Neighbourhood Representative Sampling for Efficient End-to-End Video Quality Assessment

Haoning Wu, Chaofeng Chen, Liang Liao, Senior Member, IEEE, Jingwen Hou, Student Member, IEEE, Wenxiu Sun, Qiong Yan, Jinwei Gu, Senior Member, IEEE, and Weisi Lin, Fellow, IEEE

Abstract—The increased resolution of real-world videos presents a dilemma between efficiency and accuracy for deep Video Quality Assessment (VQA). On the one hand, keeping the original resolution will lead to unacceptable computational costs. On the other hand, existing practices, such as resizing or cropping, will change the quality of original videos due to difference in details or loss of contents, and are henceforth harmful to quality assessment. With obtained insight from the studies of spatial-temporal redundancy in the human visual system, visual quality around a neighbourhood has high probability to be similar, and this motivates us to investigate an effective quality-sensitive neighbourhood representative sampling scheme for VQA. In this work, we propose a unified scheme, spatial-temporal grid mini-cube sampling (St-GMS), and the resultant samples are named fragments. In St-GMS, full-resolution videos are first divided into mini-cubes with predefined spatial-temporal grids, then the temporal-aligned quality representatives are sampled to compose the fragments that serve as inputs for VQA. In addition, we design the Fragment Attention Network (FANet), a network architecture tailored specifically for fragments. With fragments and FANet, the proposed FAST-VQA and FasterVQA (with an improved sampling scheme) achieves up to 1612× efficiency than the existing state-of-the-art, meanwhile achieving significantly better performance on all relevant VQA benchmarks.

Index Terms—Fragments, quality-sensitive neighbourhood representatives, sampling, video quality assessment.

I. INTRODUCTION

VISUAL content with a large spatial resolution has always been the pursuit of humans. Indeed, with the proliferation of high-definition photographing devices and significant advancements in various technologies such as video compression and 4G/5G, the videos shot by most common users have greatly increased in resolution (e.g., 1080P, 4K, or even 8K), thereby largely enriching human perception and entertainment styles. Nevertheless, the increased size of real-world videos has posed a number of practical obstacles for machine algorithms in terms of capture, transmission, storage, analysis, and evaluation of those videos. Video Quality Assessment (VQA), also known as the quantification of human perception of video quality, severely suffers from the growing video sizes.

While classical shallow VQA algorithms [1], [2], [3], [4] based on handcrafted features struggle to handle in-the-wild videos with diverse contents and degradation types, the most recent and effective approaches on in-the-wild VQA are based on deep neural networks [5], [6], [7], [8], [9], [10]. However, the computational complexity of deep neural networks usually grows with the video size, i.e., quadratically with the resolution, making them intolerable on high-resolution videos. Taking a 10-second-long 1080P video clip as an example, a plain ResNet-50 [11] as the network backbone will require 40,919 GFLOPs computational cost for inference and 217 GB graphic memory cost during training with a batch size of 1 (Fig. 1), which exceeds the memory limits of all GPUs at present. In order to alleviate computational resource and memory shortage issues on GPUs, the majority of deep VQA methods [5], [6], [7], [8], [12], [13], [14] choose to regress quality scores with fixed features extracted from pre-trained networks of classification tasks [11], [15], [16] instead of end-to-end training, resulting in these methods lacking effective representation learning and essentially only training a shallow regressor for VQA.

Meanwhile, various sampling strategies are employed in other video-related tasks to mitigate the computational burden. Most of them obtained their insight from studies on the human visual system (HVS) [17] or visual coding theories [18], [19], [20], which proved that visual content tends to be similar around a local region, i.e., a neighbourhood. For example, image and video compression standards, e.g., JPEG [21] and H264/AVC [22], and resizing algorithms, such as Bicubic [23], generally extract...
Fragments, in spatial view (compared with resizing and cropping) (a) and temporal view. (b) Zoom-in views of mini-patches show that fragments can retain sensitive to local quality issues (e.g., blurs in video 1/2) (a), and spot temporal variations such as shaking across frames (b).

representatives for partitioned neighbourhoods to ensure that the resampled information can represent the original contents globally. As a result, high-level video recognition (e.g., classification, detection) methods [24], [25], [26], [27] have widely adopted resizing to reduce the complexity of inputs. However, as illustrated in Fig. 2(a), resizing corrupts quality-related local textures such as blurs and artifacts in video 1&2 which is significant in VQA and other low-level tasks. On the other hand, in order to preserve these local textures, several works [28], [29] attempt to crop a single continuous patch. Nevertheless, these samples lose a large proportion of quality information, e.g., video 2&3 in Fig. 2(a), thus also not suitable for the VQA task. To build good samples for VQA, we need to ensure that they are representative of global quality information while also preserving the sensitivity to quality information on local textures and temporal variations.

In this paper, we propose a new sampling paradigm for VQA, so as to well preserve quality-related information while significantly reducing the input complexity. The samples, named quality-sensitive neighbourhood representatives, are extracted within partitioned neighbourhoods to be representative to global quality, and made up with raw resolution patches in continuous frames to preserve local spatial-temporal quality. These representatives are achieved through a unified Spatial-temporal Grid Mini-cubes Sampling (St-GMS, Fig. 3) pipeline. Spatially, it cuts a video frame into uniform non-overlapping grids, and samples a mini-patch randomly from each grid. Temporally, it cuts a video into uniform segments and samples multiple continuous frames within each segment. To better preserve temporal continuity between frames, we also constrain that mini-patches in each spatial grid and temporal segment should be aligned to form a mini-cube. Finally, all the mini-cubes are stitched to an integrated sample, termed fragments (Fig. 2).

Fig. 2(a) illustrates the spatial view of fragments. First, they preserve the local texture-related quality information (e.g., spot blurs happened in video 1&2) by retaining the patches in original resolution. Second, benefiting from the globally uniformly partitioned grids, fragments cover the global quality even though different regions have different qualities (e.g., video 2&3). Third, by splicing the mini-cubes, fragments retain contextual relations among them so that the model can learn global scene information and rough semantic information of the original frames. As for the temporal view of fragments, as shown in Fig. 2(b), with the continuous frames and aligned mini-patches in each segment, fragments can also spot temporal variations in videos, e.g., distinguish between severely shaking videos (e.g., video 5) from relatively stable shots (e.g., video 6). The segment-wise sampling on the temporal dimension also ensures temporally uniform coverage of quality information.

It is non-trivial to design deep networks for fragments, as the mini-cubes are actually independent and the edges in between may be misinterpreted as quality defects. To avoid uncontrolled fusion of pixels in different mini-cubes, we propose a rule for building networks on fragments, the match constraint, to align the pooling operations with sampled mini-cubes. Specifically, we choose Video Swin Transformer [24] as the backbone and improve the Relative Position Biases in the backbone into Gated Relative Position Biases (GRPB) to correctly represent the
positions of pixels in fragments. Based on the characteristic of fragments that quality is diverse among mini-cubes, we further replace the pool-first head that is usually used in high-level tasks with a pool-last Intra-Patch Non-linear Regression (IP-NLR) head, to get better performance and predict local quality maps beyond quality scores. In general, with a Tiny Swin Transformer (abbrev. as Swin-T) as baseline backbone and the proposed GRPB & IP-NLR modules as modifications, we propose the Fragment Attention Network (FANet) that best extracts the quality-sensitive information in fragments.

This work is a substantial extension to our earlier conference version FAST-VQA [30] which proposes a spatial-only sampling scheme and the accommodated network structure (FANet). In comparison to the conference version, we include a significant amount of improvements:

1) To further improve efficiency, we extend spatial-only sampling into the spatial-temporal sampling scheme (St-GMS), based on which we improve FAST-VQA into the Fragment spatial-temporal Video Quality Assessment (FasterVQA) that performs comparable to FAST-VQA with only 25% of FLOPs

2) We propose the Adaptive Multi-scale Inference (AMI) on FANet for adaptively inferring on different scales with one model trained on a fixed scale while keeping competitive performance.

3) We add extensive ablation studies to further analyze the effects of sampling granularity, end-to-end training and semantic pre-training in the proposed methods. The main contributions of this work are listed as follows:

- We propose the quality-sensitive neighbourhood representatives, a novel sampling paradigm for VQA, and design a unified Spatial-temporal Grid Mini-cube Sampling (St-GMS) scheme to sample fragments. The fragments enable deep VQA methods to efficiently and effectively evaluate videos of any resolution.

- We propose and evaluate the match constraint for pooling layers as guidance for building networks for fragments. Based on this constraint, we propose the Fragment Attention Network (FANet) with newly designed GRPB and IP-NLR modules to best accommodate the characteristics of fragments.

- The proposed FAST-VQA and FasterVQA outperform existing VQA methods by a large margin (up to 7%) with up to 1612× efficiency. When we infer on smaller scales via AMI, we can reach 13.6× real-time on CPU (still more accurate than existing SOTA).

II. RELATED WORKS

Classical VQA Methods: Classical VQA methods [31], [32], [33], [34] employ handcraft features to evaluate video quality. Some methods hypothesize [1], [2], [35], [36] that natural videos follow specific statistical rules, while the defect videos do not, and compute quality scores only from statistical evidence without regression from any subjective labels. In recent years, several methods [3], [4], [37] choose to first handcraft quality-sensitive features and then regress them to subjective mean opinion scores (MOS), in order to better fit the human perception. Among them, TLVQM [3] uses a combination of two levels of handcraft features, including high-complexity spatial features computed on sparse frames for measuring spatial distortions, and low-complexity temporal features computed for each frame for assessing temporal variations. VIDEVAL [4] ensembles various
handcraft features to model the diverse authentic distortions and also reduces the feature dimensions to reduce the computational burden. Spatial-temporal chips are sampled in a recent work called ChipQA [38] for more efficient handcraft feature extraction. These classical approaches suggest that it is possible to reduce the size of videos while retaining their quality information. Nevertheless, since the factors affecting the in-the-wild video quality are quite complicated and usually cannot be concluded by finite handcraft features, the performance of these classical methods are constrained.

