

Assessing Image Quality Using a Simple Generative Representation

Simon Raviv¹ Gal Chechik^{1,2}

Abstract

Perceptual image quality assessment (IQA) is the task of predicting the visual quality of an image as perceived by a human observer. Current state-of-the-art techniques are based on deep representations trained in discriminative manner. Such representations may ignore visually important features, if they are not predictive of class labels. Recent generative models successfully learn low-dimensional representations using auto-encoding and have been argued to preserve better visual features. Here we leverage existing auto-encoders and propose VAE-QA, a simple and efficient method for predicting image quality in the presence of a full-reference. We evaluate our approach on four standard benchmarks and find that it significantly improves generalization across datasets, has fewer trainable parameters, a smaller memory footprint and faster run time.

1. Introduction

Assessing the visual quality of an image is a key problem with applications in numerous computer vision fields, from image restoration and enhancement to generative text-to-image models. Current state-of-the-art methods for *Image Quality Assessment* (IQA), are built on top of deep representations trained for discriminative tasks [38; 9; 24; 4; 13], taking distorted images and predicting human quality judgment.

Broadly speaking, these deep discriminative models predict perceived image quality well, but suffer from various drawbacks. They require labeled data for training the representation, they tend to be heavy and complex, and, most importantly, their discriminative representations may remove features that may be predictive about image quality, but not about class labels. Finally, discriminative representations also tend to generalize poorly to data from a different distribution [38; 2; 24; 5; 13]. It therefore remains a hard

problem to train models that predict image quality in a way that generalizes to new datasets.

Unlike discriminative representations, recent approaches to image generation learn representations that preserve fine image content [27]. These representations can be trained self-supervised without class labels, and presumably preserve all information about image content, which may be removed by discriminative representations [16].

Here we propose a simple approach for predicting image quality based on a *Variational Auto Encoder* (VAE) generative model. Given a pre-trained VAE representation, we learn how to use its latent activation for predicting human judgment of image quality. Our approach, which we name VAE-QA, has significantly fewer trainable parameters and a smaller memory footprint, easily fitting on a standard consumer GPU. It also achieves state-of-the-art prediction accuracy on standard benchmarks in the field, both on new images from the same distribution and when generalizing to new datasets.

In summary, this paper makes the following contributions: (1) We put forward the idea that predicting image quality would be superior using a representation learned in a generative way. (2) A new architecture for predicting image quality built on top of a VAE model, which learns to align features from several different layers of the VAE. (3) Standardization of evaluation protocol. First, we release a data split so that future papers can compare data consistently. Second, we analyze aspects of the inference protocol, like the effect of the number of crops on prediction accuracy. (4) New SoTA results for cross-dataset generalization.

2. Related Work

2.1. Learning-based IQA

Various approaches were proposed for using deep representations for IQA. The most known approach is *Learned Perceptual Image Patch Similarity* (LPIPS) [38]. They were the first to highlight the potential of using representations learned during a classification task on ImageNet [28] for quality prediction. DeepQA [9] uses CNN to predict a visual sensitivity map that weights pixel importance in distorted images, aligning with human subjective opinions. WaDIQaM [2] introduces a comprehensive end-to-end deep neural network that allows for simultaneous local quality

¹Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel ²NVIDIA, Tel-Aviv, Israel. Correspondence to: Simon Raviv <simon1raviv@gmail.com>, Gal Chechik <gal.chechik@gmail.com>.

and local weight learning. Different from these methods, PieAPP [24] focuses on learning to rank. This means the network is trained to understand the probability of one image being preferred over another.

Recent advancements in deep learning-based IQA methods have led to significant improvements in the field. [4] proposed an *Image Quality Transformer* (IQT) that applies a transformer architecture to a perceptual full-reference IQA task. The proposed model extracts perceptual feature representations from each input image using a CNN backbone. It feeds the extracted feature maps into the transformer encoder and decoder to compare reference and distorted images and predict the final score. [13] proposed an *Attention-based Hybrid Image Quality Assessment* (AHIQ) network that uses hybrid image representations learned by CNN and transformer-based networks. To predict the final score, the final image representation is learned through simple attention mechanisms.

2.2. Non-learned IQA

Besides learning-based IQA methods, it is common to evaluate image quality using predefined methods, including *Peak Signal-to-Noise Ratio* (PSNR), *Structural Similarity Index Measure* (SSIM) [33], and *Feature Similarity Index Measure* (FSIM) [36]. These methods are easy to use and fast to compute, but their predictive power is very limited compared with learned approaches [38; 13]. It remains a challenge to simplify training of deep IQA predictors while maintaining prediction quality.

3. Background

To keep this paper self-contained, we describe the concepts of generative models, *Variational Auto Encoder* (VAE) [11] and the various setups used for image quality assessment.

3.1. Generative Models

Generative models learn the underlying data distribution. The most common generative models today are diffusion models [8; 31; 26]. Diffusion models add noise through diffusion, then de-noise their data by removing noise through denoising networks. The diffusion process is reversed to generate data. A denoising network iteratively refines a simple noise distribution, effectively ‘undoing’ the diffusion, until it finds a sample that closely matches the learned data distribution. As a result of this process, diffusion models generate diverse and high-quality samples that capture the complexities of the training data.

In recent years, diffusion models have shown impressive results in image generation tasks and are considered state-of-the-art. The *Latent Diffusion Model* (LDM) [26] is a diffusion model that uses a latent representation of the data

instead of pixels. The latent representation is learned by a *Variational Auto Encoder* (VAE) [11] network that maps the raw samples to a latent space. The main advantage of the LDM is its run-time efficiency, as it operates on the latent space, which is much smaller than the pixel space.

3.2. Variational Auto Encoders

A VAE is a type of generative model [3; 25; 19] that learns to represent data in a lower-dimensional latent space. It does this by encoding input data into a latent space and then decoding or reconstructing it from this latent space. The VAE is trained to minimize the difference between the input data and the reconstructed data. The objective function of VAEs is the *Evidence Lower Bound* (ELBO), which includes two components: reconstruction loss and regularization terms such as KL divergence. The reconstruction loss measures the fidelity of the reconstructed data to the original input data, while the KL divergence acts as a regularization term, encouraging the learned latent distribution to be close to a prior distribution. This balance between reconstruction and regularization allows the VAE to effectively learn the underlying data distribution.

