Appendix for "MUSIQ: Multi-scale Image Quality Transformer"

A. Transformer Encoder

A.1. Transformer Encoder Structure

We use the classic Transformer encoder [9] in our experiments. As illustrated in Figure 1, the Transformer block layer consists of multi-head self-attention (MSA), Layernorm (LN) and MLP layers. Residual connections are added in between the layers.

In MST-IQA, the multi-scale patches are encoded as $\mathbf{x}_k^n$ where $k = 0 \cdots K$ is the scale index and n is the patch index in the scale. $k = 0$ represents the full-size image. We then add HSE and SCE to the patch embeddings, forming the multi-scale representation input. Similar to previous works [2], we prepend a learnable [class] token embedding to the sequence of embedded tokens ($\mathbf{x}_{class}$).

The Transformer encoder can be formulated as:

$$\mathbf{E}_p = [\mathbf{x}_0^1; \cdots \mathbf{x}_0^l; \mathbf{x}_1^1; \cdots; \mathbf{x}_1^{m_1}; \cdots; \mathbf{x}_K^1; \cdots; \mathbf{x}_K^{m_K}] \quad (1)$$

$$\mathbf{z}_0 = [\mathbf{x}_{class}; \mathbf{E}_p + \mathbf{E}_{\text{HSE}} + \mathbf{E}_{\text{SCE}}] \quad (2)$$

$$\mathbf{z}'_q = \text{MSA}(\text{LN}(\mathbf{z}_{q-1})) + \mathbf{z}_{q-1}, \qquad q = 1 \cdots L \quad (3)$$

$$\mathbf{z}_q = \text{MLP}(\text{LN}(\mathbf{z}'_q)) + \mathbf{z}'_q, \qquad q = 1 \cdots L \quad (4)$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \quad (5)$$

$\mathbf{E}_p$ is the patch embedding. $\mathbf{E}_{\text{HSE}}$ and $\mathbf{E}_{\text{SCE}}$ are the spatial embedding and scale embedding respectively. l is the number of patches from original resolution. $m_1 \cdots m_K$ are the number of patches from resized variants. $\mathbf{z}_0$ is the input to the Transformer encoder. $\mathbf{z}_q$ is the output of each Transformer layer and L is the total number of Transformer layers.

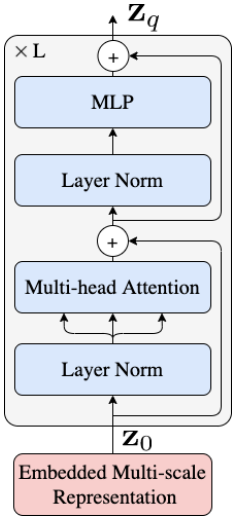

Figure 1. Transformer encoder illustration. Graph inspired by [3, 9].

A.2. Multi-head Self-Attention (MSA)

In this section we introduce the standard **QKV** self-attention (SA) [9] (Figure 2) and its multi-head version (MSA). Suppose the input sequence is represented by $\mathbf{z} \in \mathbb{R}^{N \times D}$, $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are its query, key, and value representations, respectively. They are generated by projecting the input sequence with a learnable matrix $\mathbf{U}_q, \mathbf{U}_k, \mathbf{U}_v \in \mathbb{R}^{D \times D_h}$, respectively. D_h is the inner dimension for $\mathbf{Q}, \mathbf{K}, \mathbf{V}$. We then compute a weighted sum over $\mathbf{V}$ using attention weights $\mathbf{A} \in \mathbb{R}^{N \times N}$ which are pairwise similarities between $\mathbf{Q}$ and $\mathbf{K}$.

$$\mathbf{Q} = \mathbf{z}\mathbf{U}_q, \quad \mathbf{K} = \mathbf{z}\mathbf{U}_k, \quad \mathbf{V} = \mathbf{z}\mathbf{U}_v \quad (6)$$

$$\mathbf{A} = \text{softmax}(\mathbf{Q}\mathbf{K}^T / \sqrt{D_h}) \quad (7)$$

$$\text{SA}(\mathbf{z}) = \mathbf{A}\mathbf{V} \quad (8)$$

MSA is an extension of SA where s self-attention operations (heads) are conducted in parallel. The outputs from all heads are concatenated together and then projected to the final output with a learnable matrix $\mathbf{U}_m \in \mathbb{R}^{s \cdot D_h \times D}$. D_h is typically set to D/s to keep computation and number of parameters constant for each s.