Deep VQA Methods: Benefiting from the semantic awareness of deep neural network features, deep VQA methods [9], [39] are becoming predominant. For example, VSFA [5] uses the features extracted by pre-trained ResNet-50 [11] from ImageNet-1k dataset [40] and adopts Gate Recurrent Unit (GRU) [41] for quality regression. However, due to the extremely high memory cost of deep networks on high-resolution videos (as shown in Fig. 1), most existing deep VQA methods [5], [8], [42], [43], [44] can only extract fixed features instead of updating them. Without end-to-end training, existing methods generally improve features in the three following ways. 1) Introducing heavier backbones, e.g., MLSP-FF [8] includes heavier Inception-ResNet-V2 [15] for feature extraction. 2) Using multiple backbone networks instead of one, e.g., PVQ [7] uses an additional ResNet-3d-18 [16] network to extract temporal quality features. 3) Including frame-wise pre-training [7], [10], [12] from IQA databases [45], [46]. A most recent method, BVQA-TCSVT-2022 [13], combines all these three ways to reach better performance, while it requires up to 26 minutes on CPU to assess the quality for an 8-second-long video, 200× slower than video playback. While improving performance, these practices significantly sacrifice the final computational efficiency. These practices further highlight the value of the proposed method with effective end-to-end training via efficient quality-retained sampling, so as to improve performance in an efficient manner for training and inference.

Cropping in VQA: In contrast to the rare use of resizing in VQA due to its corruption on local textures, cropping has been explored by a few existing VQA methods [7], [9], [38]. Nevertheless, as the quality of local patches does not match the MOS values of global videos, PVQ [7] collects additional human labels for each patch, while CNN+LSTM [9] requires a two-stage training to reduce inaccuracy. Moreover, during inference, existing approaches typically still need the original videos (or samples with equivalent complexity as them) as inputs. In contrast, we propose to sample grid mini-patches that can well-represent the quality of original videos, and significantly improve efficiency for deep VQA methods during both training and inference.

III. APPROACH

In this section, we introduce the proposed FAST-VQA and FasterVQA. We first define the paradigm of sampling quality-sensitive neighbourhood representatives (Section III-A), and introduce the corresponding Spatial-temporal Grid Mini-cube Sampling (St-GMS, Section III-B) scheme to resample the videos into fragments. After sampling, the fragments are fed into the Fragment Attention Network (FANet, discussed in Section III-C) which is designed based on the match constraint. We also propose an Adaptive Multi-scale Inference (AMI, Section III-D) strategy for adaptive-scale inference on the model trained at a single scale. Lastly, we present the associated objective functions (Section III-E) for model training.

A. Sampling Representatives From Neighbourhoods

In visual tasks, sampling is widely applied. Specifically, uniform sampling schemes, such as spatial nearest/bicubic down-sampling and temporal uniform sampling, are widely applied in high-level recognition tasks. In general, these methods can be concluded by two steps: 1) segmenting the image/video into various local areas (referred to as neighbourhoods), and 2) sampling a representative from each neighbourhood. We conclude the overall unified paradigm as neighbourhood representatives ($\mathcal{R}$) which can be specified to either spatial or temporal dimensions. Given a target sampled size $S$ and a single representative size $S_r$, the paradigm first divides the visual contents into neighbourhods $\mathcal{N} = \{n^i| i = 0, 1, 2, \ldots, \frac{S}{S_r} - 1\}$, and then the neighbourhood representatives $\mathcal{R}$ can be formulated as,

$$\mathcal{R} = \{r(n^i)| i = 0, 1, 2, \ldots, \frac{S}{S_r} - 1\} \quad (1)$$

where $r(n^i)$ denotes the function that samples a representative from neighbourhood $n^i$.

As neighbourhood redundancy also occurs for quality-related information, the neighbourhood representatives can also be applied to quality tasks, in which we aim to collect an efficient low-cost sample that can represent the original video’s quality. Specifically, according to many widely acknowledged studies [3], [4], [38], continuous local textures and local temporal variations are significant and necessary to correctly evaluate video quality, which will hardly be preserved if we apply resizing or uniform frame sampling. Constrained by this specific characteristic of VQA, we propose to sample quality-sensitive neighbourhood representatives ($\mathcal{R}_q$), which should further satisfy following principles: 1) they should contain raw pixels in videos instead of pooled or averaged results; and 2) the raw pixels in one representative $r(n^i)$ should form a continuous patch or clip that is large enough to distinguish spatial or temporal local quality information. As a result, these representatives $\mathcal{R}_q$ can represent both the unbiased global quality information and the sensitive local quality information (e.g., spatial local textures, temporal variations among adjacent frames) that are vital for VQA.

B. Spatial-Temporal Grid Mini-Cube Sampling

We propose the unified Spatial-temporal Grid Mini-cube Sampling (St-GMS) scheme which follows the principle of quality-sensitive neighbourhood representatives in both spatial and temporal dimensions. The pipeline for St-GMS is illustrated in Fig. 3 and discussed as follows.
1) Spatial Sampling: Grid Mini-Patch Sampling\(^{(GMS)}\): In the first part, we discuss the Grid Mini-patch Sampling (GMS, Fig. 3(a)), i.e., the spatial sampling operations in St-GMS, together with the corresponding principles.

**Representing 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 each video frame into uniform grids with each grid having the same size (as shown in Fig 3(a)). In particular, we cut the video frame \( I \) with size \( H \times W \) into \( G_f \times G_f \) uniform grids with the same sizes, denoted as \( \{g^{i,j} | 0 < i < G_f, 0 < j < G_f \} \), where \( g^{i,j} \) refers to the grid in \( i \)-th row and \( j \)-th column. The partition is formalized as follows.\(^{1}\)

\[
g^{i,j} = I_{\left[ \frac{iH}{G_f} \leq y < \frac{(i+1)H}{G_f} , \frac{jW}{G_f} \leq x < \frac{(j+1)W}{G_f} \right]} \quad (2)
\]

**Sensitive to Local Quality. Raw Patch Sampling:** To preserve the local textures (e.g., blurs, noises, artifacts) that are vital in VQA, we avoid resizing operations and select only original resolution patches, which can better represent local textural quality in grids. Henceforth, we randomly sample one small patch in its original resolution (denoted as mini-patch, \( MP^{i,j} \), with size \( S_f \times S_f \) from each spatial grid \( g^{i,j} \) and follow uniform random sampling in each grid. The spatial patch sampling (\( S_a \)) is formulated as follows.

\[
MP^{i,j} = S_a^{i,j}(g^{i,j}), \quad 0 \leq i,j < G_f \quad (3)
\]

**Preserving Contextual Relations. Patch Splicing:** Existing works\(^{[5,8,47]}\) have shown that global scene information notably affects quality-related perception, that even the same textures under different semantic background can relate to different quality\(^{[48]}\). To preserve the background information about the global scene, we retain the contextual relations among mini-patches by splicing them together:

\[
F^{i,j} = F_{\left[ i \times S_f, (i+1) \times S_f \right] \times \left[ j \times S_f, (j+1) \times S_f \right]} = MP^{i,j}, \quad 0 \leq i,j < G_f \quad (4)
\]

where \( F \) denotes the spliced mini-patches from frame \( I \) after spatial GMS pipeline, as in our conference version\(^{[30]}\). Furthermore, we extend GMS into the temporal dimension for more efficient quality evaluation, elaborated as follows.

2) Extending GMS Into the Temporal Dimension: We extend the GMS into the temporal dimension based on unified quality-sensitive neighbourhood representatives, as illustrated in Fig. 3(b). We discuss the detailed principles and operations in the temporal dimension as follows.

**Temporal Representative. Uniform Segment Partition:** Similar to the spatial case, an accurate VQA method also need to uniformly assess quality along the temporal dimension. For uniformity, TSN\(^{[49]}\) proposed general segment-wise sampling for videos which had been applied by many existing VQA methods\(^{[3,4,10]}\). Thus, we divide the video \( \mathcal{V} \) with \( T \) total frames into \( G_t \) uniform non-overlapping temporal segments (as shown in Fig. 3(b)). Overall, we extend the uniform grid partition as defined in (2) into spatial-temporal uniform grid partition, as follows.

\[
g^{k,i,j} = \mathcal{V}_{\left[ \frac{kH}{G_t} \leq y < \frac{(k+1)H}{G_t} , \frac{iW}{G_f} \leq x < \frac{(i+1)W}{G_f} , \frac{jH}{G_t} \leq t < \frac{(j+1)H}{G_t} \right]} \quad (5)
\]

where \( g^{k,i,j} \) denotes the spatial-temporal grid in \( k \)-th temporal segment, \( i \)-th row and \( j \)-th column.

**Sensitive to Inter-Frame Variations. Continuous Frames:** It is widely recognized by early works\(^{[3,7,39]}\) that inter-frame temporal variations are influential to video quality. To retain the raw temporal variations in videos, we would like the frames sampled in each segment to be continuous and the corresponding mini-patches to be aligned so that the temporal variation inside the segment can be reflected by these samples. Thus, we apply temporal continuous frame sampling (\( S_t \)) to sample \( T_f \) continuous frames in each segment. Combining with the spatial raw-patch sampling (\( S_s \), (3)), we sample a mini-cube \( MC^{k,i,j} \) of size \( T_f \times S_f \times S_f \) from each spatial-temporal grid \( g^{k,i,j} \) as follows:

\[
MC^{k,i,j} = S_t^{i,j}(S_s^{k,i,j}), \quad 0 \leq i,j < G_s, 0 \leq k < G_t \quad (6)
\]

**Long-Term Dependencies. Temporal Splicing:** Although there are no consensus on explanations of the long-term temporal dependencies in VQA, plenty of existing methods\(^{[5,12,14]}\) have proved that they are practically influential to the video quality. Therefore, we include temporal splicing into the whole splicing operation as follows:

\[
F_{3D}^{k,i,j} = F_{3D}[k \times T_f; (k+1) \times T_f, i \times S_f, (i+1) \times S_f, (j+1) \times S_f] = MC_{3D}^{k,i,j} \quad 0 \leq i,j < G_s, 0 \leq k < G_t \quad (7)
\]

where \( F_{3D} \) denotes the spliced spatial-temporal mini-cubes after the St-GMS pipeline, as space-time-unified fragments.