3.3. IQA setups

Work in the field of IQA considers three setups: full-reference (FR-IQA) – comparing the quality of a distorted image to a reference clean image, reduced-reference (RR-IQA) – assessing the quality of a distorted image using some information from the reference image, and no-reference (NR-IQA) – evaluating the distorted image is evaluated without any reference image.

4. Our Method

We present a model we call *Variational Auto Encoder Quality Assessment* (VAE-QA). It is a deep neural network that learns to predict the quality of a distorted image, given the original reference image (Full-Reference).

Our key idea is to use a pre-trained VAE to compute generative representations of the original image and the distorted image. The deep network is trained to predict image quality by taking as input representations from multiple layers of VAE.

FR-IQA methods typically have three main components: feature extraction, feature fusion, and quality prediction. We describe these three components of our method below. Figure 1 illustrates the architecture of our VAE-QA. A detailed description of the method’s components can be found in the supplementary material.

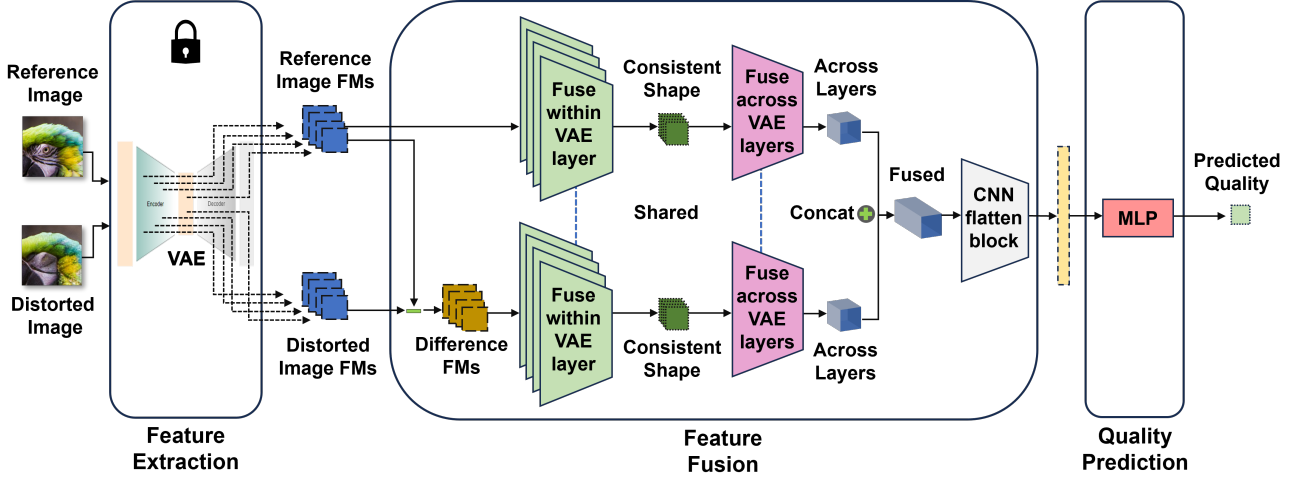


Figure 1. VAE-QA architecture: **Feature extraction module** extracts image representations from input images using a VAE. **Feature fusion module** combines the extracted image representations to form a compressed representation using within & across VAE layer(s) components. **Quality prediction module** uses the compressed representation to predict the quality score of the input images using a MLP network.

4.1. Feature Extraction Module

The feature extraction module extracts image representations from input images using a VAE. Specifically, we use the VAE encoder from LDM [27] to get image representations.

By encoding a reference image and a distorted image with the VAE encoder, several feature maps are obtained from its intermediate layers. Six feature maps were chosen, spanning a broad spectrum of abstractions and layers, to encode various levels of image information.

The first two layers (image, enc1) capture basic image features in the input spatial dimension. The next three layers (enc4, enc7, enc12) are from the down-sampling part of the encoder, which extracts high-level features in reduced spatial dimensions. The next layer (enc15) is from the middle part of the encoder, which further processes the encoded representation and adds more complex features in the same spatial dimension. The last layer (enc19) is the quantization part of the encoder, which represents the latent encoded representation of the input image. These layers capture features at different abstraction levels and spatial dimensions.

4.2. Feature Fusion Module

The feature-fusion module is the core of the architecture. It learns to combine features extracted from the VAE into a single unified representation. In VAE-QA, it has three components: Fusion within VAE layer, fusion across VAE layers, and CNN flattening block. First, a difference feature map is created by subtracting the reference feature map from the distorted feature map. Then, those feature maps

are passed to the fusion within VAE layer component.

Fusion within VAE layer. This component learns a joint feature representation combining features within the same VAE layer, taking into account its spatial structure. We use a CNN with ReLU activation and average pooling layers, followed by group normalization over channels [34]. All feature maps regardless of layer and input dimension are mapped to an output feature dimension of $\mathbb{R}^{(L+1)*128 \times 16 \times 16}$, where L is the number of layers from the VAE, and we also include the original image. This mapping allows us to combine features across layers in the next step.

Fusion across VAE layers. This component learns a joint feature representation combining features across different VAE layers, taking into account information from different spatial dimensions. We use a CNN with ReLU activation and average pooling layers, followed by group normalization over channels. We get a consistent shape feature map with a dimension of $\mathbb{R}^{1024 \times 16 \times 16}$. Finally, we concatenate reference and difference feature maps to form a single feature map with a dimension of $\mathbb{R}^{2048 \times 16 \times 16}$.

CNN flatten block. This component transforms a 2D features map into a 1D vector, used for quality prediction. We use a CNN with ReLU activation and average pooling layers to compress the feature maps. Features are also normalized using group normalization. Finally, we flatten the feature maps to form a compressed representation vector with a dimension of $\mathbb{R}^{4096}$.

Table 1. IQA datasets for model training and performance evaluation.

