

# Deep-BVQM: A Deep-learning Bitstream-based Video Quality Model



*Nasim Jamshidi Avanaki<sup>1</sup>, Steven Schmidt<sup>2</sup>, Thilo Michael<sup>1</sup>, Saman Zadtootaghaj<sup>3</sup>, Sebastian Moller<sup>1</sup>*

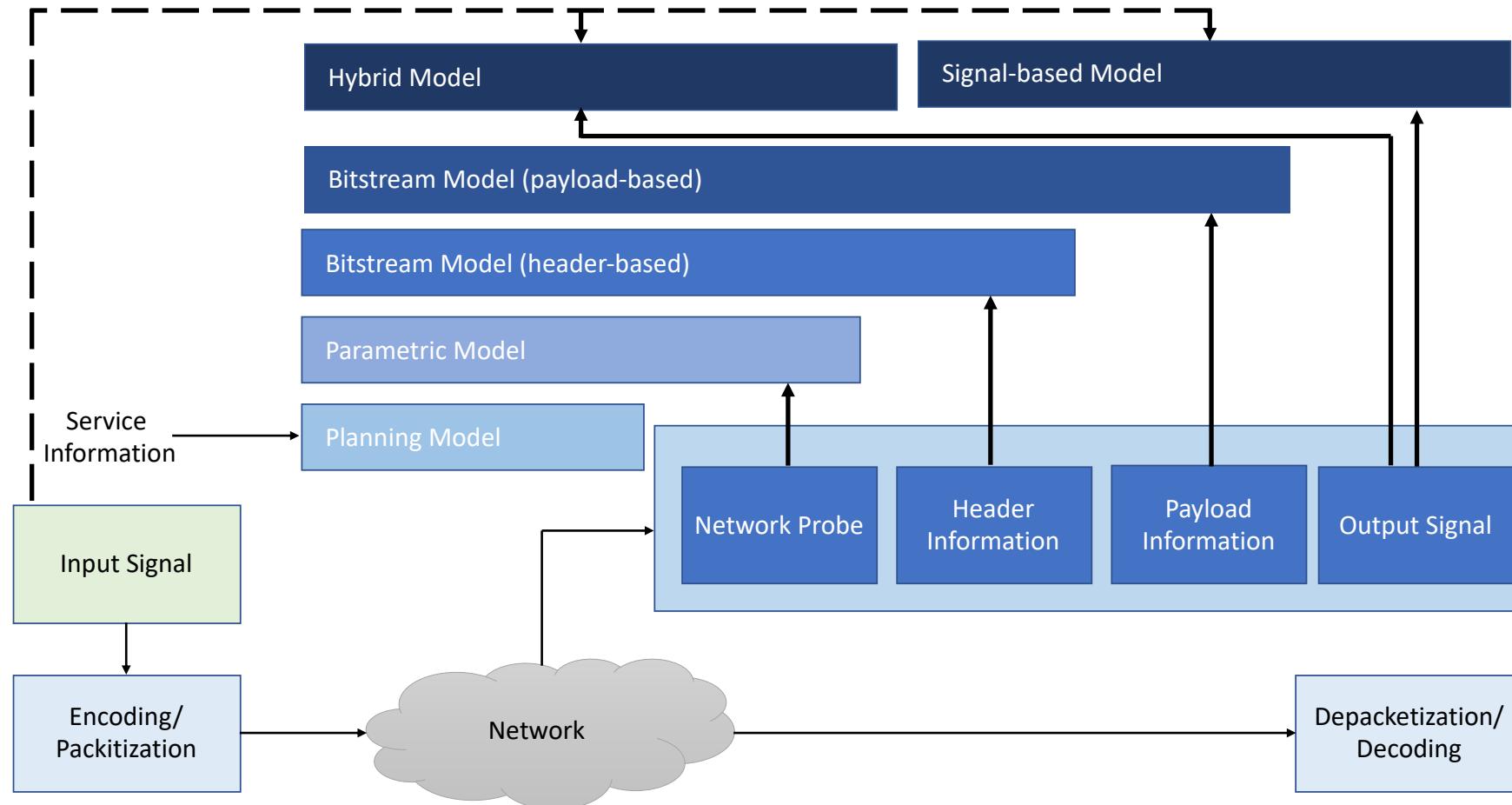
<sup>1</sup> Quality and Usability Lab, Technische Universität Berlin