The GMS and the following FANet (Section III-C, Fig. 5) together constitute the proposed FAST-VQA, which only includes the proposed spatial sampling operations and selects dense frames in the temporal dimension for inference. With unified spatial and temporal sampling strategies, we improve FAST-VQA into FasterVQA by replacing the GMS with the St-GMS. FasterVQA has 4X efficiency than FAST-VQA yet comparable accuracy. Both FAST-VQA and FasterVQA include the FANet structure, discussed as follows.

C. Quality Regression Network for Fragments

In this section, we discuss our efforts on designing a powerful quality regression network accommodated for fragments. We first discuss the rationale behind choosing backbones (the match constraint, Section III-C1). Then, we discuss our modifications on the baseline network, which finally yields the Fragment Attention Network (FANet, Section III-C2).

1) Motivation. match Constraint for Pooling Layers: It is non-trivial to build a network using the proposed fragments as inputs. Like most quality assessment networks, it should be able to effectively extract the quality information preserved in fragments, including the local textures inside mini-cubes and the contextual relationships between them. Moreover, it should

\(^{1}\)In this section, all square brackets \( (\cdot) \) denote the slicing operations, and all superscripts (e.g., \( ^{1} \)) denote position indices.
specifically avoid misinterpreting the artificial discontinuity between mini-cubes (as illustrated in Fig. 4(a)), resulted by random sampling and splicing) as local artifacts. This requirement calls for more careful network design, especially on the pooling layers that exist in the original designs of almost every deep neural network (especially multi-scale networks) for computer vision, which merge multiple feature pixels into subsequent, downsampled feature pixels. In these pooling layers, we should refrain merging feature pixels from different mini-cubes into one downsampled feature pixel, so as to avoid aforementioned misinterpretation. As a result, we impose the match constraint, which constrains that each pooling kernel should only include pixels inside of an individual mini-cube as green boxes in Fig. 4(a), but not across parts of mini-cubes (red boxes), before each mini-cube is finally downsampled as a single pixel. Formally, take any pooling kernel at any layer (before mini-cubes have been downsampled as single pixels), denote the set of original pixels that falls into the kernel as \( \mathcal{P} \), the constraint can be formulated as:

\[
\exists k, i, j, \; \text{s.t.} \; \mathcal{P} \subset \mathcal{M}^{k, i, j} \tag{8}
\]

To follow the match constraint, we require the networks that use non-overlapping pooling kernels. Many backbone structures can meet this requirement, including transformer-based structures [24], [26], [27], [50] and part of modern convolution-based structures such as ConvNeXt [51], while it is possible to match their pooling kernels with mini-cubes. Our experiments show that either 1) using conventional backbones (i.e., ResNet [11] and MobileNet [52]) with overlapping pooling kernels or 2) failing to align mini-cubes with pooled pixels leads to a notable performance drop, suggesting the significance of match constraint for pooling layers. Finally, we choose the Video Swin Transformer Tiny (Swin-T) backbone which follows the match constraint as the backbone of the quality regression network for fragments. We also make several modification on the Swin-T to better accommodate it for fragments, discussed as follows.

2) Fragment Attention Network (FANet). The Overall Framework: Fig. 5 shows the overall framework of Fragment Attention Network (FANet), the proposed end-to-end quality regression network for fragments. It includes a four-layer Swin-T with first three window self-attention layers modified by GRPB as the backbone (abbr. as Swin-GRPB), and an IP-NLR quality-regression head.

Gated Relative Position Biases (GRPB): In Swin-T, the window self-attention layers are built across mini-cubes to learn contextual relations between them. However, in these window self-attention layers, representing the positions of pixels of
fragments differs from those of normal inputs. While original Swin-T proposes relative position bias (RPB) that uses learnable Relative Bias Table \((T)\) to represent the relative positions of pixels in attention pairs \((QK^T)\), they cannot well represent the relative positions of different pixels in fragments. Specifically, considering that some pairs in the same attention window might have the same relative position (e.g., Fig. 4(b) A-C, D-E, A-B), but the cross-patch attention pairs (A-C, D-E, two pixels from different mini-cubes) are in far actual distances while intra-patch attention pairs (A-B, two pixels from the same mini-cube). Therefore, we distinguish the two type of attention pairs and propose the gated relative position biases (GRPB) as shown in Fig. 5(b) that uses two learnable real position bias table \((T_{\text{real}})\) and pseudo position bias table \((T_{\text{pseudo}})\) to replace \(T\). Denote any two pixels in positions \((p,p)\), \((p \in MC^{k,i,j}, \hat{p} \in MC^{k,i,j})\), the GRPB between them \((B(p, \hat{p}))\) can be formulated as

\[
G(p, \hat{p}) = \begin{cases} 
1, & i = i \land j = j \land k = k, \\
0, & \text{else}
\end{cases} \tag{9}
\]

\[
B(p, \hat{p}) = G(p, \hat{p})T_{\text{real}}^{p-\hat{p}} + (1 - G(p, \hat{p}))T_{\text{pseudo}}^{p-\hat{p}} \tag{10}
\]

The two bias tables \(T_{\text{pseudo}}\) and \(T_{\text{real}}\) are learnable parameters, both with shape \((2 \times 2 \times W_T - 1) \times (2 \times W_h - 1) \times (2 \times W_w - 1)\), \(^2\) where \(W_t, W_h, W_w\) are the temporal length, height, and width of the attention window in the Swin-T backbone. The \(p - \hat{p}\) is the vector difference between the two positions \(p\) and \(\hat{p}\), and is used to index the two position bias tables.

Intra-Patch Non-Linear Regression (IP-NLR) Head: Several recent quality assessment methods \([7, 46]\) apply patch-independent regression heads to obtain local quality. Based on the match constraint \((8)\), feature pixels are aligned with mini-cubes, so it is also possible to regress qualities for each mini-cube to obtain local quality maps. Furthermore, as shown in Fig. 4(c), the quality-related features in different mini-cubes should be diverse even in the same video as their original positions are far apart. Therefore, averaging them before regression as commonly practised in video recognition may have the potential risk to lose the sensitivity to the diverse quality information, while regressing them independently can avoid this problem. Based on the two reasons above, we design the Intra-Patch Non-Linear Regression (IP-NLR, Fig. 5(c)) to regress the features via a two-layer MLP first and perform pooling on the regressed local quality scores. Denote final backbone features as \(f_{\text{final}}\), the local quality map as \(l_{pr}\), the global quality scores (final output of FANet) as \(g_{pr}\), linear layers as \(L_1, L_2\), the IP-NLR can be expressed as follows:

\[
p^{l,h,w}_{pr} = L_2(\text{GeLU}(L_1(f^{l,h,w}_{\text{final}}))) \tag{11}
\]

\[
g_{pr} = \sqrt{l_{pr}} \tag{12}
\]

D. Adaptive Multi-Scale Inference

The proposed models can adapt to various computing resources by changing the sampling densities (scales) of fragments. However, our conference version \([30]\) (FAST-VQA) still requires training different models for different scales of fragments. This could be inefficient when the input scale needs to be changed frequently, or adaptively. Therefore, with the objective of training at only one scale (least cost) and infer at any different scale (most flexible), we propose the Adaptive Multi-scale Inference (AMI) for FasterVQA.

To perform AMI, we adaptively modify the backbone structure of FANet with respect to different sizes of inference inputs. Generally, we keep all the linear and pooling layers unchanged as they mainly focus on local textures. For the window-based self-attention layers, we adaptively rescale the attention windows to ensure that the proportion of the window size to the global size is conserved when the input scale changes, which simulates self-attention-based approaches \([50, 53]\) in dealing with variable-length inputs. Formally, the attention window sizes given new scales of fragments are computed as follows:

\[
\hat{W} = \frac{W_0 \odot \hat{G}}{G_0} \tag{13}
\]

where \(\hat{W}\) and \(W_0\) are the rescaled and base window sizes, and \(\hat{G}\) and \(G_0\) are the actual and preset base number of grids (to meet the match constraint, the sizes of mini-cubes are kept the same). For GRPB, we also look up from the shared \(T_{\text{real}}\) and \(T_{\text{pseudo}}\) as defined in \((10)\), and the gates \(G\) are computed from partitions of actual inputs. Our experiments demonstrate that the proposed FasterVQA with AMI can still infer with high accuracy at a certain scale even without training on fragments on the corresponding scale.

Moreover, the AMI strategy enables the proposed approach to dynamically select cost-effective samples for videos with lower complexity. Specifically, we introduce the resolution-varied AMI to achieve adaptive sampling on cross-resolution scenarios \([7, 54]\): it assigns different sample sizes for videos at different spatial scales, where the adaptive spatial grid number \(G_{s, \text{adaptive}}\) for a given video with short-edge resolution \(R\) is calculated as follows:

\[
G_{s, \text{adaptive}} = \left[ \frac{(R - R_{\text{min}})}{(R_{\text{max}} - R_{\text{min}})} \times (G_{s, \text{max}} - G_{s, \text{min}}) + G_{s, \text{min}} \right] \tag{14}
\]

where the brackets \([\) indicate rounding off to the nearest integer, \(R_{\text{max}}\) and \(R_{\text{min}}\) represent the maximum and minimum resolution in the whole dataset, and \(G_{s, \text{max}}\) and \(G_{s, \text{min}}\) represent the maximum and minimum number of allowable grids, respectively. With similar computation costs, resolution-varied AMI is more competitive under cross-resolution settings than its fixed-size variants.

