FAST-VQA: Efficient End-to-end Video Quality Assessment with Fragment Sampling

Haoning Wu[1,3], Chaofeng Chen[1], Jingwen Hou[2], Liang Liao[1], Annan Wang[2], Wenxiu Sun[3], Qiong Yan[3], and Weisi Lin[2]

[1] S-Lab, Nanyang Technological University
[2] School of Computer Science and Engineering, Nanyang Technological University
[3] Tetras AI

Abstract. Current deep video quality assessment (VQA) methods are usually with high computational cost when evaluating high-resolution videos. This cost hinders them from learning better video-quality-related features via end-to-end training. Existing approaches usually apply naive sampling to reduce the computational cost, such as *resizing* and *cropping*. However, they cause artificial quality corruptions before the network and are thus not optimal for quality assessment. In this paper, we propose a new sampling scheme, Grid Mini-patch Sampling (GMS), that allows consideration of local quality by sampling patches at their raw resolution and covers global quality with contextual relations via mini-patches sampled in uniform grids. These mini-patches are spliced and aligned temporally, named as *fragments*. We further build the Fragment Attention Network (FANet) specially designed to accommodate *fragments* with Gated Relative Position Biases (GRPB) and Intra-Patch Non-Linear Regression (IP-NLR) modules. Consisting of *fragments* and FANet, the proposed FrAgment Sample Transformer for VQA (**FAST-VQA**) enables efficient end-to-end deep VQA and learns effective video-quality-related features. It improves state-of-the-art accuracy by around 10% while reducing 99.5% FLOPs on 1080P high-resolution videos. The newly learned video-quality-related features can also be transferred into smaller VQA datasets. Extensive experiments show that FAST-VQA has good performance on inputs of various resolutions while retaining high efficiency. We publish our code at https://github.com/timothyhtimothy/FAST-VQA.

Keywords: Video Quality Assessment, Quality-retained Sampling, End-to-End Learning, State-of-the-Art Performance, High Efficiency

1 Introduction

More and more videos with a variety of contents are collected in-the-wild and uploaded to the Internet every day. With the growth of high-definition video recording devices, a growing proportion of these videos are in high resolution (e.g. $\geq 1080P$). Classical video quality assessment (VQA) algorithms based on handcrafted features are difficult to handle these videos with diverse content and degradation. In recent years, deep-learning-based VQA methods [22,23,40,8,42,21]

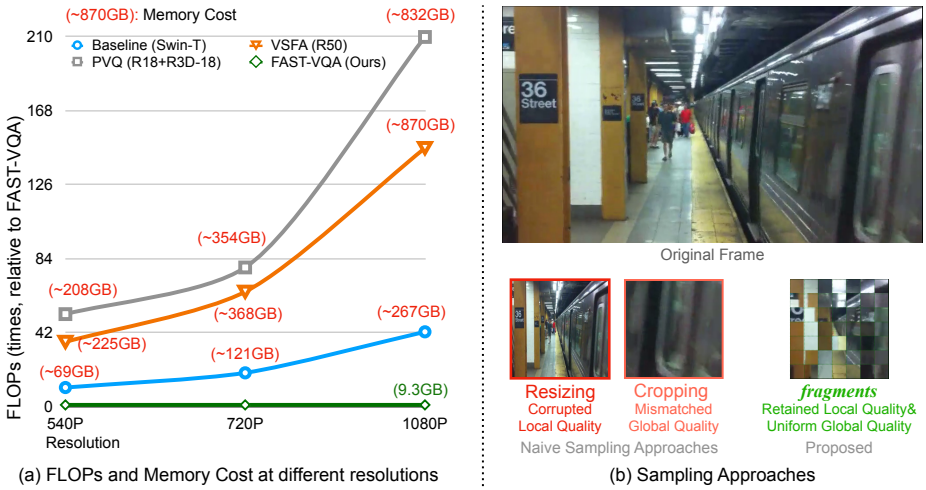

Fig. 1: Motivation for *fragments*: (a) The computational cost (FLOPs&Memory) for existing VQA methods is high especially on high-resolution videos. The memory costs shown in parenthesis are for batch size at 4. Complete numbers are listed in Tab. 4 and Tab. 16. (b) Sampling approaches. Naive approaches such as *resizing* [17,43] and *cropping* [14,15] cannot preserve video quality well. Zoom in for clearer view.

have shown better performance on in-the-wild VQA benchmarks [31,12,38,40]. However, the computational cost of deep VQA methods increases quadratically when applied to high resolution videos, and a video of size 1080×1920 would require $\mathbf{42.5}\times$ floating point operations (FLOPs) than normal 224×224 inputs (as Fig. 1(a) shows), limiting these methods from practical applications. It is urgent to develop new VQA methods that are both effective and efficient.

Meanwhile, with high memory cost noted in Fig. 1(a), existing methods usually regress quality scores with fixed features extracted from pre-trained networks for classification tasks [11,32,10] to alleviate memory shortage problem on GPUs instead of end-to-end training, preventing them from learning *video-quality-related features* that better represent quality information and limiting their accuracy. Existing approaches apply naive sampling on images or videos by resizing [17,43] or cropping [14,15] (as Fig. 1(b) shows) to reduce this cost and enable end-to-end training. However, they both cause artificial quality corruptions or changes during sampling, *e.g.*, resizing corrupts local textures that are significant for predicting video quality, while cropping causes mismatched global quality with local regions. Moreover, the severity of these problems increases with the raw resolution of the video, making them unsuitable for VQA tasks.

To improve the practical efficiency and the training effectiveness of deep VQA methods, we propose a new sampling scheme, Grid Mini-patch Sampling (GMS), to retain the sensitivity to original video quality. GMS cuts videos into spatially uniform non-overlapping grids, randomly sample a mini-patch from each grid, and then splice mini-patches together. In temporal view, we constrain the position of mini-patches to align across frames, in order to ensure the sensitivity

Fig. 2: *Fragments*, in spatial view (a) and temporal view (b). Zoom-in views of mini-patches show that *fragments* can retain spatial local quality information (a), and spot temporal variations such as shaking across frames (b). In (a), spliced mini-patches also keep the global scene information of original frames.

on temporal variations. We name these temporally aligned and spatially spliced mini-patches as *fragments*. As shown in Fig. 2, The proposed fragments can well preserve the sensitivity on both spatial and temporal quality. First, it preserves the local texture-related quality information (*e.g.*, spot blurs happened in *video 1/2*) by retaining the original resolution in patches. Second, benefiting from the globally uniformly sampled grids, it covers the global quality even though different regions have different qualities (*e.g.*, *video 3*). Third, by splicing the mini-patches, *fragments* retains contextual relations of patches so that the model can learn global scene information of the original frames. At last, with temporal alignment, *fragments* preserve temporal quality sensitivity by retaining the inter-frame variations in mini-patches from raw resolution, so they can be used to spot the temporal distortions in videos and distinguish between severely shaking videos (*e.g.*, *video 5*) from relatively stable shots (*e.g.*, *video 6*).

However, it is non-trivial to build a network using the proposed *fragments* as inputs. The network should follow two principles: 1) It should better extract the quality-related information preserved in *fragments*, including the retained local textures inside the raw resolution patches and the contextual relations between the spliced mini-patches; 2) It should distinguish the artificial discontinuity between mini-patches in *fragments* from the authentic quality degradation in the original videos. Based on these two principles, we propose a Fragment Attention Network (FANet) with Video Swin Transformer Tiny (Swin-T) [26] as the backbone. Swin-T has a hierarchical structure and processes inputs with patch-wise operations, which is naturally suitable for proceeding with proposed *fragments*.