<sup>2</sup> Sony Interactive Entertainment

<sup>3</sup> Dolby Laboratories

**2022**

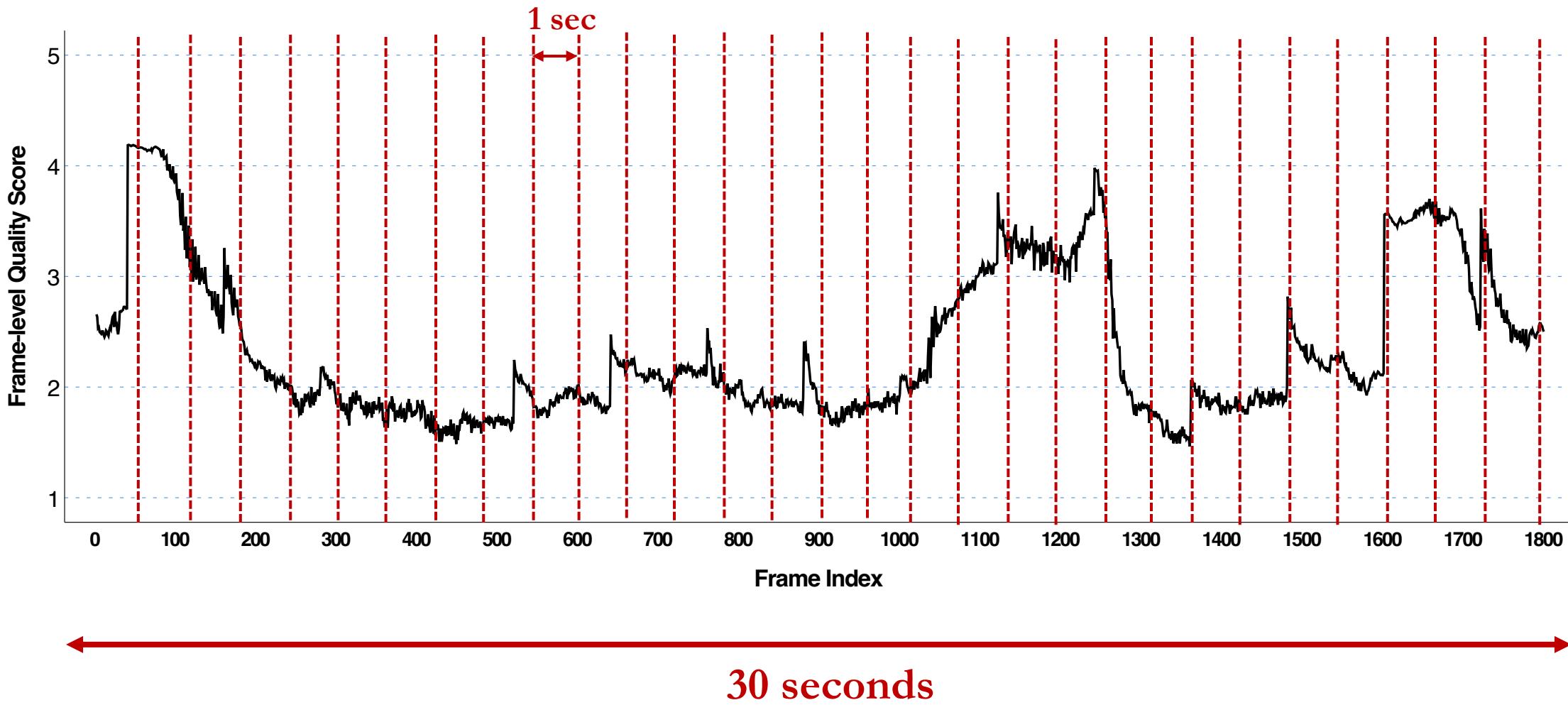
# Motivation



## *Motivation*

- Bitstream-based models are lightweight, accurate and good choice for monitoring models.
- Each bitstream model takes a few years to be developed.
  - Require collecting massive subjective tests.
  - Any codec updates requires development of a new model.
- They predict the quality at minimum 1 second not at frame-level.