E. Objective Functions

Many existing studies \([55, 56, 57]\) have suggested that information about relative quality comparison among different videos is more indicative than the absolute quality score of a single video. To utilize the relative quality among videos during training, we adopt the monotonicity and linearity fusion loss objective \([6, 58]\), as the weighted sum of monotonicity loss
The resolution-varied AMI, we set $G_{\text{max}}^* = 7, G_{\text{min}}^* = 4$ to keep similar inference cost with fixed FasterVQA-MS.

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

3) Training & Benchmark Sets: We use the large-scale LSVQ_{train} [7] dataset with 28,056 videos for training FAST-VQA/FasterVQA. For evaluation, we choose 4 testing sets to test the model trained on LSVQ. The first two sets, LSVQ_{test} and LSVQ_{1080p} are official intra-dataset test subsets for LSVQ, while the LSVQ_{test} consists of 7,400 various resolution videos from 240P to 720P, and LSVQ_{1080p} consists of 3,600 1080P high resolution videos. We directly evaluate the generalization ability of proposed models on cross-dataset evaluations on KoNViD-1k [60] and LIVE-VQC [54], two widely-recognized in-the-wild VQA benchmark datasets composed of natural videos. We also discuss the fine-tuning results on several non-natural VQA datasets, including lab-collected datasets [61], [62] and datasets with computer-generated videos [63], in Section IV-B3.

### Table I

| Methods       | Number of Frames (T) | Size of Mini-patch $(S_x, S_y)$ | Number of Grade $O_G$ | Window Size in FANet $W$ | FLOPs (fer) |
|---------------|----------------------|---------------------------------|-----------------------|--------------------------|-------------|
| FAST-VQA      | 24                   | (32, 32)                        | 7                     | (6, 1, 7)                | 279G        |
| FAST-VQA-M    | 24                   | (32, 32)                        | 4                     | (4, 4, 4)               | 35G         |

Both variants require 4 clips at minimum to cover whole video.

### Table II

| Methods       | Size of Mini-Cube $(T_x, T_y, T_z)$ | Segments and Grids $(G_x, G_y, G_z)$ | Associated Window Size in FANet $(W)$ | FLOPs (fer) |
|---------------|-----------------------------------|-------------------------------------|---------------------------------------|-------------|
| FasterVQA     | (4, 32, 32)                       | (6, 7, 7)                           | (6, 7, 7)                             | 69G         |
| FasterVQA-MT  | (4, 32, 32)                       | (4, 7, 7)                           | (4, 7, 7)                             | 35G         |
| FasterVQA-MS  | (4, 32, 32)                       | (5, 5, 5)                           | (8, 5, 5)                             | 35G         |

In the experiment part, we conduct experiments for the proposed concept and methods in the following aspects:

- Benchmark comparison with existing approaches (Section IV-B), in terms of both accuracy and efficiency.
- Detailed evaluation on sampling (Section IV-C), compared to naive sampling approaches and different variants.
- Ablation studies on match constraint, FANet structure, training and inference strategies, e.g., AMI (Section IV-D).
- Extra justifications to our methods: irreplaceable role of semantics (Section IV-E), evaluation on high-resolution cases (Section IV-F) and stability analysis (Section IV-G).
- Quantitative studies for local quality maps (Section IV-H).

A. Evaluation Setup

1) Implementation Details: We use the Swin-T [24] as the backbone of our FANet, which is initialized by pretraining on Kinetics-400 dataset [59]. For FAST-VQA, we implement two sampling densities for fragments and adjust the window sizes in FANet to the input sizes: FAST-VQA (better accuracy) and FAST-VQA-M (mobile-friendly), as listed in Table I. For FasterVQA, as we practice Adaptive Multi-scale Inference (AMI), we unify different sample densities in one single model. Still, we benchmark the performance of FasterVQA on two mobile-friendly scales with reduced size on either spatial (FasterVQA-MS) or temporal (FasterVQA-MT) dimensions together with the base scale (FasterVQA), as listed in Table II. All $S_x$ and $T_y$ are selected to follow the match constraint (8). The $\lambda$ in (17) is set as 0.3, with initial learning rate set as 0.001 for IP-NLR head and 0.0001 for the Swin-GRPB backbone respectively. For

$$L_{\text{mono}} \text{ and linearity loss } L_{\text{lin}}, \text{ as follows:}$$

$$L_{\text{mono}} = \sum_{i,j} \max (s_{i,j}^{{\text{pred}}}, s_{i,j}^{{\text{gt}}}) \text{sgn} (s_{i,j}^{{\text{gt}}} - s_{i,j}^{{\text{gt}}}) = 0$$

$$L_{\text{lin}} = \left( 1 - \frac{1}{\sum_{i,j} \text{sgn} \left( s_{i,j}^{{\text{pred}}} - s_{i,j}^{{\text{gt}}} - s_{i,j}^{{\text{gt}}} - s_{i,j}^{{\text{gt}}} > 0 \right)} \right) / 2$$

$$L_{\text{fusion}} = L_{\text{lin}} + \lambda L_{\text{mono}}$$

where $\text{sgn}(\cdot)$ denotes the sign function, $<>$ denotes the inner product of two vectors, and $s_{i,j}^{{\text{pred}}}$ and $s_{i,j}^{{\text{gt}}}$ are vectors that refer to predictions and ground truth labels in a batch.

IV. EXPERIMENTS

In the experiment part, we conduct experiments for the proposed concepts and methods in the following aspects:

- Benchmark comparison with existing approaches (Section IV-B), in terms of both accuracy and efficiency.
- Detailed evaluation on sampling (Section IV-C), compared to naive sampling approaches and different variants.
- Ablation studies on match constraint, FANet structure, training and inference strategies, e.g., AMI (Section IV-D).
- Extra justifications to our methods: irreplaceable role of semantics (Section IV-E), evaluation on high-resolution cases (Section IV-F) and stability analysis (Section IV-G).
- Quantitative studies for local quality maps (Section IV-H).

A. Evaluation Setup

1) Implementation Details: We use the Swin-T [24] as the backbone of our FANet, which is initialized by pretraining on Kinetics-400 dataset [59]. For FAST-VQA, we implement two sampling densities for fragments and adjust the window sizes in FANet to the input sizes: FAST-VQA (better accuracy) and FAST-VQA-M (mobile-friendly), as listed in Table I. For FasterVQA, as we practice Adaptive Multi-scale Inference (AMI), we unify different sample densities in one single model. Still, we benchmark the performance of FasterVQA on two mobile-friendly scales with reduced size on either spatial (FasterVQA-MS) or temporal (FasterVQA-MT) dimensions together with the base scale (FasterVQA), as listed in Table II. All $S_x$ and $T_y$ are selected to follow the match constraint (8). The $\lambda$ in (17) is set as 0.3, with initial learning rate set as 0.001 for IP-NLR head and 0.0001 for the Swin-GRPB backbone respectively. For

$$L_{\text{mono}} \text{ and linearity loss } L_{\text{lin}}, \text{ as follows:}$$

$$L_{\text{mono}} = \sum_{i,j} \max (s_{i,j}^{{\text{pred}}}, s_{i,j}^{{\text{gt}}}) \text{sgn} (s_{i,j}^{{\text{gt}}} - s_{i,j}^{{\text{gt}}}) = 0$$

$$L_{\text{lin}} = \left( 1 - \frac{1}{\sum_{i,j} \text{sgn} \left( s_{i,j}^{{\text{pred}}} - s_{i,j}^{{\text{gt}}} - s_{i,j}^{{\text{gt}}} - s_{i,j}^{{\text{gt}}} > 0 \right)} \right) / 2$$

$$L_{\text{fusion}} = L_{\text{lin}} + \lambda L_{\text{mono}}$$

where $\text{sgn}(\cdot)$ denotes the sign function, $<>$ denotes the inner product of two vectors, and $s_{i,j}^{{\text{pred}}}$ and $s_{i,j}^{{\text{gt}}}$ are vectors that refer to predictions and ground truth labels in a batch.
TABLE III

| Type/Testing Set/ | FLOPs on 1080P/8-bpc | Intra dataset Test Sets | Cross-dataset Test Sets |
|------------------|----------------------|-------------------------|-------------------------|
|                  | SRC | PLCC | SRC | PLCC | SRC | PLCC | SRC | PLCC | SRC | PLCC |
| Groups           | relative to FAST-VQA |                        |                         |             |     |      |     |      |     |      |
| Existing         | Classical | NA | 0.567 | 0.576 | 0.497 | 0.531 | 0.646 | 0.647 | 0.524 | 0.536 |
|                  | Existing | NA | 0.772 | 0.774 | 0.589 | 0.616 | 0.732 | 0.724 | 0.670 | 0.691 |
|                  | Existing | NA | 0.794 | 0.783 | 0.545 | 0.554 | 0.751 | 0.741 | 0.630 | 0.640 |
| Fixed            | Existing | 147× | 0.801 | 0.796 | 0.675 | 0.704 | 0.764 | 0.794 | 0.734 | 0.772 |
| Deep             | Fixed    | 210× | 0.814 | 0.816 | 0.686 | 0.708 | 0.781 | 0.781 | 0.747 | 0.776 |
| BVQA-TCSVT-2022 | Fixed    | 210× | 0.827 | 0.828 | 0.711 | 0.739 | 0.791 | 0.795 | 0.770 | 0.807 |
| Full-res Swin-T  | Fixed    | 403× | 0.852 | 0.854 | 0.777 | 0.782 | 0.834 | 0.837 | 0.816 | 0.824 |
| Baseline         | FAST-VQA | 42.5× | 0.835 | 0.833 | 0.759 | 0.753 | 0.825 | 0.826 | 0.794 | 0.809 |
|                  | Faster-VQA-M | 0.165× | 0.852 | 0.854 | 0.759 | 0.773 | 0.861 | 0.832 | 0.788 | 0.810 |
|                  | Faster-VQA-AM | 0.230× | 0.866 | 0.850 | 0.758 | 0.784 | 0.862 | 0.854 | 0.791 | 0.818 |
|                  | Faster-VQA-AM (AMI) | 0.125× | 0.860 | 0.861 | 0.753 | 0.729 | 0.846 | 0.849 | 0.803 | 0.826 |
|                  | Faster-VQA-AM (AMI) | 0.125× | 0.876 | 0.877 | 0.779 | 0.814 | 0.852 | 0.855 | 0.823 | 0.844 |
|                  | Faster-VQA | 0.25× | 0.873 | 0.874 | 0.772 | 0.811 | 0.863 | 0.863 | 0.813 | 0.837 |

The 1st and 2nd best scores are denoted in boldface, underlined, and italic, respectively. We refer to State/VQA with multiple scales as AMI.