DATASET	# REFERENCE IMAGES	# DISTORTED IMAGES	# DISTORTION TYPE	# RATING	RATING TYPE
LIVE [30]	29	779	5	25K	DMOS
CSIQ [15]	30	866	6	5K	DMOS
TID2013 [23]	25	3,000	24	500K	MOS
KADID-10K [17]	81	10,125	25	30.4K	DMOS

4.3. Quality Prediction Module

The quality prediction module is a 3 layer MLP network. It takes the fused representation vector as input and predicts the quality score of the input images.

5. Experiments

5.1. Compared Methods

We compared VAE-QA with four recent baselines that use deep representations and achieve high prediction accuracy. (1) LPIPS [38] A pioneering paper that showed the advantage of using deep representations to predict perceptual image quality. (2) DeepQA [9] predicts a visual sensitivity map that weights pixel importance in distorted images, aligning with human subjective opinions. (3) PieAPP [24], trained using ranking loss. (4) AHIQ [13], added features from vision transformers to standard CNN features.

Notably, [29] showed that quality predictions can be improved by predicting a saliency map designed to mimic human perception, and adding that as a side-channel to deep representations. Since this paper focuses on finding good deep features, and since we could not obtain their code for comparisons, we do not report their results in the main table, but discuss the effect of salience maps in the supplementary.

We also report standard non-deep predictors, like PSNR and SSIM [33], since these are commonly used.

5.2. Evaluation Protocol

DATASETS:

We evaluate our approach with the four datasets commonly used in evaluations of FR-IQA methods LIVE [30], CSIQ [15], TID2013 [23], and KADID-10k [17]. Table 1 provides details about each dataset.

Usually, LIVE, CSIQ, and TID2013 are used for "in-distribution" evaluations. LIVE and CSIQ contain several distortion types, such as blurring, noise, and JPEG compression. TID2013, is much more comprehensive and contains 24 distortion types. These include new noise types, such as impulse noise and high frequency noise, which negatively affect image quality. Including spatial and color shifts, such as local block-wise distortions, mean shifts, and contrast

changes, which alter an image's appearance. This makes TID2013 more challenging.

Recent papers [2; 24; 5; 13] used KADID-10k to evaluate cross-dataset generalization. KADID-10k is more than three times larger than TID2013, and includes more reference images.

TARGET VALUES:

In FR-IQA, *Mean Opinion Score* (MOS) refers to the average score a group of people give an image when comparing its original version to another image. The objective of our method is to predict MOS. Details on MOS vs. *Difference MOS* (DMOS) are available in the supplementary.

DATA SPLITS:

There is no standard data split in this domain. Therefore we split each dataset randomly into training (80%), and test (20%) sets and make that split available at [IQA Standard Datasets Splits](#).

We repeated the split three times and reported below the average over the three random splits. Splitting is done according to the reference image, so all distortions of the same image are in the same split. Images from test data are not visible during training.

To tune the hyperparameters of all models, we further split the training set into train and validation (60% of the original data into train and 20% into validation).

EVALUATION METRICS:

We follow the standard protocol in this field and report the correlation between predicted values and ground truth human judgment of image quality for each distorted image. Specifically, we compute *Pearson Linear Correlation Coefficient* (PLCC) and the (non-linear) *Spearman Rank Correlation Coefficient* (SRCC).

INFERENCE PROTOCOL:

There is no single standardized evaluation protocol in this domain. Different previous papers take different approaches to how they handled images of different sizes. They use various sizes and numbers of crops from images and integrating

Table 2. **Within-dataset evaluation:** Performance scores on three standard IQA datasets: LIVE, CSIQ, and TID2013. Scores for our method are averages across three seeds, the standard error is the order of 0.01, and is reported in the supplementary for clarity. Scores for [13] and [38] were reproduced using authors code and are marked by (*), see also Table 5. Scores for all other methods were taken from original papers.

METHOD	LIVE		CSIQ		TID2013		OVERALL	
	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC
PSNR	.865	.873	.819	.810	.677	.687	.787	.790
SSIM [33]	.937	.948	.852	.865	.777	.727	.855	.847
FSIMc [36]	.961	.965	.919	.931	.877	.851	.919	.916
VSI [37]	.948	.952	.928	.942	.900	.897	.925	.930
NLPD [14]	.932	.937	.923	.932	.839	.800	.898	.890
GMSD [35]	.957	.960	.945	.950	.855	.804	.919	.905
SCQI [1]	.937	.948	.927	.943	.907	.905	.924	.932
DEEP METHODS								
LPIPS [38] (*)	.823	.911	.873	.934	.758	.779	.818	.875
DEEPPQA [9]	.982	.981	.965	.961	.947	.939	.818	.875
PIEAPP [24]	.986	.977	.975	.973	.946	.945	.969	.965
AHIQ [13] (*)	.988	.985	.971	.968	.927	.922	.962	.958
VAE-QA (OURS)	.984	.981	.974	.968	.961	.958	.973	.969

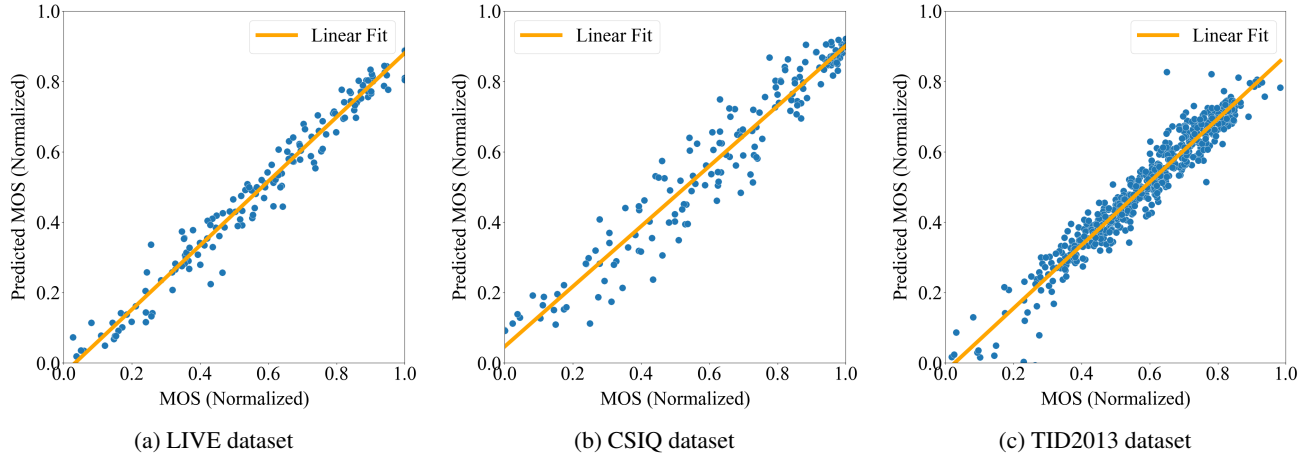


Figure 2. MOS vs. Predicted MOS for three IQA datasets.

information from crops in various ways.

