Appendix to "Toward Generalized Image Quality Assessment: Relaxing the Perfect Reference Quality Assumption"

Du Chen[1,3,*], Tianhe Wu[2,3,*], Kede Ma[2,†], and Lei Zhang[1,3,†]
[1]The Hong Kong Polytechnic University [2]City University of Hong Kong [3]OPPO Research Institute
csdud.chen@connet.polyu.hk, {tianhewu, kede.ma}@cityu.edu.hk, cslzhang@comp.polyu.edu.hk

In this appendix, we provide the following material:
- Training details of our generative image enhancer (please refer to Sec.3.1 of the main paper);
- Inference details of our generative image enhancer (please refer to Sec.3.2 of the main paper);
- Details of subjective experimental setups for constructing DiffIQA (please refer to Sec.3.2 of the main paper);
- Details of subjective experimental setups for constructing SRIQA-Bench (please refer to Sec.5.2 of the main paper);
- Discussions on the generated "fake" details.

1. Training Details of the Generative Image Enhancer

1.1. Architecture

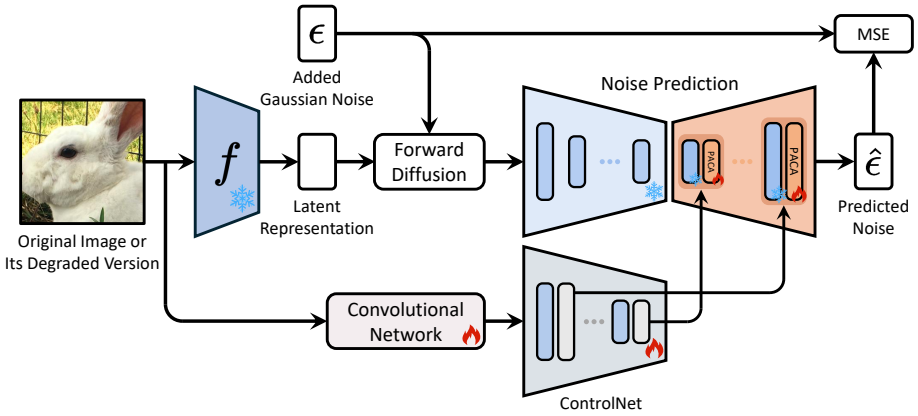


Fig. 1. Training of our generative image enhancer.

The overall architecture of the enhancer during the training phase is illustrated in Fig. 1. The original image is passed through a lightweight convolutional network, with features fed into a ControlNet [6] to provide content-aware conditional signal to the diffusion UNet [2]. Meanwhile, the original image or its degraded version is passed through the image encoder [1] to produce a latent representation. In forward diffusion, Gaussian noise is added to the latent image, which serves as the input to the diffusion UNet. The conditional signal from ControlNet interacts with the diffusion UNet via pixel-aware cross-attention (PACA) [4]. Finally, we compute the MSE between the predicted noise by the diffusion UNet and the added Gaussian noise, which is treated as the ground-truth. During training, only the convolutional network, ControlNet, and PACA modules are trainable.

[*]Equal contribution.
[†]Corresponding authors.

