



Video Transformer based Video Quality Assessment with Spatiotemporally adaptive Token Selection and Assembly

Shiling Zhao*, Haibing Yin**+, Hongkui Wang**+, Yang Zhou*

*Hangzhou Dianzi University, Hangzhou, China

+Lishui Research Institute of Hangzhou Dianzi University, Lishui, China

目录

CONTENTS

01 Background

02 Approach

03 Experimental Results

04 Conclusion



01

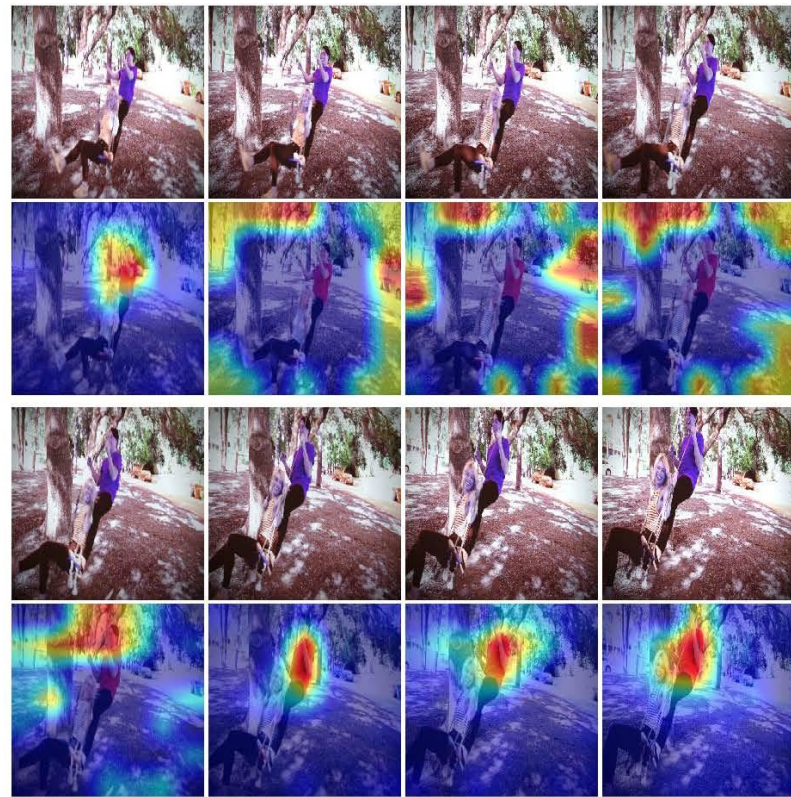
Background

Background

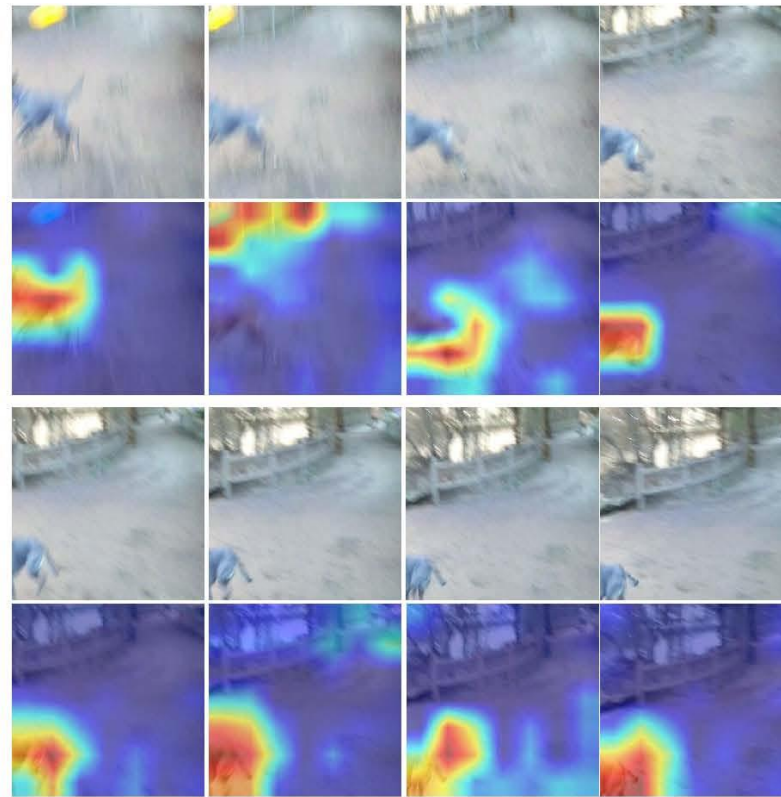
Existing sampling strategies for VQA:

Regular sampling or Random sampling etc.