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(\mathbf{z}); \cdots; \text{SA}_s(\mathbf{z})]\mathbf{U}_m \quad (9)$$

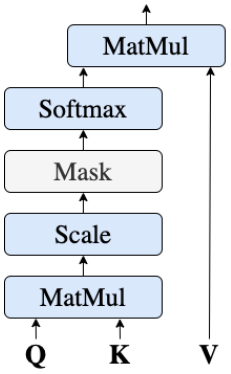

Figure 2. Single head self-attention (SA) illustration.

A.3. Masked Self-Attention

Masking is often used in self-attention [2, 9] to ignore padding elements or to restrict attention positions and prevent data leakage (*e.g.* in causal or temporal predictions). In batch training, we use the input mask to indicate the effective input and to ignore padding tokens. As shown in Figure 2, the mask is added on attention weights before the softmax. By setting the corresponding elements to $-\text{inf}$ before the softmax step in Equation 7, the attention weights on invalid positions are close to zero.

The attention mask is constructed as $\mathbf{M} \in \mathbb{R}^{N \times N}$ where

$$\mathbf{M}_{i,j} = \begin{cases} 0 & \text{if attention pos}_i \rightarrow \text{pos}_j \text{ valid} \\ -\text{inf} & \text{if attention pos}_i \rightarrow \text{pos}_j \text{ invalid} \end{cases} \quad (10)$$

Then the masked self-attention weight matrix is calculated as

$$\mathbf{A}_m = \text{softmax}((\mathbf{Q}\mathbf{K}^T + \mathbf{M})/\sqrt{D_h}). \quad (11)$$

A.4. Different Transformer Encoder Settings

We use a lightweight parameters setting for Transformer encoder in the main experiments to make the model size comparable to ResNet-50. Here we also report the results from different Transformer encoder settings. The model variants are shown as Table 1. The MST-IQA-Small model is the one used in our main experiments in the paper. The performance of these variants on the AVA dataset is shown in Table 2. Overall, these models have similar performance when pre-trained on ImageNet [6]. Larger Transformer backbones might need more data to pre-train in order to get better performance. As shown in experiments from [3], larger Transformer backbones get better performance when pre-trained on ImageNet21k [1] or JFT-300m [8].

Model	Layers	Hidden size D	MLP size	Heads	Params
MST-IQA-Small	14	384	1152	6	27M
MST-IQA-Medium	8	768	2358	8	61M
MST-IQA-Large	12	768	3072	12	98M

Table 1. MST-IQA variants with different Transformer encoder settings.

Model	SRCC	PLCC
MST-IQA-Small	0.916	**0.928**
MST-IQA-Medium	**0.918**	**0.928**
MST-IQA-Large	0.916	0.927

Table 2. Performance of different MST-IQA variants on the KonIQ-10k dataset.

B. Additional Studies for HSE

B.1. Grid Size G in HSE

We run ablation studies for the grid size G in the proposed hash-based 2D spatial embedding (HSE). Results are shown in Table 3. Small G may result in collision and therefore the model cannot distinguish spatially close patches. Large G means the hashing is more sparse and therefore needs more diverse resolutions to train, otherwise some positions may not have enough data to learn good representations. One can potentially generate fixed T for larger G when detailed positions really matter (*e.g.* using sinusoidal function, see Appendix B.2). With a learnable T, a good rule of thumb is to let grid size times the number of patches P roughly equals the average resolution, *i.e.* $G \times G \times P \times P = H \times W$. Since the average resolution across 4 datasets is around 450×500 and we use patch size 32, we use grid size around 10 to 15. Overall, we find different G does not change the performance too much once it is large enough, showing that rough spatial encoding is sufficient for IQA tasks.