For instance, JND-SalCAR [29] used 32×32 patches sampled from reference and distorted images, to train and test their network. However, this approach may not fully capture the global quality of the image, as it ignores the spatial relationships between patches. Moreover, it may introduce bias due to the patch selection strategy, as some regions may be over- or under-represented. In contrast, AHIQ [13] used the entire image as input, and reported the average of 20 random crops from the image as the final score.

To disentangle the effect of the number of crops, we report the quality of our method using the same protocol as [13], and later study the effect of various crops.

Out of all previous strong work in this area, only [13] provided code that allowed us to reproduce the results using the

same protocol.

5.3. Implementation Details

PRE-TRAINED VAE:

We use a VAE [11] from LDM [27] that was pre-trained on OpenImages [12]. Specifically, the VAE variant we used was trained with KL divergence regularized latent space and a down sample factor of 8 (aka LDM-8).

DATA PROCESSING:

Input images are normalized to the range $[-1, 1]$ and cropped to 256×256 . During training, we randomly crop input images with a uniform distribution and flip them horizontally with a probability of 0.5.

Table 3. **Cross dataset evaluation.** Trained on KADID-10k dataset and tested on LIVE, CSIQ, and TID2013 datasets. Performance scores for all methods except of AHIQ were taken from their papers. (*) stands for results that we reproduced using the authors code. (**) stands for results borrowed from [5].

METHOD	KADID → LIVE		KADID → CSIQ		KADID → TID2013		OVERALL	
	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC
LPIPS [38] (**)	.934	.932	.896	.876	.749	.670	.860	.826
WADIQAM [2]	.940	.947	.901	.909	.834	.831	.892	.896
PIPIEAPP [24]	.908	.919	.877	.892	.859	.876	.881	.896
DISTS [5]	.934	.932	.896	.876	.749	.670	.860	.826
AHIQ [13] (*)	.956	.954	.945	.940	.891	.888	.931	.927
VAE-QA (OURS)	.945	.967	.952	.960	.905	.900	.934	.942

Table 4. **Cross dataset evaluation.** Trained on TID2013 dataset and tested on LIVE and CSIQ datasets. "CENTER" refers to using a single center crop. Performance scores for all methods were taken from their papers.

METHOD	TID2013 → LIVE		TID2013 → CSIQ		OVERALL	
	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC
WADIQAM [2]	-	.936	-	.931	-	.934
DOG-SSIM [22]	-	.948	-	.925	-	.937
VAE-QA (CENTER)	.923	.942	.938	.935	.931	.939
VAE-QA (OURS)	.930	.948	.943	.941	.937	.945

IMAGE REPRESENTATION:

Each image was represented using 7 maps: the 256×256 cropped image itself and six layers of VAE encoder outputs, layers {1, 4, 7, 12, 15, 19}. These layers include the initial CNN layer (1), 3 down-sampling stages (4, 7, 12), a mid-stage (15), and an end part (19) that reduces channels and quantizes to the final latent space representation.

NORMALIZATION AND REGULARIZATION:

The signal in the model is normalized by group normalization [34]. We use 32 groups for normalization layers. For 2D dropout [32] layers in the feature fusion module, we use a dropout rate of 0.2, and for the quality prediction module dropout [7] layers, we apply a dropout rate of 0.3.

OPTIMIZATION:

As a training loss, we minimized the mean squared error between the predicted and true MOS labels, normalized to the range [0, 1]. We use Adam [10] optimizer with an initial learning rate of 10^{-4} , weight decay of 10^{-5} , and batch size of 8. Learning rate was scheduled with Cosine Annealing [18] with a minimum learning rate of 0 and maximum scheduler steps of 50. The model is trained for 50 epochs and the best model is selected based on the highest PLCC and SRCC evaluation metrics. Specifically, we followed the protocol by [13]: at each epoch, if the PLCC or SRCC was greater than the highest recorded till that epoch, the model

of that epoch was considered the best model.

SW & HW:

The model was trained using a single NVIDIA A6000 GPU with 48GB of memory. The model is implemented over PyTorch [21] and PyTorch Lightning [6] libraries.

6. Results

We evaluate VAE-QA in two main scenarios. First, generalizing to new images from the same dataset. Then, a more realistic scenario of generalizing to images from a different dataset not seen during training. We follow standard evaluation protocols as possible.

6.1. Within-dataset Evaluations

We first evaluate the model by training and testing it on images from the same data distribution. Table 2 shows the linear and non-linear correlation over the test split for each of the 3 standard datasets. For LIVE and CSIQ datasets, VAE-QA shows a small improvement, while it improves considerably for TID2013 and across datasets overall.

Figure 2 shows the relation between predicted and ground-truth quality. The scatter plot suggests this relation is largely linear in this regime.