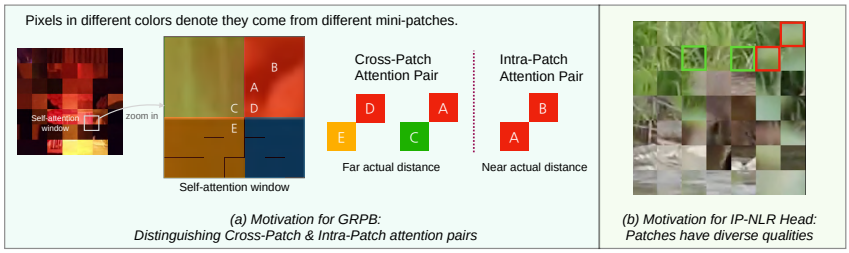

Fig. 3: Motivation for the two proposed modules in FANet: (a) Gated Relative Position Biases (GRPB); (b) Intra-Patch Non-Linear Regression (IP-NLR) head. The structures for the two modules are illustrated in Fig. 5.

Furthermore, to avoid the negative impact of discontinuity between mini-patches on quality prediction, we propose two novel modules, *i.e.*, Gated Relative Position Biases (GRPB) and Intra-Patch Non-Linear Regression (IP-NLR), to correct for the self-attention computation and the final score regression in the FANet respectively. Specifically, considering that some pairs in the same attention window might have the same relative position (*e.g.* Fig. 3(a) A-C, D-E, A-B), but the cross-patch attention pairs (A-C, D-E) are in far actual distances while intra-patch attention pairs (A-B) are in much nearer actual distances in the original video, we propose GRPB to explicitly distinguish these two kinds of attention pairs to avoid confusion of discontinuity between patches and authentic video artifacts. In addition, due to the discontinuity, different mini-patches contain diverse quality information (Fig. 3(b)), thus pooling operation before score regression applied in existing methods may confuse the information. To address this issue, we design IP-NLR as a quality-sensitive head, which first regresses the quality scores of mini-patches independently with non-linear layers and pools them after the regression.

In summary, we propose the FrAgment Sample Transformer for VQA (**FAST-VQA**), with the following contributions:

1. We propose *fragments*, a new sampling strategy for VQA that preserves both local quality and unbiased global quality with contextual relations via uniform Grid Mini-patch Sampling (GMS). The *fragments* can reduce the complexity of assessing 1080P videos by 97.6% and enables effective end-to-end training of VQA with quality-retained video samples.
2. We propose the Fragment Attention Network (FANet) to learn the local and contextual quality information from *fragments*, in which the Gated Relative Position Biases (GRPB) module is proposed to distinguish the intra-patch and cross-patch self-attention and the Intra-Patch Non-Linear Regression (IP-NLR) is proposed for better quality regression from *fragments*.
3. The proposed FAST-VQA can learn *video-quality-related features* efficiently through end-to-end training. These quality features help FAST-VQA to be **10%** more accurate than the existing state-of-the-art approaches and **8%** better than full-resolution Swin-T baseline with fixed recognition features. Through transfer learning, these quality features also significantly improve the best benchmark performance for small VQA datasets.

2 Related Works

Classical VQA Methods Classical VQA methods [30,28,20,35,34] handcrafted features to evaluate video quality. Among recent works, TLVQM [20] uses a combination of spatial high-complexity and temporal low-complexity handcraft features and VIDEVAL [35] ensembles different handcraft features to model the diverse authentic distortions. However, the reasons affecting the video quality are quite complicated and cannot be well captured with these handcrafted features.

Fixed-feature-based Deep VQA Methods Due to the extremely high computational cost of deep networks on high resolution videos, existing deep VQA methods train only a feature regression network with fixed deep features. Among them, VSFA [22] uses the features extracted by pre-trained ResNet-50 [11] from ImageNet-1k [5] and GRU [4] for temporal regression. MLSP-FF [8] also uses heavier Inception-ResNet-V2 [32] for feature extraction. Some methods [40,41] use the features extractor pre-trained with IQA datasets [13,39]. PVQ [40] also extracts features pretrained on action recognition dataset [16] for better perception on inter-frame distortion. These methods are limited by their high computational cost on high resolution videos. Additionally, without end-to-end training, fixed features pretrained by other tasks are not optimal for extracting quality-related information, which also limits the accuracy of quality assessment.

VQA Datasets Tab. 1 shows common VQA datasets, other video datasets and their sizes. The early VQA datasets [29,7] are synthesized with specialized distortion and have a very small volume. Some recent in-the-wild VQA datasets like KoNViD-1k [12], YouTube-UGC [38] and LIVE-VQC [31] are still small compared to datasets for other video tasks such as [16,2,9]. Recently, LSVQ[40], a large-scale VQA dataset with 39,076 videos is publicly available. With end-to-end deep learning of the proposed FAST-VQA, the *video-quality-related* features learnt on large-scale LSVQ dataset can be transferred into smaller VQA datasets to reach better performance.

Table 1: Common datasets in VQA and other video tasks. Most VQA datasets are much smaller than datasets for other VQA tasks.

Dataset	Task	Distortion Type	Size
Kinetics-400 [16]	Video Recognition	–	306,245
ActivityNet [2]	Video Action Localization	–	27,801
AVA [9]	Atomic Action Detection	–	386,000
CVD2014 [29]	Video Quality Assessment	Synthetic In-capture	234
KoNViD-1k [12]	Video Quality Assessment	In-the-wild	1,200
LIVE-VQC [31]	Video Quality Assessment	In-the-wild	585
Youtube-UGC [38]	Video Quality Assessment	In-the-wild	1,147
LSVQ [40]	Video Quality Assessment	In-the-wild	39,076

Vision Transformers Vision transformers [19,33,1,6,25] have shown effective on computer vision tasks. They cut images or videos into non-overlapping patches as input and perform self-attention operations between them. The patch-wise operations in vision transformers naturally distinguish the edges of mini-patches and are suitable for handling with the proposed *fragments*.

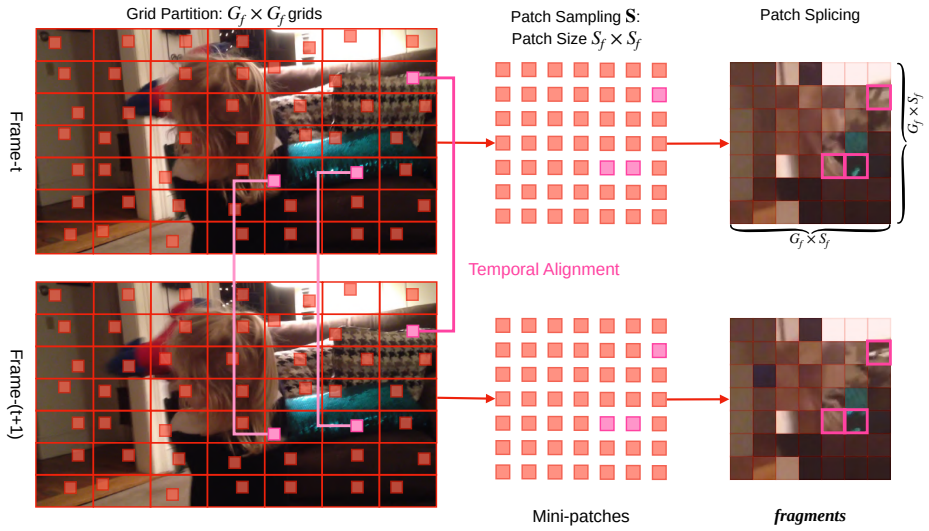

Fig. 4: The pipeline for sampling *fragments* with Grid Mini-patch Sampling (GMS), including grid partition, patch sampling, patch splicing, and temporal alignment. After GMS, the *fragments* are fed into the FANet (Fig. 5).

3 Approach

In this section, we introduce the full pipeline of the proposed FAST-VQA method. An input video is first sampled into *fragments* via Grid Mini-patch Sampling (GMS, Sec. 3.1). After sampling, the resultant fragments are fed into the Fragment Attention Network (FANet, Sec. 3.2) to get the final prediction of the video's quality. We introduce both parts in the following subsections.