TABLE IV

FLOPS AND RUNNING TIME (AVG. OF 20 RUNS) ON GPU SERVER (TESLA V100) AND CPU (APPLE M1) COMPARISON OF FAST-VQA, STATE-OF-THE-ART METHODS AND OUR BASELINE ON 8-SEC VIDEOS DIFFERENT RESOLUTIONS

| Method                  | 540P | 720P | 1080P |
|-------------------------|------|------|-------|
|                         | FLOPs | Time (GPU/s) | Time (CPU/s) | FLOPs | Time (GPU/s) | Time (CPU/s) | FLOPs | Time (GPU/s) | Time (CPU/s) |
| VQA [3]                 | 16240 | 2.603 | 152.4 | 18138 | 3.571 | 233.9 | 40191 | 11.14 | 445.6 |
| PVQ [7]                 | 44660 | 3.091 | 149.5 | 22027 | 4.143 | 247.8 | 38501 | 13.79 | 536.4 |
| BVQA-TCSVT-2022 [13]   | 46758 | 5.383 | 278.4 | 53194 | 10.43 | 592.4 | 12347 | 27.44 | 1547 |
| Full-res Swin-T [24]    | 30266 | 5.326 | 170.5 | 30973 | 5.047 | 164.5 | 30973 | 8.753 | 2347 |
| FAST-VQA (Ours)         | 279.7 | 0.046 | 8.839 | 279.7 | 0.046 | 8.930 | 279.7 | 0.045 | 8.678 |
| Faster-VQA (Ours)       | 69.4 | 0.023 | 2.794 | 69.4 | 0.022 | 2.730 | 69.4 | 0.023 | 2.697 |
| Faster-VQA-AM (Ours)    | 46.5 | 0.019 | 1.586 | 46.5 | 0.019 | 1.635 | 46.5 | 0.019 | 1.542 |
| Faster-VQA-AM (Ours)    | 36.1 | 0.016 | 0.594 | 36.1 | 0.016 | 0.587 | 36.1 | 0.018 | 0.499 |
| Faster-VQA-AM (Ours)    | 35.2 | 0.018 | 0.447 | 35.2 | 0.018 | 0.423 | 35.2 | 0.017 | 0.445 |

Fig. 6. Performance-FLOPs curve of proposed FAST-VQA / FasterVQA and baseline methods. X-Axis: GFLOPs (log scale); Y-Axis: PLCC.

On different resolutions in Table IV. We also draw the respective performance-FLOPs curves in Fig. 6. Note that we remove video loading latency for all methods.

Efficiency of Base Models: Even the base models of FAST-VQA and FasterVQA reach unprecedented efficiency. FAST-VQA reduces up to 210× FLOPs and 70× CPU running time than PVQ [7] while obtaining notably better performance, while FasterVQA can reduce up to 840× FLOPs and 284× CPU running time. FasterVQA is also 3.3× faster than FAST-VQA and obviously faster than real-time.

Efficiency of Mobile-Friendly Variants: Prior to our submission, the fastest in-the-wild VQA method (including classical methods) on CPU with relatively good accuracy was the RAPIQUE [64] model with 17.3 s CPU inference time. However, all three of our efficient versions can infer in less than one second on the Apple M1 CPU, which is the processor for several iPad modules. They enable the implementation of more accurate VQA methods on devices with limited computing resources, and we hope the proposed methods can help contribute to green computing on VQA.

3) Fine-Tuning on Small Datasets. End-to-end Pre-train&Fine-tune for VQA: With fragments, we are able to enable the pre-train&fine-tune scheme for VQA with affordable computational resources, which pre-trains on large VQA datasets to learn quality-related representations and fine-tunes on smaller datasets. This scheme is important as many VQA datasets [54], [60], [61], [62], [63] in specific scenarios are with much smaller scale than datasets for other video tasks [59], [69], [70], [71] and it is relatively hard to learn robust quality representations on these small VQA datasets alone. While some existing methods have to pre-train with IQA datasets due to their higher computational cost on videos, the exclusive feature of FAST-VQA to allow pre-training with VQA datasets can better learn the representations related to temporal distortions (e.g., flickers, stalls) that only happen in...
TABLE V
FINETUNE RESULTS ON LIVE-VQC, KoNViD, CVD2014, LIVE-QUALCOMM AND YOUTUBE-UGC, COMPARED WITH EXISTING CLASSICAL AND FIXED-BACKBONE DEEP VQA METHODS, AND ENSEMBLE OF CLASSICAL (C) AND DEEP (D) BRANCHES

| Time-tuning Dataset/ | LIVE-VQC | KoNViD-1k | CVD2014 | LIVE-Qualcomm | YouTube-UGC |
|---------------------|----------|----------|---------|---------------|--------------|
| resolution range in the dataset | (240P, 1080P) | (540P) | (400P, 720P) | (1080P) | (360P, 2160P(4K)) |
| Groups | | | | | |
| Existing | | | | | |
| Classical | N/A | TLQP [3] | 0.799 | 0.803 | 0.773 | 0.768 | 0.83 | 0.85 | 0.77 | 0.81 | 0.669 | 0.659 |
| | N/A | VIDEVAL [4] | 0.752 | 0.751 | 0.783 | 0.780 | NA | NA | NA | NA | 0.779 | 0.773 |
| | ImageNet [40] | VIDEVAL [4] | 0.755 | 0.786 | 0.803 | 0.817 | NA | NA | NA | NA | 0.759 | 0.768 |
| Ensemble | | | | | |
| C&D | ImageNet [40]-KoNViD-10k [45] (IQA) | CNN+TLQP [10] | 0.825 | 0.834 | 0.816 | 0.818 | 0.863 | 0.880 | 0.810 | 0.833 NA | NA |
| | ImageNet [40] | CNN+VIDEVAL [4] | 0.785 | 0.810 | 0.815 | 0.817 | NA | NA | NA | NA | 0.808 | 0.803 |
| Existing | | | | | |
| Fixed | ImageNet [40] | VsQA [5] | 0.773 | 0.795 | 0.773 | 0.775 | 0.870 | 0.888 | 0.735 | 0.732 | 0.724 | 0.743 |
| Deep | ImageNet [40] | CST-VQA [42] | NA | NA | 0.814 | 0.825 | 0.831 | 0.844 | 0.801 | 0.825 NA | NA |
| | Kinetics-400 [59]-PaQ2-PVQ [46] (IQA) | PVQ [7] | 0.827 | 0.837 | 0.791 | 0.786 | NA | NA | NA | NA | 0.816 | 0.802 |
| | Kinetics-400 [59]-VT-UGC [65] (IQA) | CoNViQ [65] | NA | NA | 0.767 | 0.764 | NA | NA | NA | NA | 0.816 | 0.802 |
| | Kinetics-400 [59]-Fused [45], [46], [67] (IQA) | BVQA-Li [13] | 0.831 | 0.842 | 0.834 | 0.836 | 0.872 | 0.869 | 0.817 | 0.828 | 0.639 | 0.619 |
| | Kinetics-400 [59] | Full-ref Swin-T [24] | 0.793 | 0.808 | 0.841 | 0.838 | 0.868 | 0.870 | 0.786 | 0.803 | 0.798 | 0.796 |
| Ours | Kinetics-400 [59]+LSVQ [7] (VQA) | FAST-VQA-M | 0.803 | 0.828 | 0.873 | 0.872 | 0.877 | 0.892 | 0.804 | 0.828 | 0.768 | 0.765 |
| | | -- standard deviation | ±0.031 ±0.030 | ±0.012 ±0.012 | ±0.033 ±0.019 | ±0.039 | ±0.026 | ±0.019 ±0.022 |
| | | FAST-VQA | 0.849 | 0.865 | 0.892 | 0.892 | 0.921 | 0.903 | 0.812 | 0.851 | 0.853 | 0.862 |
| | | -- standard deviation | ±0.024 ±0.018 | ±0.008 ±0.008 | ±0.030 ±0.019 | ±0.026 | ±0.024 | ±0.018 ±0.021 |
| | | FastVQA (Ours) | 0.843 | 0.868 | 0.895 | 0.899 | 0.904 | 0.904 | 0.836 | 0.864 | 0.863 | 0.859 |
| | | -- standard deviation | ±0.030 ±0.007 | ±0.010 ±0.010 | ±0.029 ±0.018 | ±0.036 | ±0.037 | ±0.014 ±0.017 |

Pre-training dataset(s) are noted in the table.

videos. We evaluate the ability of this new learning scheme as follows.

Results on Public Datasets: Practically, we use LSVQ as the large dataset and choose five small datasets representing diverse scenarios, including not only natural video datasets, i.e., LIVE-VQC (from real-world mobile photography, 240P-1080P) and KoNViD-1 k (from online social media contents, all 540P), but also non-natural datasets: CVD2014 (lab-collected in-capture distortions, 480P-720P), LIVE-Qualcomm (lab-collected videos with specific degradations, all 1080P) and YouTube-UGC (user-generated contents, including computer-generated contents, 360P-2160P(4K)). We divide each dataset into random splits for 10 times and report the average result on the test splits. As Table V shows, with the pre-train&fine-tune scheme, the proposed FAST-VQA and FasterVQA outperform the existing state-of-the-arts on all these five scenarios with a very large margin (up to 7%).