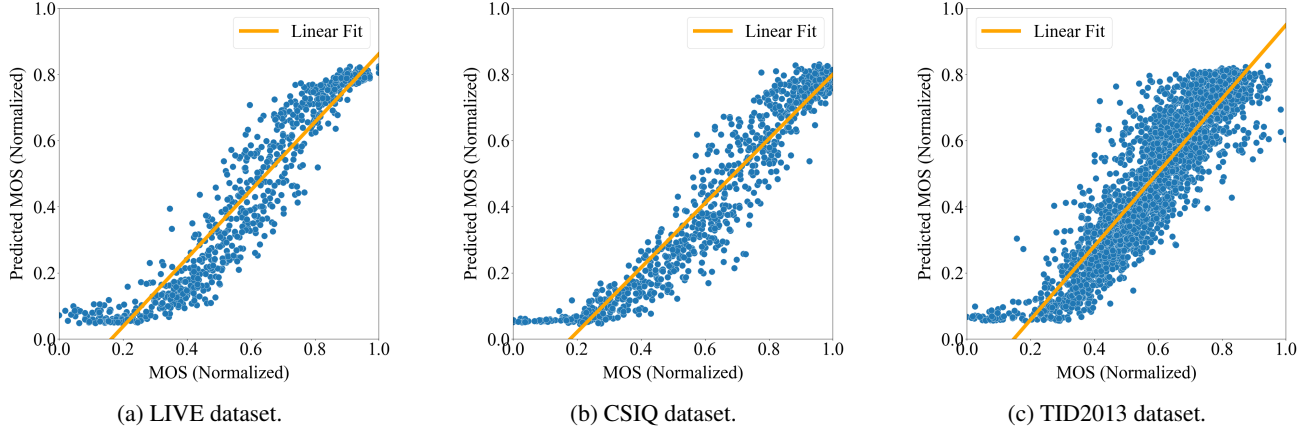


Figure 3. MOS vs. Predicted MOS. Trained on KADID-10k, tested on other IQA datasets.

Table 5. The effect of inference protocol on prediction quality for within-dataset experiments.

METHOD	LIVE		CSIQ		TID2013		OVERALL	
	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC
20 RANDOM CROPS								
AHIQ [13] (REPORTED)	.989	.984	.978	.975	.968	.962	.978	.974
AHIQ [13] (REPRODUCED)	.988	.985	.971	.968	.927	.922	.962	.958
VAE-QA (OURS)	.984	.981	.974	.968	.961	.958	.973	.969
CENTER CROP								
AHIQ [13] (REPRODUCED)	.987	.984	.970	.965	.931	.922	.963	.957
VAE-QA (OURS)	.984	.981	.971	.966	.958	.954	.971	.967

6.2. Cross dataset Evaluations

To evaluate the generalization performance of the model, we trained the model on the entire KADID-10k dataset and evaluated it on the full set of LIVE, CSIQ, and TID2013 datasets. Table 3 report the average PLCC and SRCC values of this experiment. VAE-QA consistently improves over current methods.

We further tested generalization by training the model on the TID2013 dataset and testing on the LIVE and CSIQ datasets. Table 4 shows the results of this experiment. Our model generalizes robustly to new data distributions.

Figure 3 shows the relation between predicted MOS and ground truth for the cross-dataset experiments.

6.3. Analysis

We explore various parameters that affect the quality of the results including the evaluation protocol and number of crops, and the variance across crops and seeds.

THE EFFECT OF EVALUATION PROTOCOL:

Different studies in this field [29; 24; 13] use different ways to preprocess a given image into a square input to the model. Table 5 and Table 6 reports the correlations observed when

using 20 crops compared with a single central crop. It also shows results obtained from the original AHIQ paper and the correlations obtained where we ran the authors’ code in our system. The authors have been extremely helpful, and we made an effort to reproduce all hyperparameters in the model. The differences can be attributed to “lucky” random seeds, or to hyperparameter differences that we could not trace.

THE EFFECT OF THE NUMBER OF CROPS:

Correlation results were also examined in relation to the number of crops. See Figure 5. At 20 crops, saturation is noticeable.

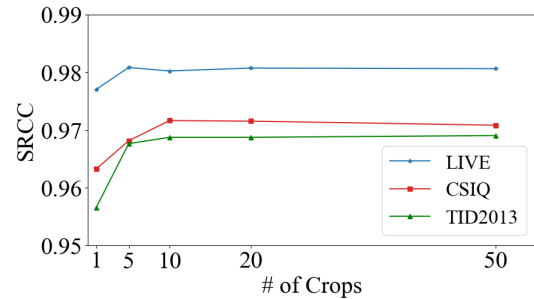


Figure 5. The effect of the number of crops on the SRCC.

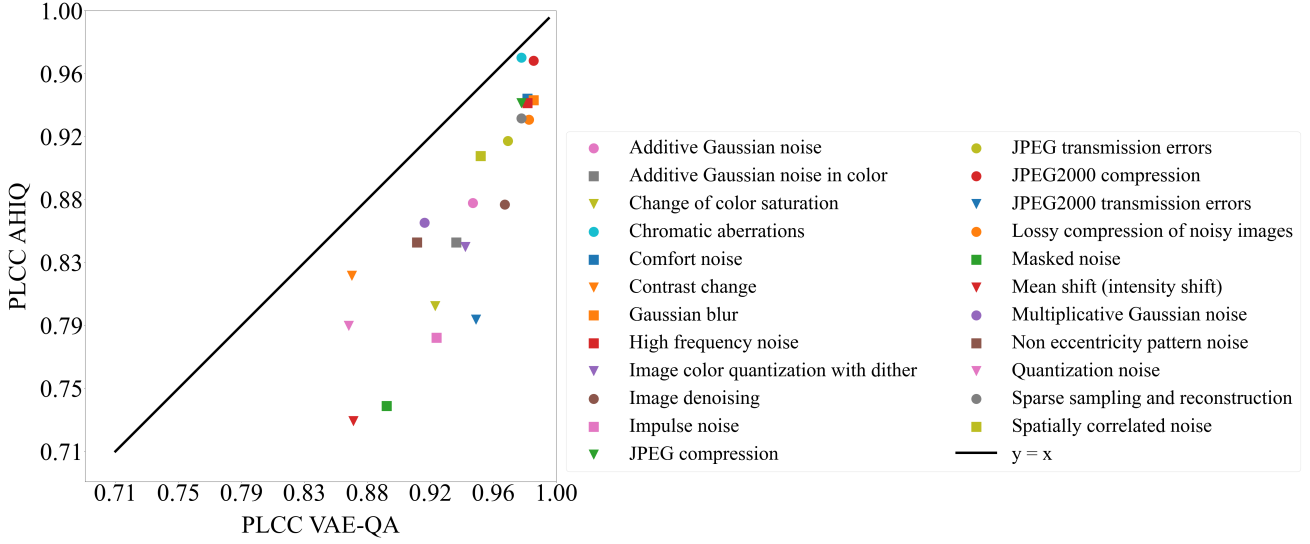


Figure 4. Quality prediction by distortion type on TID2013 dataset. The figure compares PLCC obtained with our VAE-QA and AHQ.