# *Quality variation in gaming video*

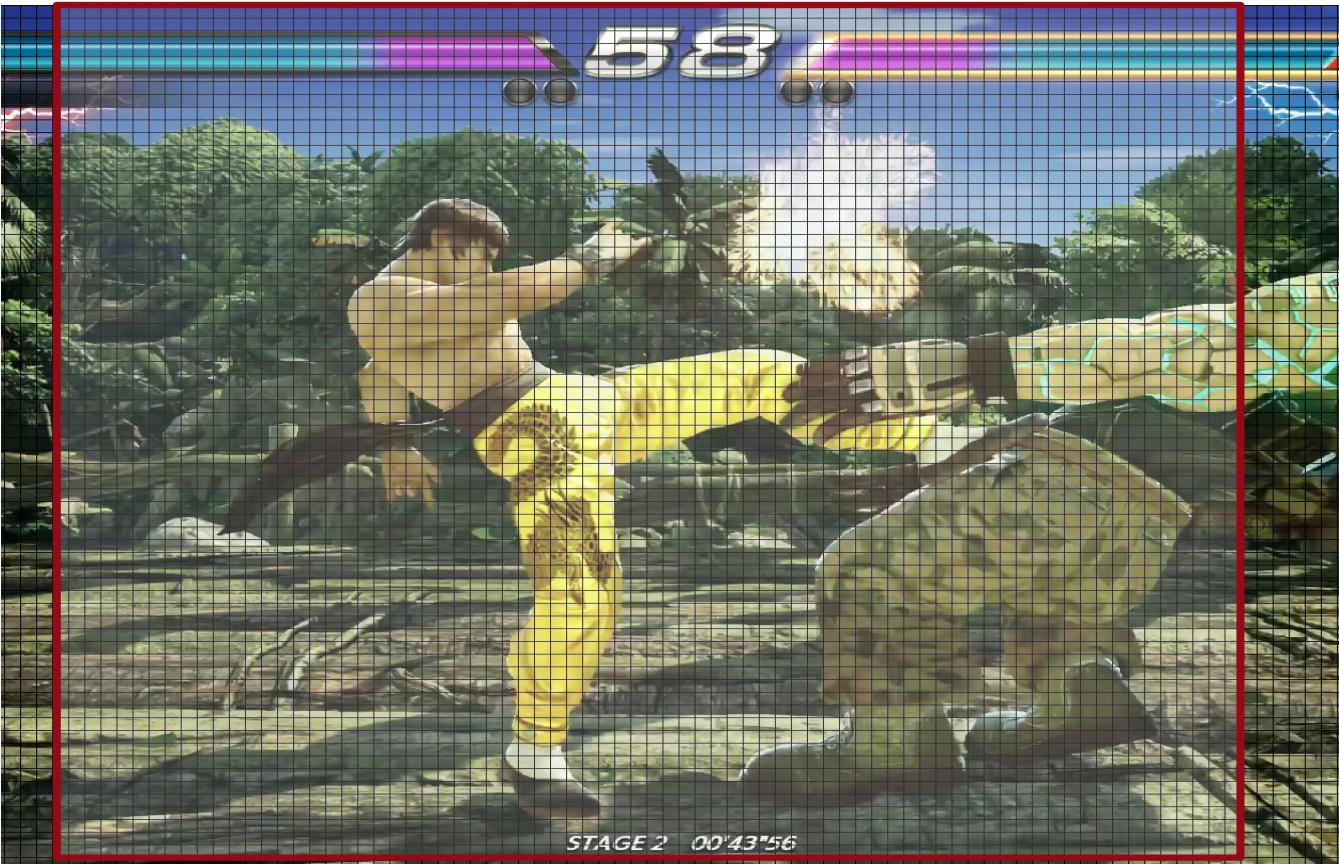




**Deep-BVQM has been developed  
at two levels:**

- **Frame Level**
  - Using CNN
- **Video Level**
  - Using LSTM

## *Deep-BVQM (frame-level)*



- Take 89 x 89 patches from the QP values.
- Train a CNN model
- Use VMAF classes for training

[0-20]	[0-17]
[20-40]	[23-37]
[40-60]	[43-57]
[60-80]	[63-77]
[80 -100]	[83 -100]