Spatial Embedding	SRCC	LCC
HPE ($G = 5$)	0.720	0.733
HPE ($G = 8$)	0.723	0.734
HPE ($G = 10$)	**0.726**	**0.738**
HPE ($G = 12$)	0.722	0.736
HPE ($G = 15$)	0.724	0.735
HPE ($G = 20$)	0.722	0.734

Table 3. Ablation study for different grid size G in HSE on AVA dataset.

	Learnable T		Fixed-Sin T	
G	SRCC	PLCC	SRCC	PLCC
10	**0.726**	**0.738**	0.719	**0.733**
15	0.724	0.735	0.716	0.730
20	0.722	0.734	**0.720**	**0.733**

Table 4. Comparison of sinusoidal HSE and learnable HSE matrix on AVA dataset.

Patch Size	16	32	48	64
SRCC	0.715	**0.726**	0.713	0.705
PLCC	0.729	**0.738**	0.727	0.719

Table 5. Comparison of different patch size on AVA dataset.

B.2. Sinusoidal HSE v.s. Learnable HSE

Besides the learnable HSE matrix $T \in \mathbb{R}^{G \times G \times D}$ introduced in the paper, another option is to generate a fixed positional encoding matrix T using the sinusoidal function as [9]. In Table 4, we show the performance comparison of using learnable T or generated sinusoidal T with different Grid size G. Overall, the learnable T gives slightly better performance than that of the fixed T.

B.3. Visualization of HSE with Different G

Figures 3 and 4 visualize the learned HSE with $G = 5$ and $G = 15$, respectively. Even with G as small as 5, the similarity matrix corresponds well to the patch position in the image, showing that HSE captures patch position in the image.

C. Effect of Patch Size

We ran ablation on different patch size P, results are shown in Table 5. In our settings, we find patch size $P = 32$ performs well across datasets.

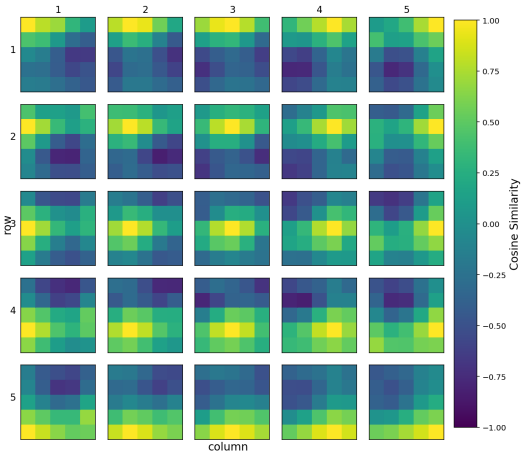

Figure 3. Visualization of the grid of HSE with $G = 5$.

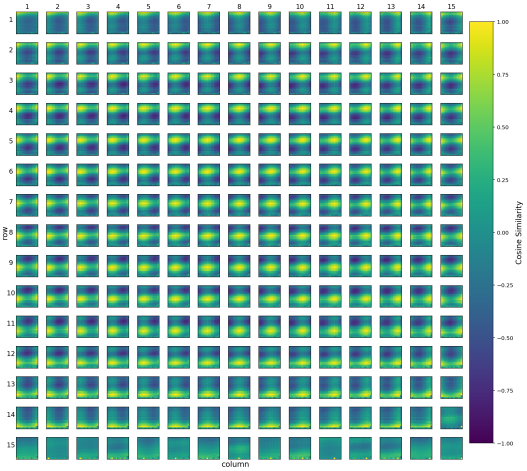

Figure 4. Visualization of the grid of HSE with $G = 15$.