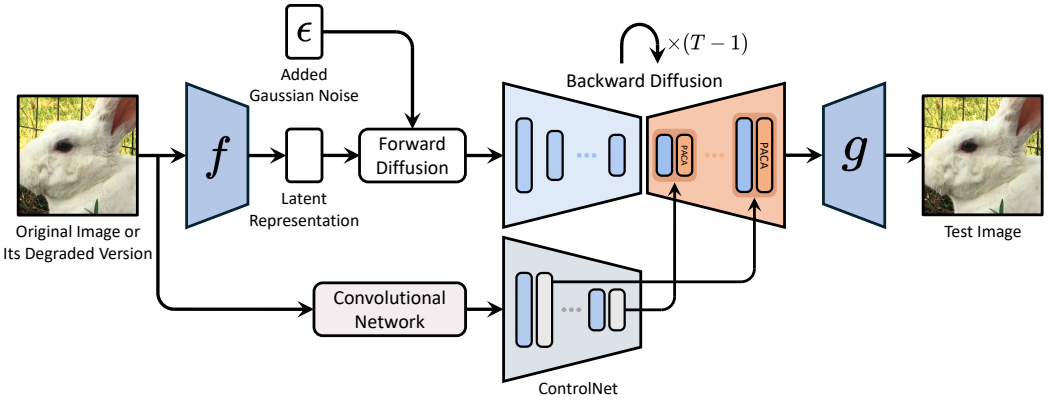


Fig. 2. Inference of our generative image enhancer.

1.2. Training Specifications

To diversify output image quality, 50% of the original inputs are directly fed into the enhancer, while the other 50% undergo slight blind degradations [3, 5]:

$$x_d = \texttt{compression}\left(\texttt{resizing}(x * h) + \epsilon\right), \tag{1}$$

where x represents the original high-quality image, and h is an (an)isotropic blur kernel. $\texttt{resizing}(\cdot)$ indicates the resizing operation, ϵ denotes the additive Gaussian or Poisson noise, and $\texttt{compression}(\cdot)$ stands for JPEG compression. The degraded image x_d is resized to the original resolution using bicubic interpolation before feeding into the enhancer. Detailed degradation configurations are provided in Table 1.

We trained our enhancer on eight NVIDIA V100 GPUs using Adam with a fixed learning rate of 5×10^{-5} for $100,000$ iterations, each GPU handling a minibatch size of 32. The training image size was configured at 512×512 pixels.

Table 1. Blind degradation settings of our enhancer. "iso" and "an-iso" denote "isotropic" and "an-isotropic," respectively.

Operation	Parameter	Setting
Blurring	Kernel size $[2m+1]$	$m \in [1, 4]$
	Kernel list	iso, an-iso, generalized iso, generalized an-iso, plateau iso, plateau an-iso
	Kernel list probability	$0.45, 0.25, 0.12, 0.03, 0.12, 0.03$
	Sinc kernel [3] probability	0.1
	Standard deviation	$[0.0, 1.2]$
Resizing	Resizing list	down-sampling, up-sampling
	Resizing list probability	$0.85, 0.05, 0.1$
	Resizing range	$[0.8, 1.1]$
	Resizing mode	area, bilinear, bicubic
Noise Contamination	Noise list	Gaussian, Poisson
	Noise list probability	$0.5, 0.5$
	Sigma of Gaussian	$[0.0, 13.0]$
	Scale of Poisson	$[0.0, 0.9]$
	Gray noise [3] probability	0.1
JPEG Compression	Quality factor	$[75, 95]$

2. Inference Details of the Generative Image Enhancer

The overall architecture of our enhancer during inference is illustrated in Fig. 2. As described in Sec. **??** of the main paper, we randomly applied the same degradations as used during training, augmented the initial image latent with additive Gaussian noise of varying intensities, and adjusted the sampling steps within range $[20, 1000]$.

Finally, we generated a total of $179,208$ test images using 20 NVIDIA V100 GPUs, with an inference batch size of one per GPU. To accelerate inference, we employed the same UniPC Scheduler in PASD [4]. The entire inference process took approximately 20 days.

3. Subjective Experimental Setups of DiffIQA

We developed a graphical user interface (GUI), as illustrated in Fig. 3, for MOS collection. This software is built using PyQt5[1], which is compatible with Windows Operating Systems from versions 8 to 11, ensuring low latency and support for screen resolutions ranging from 1,080 to 2K. Core functionalities of our GUI include 1) presentation of images in random spatial order; 2) a zoom-in feature using the mouse scroll wheel for more-detailed comparison; 3) a maximum of 10-second viewing time with the prompt of the message: "Please make your choice."; 4) a radio button group of three choices; and 5) a checkpointing feature, ensuring that the subject can stop at any time to minimize the fatigue effect, and the software will resume from the last image pair when reopened. It is important to note that our paired comparison is incomplete, as the test image is compared solely with its reference image.