# *Deep-BVQM (frame-level) (incl. CNN architecture, dataset)*

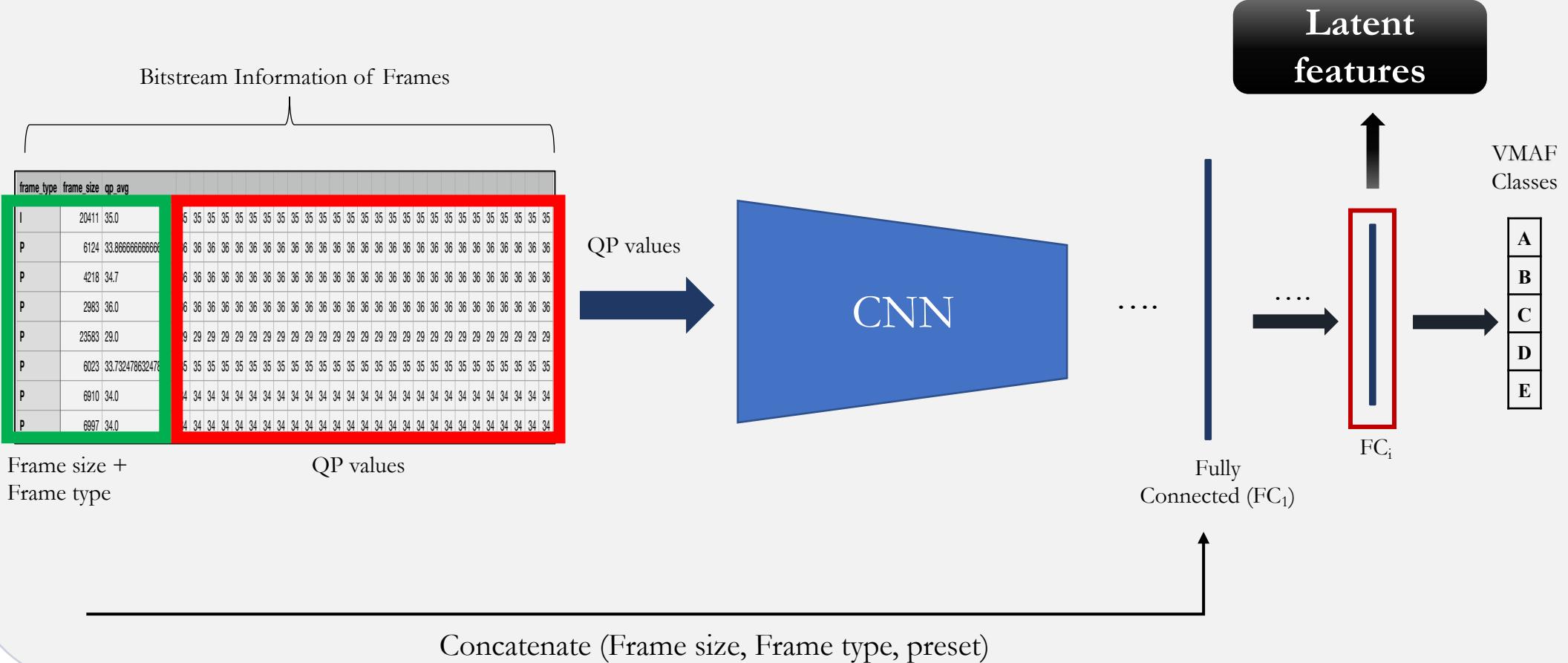
## Training Dataset used for frame-level module

Parameters	Quantity	Values
Source videos	38 SRC	CGVDS + GVSET
Rate controller	1	CBR
Preset	3	Slow, veryfast, Llhq
Bitrate	9	300 kbps to 10 mbps
Codec	1	H.264
Resolution	1	1080p
Frame rate	1	30, 60 fps
$38 \times 1 \times 3 \times 9 \times 1 \times 1 \times 1 = 1026$ video files $\approx 300$ k frames (training)		

## CNN Architecture

Layer	Output Size
Input	1x89x89
Pool	1x44x44
Conv1	16x44x44
Pool	16x22x22
Conv2	32x22x22
Conv3	64x22x22
Pool/Dropout (20%)	64x11x11
Conv4	64x11x11
Fc1	7744
Fc2	1000
Fc3	5