Table 6. The effect of inference protocol on cross-dataset generalization.

METHOD	KADID→LIVE		KADID→CSIQ		KADID→TID2013		OVERALL	
	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC
CENTER CROP								
AHIQ [13] (REPRODUCED)	.949	.942	.930	.922	.868	.862	.916	.909
VAE-QA (OURS)	.942	.964	.949	.957	.897	.893	.929	.938
20 RANDOM CROPS								
AHIQ [13] (REPORTED)	.952	.970	.955	.951	.899	.901	.935	.941
AHIQ [13] (REPRODUCED)	.956	.954	.945	.940	.891	.888	.931	.927
VAE-QA (OURS)	.945	.967	.952	.960	.905	.900	.934	.942

VARIANCE ACROSS CROPS AND SEEDS:

Table 7 examines the variance across crops and seeds together. It shows the standard deviation over 20 crops, averaged across 3 seeds, and the standard deviation across those three seeds.

We observe a small variance between seeds, and a larger variance between crops. This is consistent with the results in Figure 5.

CORRELATION BY DISTORTION TYPE

To obtain more insight into the performance of VAE-QA, we measured the quality of quality prediction for different types of distortion. Figure 4 compares the prediction accuracy measured using PLCC, for our method and AHQ, the current SoTA. The VAE representation underlying VAE-QA, appears to help in all distortion types, and provides on average larger improvements for those types that are challenging for AHQ (masked noise and intensity shift).

6.4. Runtime and Memory Footprint

In this section, we compare the memory footprint and runtime of our method with the current state-of-the-art method AHQ [13]. We report the number of parameters and model size in Table 8. We also provide the model’s memory consumption in Figure 6. Finally, we report the model’s runtime in Table 9. This shows that our method has a smaller memory footprint and faster inference time than the current state-of-the-art method.

Table 9. Inference time per image.

METHOD	RUNTIME [MS]
AHIQ [13]	35.16
VAE-QA (OURS)	26.56
DIFF [%]	32.37

7. Conclusion

We described VAE-QA a deep architecture for image quality assessment that is based on deep features learned using

Table 7. The effect of the variance across crops and seeds on prediction quality.

METHOD	LIVE		CSIQ		TID2013	
	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC
AHIQ [13] (REPRODUCED)	.0023 $\pm$.0008	.0023 $\pm$.0007	.0030 $\pm$.0012	.0026 $\pm$.0005	.0052 $\pm$.0012	.0068 $\pm$.0015
VAE-QA (OURS)	.0023 $\pm$.0009	.0020 $\pm$.0009	.0025 $\pm$.0014	.0027 $\pm$.0007	.0018 $\pm$.0008	.0023 $\pm$.0008

Table 8. Number of parameters and model size.

METHOD	NON TRAINABLE [M]	TRAINABLE [M]	TOTAL PARAMS [M]	PARAMS SIZE [MB]
AHIQ [13]	112.1	27.2	139.3	557.2
VAE-QA (OURS)	34.2	14.4	48.5	194.2
DIFF [%]	69.49	47.06	65.18	65.15

a generative (auto-encoding) task. The intuition is that generative latent representations are designed to capture the fine details of the image in contrast with discriminative approaches which preserve information about class labels. We find that VAE-QA consistently improves the accuracy of predicted quality compared with previous methods when tested on images from different datasets providing a new SoTA in this task. It also achieves a small improvement on average on same-dataset generalization. VAE-QA also has a smaller memory footprint and faster inference time than current SoTA.

These results suggest that current generative models can be easily leveraged for quality assessment. We expect that this approach can be generalized to other applications like video quality assessment.

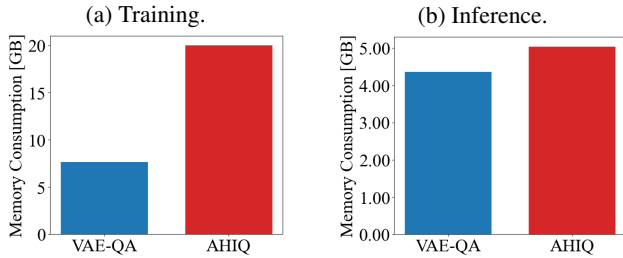


Figure 6. Memory footprint of the model.

