	0.508	0.865	0.808
HOSA [17]	0.694	0.678	0.947	0.823
BIECON [48]	0.651	0.613	0.962	0.823
WaDIQaM [45]	0.805	0.680	0.963	**0.973**
SFA [49]	0.872	0.833	0.895	0.818
PQR [27]	0.884	0.882	0.971	0.901
HyperIQA [50]	**0.922**	0.882	0.966	0.943
CNNIQA++ [28]	-	-	0.966	0.905
MEON [51]	-	-	0.955	0.864
DBCNN [30]	0.881	0.865	0.971	0.959
Ours	**0.922**	**0.886**	**0.976**	0.956

methods [45], [48] and deep learning based authentic IQA methods [27], [30], [49], [50]. All of the experiments are conducted 10 times to avoid the bias of randomness.

a) Single Database Evaluations.: The results are shown in Table III, the best results are highlighted in bold. Our approach outperforms all of the SOTA methods on most of the datasets (KonIQ-10k, LIVEC, and LIVE), and on CSIQ our approach also shows comparable performance with the SOTA methods. This suggests that our well-learned DGRs can be utilized to deal with both synthetically and authentically distorted images.

We also present the performance comparison of our approach on individual distortion types. We choose LIVE and CSIQ which are unseen during our pretraining stage for a fair comparison. The results are shown in Table IV. Compared with some methods, our approach, which is pretrained without using the annotation of MOS/DMOS (noticed as Ours (w/o) in table) can still get comparable performance on some distortion types. After finetuning with the target dataset MOS/DMOS annotations, the performance can get better. This shows that the DGRs do provide rich effective prior for IQA.

b) Generalization Evaluation.: We run the cross dataset tests on both synthetically distorted dataset pair (LIVE and CSIQ) and authentically distorted dataset pair (KoinIQ and LIVE). We select two most competing methods, DBCNN and HyperIQA for comparison. In the implementation, we use one dataset as a training set and the other one is used as a testing set. The results are shown in Table V, it can be observed that our approach can get comparable performance with the other methods. This proves the strong generalization ability of our approach.

D. Ablation Study

To evaluate the efficiency of each component in our approach. We conduct ablation study on both synthetic and authentic distortion datasets, i.e., KonIQ and LIVE.

a) Model Components.: As the scheme of using backbone ResNet50 is treated as the baseline, all the components are integrated to it, as shown in Table VI. We first prove the effectiveness of DGR and its components. The performance of utilizing edge embedding and node embedding separately is not as good as using them in combination, especially for edge embedding. This is because the most of information on edges has been aggregated into nodes during the training stage. When combined with both of them, the GraphIQA can get improved performance compared with baseline. At last, we achieve the regression based on Gaussian prior, the SRCC is further improved to the highest result on KonIQ dataset, and for LIVE dataset the performance without Gaussian prior is better. That is because the DGR, which is trained on synthetically distorted dataset, already has strong representation ability to handle synthetic distortion images.

b) Edge Embedding Size.: Then, we compare the performance on different size of edge embedding Table VIII, which are 1, 16, 32, 64 and 96. We can see that DGRs with edge embedding set as 64 can get better performance, which proves the effectiveness of the proposed 3D adjacency matrix.

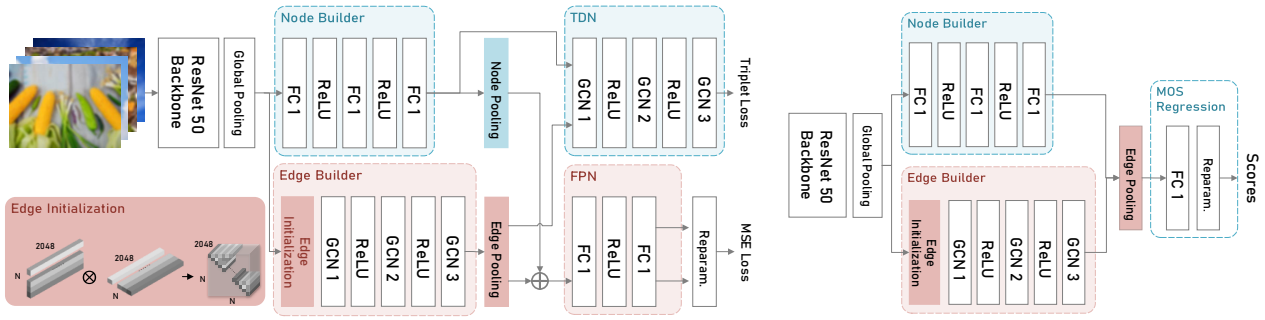
Some hyper-parameter ablation studies including margin of triplet loss, λ in loss function, node embedding size, batch size, and learning rate for both pretraining and finetune can be found in our supplementary material.