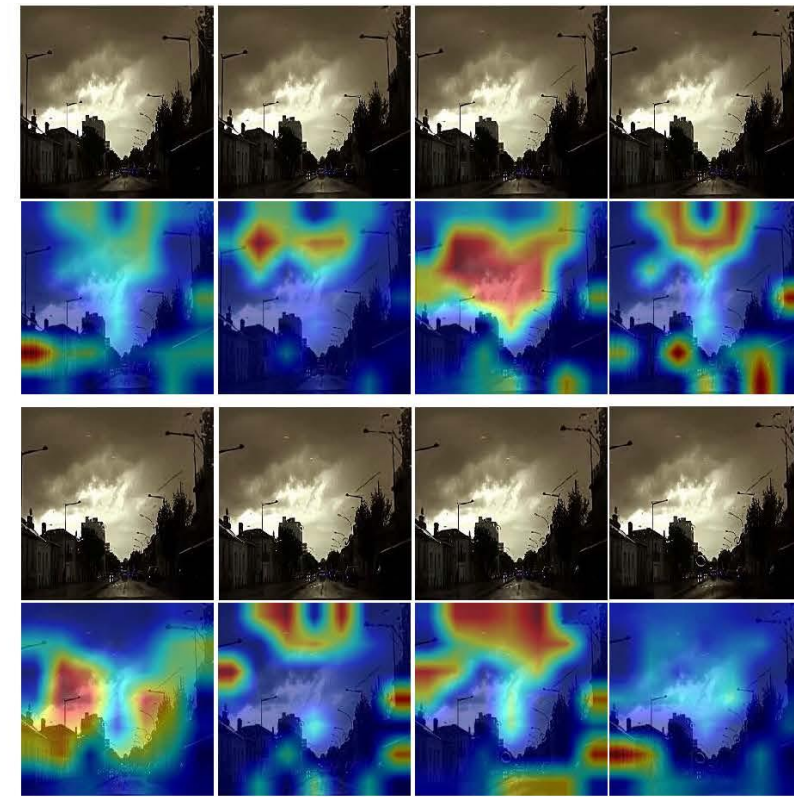
Distribution of distorted information:



LIVE-VQC



YouTube-UGC



KoNViD-1k

Background

Why do we use the transformer architecture ?

Transformer models are able to better capture the global relationships and dependencies within the sequence, and can perform parallel computations faster than CNN models.

Where does the transformer architecture fall short in the video quality assessment task?

The self-attention calculation for each layer of a traditional Transformer is uniformly executed on the entire token set, which falls short in ability to suppress unimportant regions.