3.1 Grid Mini-patch Sampling (GMS)

To well preserve the original video quality after sampling, we follow several important principles when designing the sampling process for *fragments*. We will illustrate the process along with these principles below.

Preserving global quality: uniform grid partition. To include each region for quality assessment and uniformly assess quality in different areas, we design the grid partition to cut video frames into uniform grids with each grid having the same size (as shown in Fig 4). We cut the t-th video frame $\mathcal{V}_t$ into $G_f \times G_f$ grids, denoted as $\mathcal{G}_t = \{g_t^{0,0}, ..g_t^{i,j}, ..g_t^{M-1,N-1}\}$, where $G_t^{i,j}$ denotes the grid for the i-th row and j-th column, and M and N are the total numbers of rows and columns, respectively. The partition process for each G_i is formalized as follows.

$$g_t^{i,j} = \mathcal{V}_t[\frac{i \times H}{G_f} : \frac{(i+1) \times H}{G_f}, \frac{j \times W}{G_f} : \frac{(j+1) \times W}{G_f}] \tag{1}$$

where H and W denote the height and weight of the video frame.

Preserving local quality: raw patch sampling. To preserve the local textures (*e.g.* blurs, noises, artifacts) that are vital in VQA, we select raw resolution patches without any resizing operations to represent local textural quality in grids. We employ random patch sampling to select one mini-patch $\mathcal{MP}_t^{i,j}$ of size of $S_f \times S_f$ from each grid $g_t^{i,j}$. The patch sampling process is as follows.

$$\mathcal{MP}_t^{i,j} = \mathbf{S}_t^{i,j}(G_t^{i,j}) \tag{2}$$

where $\mathbf{S}_t^{i,j}$ is the patch sampling operation for frame t and grid i, j.

Preserving temporal quality: temporal alignment. It is widely recognized by early works [18,20,40] that inter-frame temporal variations are influential to video qualities. To retain the raw temporal variations in videos (with T frames), we strictly align the sample areas during patch sampling operations $\mathbf{S}$ in different frames, as the following constraint shows.

$$\mathbf{S}_t^{i,j} = \mathbf{S}_{\hat{t}}^{i,j} \qquad \forall\ 0 \le t, \hat{t} < T,\ 0 \le i, j < G_f \tag{3}$$

Preserving contextual relations: patch splicing. Existing works [24,22,8] have shown that the global scene information and contextual information affects quality predictions. To keep the global scene information of the original videos, we keep the contextual relations of mini-patches by splicing them into their original positions, as the following equation shows:

$$\begin{aligned} \mathcal{F}_t^{i,j} &= \mathcal{F}_t[i \times S_f : (i+1) \times S_f, j \times S_f : (j+1) \times S_f] \\ &= \mathcal{MP}_t^{i,j}, \qquad\qquad 0 \le i, j < G_f \end{aligned} \tag{4}$$

where $\mathcal{F}$ denote the spliced and temporally aligned mini-patches after the Grid Mini-patch Sampling (GMS) pipeline, named as *fragments*.

3.2 Fragment Attention Network (FANet)

The Overall Framework. Fig. 5 shows the overall framework of FANet. It uses a Swin-T with four hierarchical self-attention layers as backbone. We also design the following modules to adapt it to fragments well.

Gated Relative Position Biases. Swin-T adds relative position bias (RPB) that uses learnable Relative Bias Table ($\mathbf{T}$) to represent the relative positions of pixels in attention pairs (QK^T). For *fragments*, however, as discussed in Fig. 3(a), the cross-patch pairs have much large actual distances than intra-patch pairs and should not be modeled with the same bias table. Therefore, we propose the gated relative position biases (GRPB, Fig. 5(b)) that uses learnable real position bias table ($\mathbf{T}_{\text{real}}$) and pseudo position bias table ($\mathbf{T}_{\text{pseudo}}$) to replace $\mathbf{T}$. The mechanisms of them are the same as $\mathbf{T}$ but they are learnt separately and used for intra-patch and cross-patch attention pairs respectively. Denote $\mathbf{G}$ as the intra-patch gate ($\mathbf{G}_{i,j} = 1$ if i, j are in the same mini-patch else $\mathbf{G}_{i,j} = 0$), the self-attention matrix (M_A) with GRPB is calculated as:

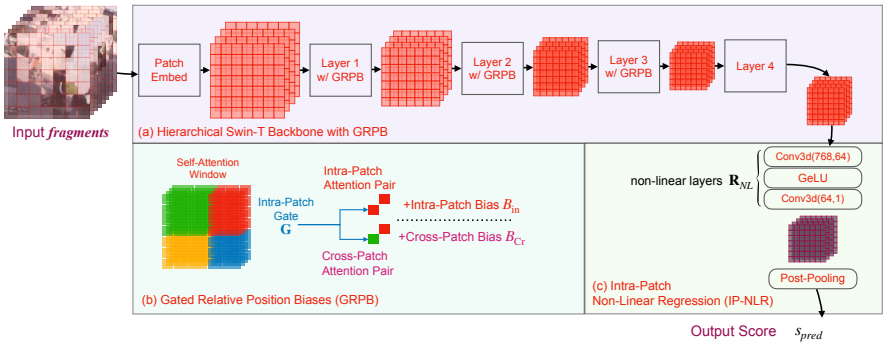

Fig. 5: The overall framework for FANet, including the Gated Relative Position Biases (GRPB) and Intra-Patch Non-Linear Regression (IP-NLR) modules. The input *fragments* come from Grid Mini-patch Sampling (Fig. 4).

$$B_{\mathrm{In},(i,j)} = \mathbf{T}_{\mathrm{real,FRP}(i,j)}; B_{\mathrm{Cr},(i,j)} = \mathbf{T}_{\mathrm{pseudo,FRP}(i,j)} \tag{5}$$

$$M_A = QK^T + \mathbf{G} \otimes B_{\mathrm{In}} + (\mathbf{1} - \mathbf{G}) \otimes B_{\mathrm{Cr}} \tag{6}$$

where $\mathrm{FRP}(i,j)$ is the relative position of pair (i,j) in *fragments*.

Intra-Patch Non-Linear Regression. As illustrated in Fig. 3(b), different mini-patches have diverse qualities due to discontinuity between them. If we pool features from different patches before regression, the quality representations of mini-patches will be confused with each other. To avoid this problem, we design the Intra-Patch Non-Linear Regression (IP-NLR, Fig. 5(c)) to regress the features via non-linear layers ($\mathbf{R}_{NL}$) first, and perform pooling following the regression. Denote features as f, output score as s_{pred}, pooling operation as Pool, the IP-NLR can be expressed as follows:

$$s_{pred} = \mathrm{Pool}(\mathbf{R}_{NL}(f)) \tag{7}$$

4 Experiments

In the experiment part, we conduct several experiments to evaluate and analyze the performance of the proposed FAST-VQA model.

4.1 Evaluation Setup

Implementation Details We use the Swin-T [26] pretrained on Kinetics-400 [16] dataset to initialize the backbone in FANet. As Tab. 2 shows, we implement two sampling densities for *fragments*: FAST-VQA (normal density) and FAST-VQA-M (lower density & higher efficiency), and accomodate window sizes in FANet to the input sizes. Without special notes, all ablation studies are on variants of FAST-VQA. We use PLCC (Pearson linear correlation coef.) and SRCC (Spearman rank correlation coef.) as metrics and use differentiable PLCC loss $l = \frac{(1-\mathrm{PLCC}(s_{pred},s_{gt}))}{2}$ as loss function. We set the training batch size as 16.

Table 2: Comparison of FAST-VQA and FAST-VQA-M with lower sampling density.