References

- [1] Bae, S.-H. and Kim, M. A novel image quality assessment with globally and locally consistent visual quality perception. *IEEE Transactions on Image Processing*, 25(5):2392–2406, 2016.
- [2] Bosse, S., Maniry, D., Müller, K.-R., Wiegand, T., and Samek, W. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Transactions on Image Processing*, 27(1):206–219, January 2018. ISSN 1941-0042.
- [3] Cai, L., Gao, H., and Ji, S. Multi-stage variational auto-encoders for coarse-to-fine image generation, 2017.
- [4] Cheon, M., Yoon, S.-J., Kang, B., and Lee, J. Perceptual image quality assessment with transformers, 2021.
- [5] Ding, K., Ma, K., Wang, S., and Simoncelli, E. P. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020. ISSN 1939-3539.
- [6] Falcon, W. and The PyTorch Lightning team. PyTorch Lightning, 2019. Version 1.4.
- [7] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors, 2012.
- [8] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.
- [9] Kim, J. and Lee, S. Deep learning of human visual sensitivity in image quality assessment framework. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1969–1977, 2017.
- [10] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017.
- [11] Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.
- [12] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., Kolesnikov, A., Duerig, T., and Ferrari, V. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, March 2020. ISSN 1573-1405.
- [13] Lao, S., Gong, Y., Shi, S., Yang, S., Wu, T., Wang, J., Xia, W., and Yang, Y. Attentions help cnns see better: Attention-based hybrid image quality assessment network, 2022.
- [14] Laparra, V., Ballé, J., Berardino, A., and Simoncelli, E. P. Perceptual image quality assessment using a normalized laplacian pyramid. In *Human Vision and Electronic Imaging*, 2016.
- [15] Larson, E. C. and Chandler, D. M. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006, 2010.
- [16] Li, D., Ling, H., Kar, A., Acuna, D., Kim, S. W., Kreis, K., Torralba, A., and Fidler, S. Dreamteacher: Pre-training image backbones with deep generative models, 2023.
- [17] Lin, H., Gu, J., Li, C., and Dong, C. Kadid-10k: A large-scale artificially distorted iqa database. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–3. IEEE, 2019.
- [18] Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts, 2017.
- [19] Luhman, T. and Luhman, E. High fidelity image synthesis with deep vaes in latent space, 2023.
- [20] Mohammadi, P., Ebrahimi-Moghadam, A., and Shirani, S. Subjective and objective quality assessment of image: A survey, 2014.
- [21] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library, 2019.
- [22] Pei, S.-C. and Chen, L.-H. Image quality assessment using human visual dog model fused with random forest. *IEEE Transactions on Image Processing*, 24(11):3282–3292, 2015.
- [23] Ponomarenko, N., Jin, L., Ieremeiev, O., Lukin, V., Egiazarian, K., Astola, J., Vozel, B., Chehdi, K., Carli, M., Battisti, F., et al. Image database tid2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, 30:57–77, 2015.
- [24] Prashnani, E., Cai, H., Mostofi, Y., and Sen, P. Pieapp: Perceptual image-error assessment through pairwise preference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1808–1817, 2018.
- [25] Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2, 2019.

- [26] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2021.
- [27] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2022.
- [28] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge, 2015.
- [29] Seo, S., Ki, S., and Kim, M. A novel just-noticeable-difference-based saliency-channel attention residual network for full-reference image quality predictions, 2020.
- [30] Sheikh, H. R., Sabir, M. F., and Bovik, A. C. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451, 2006.
- [31] Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models, 2022.
- [32] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. Efficient object localization using convolutional networks, 2015.
- [33] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [34] Wu, Y. and He, K. Group normalization, 2018.
- [35] Xue, W., Zhang, L., Mou, X., and Bovik, A. C. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE Transactions on Image Processing*, 23(2):684–695, February 2014. ISSN 1941-0042.
- [36] Zhang, L., Zhang, L., Mou, X., and Zhang, D. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.
- [37] Zhang, L., Shen, Y., and Li, H. Vsi: A visual saliency-induced index for perceptual image quality assessment. *IEEE Transactions on Image Processing*, 23(10):4270–4281, 2014.
- [38] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric, 2018.

A. Additional Results

A.1. Correlation by Distortion Type

To obtain more insight into the performance of VAE-QA, we measured the quality of quality prediction for different types of distortion. Figure 7 shows the performance of our method and AHIQ on the LIVE and CSIQ datasets. These datasets contain simpler distortions compared to TID2013, and VAE-QA reaches comparable performance to SoTA.

A.2. Variability

Tables 10, 11, and 12 provide the standard deviation of our method across three seeds.

B. Limitations

B.1. Dataset Bias

KADID-10k [17] was used for cross-dataset generalization, which is a large dataset with a range of distortions. However, the dataset does not necessarily represent the entire distortion distribution in reality. Furthermore, all datasets have a limited number of unique images, which makes generalization to real-world images difficult. Therefore, our method may suffer from some generalization errors when applied to real-world scenarios.

B.2. Image Type

Our method uses the VAE from [26] to extract features from images. The VAE is trained on a OpenImages [12] dataset, which contains natural images. Therefore, our method may not work well on images that are significantly different from natural ones, such as medical images, satellite images, or images of a particular domain.

C. Implementation Details

This section describes in detail the components of our method.

C.1. Feature Extraction Module

We use feature maps from the VAE with the following dimensions: $f_{img} \in \mathbb{R}^{c_0 \times r_0 \times r_0}$, $f_{enc1} \in \mathbb{R}^{c_1 \times r_0 \times r_0}$, $f_{enc4} \in \mathbb{R}^{c_1 \times r_1 \times r_1}$, $f_{enc7} \in \mathbb{R}^{c_2 \times r_2 \times r_2}$, $f_{enc12} \in \mathbb{R}^{c_3 \times r_3 \times r_3}$, $f_{enc15} \in \mathbb{R}^{c_3 \times r_3 \times r_3}$, and $f_{enc19} \in \mathbb{R}^{c_4 \times r_3 \times r_3}$.

Where c_i is the number of channels and r_i is the spatial resolution of the feature map. Channel dimensions are $c_i = \{3, 128, 256, 512, 8\}$ and spatial resolutions are $r_i = \{256, 128, 64, 32\}$. The feature fusion module uses these feature maps as input.

C.2. Feature Fusion Module

The feature-fusion module combines features extracted from the VAE into a single unified representation. First, a difference feature map is created by subtracting the reference feature map from the distorted feature map. Then, those feature maps are passed to the fusion within VAE layer component.

FUSION WITHIN VAE LAYER:

Algorithm 1 *FusionWithinVAELayer_i*

Input: feature map F_i

Output: fused feature map F_i^{fused}

$h = Conv2d_i(F_i)$

$h = ReLU(h)$

$h = Dropout2d(h)$

$h = GroupNorm_i(h)$

$F_i^{fused} = AdaptiveAvgPool2d(h)$

The input to the fusion across VAE layers component is the concatenated feature maps:

$$F_{fused}^{ref} = \text{Concat}(\text{FusionWithinVAELayer}_i(F_i^{ref})), \quad (1)$$

$$F_{fused}^{diff} = \text{Concat}(\text{FusionWithinVAELayer}_i(F_i^{diff})), \quad (2)$$

where $F_{fused}^{ref}, F_{fused}^{diff} \in \mathbb{R}^{L \times 128 \times 16 \times 16}$, where L is the number of layers from the VAE plus the input image.