D. The Maximum Number of Patches (l) from Full-size Image

We run ablation with different l during training. As shown in the Table 6, using large l in the fine-tuning can improve the model performance. Since larger resolution images have more patches than low resolution ones, when l is too small, some larger images might be cutoff, thus the model performance will degrade.

l	SRCC	LCC
128	0.876	0.895
256	0.906	0.923
512	**0.916**	**0.928**

Table 6. Comparison of maximum #patches l from full-size image on KonIQ-10k dataset.

E. KonIQ-10k More Results

In our main experiment on KonIQ-10k, we followed BIQA [7] and MetaIQA [10] to report the average of 10 random 80/20 train-test splits to avoid the bias. On the other hand, methods like KonCept512 [5] uses a fixed split instead of averaging. In Table 7, we report our results using the same fixed split. Images in KonIQ-10k are of the same resolution and CNN models like KonCept512 usually need a cherry-picked fixed size to work well. Unlike CNN models that are constrained by fixed size, MST-IQA does not need tuning the input size and generalizes well for diverse resolutions.

method	SRCC	LCC
KonCept512 [5]	0.921	**0.937**
MST-IQA (Ours)	**0.924**	**0.937**

Table 7. Results on KonIQ-10k dataset using same fixed split as KonCept512 [5].

F. SPAQ Full-size Results

As mentioned in Section 4.1, we follow [4] to resize the raw images such that the shorter side is 512 for a fair comparison with the reference methods. Since our model can be applied directly on the images without resizing, we also report the performance on the SPAQ full-size test in Table 8 when training on the SPAQ full-size train. The results only have very little difference.

	SRCC	PLCC
Full-size train and test	0.916 ($\pm$0.001)	0.919 ($\pm$0.001)
Resized train and test	0.917 ($\pm$0.002)	0.921 ($\pm$0.002)

Table 8. Comparison of MST-IQA train and evaluate on full-size SPAQ dataset or the 512 shorter side resized SPAQ dataset.

G. Computation Complexity

For the default MST-IQA model, the number of parameters is around 27M. For a 224x224 image, its FLOPS is 8.86×10^9, which is at the same level as SOTA CNN-based models (23M parameters and 3.8×10^9 FLOPS for ResNet50). Training IQA takes 0.8 TPUv3-core-days on average. MST-IQA is compatible with the efficient Transformer backbones like Linformer and Performer, which greatly reduce the complexity of the original Transformer. We leave model speedup as the future work.

H. Multi-scale Attention Visualization

To understand how MST-IQA uses self-attention to integrate information across different scales, we visualize the

average attention weights from the output tokens to each image in the multi-scale representation as Figure 5. We follow [3] for the attention map computation. In short, the attention weights are averaged across all heads and then recursively multiplied, accounting for the mixing of attention across tokens through all layers.