Methods	Number of Frames (T)	Patch Size (S_f)	Number of Grids (G_f)	Window Size in FANet	FLOPs	Parameters
FAST-VQA	32	32	7	(8,7,7)	279G	27.7M
FAST-VQA-M	16	32	4	(4,4,4)	46G	27.5M

Training & Benchmark Sets We use the large-scale $\text{LSVQ}_{\text{train}}$[40] dataset with 28,056 videos for training FAST-VQA. For evaluation, we choose 4 testing sets to test the model trained on LSVQ. The first two sets, $\text{LSVQ}_{\text{test}}$ and $\text{LSVQ}_{1080\text{p}}$ are official intra-dataset test subsets for LSVQ, while the $\text{LSVQ}_{\text{test}}$ consists of 7,400 various resolution videos from 240P to 720P, and $\text{LSVQ}_{1080\text{p}}$ consists of 3,600 1080P high resolution videos. We also evaluate the generalization ability of FAST-VQA on cross-dataset evaluations on KoNViD-1k [12] and LIVE-VQC [31], two widely-recognized in-the-wild VQA benchmark datasets.

4.2 Benchmark Results

Table 3: Comparison with existing methods (classical and deep) and our baseline (Full-res Swin-T *features*). The 1st/2nd best scores are colored in **red** and blue, respectively.

Type/		Intra-dataset Test Sets				Cross-dataset Test Sets			
Testing Set/		**$\text{LSVQ}_{\text{test}}$**		**$\text{LSVQ}_{1080\text{p}}$**		**KoNViD-1k**		**LIVE-VQC**	
Groups	Methods	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
Existing Classical	BRISQUE[27]	0.569	0.576	0.497	0.531	0.646	0.647	0.524	0.536
	TLVQM[20]	0.772	0.774	0.589	0.616	0.732	0.724	0.670	0.691
	VIDEVAL[35]	0.794	0.783	0.545	0.554	0.751	0.741	0.630	0.640
Existing Deep	VSFA[22]	0.801	0.796	0.675	0.704	0.784	0.794	0.734	0.772
	$\text{PVQ}_{\text{wo/ patch}}$[40]	0.814	0.816	0.686	0.708	0.781	0.781	0.747	0.776
	$\text{PVQ}_{\text{w/ patch}}$[40]	0.827	0.828	0.711	0.739	0.791	0.795	0.770	0.807
Full-res Swin-T[26] *features*		0.835	0.833	0.739	0.753	0.825	0.828	0.794	0.809
FAST-VQA-M (Ours)		0.852	0.854	0.739	0.773	0.841	0.832	0.788	0.810
FAST-VQA (Ours)		**0.876**	**0.877**	**0.779**	**0.814**	**0.859**	**0.855**	**0.823**	**0.844**
Improvement to $\text{PVQ}_{\text{w/ patch}}$		*+6%*	*+6%*	*+10%*	*+10%*	*+9%*	*+8%*	*+7%*	*+5%*

In Tab. 3, we compare with existing classical and deep VQA methods and our baseline, the full-resolution Swin-T with feature regression instead of end-to-end training (denoted as 'Full-res Swin-T *features*'). With its video-quality-related features, FAST-VQA achieves at most 10% improvement to PVQ, the existing state-of-the-art on $\text{LSVQ}_{1080\text{p}}$. Even the efficient version FAST-VQA-M can outperform existing state-of-the-art. FAST-VQA also shows significant improvement to its fixed-feature-based baseline with the same backbone, demonstrating that the proposed new quality-retained sampling with end-to-end training scheme for VQA is not only much more efficient (with only 2.36% FLOPs required on 1080P videos) but also notably more accurate (with 8.10% improvement on PLCC metric for $\text{LSVQ}_{1080\text{p}}$) than the existing fixed-feature-based paradigm.

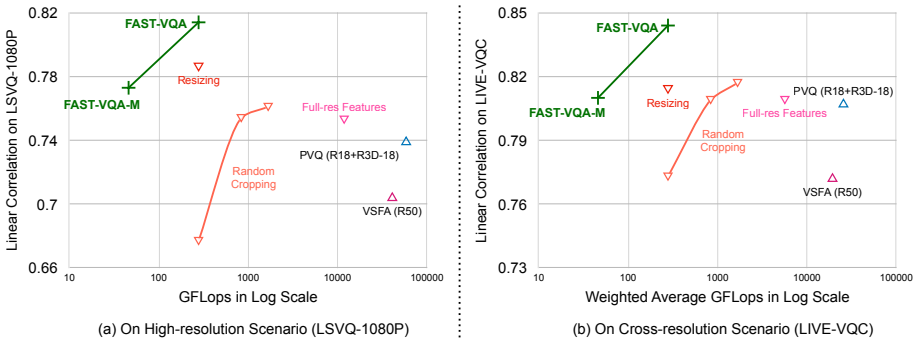

Fig. 6: The Performance-FLOPs curve of proposed **FAST-VQA** and baseline methods.

Table 4: FLOPs and running time (on GPU/CPU, average of ten runs) comparison of FAST-VQA, state-of-the-art methods and our baseline on different resolutions. We **boldface** FLOPs $\leq$ 500G and running time $\leq$ 1s.

Method	540P		720P		1080P	
	FLOPs(G)	Time(s)	FLOPs(G)	Time(s)	FLOPs(G)	Time(s)
VSFA[22]	$10249_{36.7\times}$	2.603/92.761	$18184_{65.2\times}$	3.571/134.9	$40919_{147\times}$	11.14/465.6
PVQ[40]	$14646_{52.5\times}$	3.091/97.85	$22029_{79.0\times}$	4.143/144.6	$58501_{210\times}$	13.79/538.4
Full-res Swin-T[26] *feat.*	$3032_{10.9\times}$	3.226/102.0	$5357_{19.2\times}$	5.049/166.2	$11852_{42.5\times}$	8.753/234.9
FAST-VQA (Ours)	$\mathbf{279}_{1\times}$	**0.044**/9.019	$\mathbf{279}_{1\times}$	**0.043**/9.530	$\mathbf{279}_{1\times}$	**0.045**/9.142
FAST-VQA-M (Ours)	$\mathbf{46}_{0.165\times}$	**0.019/0.729**	$\mathbf{46}_{0.165\times}$	**0.019/0.613**	$\mathbf{46}_{0.165\times}$	**0.019/0.714**

4.3 Efficiency of FAST-VQA

To demonstrate the efficiency of FAST-VQA, we compare the FLOPs and running times on CPU/GPU (average of ten runs per sample) of the proposed FAST-VQA with existing approaches on different resolutions, see Tab. 4. We also draw the performance-FLOPs curve on LSVQ_{1080p} and LIVE-VQC in Fig. 6. As we can see, FAST-VQA reduces up to 210× FLOPs and 247× running time than PVQ while obtaining notably better performance. The more efficient version, FAST-VQA-M, only requires 1/1273 FLOPs of PVQ and 1/258 FLOPs of our full-resolution baseline while still achieving slightly better performance. Moreover, FAST-VQA (especially FAST-VQA-M) also runs very fast even on CPU, which reduces the hardware requirements for the applications of deep VQA methods. All these comparisons show the unprecedented efficiency of proposed FAST-VQA.

4.4 Transfer Learning with video-quality-related features

FAST-VQA also makes the pretrain-finetune scheme on VQA possible with affordable computation resources. With FAST-VQA, we can pretrain with large VQA datasets in end-to-end manner to learn quality related features, and then transfer to specific VQA scenarios where only small datasets are available. We use

Table 5: The finetune results on LIVE-VQC, KoNViD, CVD2014 and YouTube-UGC datasets, compared with existing classical and deep VQA methods.