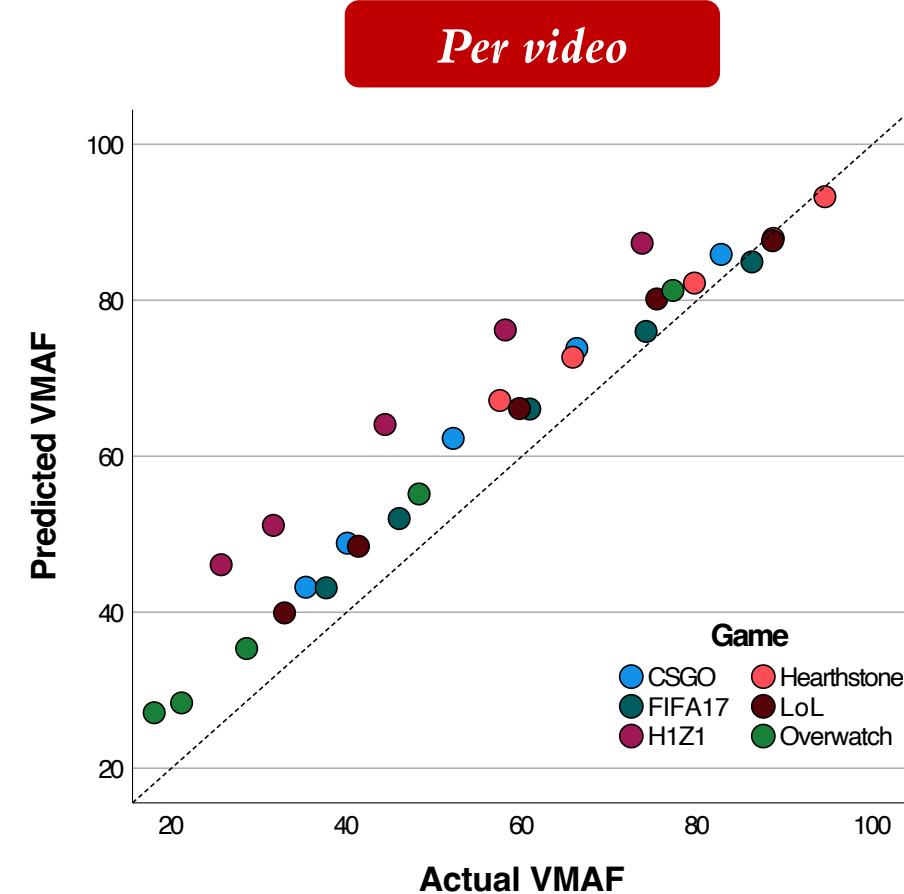
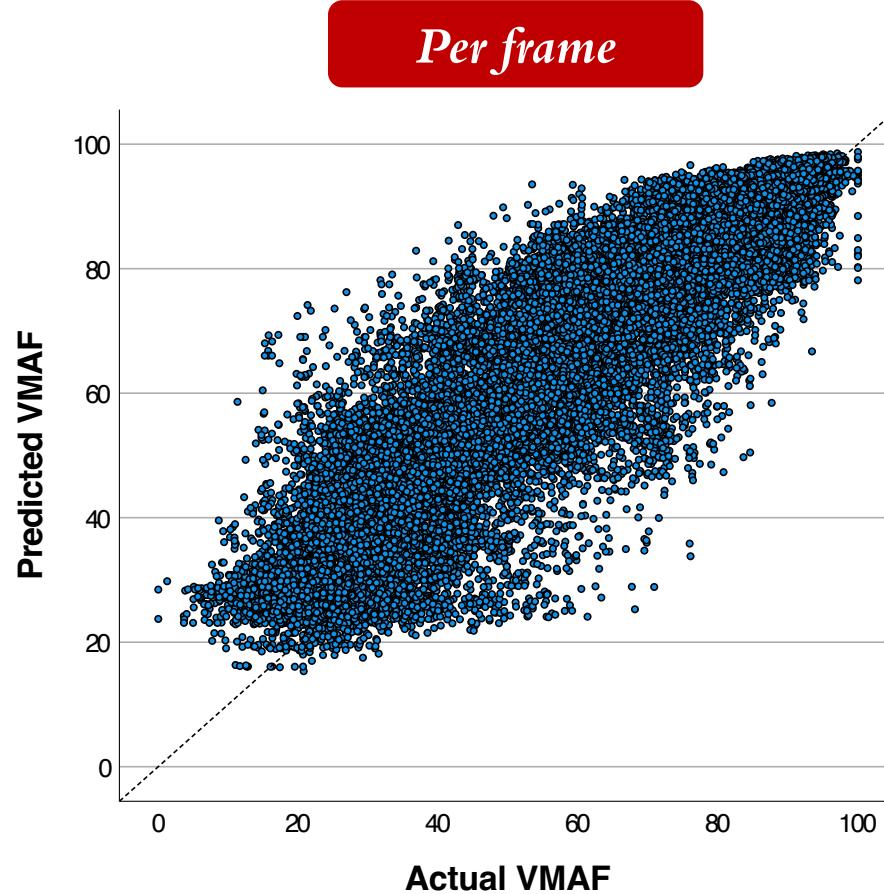
# Deep-BVQM (frame-level)