Results on ICME2021 UGC-VQA Challenge: We also evaluated the fine-tune performance of the proposed FAST-VQA on the ICME2021 UGC-VQA challenge [68], where the ground truths are hidden and all the methods are fairly evaluated by the challenge server. As shown in Table VI, while the top methods show very similar performance, FAST-VQA outperforms the existing state-of-the-arts on all these five scenarios with a very large margin (up to 7%), while obtaining much higher efficiency. Note that YouTube-UGC contains 4K videos with 600-frame long but even FasterVQA still performs well (+5% to existing best). Compared with existing state-of-the-arts [7], [13] that pre-train on IQA datasets [45], [46], the more direct pure VQA pre-train&fine-tune scheme enabled by fragments is more effective on all smaller VQA datasets.

C. Evaluation on Sampling Approaches

We specifically discuss the effects of the proposed sampling paradigm, quality-sensitive neighbourhood representatives, and the St-GMS (Section III-B) scheme to get fragments. We first show the effectiveness of spatial GMS by comparing it to different spatial sampling variants (Table VII), and the effectiveness of unified St-GMS by comparing it to different temporal sampling variants (Table VIII). We also discuss the sampling granularity (Fig. 7) to support the general paradigm of selecting quality-sensitive neighbourhood representatives.

1) Effects of GMS: In the Spatial Dimension: Comparing with resizing & cropping. In Group 1 of Table VII, we compare the proposed fragments with spatial GMS with two common sampling approaches: bilinear resizing and random cropping. The proposed fragments are notably superior to bilinear resizing on high-resolution (LSVQ1080P) (+4%) and cross-resolution...
TABLE VII
ABLATION STUDY FOR GMS IN SPATIAL DIMENSION: COMPARISON WITH NAÏVE APPROACHES AND VARIANTS

| Testing Set/Video Resolutions | Methods/Metric | Relative FLOPs | LSVQext 240p to 720p | LSVQtemp 1080p | KoNVid-1k 540p | LIVE-VQC 240p to 1080p |
|-----------------------------|----------------|---------------|----------------------|----------------|----------------|-----------------------|
|                            |                |               | SRCC | PLCC | SRCC | PLCC | SRCC | PLCC | SRCC | PLCC |
| Group 1: naïve Sampling Approaches |                |               |            |      |      |      |      |      |      |      |
| - bilinear resizing |                |               | 3x   | 0.857 | 0.859 | 0.752 | 0.786 | 0.841 | 0.840 | 0.772 | 0.814 |
| - random cropping |                |               | 3x   | 0.807 | 0.612 | 0.643 | 0.679 | 0.734 | 0.776 | 0.740 | 0.773 |
| - test with 3 crops |                |               | 3x   | 0.838 | 0.835 | 0.727 | 0.754 | 0.841 | 0.827 | 0.785 | 0.809 |
| - test with 6 crops |                |               | 6x   | 0.843 | 0.844 | 0.734 | 0.761 | 0.845 | 0.834 | 0.796 | 0.817 |
| - resizing+cropping with 3 crops, as in [24] |                |               | 3x   | 0.860 | 0.662 | 0.758 | 0.795 | 0.845 | 0.846 | 0.768 | 0.817 |
| Group 2: Variants of fragments in the spatial dimension |                |               |      |      |      |      |      |      |      |      |
| - random mini-patches |                |               | 1x   | 0.867 | 0.861 | 0.754 | 0.790 | 0.844 | 0.845 | 0.792 | 0.818 |
| - shuffled mini-patches |                |               | 1x   | 0.868 | 0.863 | 0.761 | 0.799 | 0.849 | 0.847 | 0.796 | 0.821 |
| - fixed center sampling per grid during training |                |               | 1x   | 0.870 | 0.871 | 0.772 | 0.805 | 0.854 | 0.853 | 0.812 | 0.834 |
| - with temporal alignment |                |               | 1x   | 0.850 | 0.853 | 0.776 | 0.797 | 0.823 | 0.816 | 0.764 | 0.802 |
| GMS (FasterVQA, Ours) |                |               | 1x   | 0.876 | 0.877 | 0.779 | 0.814 | 0.859 | 0.855 | 0.823 | 0.844 |

TABLE VIII
ABLATION STUDY FOR ST-GMS ON THE TEMPORAL DIMENSION: COMPARISON WITH NAÏVE APPROACHES AND VARIANTS

| Testing Set/Inter-frame Variations | Methods/Metric | Relative FLOPs | LSVQext weak to medium | LSVQtemp medium | KoNVid-1k weak | LIVE-VQC strong |
|-----------------------------------|----------------|---------------|------------------------|----------------|----------------|----------------|
|                                   |                |               | SRCC | PLCC | SRCC | PLCC | SRCC | PLCC | SRCC | PLCC |
| Group 1: naïve Sampling Approaches |                |               |      |      |      |      |      |      |      |      |
| - sampling a continuous short clip |                |               | 0.25x | 0.856 | 0.856 | 0.780 | 0.785 | 0.833 | 0.834 | 0.782 | 0.812 |
| - uniform sampling (sparse, no continuous frames) |                |               | 0.25x | 0.859 | 0.856 | 0.753 | 0.790 | 0.843 | 0.842 | 0.774 | 0.808 |
| Group 2: Variants of fragments in the temporal dimension |                |               |      |      |      |      |      |      |      |      |
| - temporally random mini-cubes |                |               | 0.25x | 0.863 | 0.866 | 0.798 | 0.797 | 0.851 | 0.852 | 0.803 | 0.827 |
| - temporally shuffled mini-cubes |                |               | 0.25x | 0.864 | 0.866 | 0.756 | 0.793 | 0.853 | 0.854 | 0.807 | 0.828 |
| St-GMS (FasterVQA, Ours) |                |               | 0.25x | 0.873 | 0.874 | 0.772 | 0.811 | 0.864 | 0.863 | 0.813 | 0.837 |

(LIVE-VQC) scenarios (+4%). Fragments still lead to non-trivial 2% improvements over resizing on lower-resolution scenarios where the problems of resizing are not that severe. This proves that keeping local textures is vital for VQA. Fragments also largely outperform single random crops as well as ensembles of multiple crops, suggesting that retaining uniform global quality is also critical to VQA. We additionally compare with Swin-T’s original inference samples for video recognition, resizing+cropping with three crops, which need 3× computational cost but still perform notably worse than fragments.

Comparing With Spatial Variants of Fragments: We also compare with three variants of fragments in Table VII, 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 (sample mini-patches in grids but shuffle them while splicing). The proposed GMS is markedly superior to both variants. Furthermore, we explore the variant with fixed center samples in each grid during training, which is notably worse than the adopted uniform random sampling strategy, probably due to its compromised data diversity that fails to sufficiently train the GRPB module in FANet. Moreover, the variant without alignment are significantly worse than our default aligned design, especially on high-resolution videos, indicating that preserving temporal variations between frames is necessary for accurate video quality evaluations.

2) Effects of St-GMS: In the Temporal Dimension. Comparing With Uniform & Short-Clip Sampling: In Group 1 of Table VIII, we compare the proposed spatial-temporal fragments with St-GMS in the temporal dimension with two prevalent temporal sampling strategies: sampling a short clip and uniform sampling. A short clip leads to a notable performance drop on KoNVid-1k [60], where a non-uniform sample is insufficient to account for the changing content over time. Uniform sampling lacks continuous frames and is especially inaccurate on LIVE-VQC [54], where inter-frame variations are very complicated. The proposed FasterVQA with St-GMS is representative and sensitive to temporal quality and performs better in a variety of situations.

Comparing With Temporal Variants of Fragments: Similar to the spatial situation, we also discussed random (ignore segments while sampling) and shuffled mini-cubes. The results suggest that preserving contextual relations is still important in the temporal dimension and leads to a performance gap of around 1% across all datasets. However, the gap is notably smaller than in the spatial dimension, indicating that the temporal contextual relations may be less influential on quality than their spatial counterparts.

3) Discussion on Sampling Granularity: We sample the fragments based on the paradigm of quality-sensitive neighbourhood representatives, where we stress two important factors: 1) partitioned neighbourhoods (the more, the better representative); 2) continuous representatives (the larger, the better textural sensitivity). They have to be balanced during practical sampling.
We discuss the two important factors by evaluating the spatial and temporal granularity of sampling given a fixed total sample size.

Spatial Granularity: $G_i \& S_f$ in GMS. We discuss different combinations of number of grids ($G_i$) and size of mini-patches ($S_f$) for GMS, including combinations that follow (solid curves) or not follow (dashed curves) the match constraint (8). We notice that setting $S_f = 32$ shows best performance and is better than smaller patches which gradually becomes insensitive to local textures and degenerates into resizing), or larger patches which gradually cedes to be representative to global quality and degenerates into cropping. (Results of cropping are in Table VII).

Temporal Granularity: $G_i \& T_f$ in St-GMS. We also discuss the combinations of number of $G_i$ and $T_f$ for St-GMS given the same total frames. As no temporal pooling is operated in FANet, we only have the matched group, as shown in Fig. 7(b). The $T_f = 4$ shows best performance on both datasets which is comparable to dense temporal sampling (FAST-VQA), which follows our observation that a few continuous frames can be sensitive to temporal variations.

D. Ablation Studies on FANet, Training and Inference

1) Effects of the Match Constraint. Effects of Appropriate Backbones: In the first part of our ablation studies on FANet, we discuss the effects of different backbone structures by dividing them into two groups: those with non-overlapping pooling layers and can comply with the match constraint (Swin-T, inflated ConvNeXt-Tiny) and others (I3D [25] with ResNet-50 backbone under a modern initialization [72]). The IP-NLR is included in all variants, while the GRPB is excluded as it is particularly designed for Swin-T. As shown in Table IX, the matched backbones are significantly more effective at processing fragments as inputs given similar computational cost, demonstrating our analysis for the match constraint (8).

Effects of Matching Mini-Cubes With Pooling: We further discuss the match constraint by comparing the spatial matched (solid lines) vs mis-matched mini-cubes (dashed lines) with the same backbone structure. As Fig. 7(a) shows, the non-matched combinations of pooling kernels and mini-cubes show notably worse performance in all situations, again proving the importance of the match constraint.