Finetune Dataset/		LIVE-VQC		KoNViD-1k		CVD2014		YouTube-UGC	
Groups	Methods	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
Existing Classical	TLVQM[20]	0.799	0.803	0.773	0.768	0.83	0.85	0.669	0.659
	VIDEVAL[35]	0.794	0.783	0.783	0.780	—	—	0.779	0.773
	RAPIQUE[34]	0.755	0.786	0.803	0.817	—	—	0.759	0.768
Existing Deep	VSFA[22]	0.773	0.795	0.773	0.775	0.870	0.868	0.724	0.743
	PVQ[40]	0.827	0.837	0.791	0.786	—	—	—	—
	GST-VQA[3]	—	—	0.814	0.825	0.831	0.844	—	—
	CoINVQ[37]	—	—	0.767	0.764	—	—	0.816	0.802
Full-res Swin-T[26] *features*		0.799	0.808	0.841	0.838	0.868	0.870	0.798	0.796
FAST-VQA-M (Ours)		0.803	0.828	0.873	0.872	0.877	0.892	0.768	0.765
FAST-VQA (ours)		**0.849**	**0.865**	**0.891**	**0.892**	**0.891**	**0.903**	**0.855**	**0.852**

LSVQ as the large dataset and choose four small datasets representing diverse scenarios, including LIVE-VQC (real-world mobile photography, 240P-1080P), KoNViD-1k (various contents collected online, 540P), CVD2014 (synthetic in-capture distortions, 480P-720P) and YouTube-UGC (user-generated contents, including computer graphic contents, 360P-2160P[4]). We divide each dataset into random splits for 10 times and report the average result on the test splits. As Tab. 5 shows, with video-quality-related features, the proposed FAST-VQA outperforms the existing state-of-the-arts by a large margin. We believe our pre-trained FAST-VQA can serve as a strong backbone network and boost scenario-specific VQA research. The full comparison and analysis of transfer learning results are in Appendix Sec. A.2.

4.5 Ablation Studies on *fragments*

For the first part of ablation studies, we prove the effectiveness of *fragments* by comparing with other common sampling approaches (Tab. 6) and different variants of fragments (Tab. 6). We keep the FANet structure fixed during this part.

Comparing with resizing/cropping In Group 1 of Tab. 6, we compare the proposed fragments with two common sampling approaches: *bilinear resizing* and *random cropping*. The proposed *fragments* are notably better than bilinear resizing on **high-resolution** (LSVQ_{1080p}) (+4%) and **cross-resolution** (LIVE-VQC) scenarios (+4%). Fragments still lead to non-trivial 2% improvements than resizing on lower-resolution scenarios where the problems of resizing is not that severe. This proves that keeping local textures is vital for VQA. Fragments also largely outperform single random crop as well as ensemble of multiple crops, suggesting that retaining the uniform global quality is also critical to VQA.

[4] Due to privacy reasons, the current public version of YouTube-UGC is incomplete and only with 1147 videos. The peer comparison is only for reference.

Table 6: Ablation study on *fragments*: comparison with resizing, cropping (Group 1) and different variants for fragments (Group 2).

Testing Set/	**LSVQ$_{test}$**		**LSVQ$_{1080p}$**		**KoNViD-1k**		**LIVE-VQC**	
Methods/Metric	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
Group 1: Naive Sampling Approaches								
bilinear resizing	0.857	0.859	0.752	0.786	0.841	0.840	0.772	0.814
random cropping	0.807	0.812	0.643	0.677	0.734	0.776	0.740	0.773
- test with 3 crops	0.838	0.835	0.727	0.754	0.841	0.827	0.785	0.809
- test with 6 crops	0.843	0.844	0.734	0.761	0.845	0.834	0.796	0.817
Group 2: Variants of *fragments*								
random mini-patches	0.857	0.861	0.754	0.790	0.844	0.845	0.792	0.818
shuffled mini-patches	0.858	0.863	0.761	0.799	0.849	0.847	0.796	0.821
w/o temporal alignment	0.850	0.853	0.736	0.779	0.823	0.816	0.764	0.802
fragments (ours)	**0.876**	**0.877**	**0.779**	**0.814**	**0.859**	**0.855**	**0.823**	**0.844**

Comparing with variants of fragments We also compare with three variants of *fragments* in Tab. 6, Group 2. We prove the effectiveness of uniform grid partition by comparing with *random mini-patches* (ignore grids while sampling), and the importance of retaining contextual relations by comparing with *shuffled mini-patches*. Fragments show notable improvements than both variants. Moreover, the proposed fragments show much better performance than the variant *without* temporal alignment especially on high resolution videos, suggesting that preserving the inter-frame temporal variations is necessary for fragments.

Table 7: Ablation study on FANet design: the effects for GRPB and IP-NLR modules.

Testing Set/	**LSVQ$_{test}$**		**LSVQ$_{1080p}$**		**KoNViD-1k**		**LIVE-VQC**	
Variants/Metric	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
w/o GRPB	0.873	0.872	0.769	0.805	0.854	0.853	0.808	0.832
semi-GRPB on Layer 1/2	0.873	0.875	0.772	0.809	0.856	0.851	0.812	0.838
linear Regression	0.872	0.873	0.768	0.803	0.847	0.849	0.810	0.835
PrePool non-linear Regression	0.873	0.874	0.771	0.805	0.851	0.850	0.813	0.834
FANet (ours)	**0.876**	**0.877**	**0.779**	**0.814**	**0.859**	**0.855**	**0.823**	**0.844**

4.6 Ablation Studies on FANet

Effects of GRPB and IP-NLR In the second part of ablation studies, we analyze the effects of two important designs in FANet: the proposed Gated Relative Position Biases (GRPB) and Intra-Patch Non-Linear Regression (IP-NLR) VQA Head as in Tab. 7. We compare the IP-NLR with two variants: the linear regression layer and the non-linear regression layers with pooling before regression (*PrePool*). Both modules lead to non-negligible improvements especially on high-resolution (LSVQ$_{1080p}$) or cross-resolution (LIVE-VQC) scenarios. As the discontinuity between mini-patches is more obvious in high-resolution videos, this result suggests that the corrected position biases and regression head are helpful on solving the problems caused by such discontinuity.

4.7 Reliability and Robustness Analyses

As FAST-VQA is based on samples rather than original videos while a single sample for *fragments* only keeps 2.4% spatial information in 1080P videos, it is important to analyze the reliability and robustness of FAST-VQA predictions.[5]

Reliability of Single Sampling. We measure the reliability of single sampling in FAST-VQA by two metrics: 1) the assessment stability of different single samplings on the same video; 2) the relative accuracy of single sampling compared with multiple sample ensemble. As shown in Tab. 8, the normalized *std. dev.* of different sampling on a same video is only around 0.01, which means the sampled fragments are enough to make very stable predictions. Compared with 6-sample ensemble, sampling only once can already be 99.40% as accurate even on the pure high-resolution test set (LSVQ_{1080P}). They prove that a single sample of *fragments* is enough stable and reliable for quality assessment even though only a small proportion of information is kept during sampling.

Table 8: Assessment stability and relative accuracy of single sampling of *fragments*.

Testing Set/	**$\text{LSVQ}_{\text{test}}$**	**LSVQ_{1080p}**	**KoNViD-1k**	**LIVE-VQC**
Score Range	0-100	0-100	1-5	0-100
std. dev. of Single Samplings	0.65	0.79	0.046	1.07
Normalized *std. dev.*	0.0065	0.0079	0.0115	0.0107
Avg. KRCC on Single Sampling	0.6918	0.5862	0.6693	0.6296
KRCC on 6-sample ensemble	0.6947	0.5897	0.6730	0.6326
Relative Accuracy	99.59%	99.40%	99.45%	99.52%

Robustness on Different Resolutions To analyze the robustness of FAST-VQA on different resolutions, we divide the cross-resolution VQA benchmark set LIVE-VQC into three resolution groups: (A) 1080P (110 videos); (B) 720P (316 videos); (C) $\leq$540P (159 videos) to see the performance of FAST-VQA on different resolutions, compared with several variants. As the results shown in Tab. 9, the proposed FAST-VQA shows good performance ($\geq$ 0.80 SRCC&PLCC) on all resolution groups and most superior improvement than other variants on Group (A) with 1080P high-resolution videos, proving that FAST-VQA is robust and reliable on different resolutions of videos.