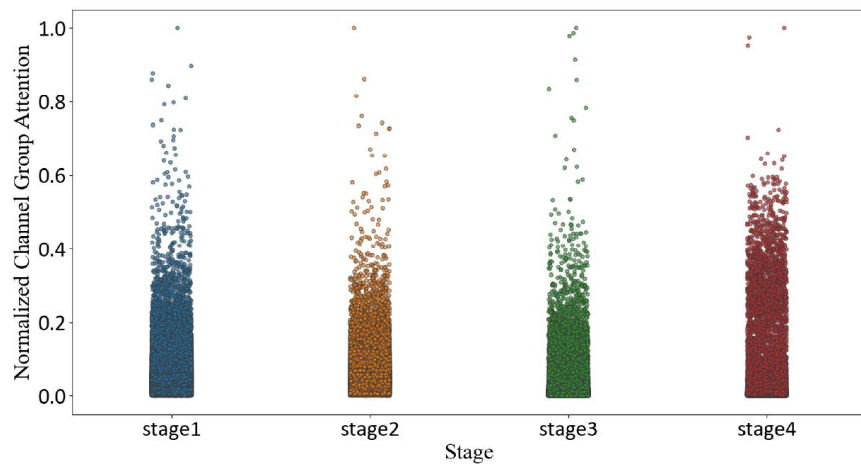


Figure 1