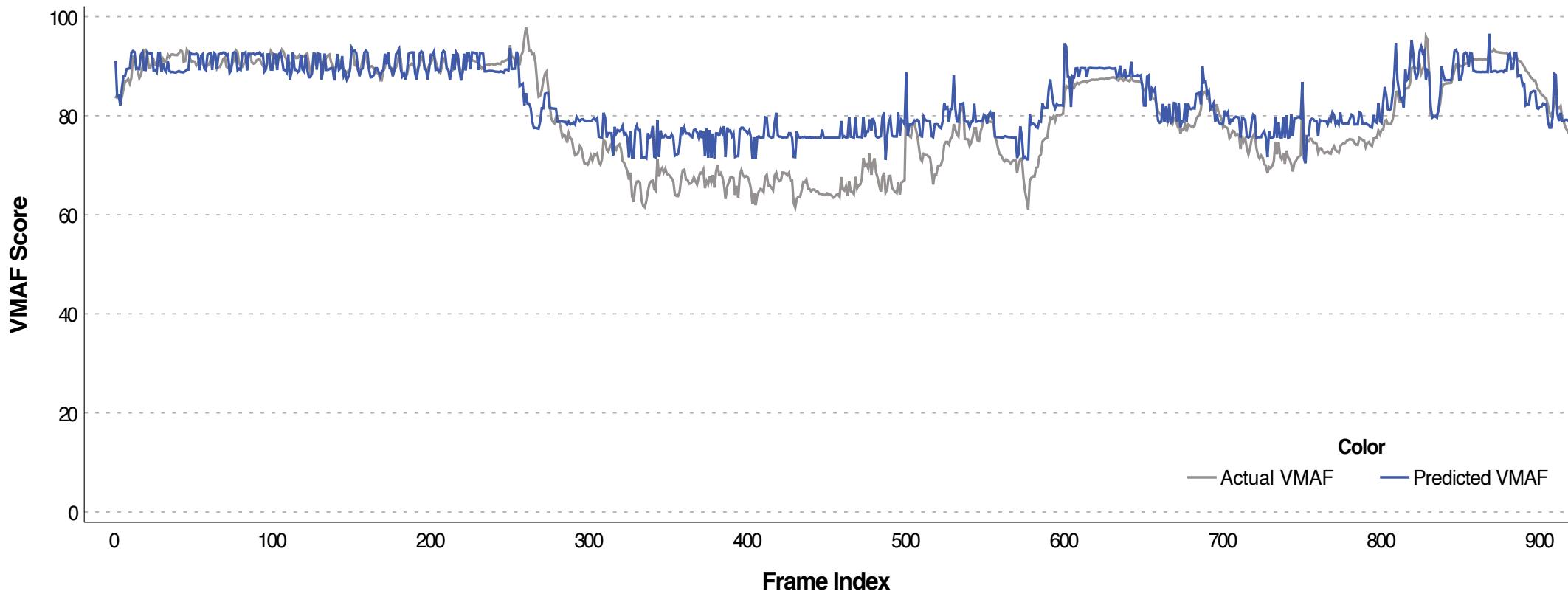
Scheme of the model design at frame level

## *Results on validation set – frame level (1)*

- Use a Random Forest model to see how well the latent features are trained.



## *Results on validation set – frame level (2)*



Comparison of predicted VMAF, blue line, vs actual VMAF scores, grey line, over 900 frames.