2) Effects of GRPB and IP-NLR: In the second part of the ablation studies on FANet, we analyze the effects of two novel modifications in it: the proposed Gated Relative Position Biases (GRPB) and Intra-Patch Non-Linear Regression (IP-NLR) Head as in Table X. 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 videos, this result suggests that the corrected position biases and regression head are helpful on solving the problems caused by such discontinuity.

3) Effects of Adaptive Multi-Scale Inference (AMI): In the third part, we evaluate the importance of Adapive Multi-scale Inference (AMI) to allow inference of FasterVQA on different scales with only training on one base scale. In Table XI, we evaluate the inference accuracy on MT and MS scales with or without AMI. The results have demonstrated the effectiveness of AMI, which allows robust inference adapted for different computing environments.

| TABLE IX | ABILATION STUDY ON BACKBONES: NETWORKS THAT FOLLOW THE MATCH CONSTRAINT ARE SIGNIFICANTLY BETTER |
|----------|--------------------------------------------------|
| Variants/Metric | LSVQtest | LSVQtrain | KeNViD-1k | LIVE-VQC |
| ------------------ |----------|-----------|-----------|-----------|
| SRCC | PLCC | SRCC | PLCC | SRCC | PLCC | SRCC | PLCC |
| "non-matched" backbone (with overlapping pooling kernels): | | | | | | | |
| I3D-ResNet50 | 0.867/0.846 | 0.717/0.704 | 0.828/0.829 | 0.775/0.708 | | | |
| ConvNeXt-Tiny | 0.869/0.870 | 0.765/0.802 | 0.851/0.852 | 0.811/0.833 | | | |
| Swin-T (w/o GRPB) | 0.873/0.872 | 0.769/0.805 | 0.855/0.853 | 0.808/0.832 | | | |
| "matched" backbone (with non-overlapping pooling kernels): | | | | | | | |
| I3D-ResNet50 | 0.873/0.872 | 0.769/0.805 | 0.855/0.853 | 0.808/0.832 | | | |
| ConvNeXt-Tiny | 0.869/0.870 | 0.765/0.802 | 0.851/0.852 | 0.811/0.833 | | | |
| Swin-T (w/o GRPB) | 0.873/0.872 | 0.769/0.805 | 0.855/0.853 | 0.808/0.832 | | | |

| TABLE X | ABILATION STUDY ON GRPB AND IP-NLR |
|----------|-----------------------------------|
| Variants/Metric | LSVQtest | LSVQtrain | KeNViD-1k | LIVE-VQC |
| ------------------ |----------|-----------|-----------|-----------|
| SRCC | PLCC | SRCC | PLCC | SRCC | PLCC | SRCC | PLCC |
| Variants of GRPB | | | | | | | |
| w/o GRPB (baseline) | 0.873/0.872 | 0.769/0.805 | 0.854/0.853 | 0.794/0.778 | | | |
| GRPB on Layers 1&2 | 0.868/0.869 | 0.763/0.802 | 0.849/0.847 | 0.780/0.773 | | | |
| Variants of IP-NLR | | | | | | | |
| w/o IP-NLR (Baseline) | 0.873/0.872 | 0.769/0.803 | 0.847/0.849 | 0.794/0.778 | | | |
| w/ non-linear, post-first | 0.873/0.872 | 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 | | | |
We then evaluate the resolution-varied AMI on cross-resolution scenarios: LIVE-VQC [54], and LSVQ cross-resolution test set (LSVQ_{test}+LSVQ_{1080p}) [7]. As the results shown in Table XIV, the resolution-varied AMI can outperform the respective naïve fixed settings (with all $G_r$ set as 5) under similar average FLOPs. We believe more advanced adaptive sampling strategies can be explored in the future.

4) Effects of End-to-End Pre-Train&Fine-Tune Scheme: We discuss the effects of pre-train&fine-tune scheme (Section IV-B3) in Table XIV in comparison with direct training on these small datasets (w/o end-to-end pre-train) and only linear regression on pre-trained features (w/o end-to-end finetune). The large-scale pre-training contributes to the performance by up to 11%, and are especially effective on cross-resolution scenarios, e.g., LIVE-VQC and YouTube-UGC. The end-to-end fine-tune also lead to up to 8% improvements, especially on non-natural videos (CVD2014, LIVE-Qualcomm, YouTube-UGC) which may contain specific quality-related issues. Both stages are undoubtedly effective and made affordable via the proposed fragments.

E. Role of Semantics in FAST-VQA/FasterVQA

Can Fragments Preserve Semantics? In our discussions in Section III-B2, one question remains unclear: can the fragments retain aware to semantic video contents that can still be recognized by deep neural networks? This can hardly be answered as for a 10-sec-long 720P video, fragments sampled by St-GMS contain only 0.58% original information. Thus, we measure the ability by experiments: we use fragments as classification inputs for videos in Kinetics-400 [59] action recognition dataset, and the results prove that simply fine-tuning the Swin-T backbone with fragments can reach 68.6% top-1 accuracy (87.4% relative to original Swin-T which needs 12 samples and requires $12 \times$ FLOPs) and 88.7% top-5 accuracy (94.8% relative to original), which has been on par with several deep VQA approaches under similar computational cost. The absolute accuracy also suggests that the fragments still contain rough scene-level semantics and can be recognized by the backbone in FANet.

Effects of Semantic Pre-Training: We further discuss the significance of semantic pre-training by training FAST-VQA/FasterVQA models from scratch (w/o semantics) as their semantic-blind variants, and the proposed models are regarded as semantic-aware (w/ semantics) variants based on discussions above. As shown in Table XIII, semantic pre-training has significantly contributed to the performance on FAST-VQA (avg. 8%) and FasterVQA (avg. 10%), especially FasterVQA. We also observed that the intra-dataset performance of the state-of-the-art classical VQA approaches is comparable to that of our variants without semantic pre-training. The results indicate the significant influence of semantics in VQA and suggest that there might exist an accuracy limit of all semantic-blind VQA methods. This further proves that semantic-aware deep VQA methods are irreplaceable, while FAST-VQA and FasterVQA fill in the blanks on improving their practical efficiency.

F. Evaluation on High-Resolution Videos

As the base version of FAST-VQA only samples 5.44% and 2.42% spatial information from 720P and 1080P videos, respectively, it is worthwhile to evaluate its performance on high-resolution videos. We use two existing databases with 1080P videos: for cross-resolution LIVE-VQC, we split the videos according to their resolutions and test the performance of different variants; for LSVQ_{1080p}, we create variants by downsampling its 1080P videos before sampling fragments and compare between them.

1) Performance on Split Resolutions: We divide the cross-resolution VQA benchmark set LIVE-VQC into three resolution groups: (A) 1080P (110 videos); (B) 720P (316 videos); and (C) ≤540P (159 videos) to evaluate the performance of FAST-VQA on different resolutions in comparison to other variants. As shown in Table XV, the proposed FAST-VQA achieves good performance on all resolution groups ($\geq$0.80 SRCC&PLCC), with the most superior improvement over other variants on Group (A) with 1080P high-resolution videos, proving that FAST-VQA is robust and reliable on videos with different resolutions.

2) Impacts of Video Downsampling: To demonstrate that keeping the raw-resolution textures is crucial in sampling fragments, we evaluate the proposed FAST-VQA/FasterVQA with multiple downsampling variants of LSVQ_{1080p} dataset. We resize these 1080P high-resolution videos into 540P(2X↓), 360P(3X↓), 270P(4X↓) and sample fragments from the resized videos. As shown in Fig. 9, although downsampling before sampling can preserve more information from these videos, the overall effect still significantly degrades the final accuracy, proving that keeping the original resolution is crucial to quality sensitivity. As the model is only trained on videos ≤720P, the result further reveals
G. Stability and Reliability Analysis

Due to the randomness of fragment sampling, the proposed FAST-VQA may produce varying predictions for the same video during inference. Therefore, we measure the stability and reliability of single random sampling in FAST-VQA using two metrics: 1) the assessment stability of multiple single samplings on the same video; 2) the relative accuracy of single sampling compared with multiple sample ensemble. As shown in Table XVI, the normalized std. dev. of different sampling on the same video is only around 0.01, indicating that the sampled fragments are enough for making highly stable predictions. Compared with a six-sample ensemble, sampling only once and randomly can be 99.40% as accurate even on the pure high-resolution test set (LSVQ\textsubscript{1080P}); compared with fixed center sampling, our default one random sample for each grid also shows comparable and even slightly better accuracy.\cite{footnote}

These results prove that a single sample of fragments is sufficiently stable and reliable for quality assessment even though only a very small proportion of information is kept during sampling.

H. Visualizations of Local Quality Maps

The proposed IP-NLR head with patch-wise independent quality regression not only improves the performance of the proposed method but also enables the generation of spatial-temporal local quality maps as \cite{footnote}. These quality maps allow us to qualitatively evaluate what can be learned during the

![Fig. 8. Local quality maps on different frames of one video, where red areas indicate low predicted quality and green areas indicate high predicted quality. This sample video is a 1080P video from LIVE-VQC \cite{footnote} dataset. Zoom in for clearer view.](image)

![Fig. 9. Impacts of downsampling 1080P videos in LSVQ\textsubscript{1080P}.](image)

\begin{table}[h]
\centering
\caption{Stability and Reliability of Single Random Sampling During Inference}
\begin{tabular}{l|c|c|c|c}
\hline
Testing Set/ Score Range & LSVQ\textsubscript{1080P} & LSVQ\textsubscript{1080P} & KeNViD-1k & LIVE-VQC \\
\hline
(a) Stability of One Random Sample during Inference & 0.01 & 0.01 & 0.01 & 0.01 \\
\hline
\hline
(b) Set-wise Accuracy (KCC) on Different Sampling Strategy during Inference: & & & & \\
\hline
one fixed center sample & 0.691 & 0.5966 & 0.6489 & 0.6256 \\
\hline
one random sample & 0.6918 & 0.5862 & 0.6693 & 0.6269 \\
6 random sample ensemble & 0.6947 & 0.5897 & 0.6730 & 0.6326 \\
Relative Accuracy (1 to 6) & 99.59% & 99.40% & 99.35% & 99.52% \\
\hline
\end{tabular}
\end{table}

\footnote{Our published code also provides options to support deterministic outputs (fixed center sampling during inference) and boost outputs (by ensembling 6 random samples, slower but more accurate).}
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 [54] dataset) in Fig. 8. As the patch-wise quality maps and re-projected quality maps in Fig. 8 (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 those of the background areas, showing that FAST-VQA effectively learns the global semantics.