Table 9: Performance comparison on different resolution groups of LIVE-VQC dataset.

Resolution	(A): 1080P			(B): 720P			(C): $\leq$540P		
Variants	SRCC	PLCC	KRCC	SRCC	PLCC	KRCC	SRCC	PLCC	KRCC
Full-res Swin *features*	0.771	0.774	0.584	0.796	0.811	0.602	0.810	0.853	0.625
bilinear resizing	0.758	0.773	0.573	0.790	0.822	0.599	0.835	0.878	0.650
random cropping	0.765	0.768	0.565	0.774	0.787	0.581	0.730	0.809	0.535
w/o GRPB	0.796	0.785	0.598	0.802	0.820	0.608	0.834	0.883	0.649
FAST-VQA (Ours)	0.807	0.806	0.610	0.803	0.825	0.610	0.840	0.885	0.654

[5] In this part, we use KRCC (Kendall rank-order correlation coef., measuiring the pair-prediction accuracy rate) as an extra metric. Details in Appendix.

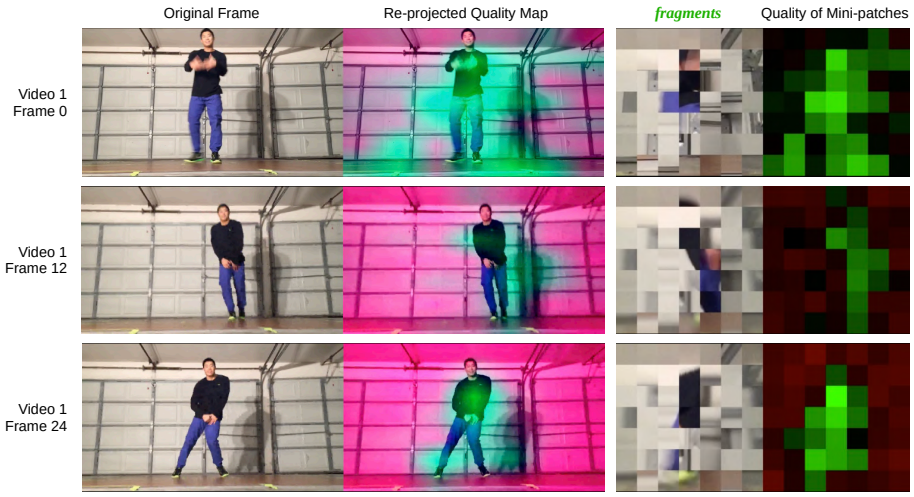

Fig. 7: Spatial-temporal patch-wise local quality maps, where **red** areas refer to low predicted quality and **green** areas refer to high predicted quality. This sample video is a 1080P video selected from LIVE-VQC [31] dataset. Zoom in for clearer view.

4.8 Qualitative Results

Patch-wise Local Quality Maps The proposed IP-NLR head with patch-wise independent quality regression not only improves the performance of FAST-VQA, but also enables it to generate patch-wise local quality maps as [40] does. These quality maps help us to qualitatively evaluate what can be learned during the end-to-end training for FAST-VQA. We show the patch-wise local quality maps and the re-projected frame quality maps for a 1080P video (from LIVE-VQC [31] dataset) in Fig. 7. As the patch-wise quality maps and re-projected quality maps in Fig. 7 (column 2&4) shows, FAST-VQA is sensitive to textural quality information and distinguishes between clear (Frame 0) and blurry textures (Frame 12/24). It demonstrates that FAST-VQA with *fragments* (column 3) as input is sensitive to local texture quality. Furthermore, the qualities of the action-related areas are notably different from the background areas, showing that FAST-VQA effectively learns the global scene information and contextual relations in the video. We show more quality maps in our project page.

5 Conclusions

Our paper has shown that proposed *fragments* are effective samples for video quality assessment (VQA) that better retain quality information in videos than naive sampling approaches, to tackle the difficulties as results of high computing and memory requirements when high-resolution videos are to be evaluated. Based on *fragments*, the proposed end-to-end FAST-VQA achieves higher efficiency (-99.5% FLOPs) and accuracy ($+10\%$ PLCC) simultaneously than existing state-of-the-art method PVQ on 1080P videos. We hope that the FAST-VQA can bring deep VQA methods into practical use for videos in any resolutions.

A Additional Results

A.1 Result with Patch Labels

The LSVQ dataset crops videos into spatial-temporal crops and additionally labels the qualities of crops, while PVQ [40] uses these *patch labels* for training and gets better performance. We also use these additional labels for training, by regarding the video crops as independent training videos, the same way as PVQ does. As shown in Tab. 10, the patch labels contribute to performance on cross-dataset scenarios such as KonViD-1k and LIVE-VQC, but reduce the intra-dataset performance on LSVQ. This might be due to the '*patching up*' adds domain gap between the original videos in test set and cropped video patches in training set, but lead to cross-dataset gain due to data augmentation effect. We report the result in our main paper without these additional labels, and will make future explorations on how these patch labels can be better used.

Table 10: Results with additional patch labels.

Testing Set/	**$LSVQ_{test}$**		**$LSVQ_{1080p}$**		**KoNViD-1k**		**LIVE-VQC**	
Methods	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
PVQ (*w/o* patch)[40]	0.814	0.816	0.686	0.708	0.781	0.781	0.747	0.776
+ *patch labels*	0.827	0.828	0.711	0.739	0.791	0.795	0.770	0.807
FAST-VQA-M (Ours)	0.852	0.854	0.739	0.773	0.841	0.832	0.788	0.810
+ *patch labels*	0.846	0.849	0.738	0.776	0.850	0.847	0.790	0.818
FAST-VQA(ours)	**0.876**	**0.877**	**0.779**	**0.814**	0.859	0.855	0.823	0.844
+ *patch labels*	0.867	0.869	0.767	0.800	**0.862**	**0.860**	**0.826**	**0.845**

A.2 Full Results on Finetune Experiments

KoNViD-1k KoNViD-1k [12] is the most commonly recognized in-the-wild VQA benchmark dataset, containing 1200 videos in the same resolution. As shown in Tab 11 KoNViD-1k, FAST-VQA can already reach good performance without video-quality-related feature, while the features contribute to 6% significant improvement. It further proves the effectiveness of these features.

LIVE-VQC LIVE-VQC [31] dataset is a cross-resolution in-the-wild dataset which contains 20% 1080P high-resolution videos. Tab. 12 shows the performance comparison on LIVE-VQC dataset. Limited by the size and diversity of the dataset, FAST-VQA cannot reach state-of-the-art performance without the video-quality-related features, yet with the features FAST-VQA can significantly outperform existing methods by a large margin.

CVD2014 CVD2014 [29] dataset is a synthetic VQA dataset that focus on in-capture distortions. Tab. 13 shows the performance comparison on CVD2014 dataset, where the FAST-VQA reaches state-of-the-art even without video-quality-related features and video-quality-related features from LSVQ training still further improves 1.5% performance.

Table 11: The mean and (*std. dev.*) of results on KoNViD-1k dataset with video-quality-related features, compared with classical & deep VQA methods.

KoNViD-1k (1200)	SRCC(std)	PLCC(std)	KRCC(std)
V-BLIINDS[30]	0.710(0.031)	0.704(0.030)	0.519(0.026)
TLVQM[20]	0.773(0.024)	0.768(0.023)	0.577(0.022)
VIDEVAL[35]	0.783(0.021)	0.780(0.021)	0.585(0.021)
RAPIQUE[34]	0.803	0.817	—
3D-CNN + LSTM[42]	0.808	0.800	—
PVQ[40]	0.791	0.786	—
VSFA[22]	0.773(0.019)	0.775(0.019)	0.578(0.019)
GST-VQA[3]	0.814(0.026)	0.825(0.043)	0.621(0.027)
MLSP-FF[8]	0.82(0.02)	—	—
Full-res Swin-T[26] *features*	0.841(0.018)	0.838(0.025)	0.648(0.019)
FAST-VQA-M$_{\text{w/o VQ-related features}}$	0.825(0.014)	0.825(0.013)	0.634(0.017)
FAST-VQA-M$_{\text{w/ VQ-related features}}$	0.873(0.012)	0.872(0.012)	0.689(0.015)
FAST-VQA$_{\text{w/o VQ-related features}}$	0.842(0.012)	0.844(0.011)	0.651(0.015)
FAST-VQA$_{\text{w/ VQ-related features}}$	**0.891(0.009)**	**0.892(0.008)**	**0.715(0.011)**

Table 12: The mean and (*std. dev.*) of results on LIVE-VQC dataset with video-quality-related features, compared with classical & deep VQA methods.