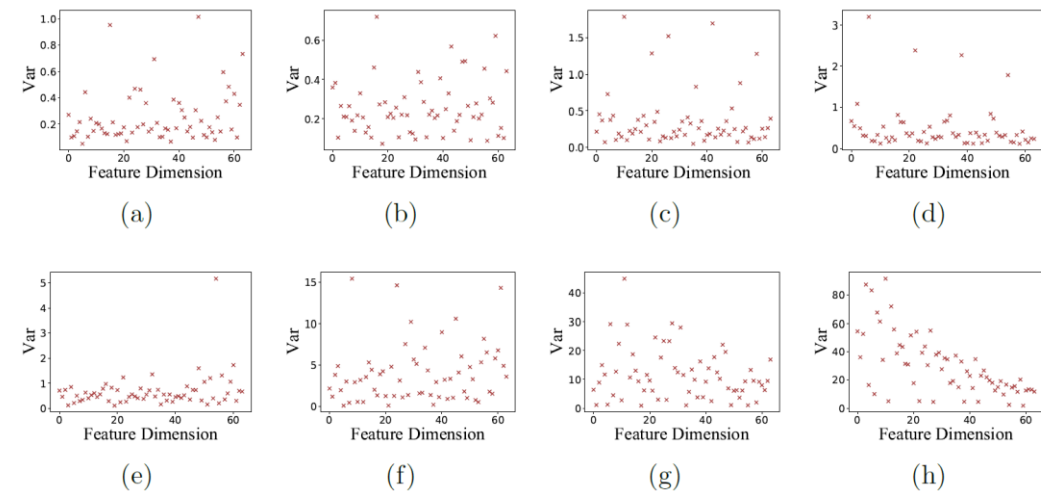
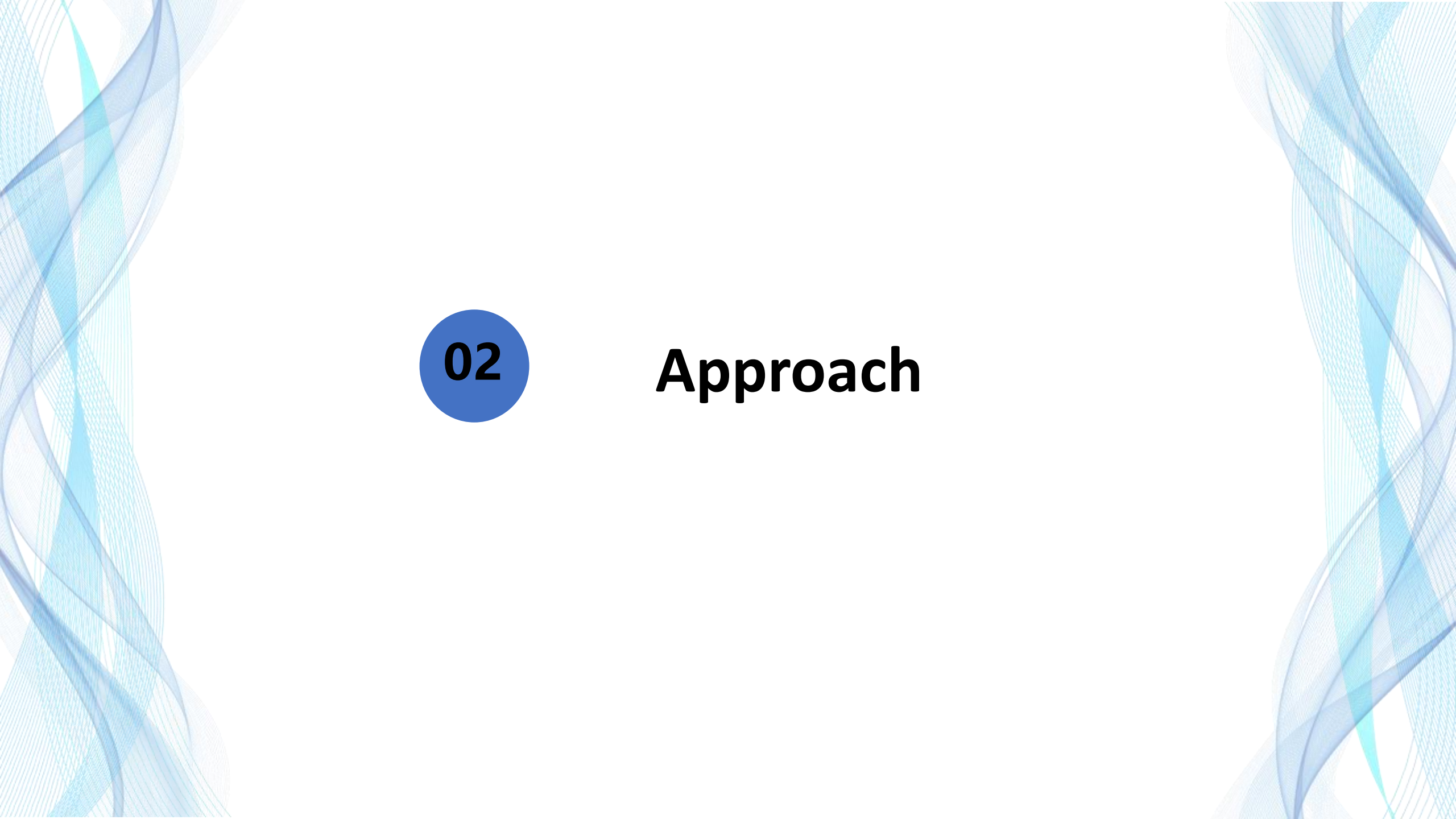