FUSION ACROSS VAE LAYERS:

Algorithm 2 *FusionAcrossVAELayers*

Input: feature map F_{fused}

Output: fused feature map F_{fused}^{across}

$h = Conv2d(F_{fused})$

$h = ReLU(h)$

$F_{fused}^{across} = GroupNorm(h)$

The input to the CNN flatten block is the concatenated feature map:

$$F_{fused}^{across} = \text{Concat}(\text{FusionAcrossVAELayers}(F_{fused}^{ref}), \text{FusionAcrossVAELayers}(F_{fused}^{diff})), \quad (3)$$

Where $F_{fused}^{across} \in \mathbb{R}^{1024 \times 2 \times 16 \times 16}$.

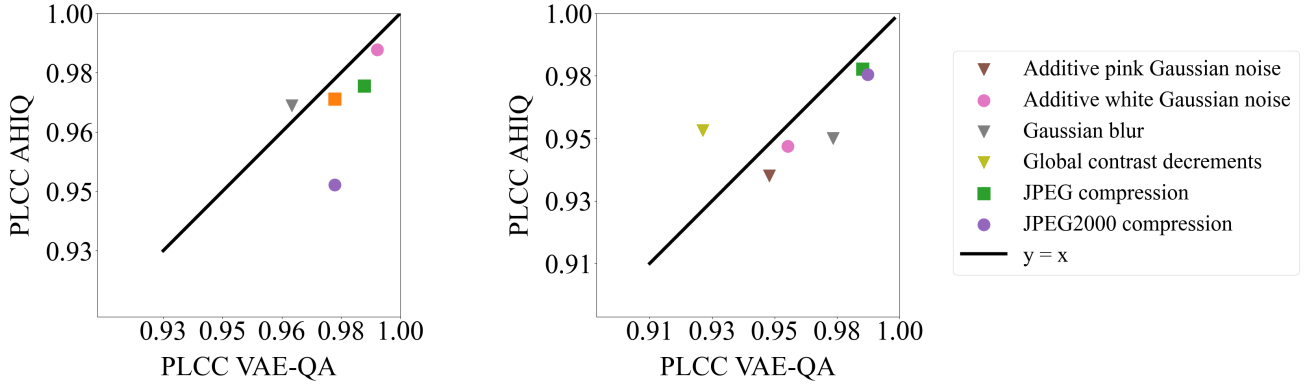


Figure 7. Quality prediction by distortion type on LIVE and CSIQ datasets. The figure compares PLCC obtained with our VAE-QA and AHQ.

Table 10. **Within-dataset evaluation - standard deviation.** Performance scores on three standard IQA datasets: LIVE, CSIQ, and TID2013. All scores are averages across three seeds. Scores for [13] were reproduced using authors code and are marked by (*).

METHOD	LIVE		CSIQ		TID2013		OVERALL	
	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC
AHIQ [13] (*)	.988±.002	.985±.002	.971±.008	.968±.008	.927±.036	.922±.045	.962	.958
VAE-QA (OURS)	.984±.005	.981±.004	.974±.009	.968±.009	.961±.016	.958±.017	.973	.969

Table 11. **Cross dataset evaluation - standard deviation.** Trained on KADID-10k dataset and tested on LIVE, CSIQ, and TID2013 datasets. All scores are averages across three seeds. Scores for [13] were reproduced using authors code and are marked by (*).

METHOD	KADID → LIVE		KADID → CSIQ		KADID → TID2013		OVERALL	
	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC
AHIQ [13] (*)	.956±.004	.954±.007	.945±.005	.940±.005	.891±.003	.888±.003	.931	.927
VAE-QA (OURS)	.945±.001	.967±.001	.952±.001	.960±.001	.905±.003	.900±.003	.934	.942

Table 12. **Cross dataset evaluation - standard deviation.** Trained on TID2013 dataset and tested on LIVE and CSIQ datasets. "CENTER" refers to using a single center crop. All scores are averages across three seeds.

METHOD	TID2013 → LIVE		TID2013 → CSIQ		OVERALL	
	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC
VAE-QA (CENTER)	.923±.001	.942±.000	.938±.002	.935±.003	.931	.939
VAE-QA (OURS)	.930±.000	.948±.000	.943±.001	.941±.002	.937	.945

CNN FLATTEN BLOCK:

Algorithm 3 *FlattenBlock*

Input: feature map F_{fused}^{across}
Output: Compressed representation C_{rep}
 $h = \text{Conv2d}(F_{fused}^{across})$
 $h = \text{ReLU}(h)$
 $h = \text{AdaptiveAvgPool2d}(h)$
 $C_{rep} = \text{Flatten}(h)$

The output of the feature fusion module is the compressed representation $C_{rep} \in \mathbb{R}^{1024 \times 2 \times 2}$.

C.3. Quality Prediction Module

Algorithm 4 *MLP*

Input: Compressed representation C_{rep}
Output: Quality prediction $q \in \mathbb{R}$
 $h = \text{Linear}(C_{rep})$
 $h = \text{GroupNorm}(h)$
 $h = \text{ReLU}(h)$
 $h = \text{Dropout}(h)$
 $h = \text{Linear}(h)$
 $h = \text{ReLU}(h)$
 $q = \text{Linear}(h)$

C.4. Mapping DMOS to MOS

Image quality can be assessed using *Mean Opinion Score* (MOS) and *Difference MOS* (DMOS) based on human opinions. For FR-IQA, MOS is the average score a group of people give after seeing an image compared to its original version. The DMOS is the difference between the raw quality scores of the reference and distorted images. The DMOS is calculated by subtracting the MOS of the reference image from the MOS of the distorted image [20]. DMOS measures the impact of a particular distortion on image quality, while MOS measures overall quality.

For datasets that provide DMOS values, we transform them to MOS using the following formula:

$$\text{MOS} = \max(\text{DMOS}) - \text{DMOS} \quad (4)$$

D. The Effect of Saliency

It has been shown that predicting perceptual quality can be improved by integrating saliency maps, which capture how much attention people pay to various areas in a given image [29].

Adding salience is likely to improve other image quality assessment than the ones tested in [29], because it helps assessment methods focus on the most important parts of an image. Unfortunately, we were unable to obtain the code from the authors and could not make a direct comparison.