E. Experiments on architecture and hyper-parameters.

a) Archtecture.: In this section, we provide experiments of network architecture on KonIQ dataset [4], which is shown

TABLE IV: SRCC comparison on individual type in LIVE and CSIQ dataset.

Dataset	LIVE						CSIQ						
Type	JP2K	JPEG	WN	GB	FF	Total	JP2K	JPEG	WN	GB	CC	FN	Total
BRISQUE [11]	0.929	0.965	0.982	**0.964**	0.828	0.939	0.840	0.806	0.723	0.820	0.804	0.378	0.746
ILNIQE [12]	0.894	0.941	0.981	0.915	0.833	0.902	0.906	0.899	0.850	0.858	0.501	0.874	0.806
HOSA [17]	0.935	0.954	0.975	0.954	**0.954**	0.946	0.818	0.733	0.604	0.841	0.716	0.500	0.741
BIECON [48]	0.952	**0.974**	0.980	0.956	0.923	0.961	0.954	**0.942**	0.902	0.946	0.523	0.884	0.815
WaDIQaM [45]	0.942	0.953	0.982	0.938	0.923	0.954	0.947	0.853	**0.974**	**0.979**	0.923	0.882	**0.955**
HyperIQA [50]	0.949	0.961	0.982	0.926	0.936	0.962	**0.960**	0.934	0.927	0.915	0.874	0.931	0.920
DBCNN [30]	0.955	0.972	0.980	0.935	0.930	0.967	0.953	0.940	0.948	0.947	0.870	**0.940**	0.946
Ours (w/o)	0.900	0.826	0.375	0.791	0.908	0.706	0.858	0.913	0.883	0.800	0.029	0.854	0.705
Ours	**0.965**	0.966	**0.984**	0.930	**0.954**	**0.976**	0.939	0.921	0.939	0.947	**0.927**	0.919	0.943



(a) GraphIQA framework (b) Finetune GraphIQA on target database

Fig. 5: The illustration of the architecture of GraphIQA, in which (a) is the architecture for pretraining while (b) is the architecture for finetuning.

TABLE V: Cross-dataset evaluation to verify the generalization.

Dataset		DBCNN	HyperIQA	Ours
Train	Test			
KonIQ	LIVEC	0.734	0.773	**0.798**
LIVEC	KonIQ	**0.788**	0.733	0.771
CSIQ	LIVE	0.909	0.940	**0.944**
LIVE	CSIQ	0.775	**0.834**	0.823

TABLE VI: SRCC evaluation of ablation study on KonIQ and LIVE dataset.

Setting				Dataset	
Res50	DGRs-Node	DGRs-Edge	Prior*	KonIQ	LIVE
✓				0.880	0.933
✓	✓			0.898	0.970
✓		✓		0.315	0.872
✓	✓	✓		0.900	**0.976**
✓	✓	✓	✓	**0.907**	0.973

* denotes adding Gaussian prior when predicting quality scores.

in Figure 5 and the results are shown in Table IX. All the results are test on models trained on 150000 epochs, and all the other parameters are kept consistent. It is observed that when Node Builder with 3 fully connected layers, Edge Builder with 3 graph convolutional layers and TDN with 3 graph convolutional layers, our model achieve the best performance.

Table X shows the relationship between the number of epoch and performance on KonIQ dataset [4]. As the training progresses, the model's ability to distinguish and represent each distortion is improved, so that the performance improved. However, long-term training cannot continue to improve the performance, because over-fitting to synthetic distortion dataset leads to poor generalization on unknown distortion types.

TABLE VII: The number of parameters of modules.

	Backbone	NB	EB	TDN	FPN
Param.	23.5M	3.7M	2.7M	2.8M	2.2M

TABLE VIII: SRCC evaluation of different dimension size of edge embedding on KonIQ and LIVE dataset.

Size	1	16	32	64	96
KonIQ	0.898	0.899	0.905	**0.907**	0.888
LIVE	0.947	0.962	0.964	**0.976**	0.958

TABLE IX: Experiments on Architecture.

Node Builder				
fc layer	1	2	3	4
KonIQ	-	-	**0.903**	0.897
Edge Builder				
GCN layer	1	2	3	4
KonIQ	0.890	0.895	**0.903**	0.893
TDN				
GCN layer	1	2	3	4
KonIQ	0.889	0.894	**0.903**	-

b) Margin of triplet loss.: The experiments on different margin of triplet loss are provided in XI. When it is set as 0.1, the GraphIQA gets best performance on both KonIQ dataset and LIVE dataset.