Before formal subjective testing, we included an approximately two-hour training session for all subjects, designed to familiarize them with the overall subjective testing procedure. Specifically, we provided a detailed demonstration of the specific functionalities of our GUI, and general guidelines to make visual comparisons. Subjects were instructed to focus primarily on image attributes closely related to perceived image quality, such as image naturalness and distortion visibility, with some visual examples (see Fig. 5).

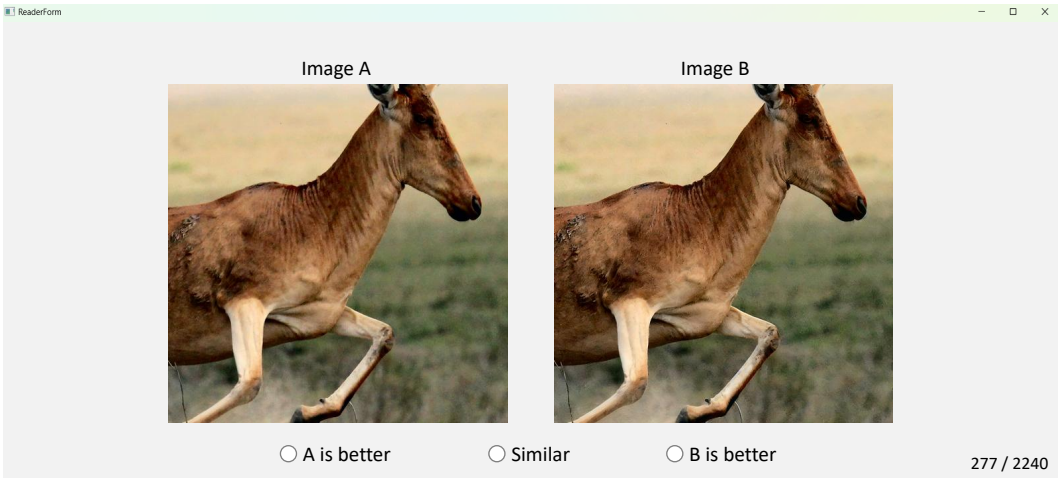
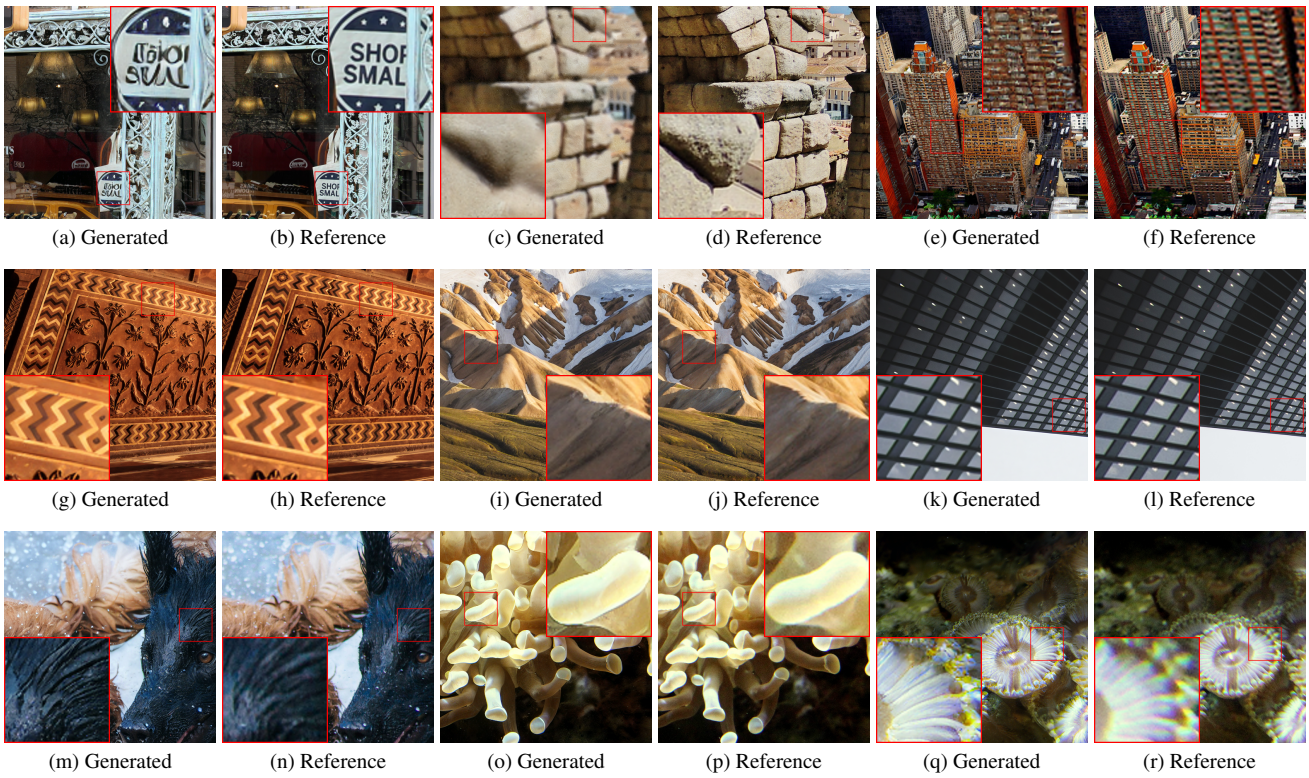