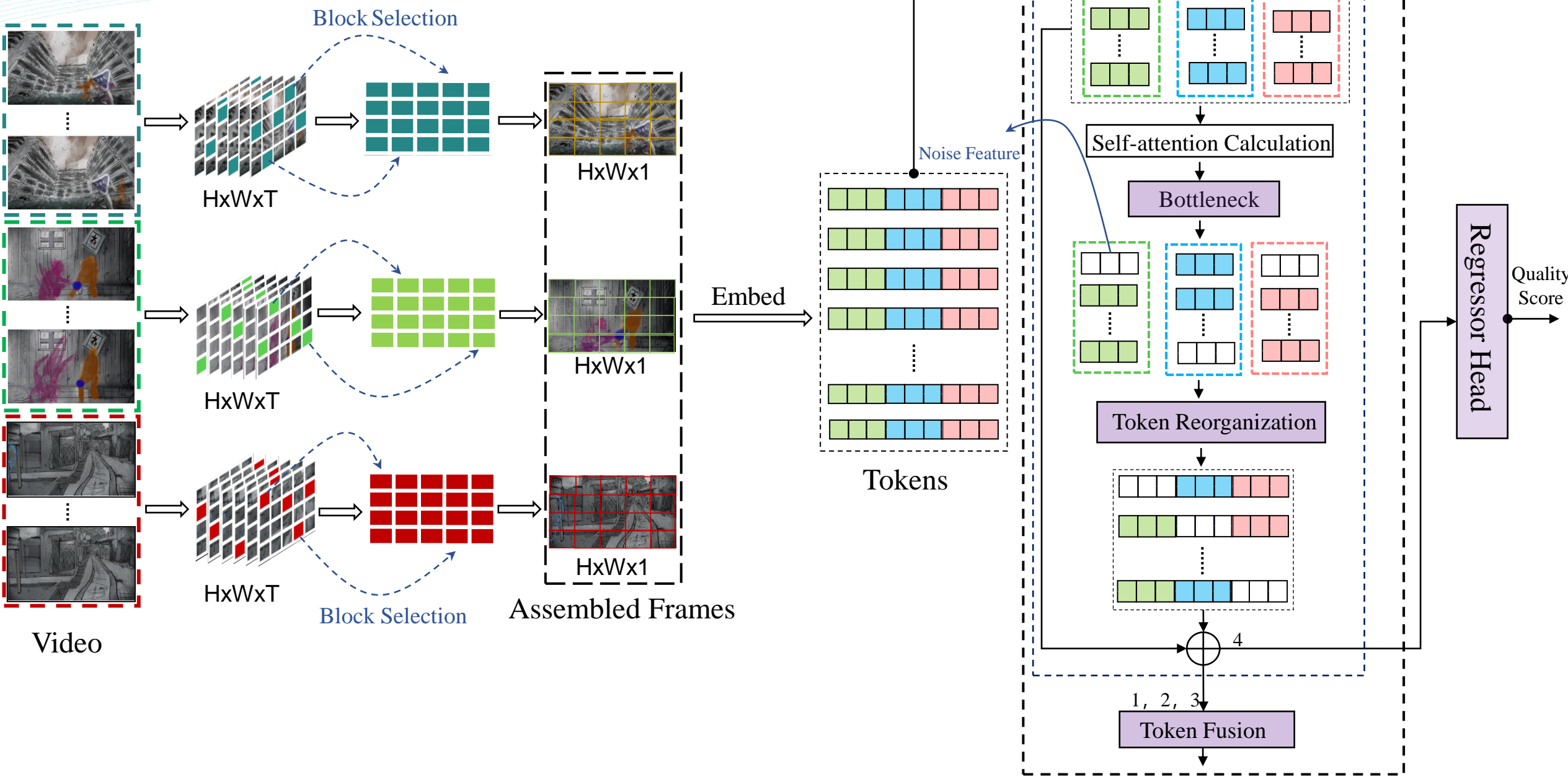
Figure 2



02

Approach

Approach

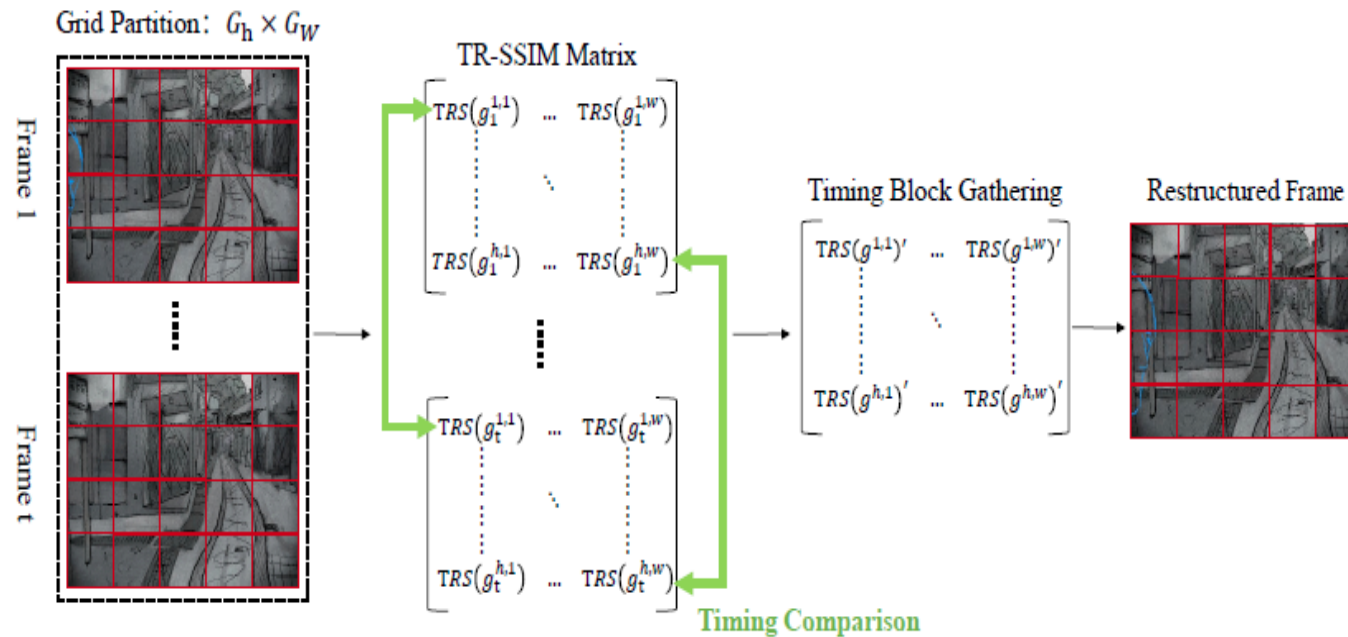


Approach

Timing Block Sampling takes into account the uneven distribution of local quality distortions focused on by the human eye in the original sequence.