TABLE X: Experiments on the number of epoch when pre-training.

Epoch	100000	150000	250000	350000	370000	400000
KonIQ	0.891	0.903	0.905	**0.907**	0.899	0.894

TABLE XI: Experiments on different margin of triplet loss.

Margin	0	0.1	0.5	1	soft margin [52]
KonIQ	0.899	**0.903**	0.903	0.900	0.898
LIVE	0.969	**0.970**	0.966	0.965	**0.970**

c) Hyper-parameters.: We test the performance with different hyper-parameters, and all the pretrained models are trained for 150000 epochs. The best results are highlighted in bold. Table XII shows that when the size of node embedding is set as 256, the performance is the best. The Table XIII shows the experimental results on mini-batch size during pre-training stage that is how many images are sampled from specific distortion type to build the graph, how weight (lambda) of $\mathcal{L}_{level}$ affects the performance on target datasets, and the performance when finetuned with different learning rate. As is shown, performance is the best with batch size 32 and Lambda 0.25. Meanwhile, as the Adam optimizer [44] is used, it is not sensitive to learning rate.

V. Conclusion

In this paper, we integrate graph representation learning into IQA and propose a novel framework GraphIQA to learn DGRs. Having the ability to represent the characteristics of each distortion and the internal structure, GraphIQA can not only generate DGRs as prior knowledge when processing known distortions but also infer the influence of unknown distortions on the perceptual image quality. For future work, we will use our DGR to challenge hybrid distortion IQA and interpretable IQA problems.

TABLE XII: Experiments on different size of node embedding.

Size	32	64	128	256	300
KonIQ	0.897	0.899	0.900	**0.903**	0.899
LIVE	0.960	0.962	0.969	**0.970**	0.962

TABLE XIII: Experiments on different mini batch size, weight (lambda) of $\mathcal{L}_{level}$ and learning rate.

bs	16	32	64	
KonIQ	0.902	**0.903**	0.905	
LIVE	0.964	**0.970**	0.960	
lambda	0.15	0.25	0.5	0.75
KonIQ	0.900	**0.903**	0.902	0.901
LIVE	0.966	**0.970**	0.969	0.969
lr	5e-5	1e-5	5e-6	1e-6
KonIQ	**0.903**	0.901	**0.903**	0.900
LIVE	0.967	0.964	**0.970**	0.968

References

[1] S. A. Golestaneh and D. M. Chandler, "No-reference quality assessment of jpeg images via a quality relevance map," *IEEE Signal Processing Letters*, vol. 21, no. 2, pp. 155–158, 2013.

[2] R. Hassen, Z. Wang, and M. M. Salama, "Image sharpness assessment based on local phase coherence," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2798–2810, 2013.

[3] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[4] H. Lin, V. Hosu, and D. Saupe, "Koniq-10k: Towards an ecologically valid and large-scale iqa database," *arXiv preprint arXiv:1803.08489*, 2018.

[5] 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*, vol. 15, no. 11, pp. 3440–3451, 2006.

[6] L. Li, H. Zhu, G. Yang, and J. Qian, "Referenceless measure of blocking artifacts by tchebichef kernel analysis," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 122–125, 2013.

[7] L. Li, W. Lin, X. Wang, G. Yang, K. Bahrami, and A. C. Kot, "No-reference image blur assessment based on discrete orthogonal moments," *IEEE transactions on cybernetics*, vol. 46, no. 1, pp. 39–50, 2015.

[8] H. Liu, N. Klomp, and I. Heynderickx, "A no-reference metric for perceived ringing artifacts in images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 4, pp. 529–539, 2009.

[9] H. Lin, V. Hosu, and D. Saupe, "Kadid-10k: A large-scale artificially distorted iqa database," in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2019, pp. 1–3.

[10] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind image quality assessment: A natural scene statistics approach in the dct domain," *IEEE transactions on Image Processing*, vol. 21, no. 8, pp. 3339–3352, 2012.

[11] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on image processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[12] L. Zhang, L. Zhang, and A. C. Bovik, "A feature-enriched completely blind image quality evaluator," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2579–2591, 2015.

[13] W. Xue, L. Zhang, and X. Mou, "Learning without human scores for blind image quality assessment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 995–1002.

[14] W. Xue, X. Mou, L. Zhang, A. C. Bovik, and X. Feng, "Blind image quality assessment using joint statistics of gradient magnitude and laplacian features," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4850–4862, 2014.

[15] K. Gu, G. Zhai, X. Yang, and W. Zhang, "Using free energy principle for blind image quality assessment," *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 50–63, 2014.