LIVE-VQC (585)	SRCC(std)	PLCC(std)	KRCC(std)
V-BLIINDS[30]	0.694(0.050)	0.718(0.050)	0.508(0.042)
TLVQM[20]	0.799(0.036)	0.803(0.036)	0.608(0.037)
VIDEVAL[35]	0.752(0.039)	0.751(0.042)	0.564(0.036)
RAPIQUE[34]	0.755	0.786	—
PVQ[40]	0.827	0.837	—
VSFA[22]	0.773(0.027)	0.795(0.026)	0.581(0.031)
MLSP-FF[8]	0.72(0.06)	—	—
Full-res Swin-T[26] *features*	0.799(0.033)	0.808(0.028)	0.613(0.036)
FAST-VQA-M$_{\text{w/o VQ-related features}}$	0.754(0.034)	0.772(0.027)	0.563(0.031)
FAST-VQA-M$_{\text{w/ VQ-related features}}$	0.803(0.032)	0.828(0.030)	0.614(0.033)
FAST-VQA$_{\text{w/o VQ-related features}}$	0.765(0.039)	0.782(0.034)	0.573(0.039)
FAST-VQA$_{\text{w/ VQ-related features}}$	**0.849(0.025)**	**0.865(0.019)**	**0.664(0.028)**

Table 13: The mean and (*std. dev.*) of results on CVD2014 dataset with video-quality-related features, compared with classical & deep VQA methods.

CVD2014 (234)	SRCC(std)	PLCC(std)	KRCC(std)
V-BLIINDS[30]	0.746(0.056)	0.753(0.053)	0.562(0.057)
TLVQM[20]	0.83(0.04)	0.85(0.04)	—
VSFA[22]	0.870(0.037)	0.868(0.032)	0.695(0.047)
MLSP-FF[8]	0.77(0.06)	—	—
Full-res Swin-T[26] *features*	0.869(0.027)	0.878(0.025)	0.698(0.035)
FAST-VQA-M$_{\text{w/o VQ-related features}}$	0.857(0.028)	0.867(0.019)	0.674(0.029)
FAST-VQA-M$_{\text{w/ VQ-related features}}$	0.877(0.035)	0.892(0.019)	0.705(0.041)
FAST-VQA$_{\text{w/o VQ-related features}}$	0.871(0.032)	0.888(0.017)	0.699(0.040)
FAST-VQA$_{\text{w/ VQ-related features}}$	**0.891(0.030)**	**0.903(0.019)**	**0.721(0.031)**

YouTube-UGC YouTube-UGC[38,36] dataset is a challenging dataset for FAST-VQA as it contains 10% **4K** videos with 20-second duration, where FAST-VQA only samples 0.1% information on these videos for evaluation (FAST-VQA-M only samples 0.02%). However, the proposed FAST-VQA still reaches state-of-the-art performance, as shown in Tab. 14, while the video-quality-related features contribute to more than 10% improvement.We also note that FAST-VQA-M *does not perform well* on this dataset, which suggests that the density of FAST-VQA-M cannot well handle 4K videos.

Table 14: The mean and (*std. dev.*) of results on YouTube-UGC dataset with video-quality-related features, compared with classical & deep VQA methods.

YouTube-UGC (1147)	SRCC(std)	PLCC(std)	KRCC(std)
TLVQM[20]	0.669(0.030)	0.659(0.030)	0.482(0.025)
RAPIQUE[34]	0.759	0.768	
VIDEVAL[35]	0.779(0.025)	0.773(0.025)	0.583(0.023)
VSFA[22]	0.724	0.743	—
CoINVQ[37]	0.816	0.802	—
Full-resolution Swin-T *features*[26]	0.798(0.027)	0.796(0.021)	0.603(0.024)
FAST-VQA-M$_{\text{w/o VQ-related features}}$	0.645(0.029)	0.628(0.034)	0.459(0.025)
FAST-VQA-M$_{\text{w/ VQ-related features}}$	0.768(0.020)	0.765(0.019)	0.572(0.022)
FAST-VQA$_{\text{w/o VQ-related features}}$	0.794(0.016)	0.784(0.016)	0.596(0.017)
FAST-VQA$_{\text{w/ VQ-related features}}$	**0.855(0.008)**	**0.852(0.011)**	**0.667(0.012)**

A.3 Resolution Sensitivity

To demonstrate that keeping the original resolution during sampling is quite important, we use LSVQ$_{1080p}$ test set in downsample evaluation to prove that *fragments* and FAST-VQA are sensitive to resolution changes: We resize these 1080P high-resolution videos into 540P(2X↓), 360P(3X↓), 270P(4X↓) and sample *fragments* from the resized videos. As the results in Tab. 15 shows, the proposed FAST-VQA is sensitive to large-scale downsampling, and suffer around 20% accuracy drop when we try to downsample these videos in large strides into lower resolution. This proves that keeping the raw resolution patches are important in sampling *fragments*.

Table 15: Relative accuracy drop on downsampling videos before FAST-VQA.

Resolution	*Original*		540P (2X↓)		360P (3X↓)		270P (4X↓)	
Results	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
Correlation *with GT Labels*	0.779	0.814	0.745	0.794	0.637	0.716	0.566	0.646
– *accuracy drop with* ↓	–	–	4.4%	2.5%	18.3%	12.1%	27.4%	20.7%
Correlation *with Original*	1.000	1.000	0.875	0.902	0.700	0.771	0.602	0.676

B Other Technical Details

B.1 Metrics & Loss Functions

Metrics We use three metrics, PLCC, SRCC and KRCC, for evaluating the accuracy of quality predictions. The Pearson Linear Correlation Coefficient (**PLCC**) computes the linear correlation of predicted scores and ground truth scores in a series. The Spearman Rank-order Correlation Coefficient (**SRCC**) will first rank the labels in both series and computes the PLCC between the two rank series. The Kendall Rank-order Correlation Coefficient (**KRCC**) computes the rank-pair accuracy and measures how much proportion relative relations of score pairs are predicted correctly.

Loss Functions We use the differentiable PLCC-induced loss

$$\mathrm{PLCC}(a,b) = \frac{\sum_i (a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_i (a_i - \overline{a})^2 \sum_i (b_i - \overline{b})^2}} \tag{8}$$

$$l = \frac{(1 - \mathrm{PLCC}(s_{pred}, s_{gt}))}{2} \tag{9}$$

as the training loss function, where $\overline{a}$ denotes the mean value for a, and s_{pred} and s_{gt} refers to predictions and ground truth labels in a batch.

B.2 Batch Size & Training/Inference Memory Cost

We show the training and inference memory cost for FAST-VQA and FAST-VQA-M in Tab. 16 and Tab. 17. The FAST-VQA with batch size $B = 16$ will require around 28GB graphic memory to be trained, which needs one Tesla V100 (32GB) GPU or four GTX1080Ti GPUs. We also provide versions for FAST-VQA with smaller batch sizes, which will require less graphic memory but might need more time for convergence during training. The more efficient variant, FAST-VQA-M with batch size $B = 16$, will only require 3.2GB graphic memory for training, and can be easily reproduced with most current GPU devices. During inference, both variants require affordable memory costs.