# Deep-BVQM (video-level)

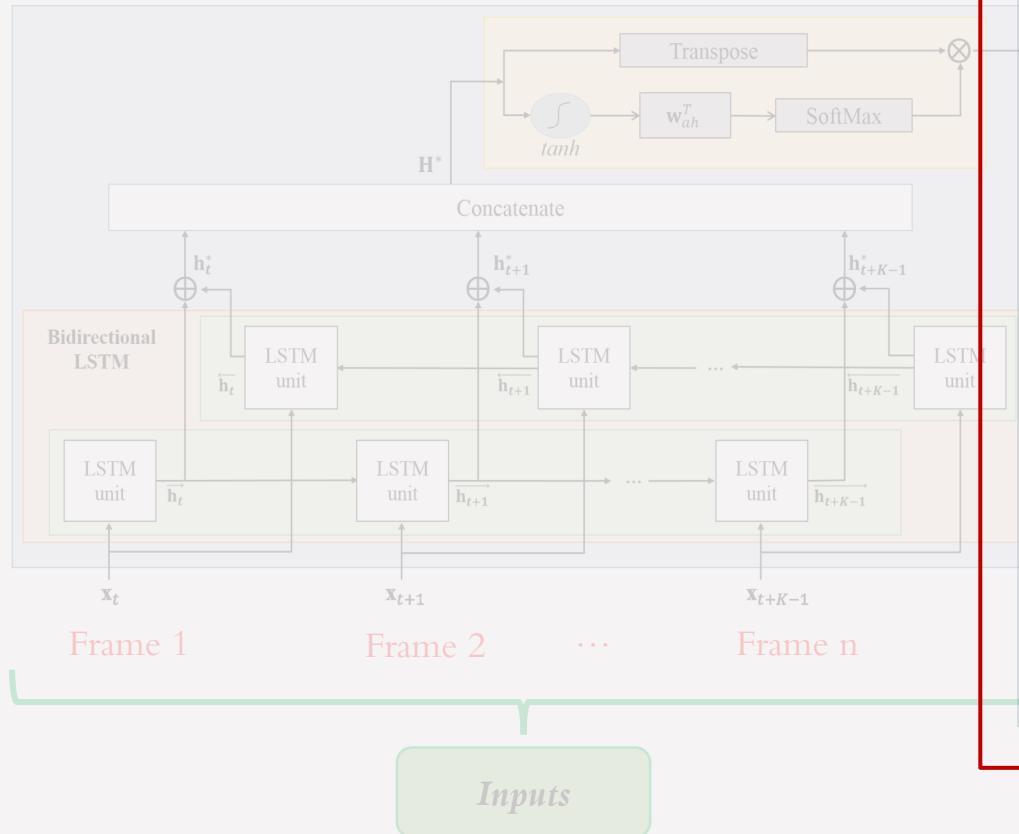
## Video quality prediction using LSTM

Attention

Concatenate

Backward layer

Forward layer



## Properties of LSTM network:

*Predicted Quality*

- Input window: 15 frame vectors taken from 1 sec duration
- Each frame vector: 8 features [5 Latent features, Frame type, Frame size, Preset]
- Training based on a small dataset of 900 input windows ( $\approx 13'500$  frame vectors) and validated on unknown datasets to the model.
- Ground truth: MOS values