[16] P. Ye, J. Kumar, L. Kang, and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1098–1105.

[17] J. Xu, P. Ye, Q. Li, H. Du, Y. Liu, and D. Doermann, "Blind image quality assessment based on high order statistics aggregation," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4444–4457, 2016.

[18] D. Ghadiyaram and A. C. Bovik, "Perceptual quality prediction on authentically distorted images using a bag of features approach," *Journal of vision*, vol. 17, no. 1, pp. 32–32, 2017.

[19] Y. Fang, K. Ma, Z. Wang, W. Lin, Z. Fang, and G. Zhai, "No-reference quality assessment of contrast-distorted images based on natural scene statistics," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 838–842, 2014.

[20] D. Ghadiyaram and A. C. Bovik, "Massive online crowdsourced study of subjective and objective picture quality," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 372–387, 2015.

[21] D. Jayaraman, A. Mittal, A. K. Moorthy, and A. C. Bovik, "Objective quality assessment of multiply distorted images," in *2012 Conference record of the forty sixth asilomar conference on signals, systems and computers (ASILOMAR)*. IEEE, 2012, pp. 1693–1697.

[22] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012.

[23] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE transactions on Image Processing*, vol. 20, no. 12, pp. 3350–3364, 2011.

[24] J. Kim, H. Zeng, D. Ghadiyaram, S. Lee, L. Zhang, and A. C. Bovik, "Deep convolutional neural models for picture-quality prediction: Challenges and solutions to data-driven image quality assessment," *IEEE Signal processing magazine*, vol. 34, no. 6, pp. 130–141, 2017.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[26] H. Talebi and P. Milanfar, "Nima: Neural image assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3998–4011, 2018.

[27] H. Zeng, L. Zhang, and A. C. Bovik, "A probabilistic quality representation approach to deep blind image quality prediction," *arXiv preprint arXiv:1708.08190*, 2017.
[28] L. Kang, P. Ye, Y. Li, and D. Doermann, "Simultaneous estimation of image quality and distortion via multi-task convolutional neural networks," in *2015 IEEE international conference on image processing (ICIP)*. IEEE, 2015, pp. 2791–2795.
[29] K. Ma, W. Liu, K. Zhang, Z. Duanmu, Z. Wang, and W. Zuo, "End-to-end blind image quality assessment using deep neural networks," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1202–1213, 2017.
[30] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind image quality assessment using a deep bilinear convolutional neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.
[31] J. Xu, W. Zhou, and Z. Chen, "Blind omnidirectional image quality assessment with viewport oriented graph convolutional networks," *arXiv preprint arXiv:2002.09140*, 2020.
[32] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.
[33] S. Yan, Z. Li, Y. Xiong, H. Yan, and D. Lin, "Convolutional sequence generation for skeleton-based action synthesis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4394–4402.
[34] J. Fu, W. Zhou, and Z. Chen, "Bayesian spatio-temporal graph convolutional network for traffic forecasting," *arXiv preprint arXiv:2010.07498*, 2020.
[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.
[36] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017, pp. 1024–1034.
[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[38] F. Hu, Y. Zhu, S. Wu, L. Wang, and T. Tan, "Hierarchical graph convolutional networks for semi-supervised node classification," *arXiv preprint arXiv:1902.06667*, 2019.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[40] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
[41] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[42] E. C. Larson and D. M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," *Journal of electronic imaging*, vol. 19, no. 1, p. 011006, 2010.
[43] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[45] S. Bosse, D. Maniry, K.-R. Müller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 206–219, 2017.
[46] H. Zhu, L. Li, J. Wu, W. Dong, and G. Shi, "Metaiqa: Deep meta-learning for no-reference image quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 143–14 152.
[47] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, and Y. Kluger, "Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data," *Nature methods*, vol. 16, no. 3, pp. 243–245, 2019.
[48] J. Kim and S. Lee, "Fully deep blind image quality predictor," *IEEE Journal of selected topics in signal processing*, vol. 11, no. 1, pp. 206–220, 2016.
[49] D. Li, T. Jiang, W. Lin, and M. Jiang, "Which has better visual quality: The clear blue sky or a blurry animal?" *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1221–1234, 2018.
[50] S. Su, Q. Yan, Y. Zhu, C. Zhang, X. Ge, J. Sun, and Y. Zhang, "Blindly assess image quality in the wild guided by a self-adaptive hyper network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3667–3676.
[51] K. Ma, W. Liu, K. Zhang, Z. Duanmu, Z. Wang, and W. Zuo, "End-to-end blind image quality assessment using deep neural networks," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1202–1213, March 2018.
[52] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.