Fig. 3. The GUI used for constructing DiffIQA.

Fig. 4. The GUI used for constructing SRIQA-Bench.

[1]https://www.riverbankcomputing.com/software/pyqt/



(a) Generated (b) Reference (c) Generated (d) Reference (e) Generated (f) Reference

(g) Generated (h) Reference (i) Generated (j) Reference (k) Generated (l) Reference

(m) Generated (n) Reference (o) Generated (p) Reference (q) Generated (r) Reference

Fig. 5. Representative images that are *worse* (**(a)** to **(e)**), *similar* (**(g)** to **(k)**), and *better* (**(m)** to **(q)**) relative to their references in our DiffIQA dataset. Zoom in for better visibility.

4. Subjective Experimental Setups of SRIQA-Bench

The GUI for SRIQA-Bench closely resembles that of DiffIQA, with the key difference being the inclusion of a reference image in the middle for facilitating comparison of the two test images, as illustrated in Fig. 4.

Unlike the training session adopted in DiffIQA, subjects were first instructed to evaluate the fidelity of the two test images relative to the reference. If the test images exhibit comparable fidelity, subjects then selected the one with better quality, following similar guidelines described in Sec. 3. Conversely, if the test images show significant differences in fidelity, subjects were instructed to choose the image with higher fidelity to the reference.

5. Discussions on the Generated "Fake" Details

It is important to note that there are instances where the enhanced image appears to have superior overall quality, but the details differ significantly from the reference. This suggests that the enhanced details are hallucinated yet plausible. To address this issue during subjective testing, participants were instructed to prioritize deformed or fake details when assessing image quality. If such details impact the image's fidelity, participants would annotate the image as having worse quality. As illustrated in Fig. 6, while the content in the blue box of the generated image appears sharper than the reference, the text in the red box is visibly distorted. Our model, A-FINE, correctly evaluates the reference image as having better quality, consistent with human judgments.

References

[1] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming Transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 1

[2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1

[3] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *IEEE/CVF International Conference on Computer Vision Workshops*, pages 1905–1914, 2021. 2

(a) Test image (b) Reference image

Fig. 6. Illustration of the "fake" generated details. In this example, the reference image is of better quality than the test image according to our subjective testing protocol.

[4] Tao Yang, Rongyuan Wu, Peiran Ren, Xuansong Xie, and Lei Zhang. Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization. In *European Conference on Computer Vision*, pages 74–91, 2024. 1, 2

[5] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE/CVF International Conference on Computer Vision*, pages 4791–4800, 2021. 2

[6] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1