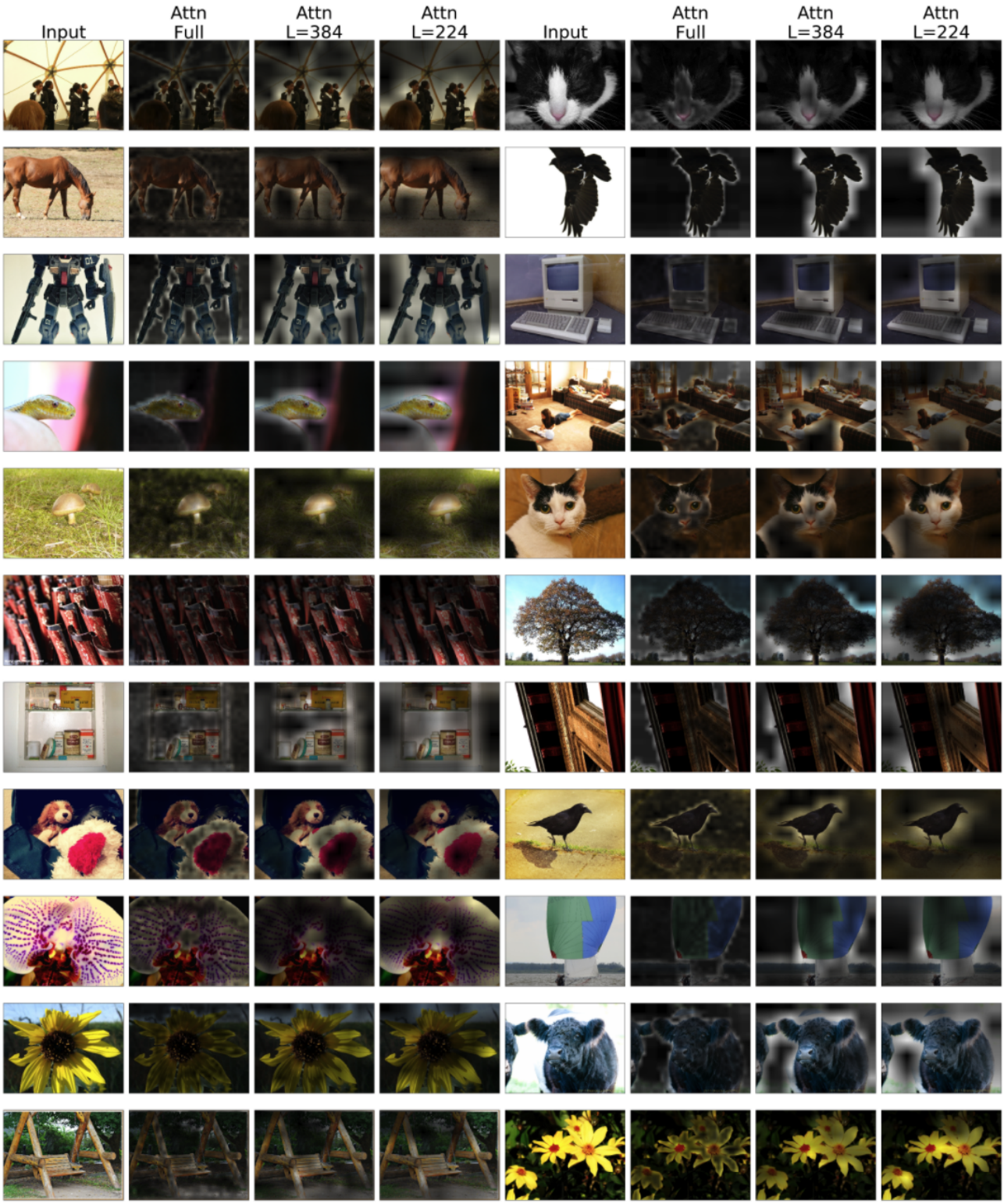

Figure 5. Visualizations of attention from the output tokens to the multi-scale representation. “Input” column shows the input image. “Attn Full” shows the attention on the full-size image. “Attn L=384” and “Attn L=224” show the attention on the ARP resized images. Note that images here are resized to fit the grid, the model inputs are 3 different resolutions.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. 1

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy. 1, 2, 4

[4] Yuming Fang, Hanwei Zhu, Yan Zeng, Kede Ma, and Zhou Wang. Perceptual quality assessment of smartphone photography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3677–3686, 2020. 3

[5] Vlad Hosu, Hanhe Lin, Tamas Sziranyi, and Dietmar Saupe. Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Transactions on Image Processing*, 29:4041–4056, 2020. 3

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. 2

[7] Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning 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*, pages 3667–3676, 2020. 3

[8] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 843–852, 2017. 2

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1, 2

[10] Hancheng Zhu, Leida Li, Jinjian Wu, Weisheng Dong, and Guangming Shi. Metaiqa: deep meta-learning for no-reference image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14143–14152, 2020. 3