To further validate the effectiveness of local quality prediction for FAST-VQA, we visualize its local quality maps on diverse contents of videos, as shown in Fig. 10. Similarly, in Video 4, the areas more related to semantic information are predicted with higher quality. Moreover, as the blurs occurred in Video 1 (Frame 12/24), other distortions such as noises (Video 2), over-exposure (Video 4), or under-exposure (Video 5) can also be detected locally. To facilitate future research, we have published the scripts to generate local quality maps on custom videos in our GitHub repository.

V. CONCLUSION

In this paper, we have discussed sampling for video quality assessment (VQA) in order to tackle the difficulties as a result of high computing and memory requirements when evaluating high-resolution videos. We propose the principle of quality-sensitive neighbourhood representatives and conduct extensive experiments to demonstrate that the proposed samples, fragments, are effective samples for VQA that retain better quality information in videos than naïve sampling approaches. Thanks to the significantly reduced input complexity in fragments, the proposed end-to-end FAST-VQA (with only spatial sampling) and FasterVQA (with unified spatial-temporal sampling) refreshed state-of-the-arts on all in-the-wild VQA benchmarks with up to $1612 \times$ efficiency than the existing state-of-the-art. The proposed methods can bring deep VQA methods into practical use with any video resolution or length. In our future work, we would like to further explore the paradigm of neighbourhood representative sampling in more related areas.

REFERENCES

[1] M. A. Saad, A. C. Bovik, and C. Charrier, “Blind image quality assessment: A natural scene statistics approach in the DCT domain,” IEEE Trans. Image Process., vol. 21, no. 8, pp. 3339–3352, Aug. 2012.
[2] A. Mittal, M. A. Saad, and A. C. Bovik, “A completely blind video integrity oracle,” IEEE Trans. Image Process., vol. 25, no. 1, pp. 289–300, Jan. 2016.
[3] J. Korhonen, “Two-level approach for no-reference consumer video quality assessment,” IEEE Trans. Image Process., vol. 28, no. 12, pp. 5923–5938, Dec. 2019.
[4] Z. Tu, Y. Wang, N. Birkbeck, B. Adsumilli, and A. C. Bovik, “UGC-VQA: Benchmarking blind video quality assessment for user generated content,” IEEE Trans. Image Process., vol. 30, pp. 4449–4464, 2021.
K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–777.

J. You, “Long short-term convolutional transformer for no-reference video quality assessment,” in Proc. ACM Int. Conf. Multimedia, 2021, pp. 21–21.

B. Li, W. Zhang, M. Tian, G. Zhai, and X. Wang, “Blindly assess quality of in-the-wild videos via quality-aware pre-training and motion perception,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 9, pp. 5944–5958, Sep. 2022.

H. Wu et al., “DisCoVQA: Temporal distortion-contrast transformers for video quality assessment,” 2022, arXiv:2206.09853.

C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in Proc. AAAI Conf. Artif. Intell., 2017, pp. 4278–4284.

K. Hara, H. Katoaka, and Y. Satoh, “Learning spatio-temporal features with 3D residual networks for action recognition,” in Proc. Int. Conf. Comput. Vis. Workshops, 2017, pp. 3154–3160.

R. A. Poldrack and M. J. Farah, “Progress and challenges in probing the human brain,” Nature, vol. 526, no. 7573, pp. 373–379, 2015.

M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira, “Exploiting spatial redundancy in pixel domain Wyner-Ziv video coding,” in Proc. IEEE Conf. Image Process., 2006, pp. 253–256.

D. J. Li, “The MPEG video compression algorithm,” Signal Process.: Image Commun., vol. 4, no. 2, pp. 129–140, 1992.

M. Buckler, P. Bedoukian, S. Jayasuriya, and A. Sampson, “EVA: Exploiting temporal redundancy in live computer vision,” in Proc. ACM/IEEE 45th Ann. Int. Symp. Comput. Archit., 2018, pp. 533–546.

G. K. Wallace, “The JPEG still picture compression standard,” Commun. ACM, vol. 34, no. 1, pp. 30–44, Apr. 1991.

T. Wiegand, “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H. 264-ISO/IEC 14496-10 AVC),” JVT-G050, 2003.

R. Keys, “Cubic convolution interpolation for digital image processing,” IEEE Trans. Acoust. Speech Signal Process., vol. 29, no. 6, pp. 1153–1160, Dec. 1981.

Z. Liu et al., “Video swim transformer,” 2021, arXiv:2106.13230.

J. Carreira and A. Zisserman, “Quo Vadis, action recognition? A new model and the kinetics dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4724–4733.

H. Fan et al., “Multiscale vision transformers,” in Proc. Int. Conf. Comput. Vis., 2021, pp. 6828–6836.

A. Ararn, M. Delhaghi, G. Heigold, C. Sun, M. Lucic, and C. Schmid, “ViViT: A video vision transformer,” in Proc. Int. Conf. Comput. Vis., 2021, pp. 6836–6846.

L. Kang, P. Ye, Y. Li, and D. Doermann, “Convolutional neural networks for no-reference image quality assessment,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 1733–1740.

L. Wang et al., “Temporal segment networks for action recognition in videos,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 11, pp. 2740–2755, Nov. 2019.

A. Kolesnikov et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Representations, 2021.

Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 11 976–11 986.

A. Howard et al., “Searching for MobileNetV3,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 1314–1324.

A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inform. Process. (Adv. Neural Inform. Process. Soc.), 2017, pp. 6000–6001.

Z. Sinno and A. C. Bovik, “Large-scale study of perceptual video quality,” IEEE Trans. Image Process., vol. 28, no. 2, pp. 612–627, Feb. 2019.

X. Liu, J. Van De Weijer, and A. D. Bagdanov, “Exploiting unlabeled data in CNNs by self-supervised learning to rank,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 12, pp. 1862–1879, Apr. 2019.

W. Zhang, K. Ma, G. Zhai, and X. Yang, “Uncertainty-aware blind image quality assessment in the laboratory and wild,” IEEE Trans. Image Process., vol. 30, pp. 3474–3486, 2021.
Haoning Wu received the BS degree from the School of Electronic Engineering and Computer Science, Peking University, Beijing, China, in 2021. He is currently working toward the PhD degree with S-Lab, School of Computer Science and Engineering, Nanyang Technological University, Singapore, supervised by Prof. Weisi Lin (Fellow, IEEE). His research interests mainly include video quality assessment, including improving its robustness, efficiency, and interpretability and connecting it with related tasks.

Liang Liao (Senior Member, IEEE) received the BS degree from the International School of Software, Wuhan University, Wuhan, China, in 2013, and the PhD degree from the National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, in 2019. He is currently a research fellow with S-lab, School of Computer Science and Engineering, Nanyang Technological University, Singapore. He was a project researcher with the National Institute of Informatics, Japan, from 2019 to 2022. His research interests include transmission, and quality assessment.

Jingwen Hou (Student Member, IEEE) received the bachelor’s degree in E-commerce from the Beijing University of Posts and Telecommunications, Beijing, China, and the master’s degree in E-business from Carnegie Mellon University, Pittsburgh, USA. He is currently pursuing working toward the PhD degree with School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include multimedia quality assessment and computer vision.

Wenxiu Sun received the BE degree from the Department of Electronic Engineering, Nanjing University, Nanjing, China, in 2009, and the PhD degree from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2014. She is currently a research director with SenseTime. Her research interests include image processing, computer vision, and machine learning.

Chaofeng Chen received the BEng degree from the Department of Computer Science, Huazhong University of Science and Technology, Wuhan, China, in 2015, and the PhD degree from the Department of Computer Science, The University of Hong Kong, Hong Kong SAR, China, in 2021. He is currently a research fellow with S-lab, School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include image/video restoration, editing and generation, quality assessment, and face related tasks.
Jinwei Gu (Senior Member, IEEE) received the BS and MS degrees from Tsinghua University, in 2002 and 2005 respectively, and the PhD degree from Columbia University, in 2010. He is the R&D executive director of SenseTime USA. His current research focuses on low-level computer vision, computational photography, smart visual sensing and perception, and robotics. Before joining SenseTime, he was a senior research scientist with NVIDIA Research from 2015 to 2018. Prior to that, he was an assistant professor with the Rochester Institute of Technology from 2010 to 2013, and a senior researcher with the Media Lab of Futurewei Technologies from 2013 to 2015. He serves as an associate editor for *IEEE Transactions on Computational Imaging* and *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Weisi Lin (Fellow, IEEE) received the PhD degree from King’s College, London University, U.K. He is currently a professor with the School of Computer Science and Engineering, Nanyang Technological University. His research interests include expertise in image processing, perceptual signal modeling, video compression, and multimedia communication, in which he has published more than 200 journal papers, more than 230 conference papers, filed seven patents, and authored two books. He has been an invited/panelist/keynote/tutorial speaker in more than 20 international conferences, as well as a distinguished lecturer with the *IEEE Circuits and Systems Society* (2016–2017), and the Asia-Pacific Signal and Information Processing Association (APSIPA) (2012–2013). He is a fellow of the IET, and an honorary fellow of Singapore Institute of Engineering Technologists. He has been a Technical Program Chair for IEEE ICME 2013, PCM 2012, QoMEX 2014, and IEEE VCIP 2017. He has been an associate editor for *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Multimedia*, and the *IEEE Signal Processing Letters*. 