Table 16: Training memory costs with different batch sizes.

Method	Input Size	Attn. Window Size	Batch Size	Training Cost
FAST-VQA	(32,224,224)	(8,7,7)	16	28.0GB
FAST-VQA-B8	(32,224,224)	(8,7,7)	8	15.8GB
FAST-VQA-B4	(32,224,224)	(8,7,7)	4	9.3GB
FAST-VQA-M	(16,128,128)	(4,4,4)	16	3.2GB
FAST-VQA-M-B8	(16,128,128)	(4,4,4)	8	2.1GB

Table 17: Inference memory costs of FAST-VQA and FAST-VQA-M.

Method	Input Size	Attn. Window Size	Inference Cost
FAST-VQA	(4,32,224,224)	(8,7,7)	6.2GB
FAST-VQA-M	(4,16,128,128)	(4,4,4)	2.5GB

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lucic, M., Schmid, C.: Vivit: A video vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6836–6846 (October 2021)
2. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
3. Chen, B., Zhu, L., Li, G., Lu, F., Fan, H., Wang, S.: Learning generalized spatial-temporal deep feature representation for no-reference video quality assessment. IEEE Transactions on Circuits and Systems for Video Technology (2021)
4. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1724–1734. ACL (2014)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
6. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6824–6835 (October 2021)
7. Ghadiyaram, D., Pan, J., Bovik, A.C., Moorthy, A.K., Panda, P., Yang, K.C.: In-capture mobile video distortions: A study of subjective behavior and objective algorithms. IEEE Transactions on Circuits and Systems for Video Technology **28**(9), 2061–2077 (2018)
8. Götz-Hahn, F., Hosu, V., Lin, H., Saupe, D.: Konvid-150k: A dataset for no-reference video quality assessment of videos in-the-wild. In: IEEE Access 9. pp. 72139–72160. IEEE (2021)
9. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
10. Hara, K., Kataoka, H., Satoh, Y.: Learning spatio-temporal features with 3d residual networks for action recognition. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 3154–3160 (2017)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
12. Hosu, V., Hahn, F., Jenadeleh, M., Lin, H., Men, H., Szirányi, T., Li, S., Saupe, D.: The konstanz natural video database (konvid-1k). In: Ninth International Conference on Quality of Multimedia Experience (QoMEX). pp. 1–6 (2017)
13. Hosu, V., Lin, H., Sziranyi, T., Saupe, D.: Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment. IEEE Transactions on Image Processing **29**, 4041–4056 (2020)
14. Kang, L., Ye, P., Li, Y., Doermann, D.: Convolutional neural networks for no-reference image quality assessment. IEEE conference on computer vision and pattern recognition (2014)

15. Kang, L., Ye, P., Li, Y., Doermann, D.: Simultaneous estimation of image quality and distortion via multi-task convolutional neural networks. IEEE international conference on image processing (ICIP) (2015)
16. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, A., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. ArXiv **abs/1705.06950** (2017)
17. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: Multi-scale image quality transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5148–5157 (October 2021)
18. Kim, W., Kim, J., Ahn, S., Kim, J., Lee, S.: Deep video quality assessor: From spatio-temporal visual sensitivity to a convolutional neural aggregation network. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
19. Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., Zhai, X.: An image is worth 16x16 words: Transformers for image recognition at scale (2021)
20. Korhonen, J.: Two-level approach for no-reference consumer video quality assessment. IEEE Transactions on Image Processing **28**(12), 5923–5938 (2019)
21. Korhonen, J., Su, Y., You, J.: Blind natural video quality prediction via statistical temporal features and deep spatial features. In: Proceedings of the 28th ACM International Conference on Multimedia. p. 3311–3319. MM '20, Association for Computing Machinery, New York, NY, USA (2020)
22. Li, D., Jiang, T., Jiang, M.: Quality assessment of in-the-wild videos. In: Proceedings of the 27th ACM International Conference on Multimedia. p. 2351–2359. MM '19, Association for Computing Machinery, New York, NY, USA (2019)
23. Li, D., Jiang, T., Jiang, M.: Unified quality assessment of in-the-wild videos with mixed datasets training. International Journal of Computer Vision **129**(4), 1238–1257 (2021)
24. Li, D., Jiang, T., Lin, W., Jiang, M.: Which has better visual quality: The clear blue sky or a blurry animal? IEEE Transactions on Multimedia **21**(5), 1221–1234 (2019)
25. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
26. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. arXiv preprint arXiv:2106.13230 (2021)
27. Mittal, A., Moorthy, A.K., Bovik, A.C.: No-reference image quality assessment in the spatial domain. IEEE Transactions on Image Processing **21**(12), 4695–4708 (2012)
28. Mittal, A., Saad, M.A., Bovik, A.C.: A completely blind video integrity oracle. IEEE Transactions on Image Processing **25**(1), 289–300 (2016)
29. Nuutinen, M., Virtanen, T., Vaahteranoksa, M., Vuori, T., Oittinen, P., Häkkinen, J.: Cvd2014—a database for evaluating no-reference video quality assessment algorithms. IEEE Transactions on Image Processing **25**(7), 3073–3086 (2016)
30. Saad, M.A., Bovik, A.C., Charrier, C.: Blind image quality assessment: A natural scene statistics approach in the dct domain. IEEE Transactions on Image Processing **21**(8), 3339–3352 (2012)
31. Sinno, Z., Bovik, A.C.: Large-scale study of perceptual video quality. IEEE Transactions on Image Processing **28**(2), 612–627 (2019)
32. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the Thirty-

First AAAI Conference on Artificial Intelligence. p. 4278–4284. AAAI'17, AAAI Press (2017)
33. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., J'egou, H.: Training data-efficient image transformers & distillation through attention. In: Proceedings of the International Conference on Machine Learning (ICML) (2021)
34. Tu, Z., Chen, C.J., Wang, Y., Birkbeck, N., Adsumilli, B., Bovik, A.C.: Efficient user-generated video quality prediction. In: 2021 Picture Coding Symposium (PCS). pp. 1–5 (2021)
35. Tu, Z., Wang, Y., Birkbeck, N., Adsumilli, B., Bovik, A.C.: Ugc-vqa: Benchmarking blind video quality assessment for user generated content. IEEE Transactions on Image Processing **30**, 4449–4464 (2021)
36. Wang, Y., Inguva, S., Adsumilli, B.: Youtube ugc dataset for video compression research. In: 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP). pp. 1–5 (2019)
37. Wang, Y., Ke, J., Talebi, H., Yim, J.G., Birkbeck, N., Adsumilli, B., Milanfar, P., Yang, F.: Rich features for perceptual quality assessment of ugc videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13435–13444 (June 2021)
38. Yim, J.G., Wang, Y., Birkbeck, N., Adsumilli, B.: Subjective quality assessment for youtube ugc dataset. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 131–135 (2020)
39. Ying, Z.a., Niu, H., Gupta, P., Mahajan, D., Ghadiyaram, D., Bovik, A.: From patches to pictures (paq-2-piq): Mapping the perceptual space of picture quality. arXiv preprint arXiv:1912.10088 (2019)
40. Ying, Z., Mandal, M., Ghadiyaram, D., Bovik, A.: Patch-vq: 'patching up' the video quality problem. In: 2021 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14019–14029 (June 2021)
41. You, J.: Long short-term convolutional transformer for no-reference video quality assessment. In: Proceedings of the 29th ACM International Conference on Multimedia. p. 2112–2120. MM '21, Association for Computing Machinery, New York, NY, USA (2021)
42. You, J., Korhonen, J.: Deep neural networks for no-reference video quality assessment. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). pp. 2349–2353 (2019)
43. Zhang, W., Ma, K., Yan, J., Deng, D., Wang, Z.: Blind image quality assessment using a deep bilinear convolutional neural network. IEEE Transactions on Circuits and Systems for Video Technology **30**(1), 36–47 (2020)