Increasing the information density of the sampled frame set.

Reducing the loss possibility of important spatial features through HVS-based fine-grained sampling.



Approach

Stage-wise adaptive Screening Network(SSNet) divides the process of visual information processing into four stages based on the filter theory of attention.

Detecting and processing the “noise” of visual information step by step to obtain better representational capacity of tokens.

Two main components:

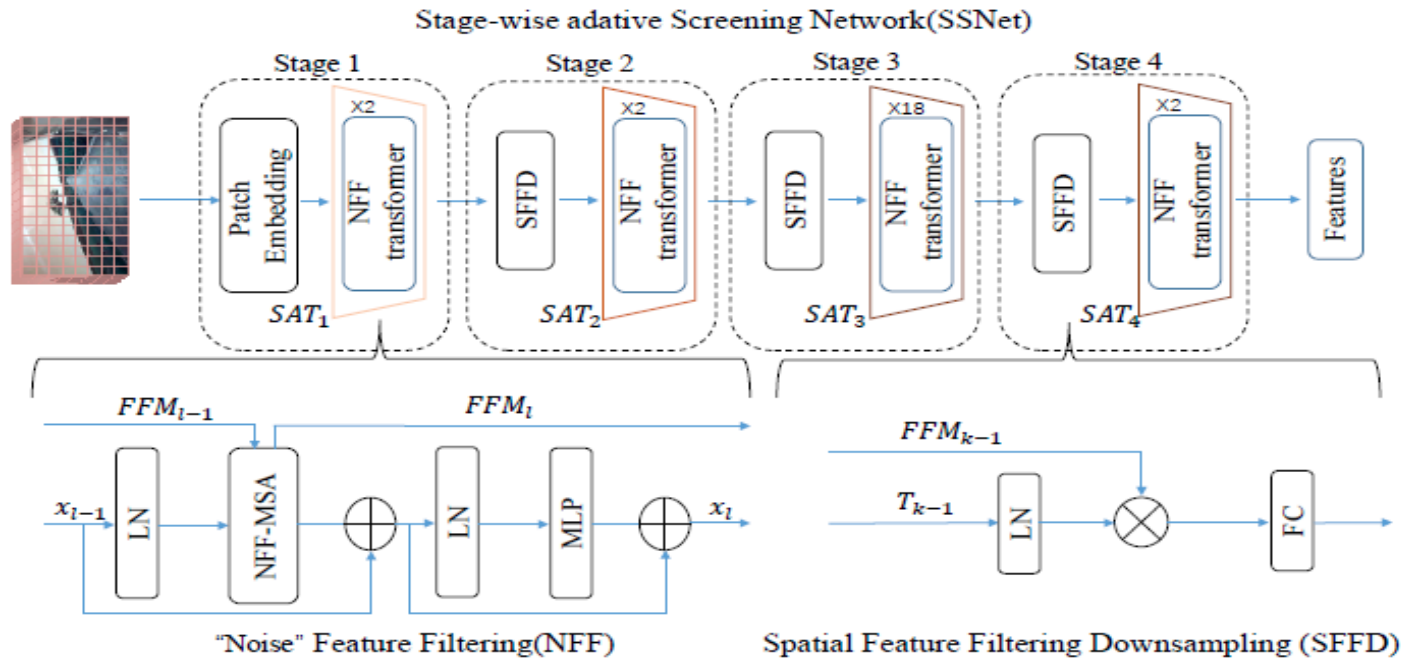
- **“Noise” Feature Filtering Transformer Layer**
- **Spatial Feature Filtering Down-sampling**

Approach

“Noise” Feature Filtering Transformer Layer

Two threshold : proportion threshold and variance threshold.

- Considering the variations of attention weights across tokens at different stages.
- Considering the differences in attention weights for each token across different layers within the same stage.





03

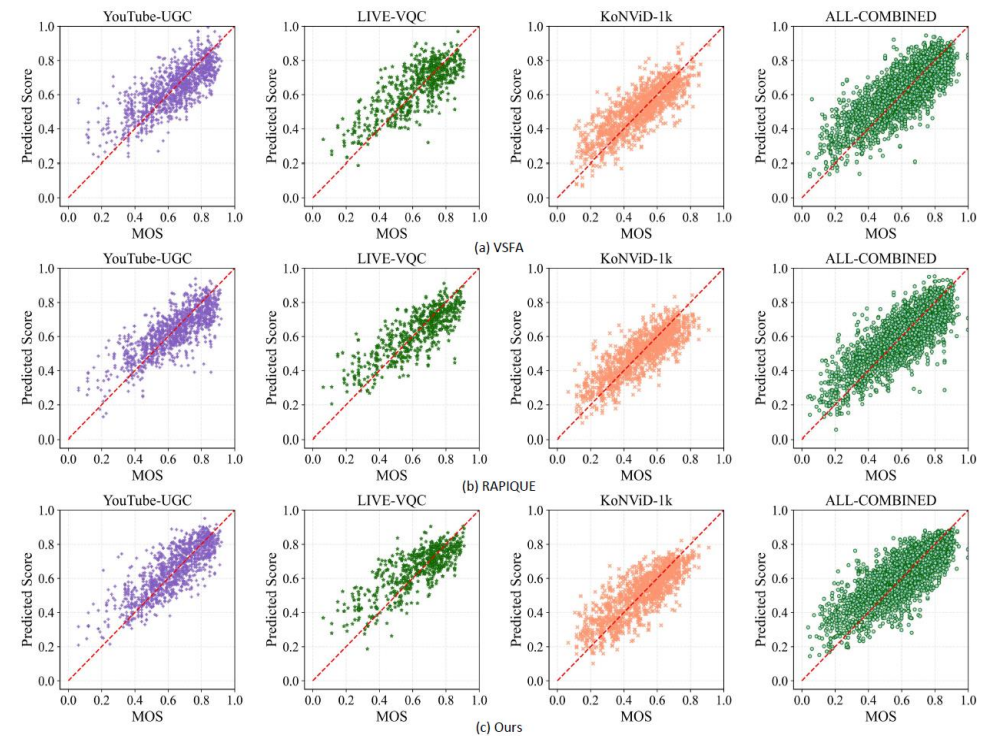
Experimental Results

Experimental Results

Method	KoNViD-1k[10]		YouTube-UGC[11]	
	SRCC	PLCC	SRCC	PLCC
BRISQUE[12]	0.656	0.657	0.382	0.395
CORNIA[13]	0.716	0.713	0.597	0.605
V-BLIINDS[14]	0.710	0.703	0.559	0.555
TLVQM[4]	0.783	0.768	0.669	0.659
RAPIQUE[15]	0.803	0.818	0.759	0.768
CoINVQ[3]	0.767	0.764	0.816	0.802
Ours	0.793	0.800	0.819	0.828

- the scattered point distribution of our method is closer to the diagonal compared with VSFA and RAPIQUE, which demonstrates the effectiveness of the proposed ATSViT.

- The proposed ATSViT achieves best performance on YouTube-UGC and performs slightly worse than RAPIQUE on KoNViD-1k.



Experimental Results

Method	KoNViD-1k[10]		YouTube-UGC[11]	
	SRCC	PLCC	SRCC	PLCC
Regular Sampling + Swin-B	0.787	0.792	0.797	0.809
Regular Sampling + SSNet	0.788	0.798	0.803	0.812
TBS + Swin-B	0.790	0.797	0.807	0.815
TBS + SSNet	0.793	0.800	0.819	0.828

- The performance of the SSNet is improved considerably compared to Swin-B, which indicates that the designed SSNet is more in line with the processing of visual information by the cognitive system.
- Together with the TBS module, all models have been given gain against regular sampling. The experimental results verify TBS effectively aggregates distortion information.



04

Conclusion

Conclusion

- We propose an ATSViT based VQA model for UGC videos.
- This model extracts and processes high-level semantic features in SSNet leveraging the reconstructed sequences with a dense distribution of distorted information in TBS, then jointly learns the feature aggregation for video quality prediction.
- Experimental results show the superior performance of the proposed VQA model compared to both conventional as well as deep learning video quality models.



THANKS!