# Deep-BVQM (video-level)(incl. LSTM architecture)

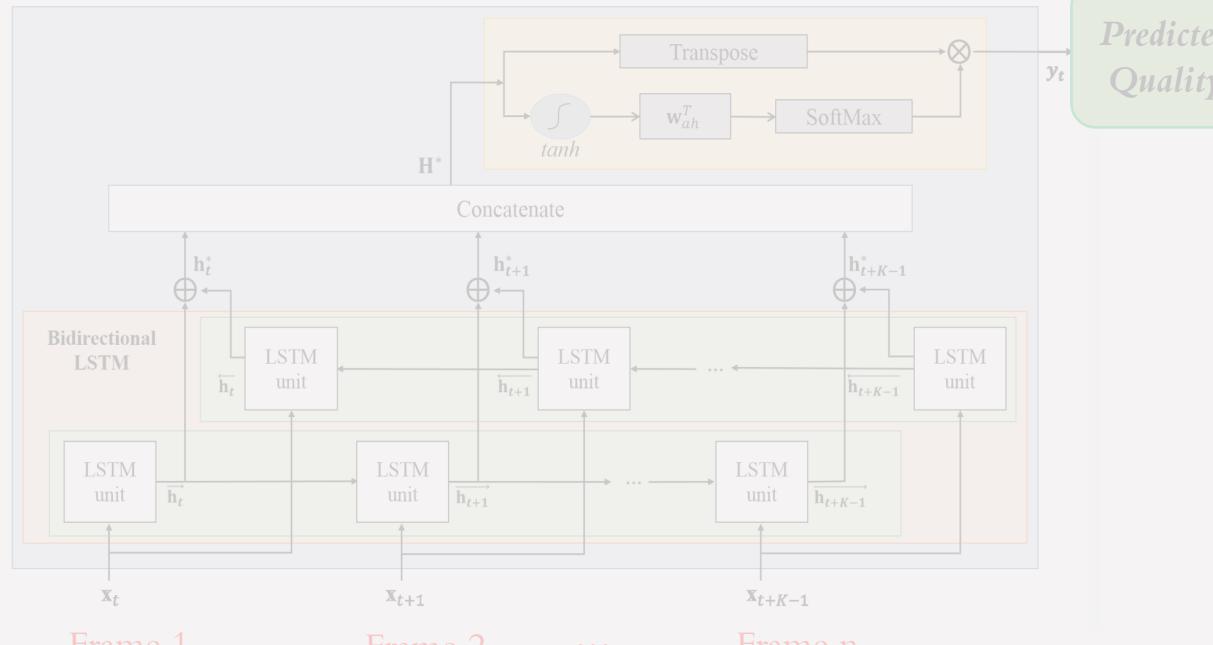
## Video quality prediction using LSTM

Attention

Concatenate

Backward layer

Forward layer

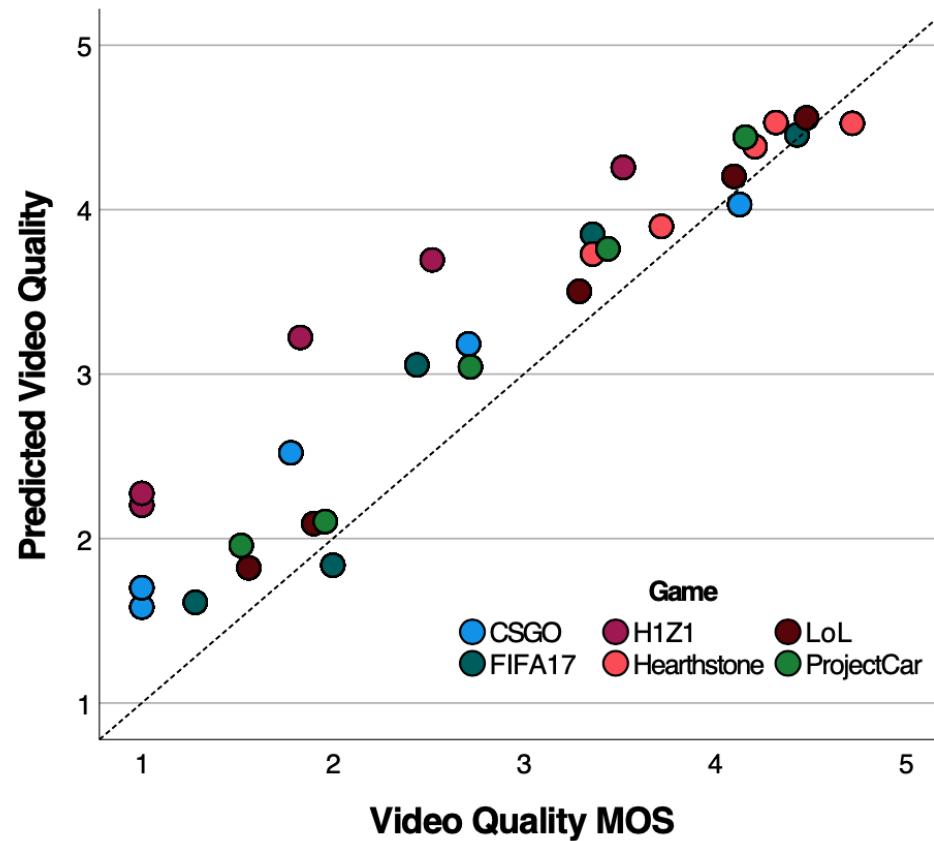


Inputs

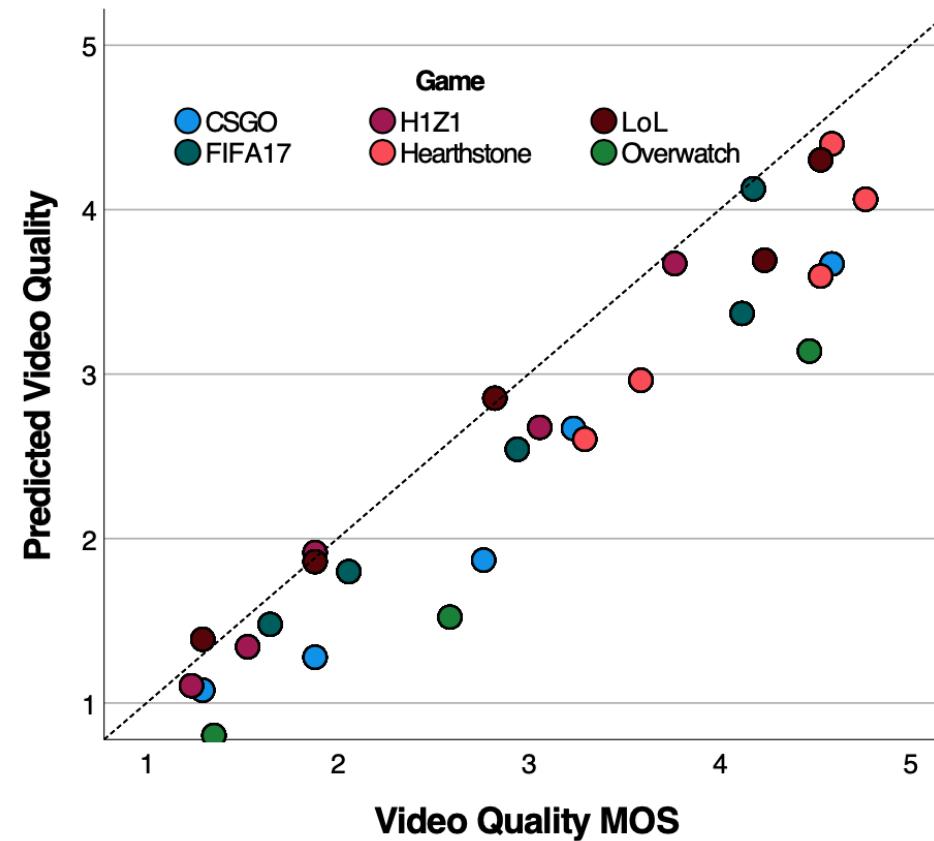
## LSTM Architecture

Layer	Output Shape	Parameters
Input	1 x 15 x 8	0
LSTM <sub>1</sub>	(None, 15, 64)	18688
LSTM <sub>2</sub>	(None, 15, 32)	12416
LSTM <sub>3</sub>	(None, 15, 32)	8320
LSTM <sub>4</sub>	(None, 16)	3136
Dropout	(None, 16)	0
Dense	(None, 17)	17
Total trainable params: 42,577		

# *Performance on two gaming datasets - video level*



GVSET Dataset



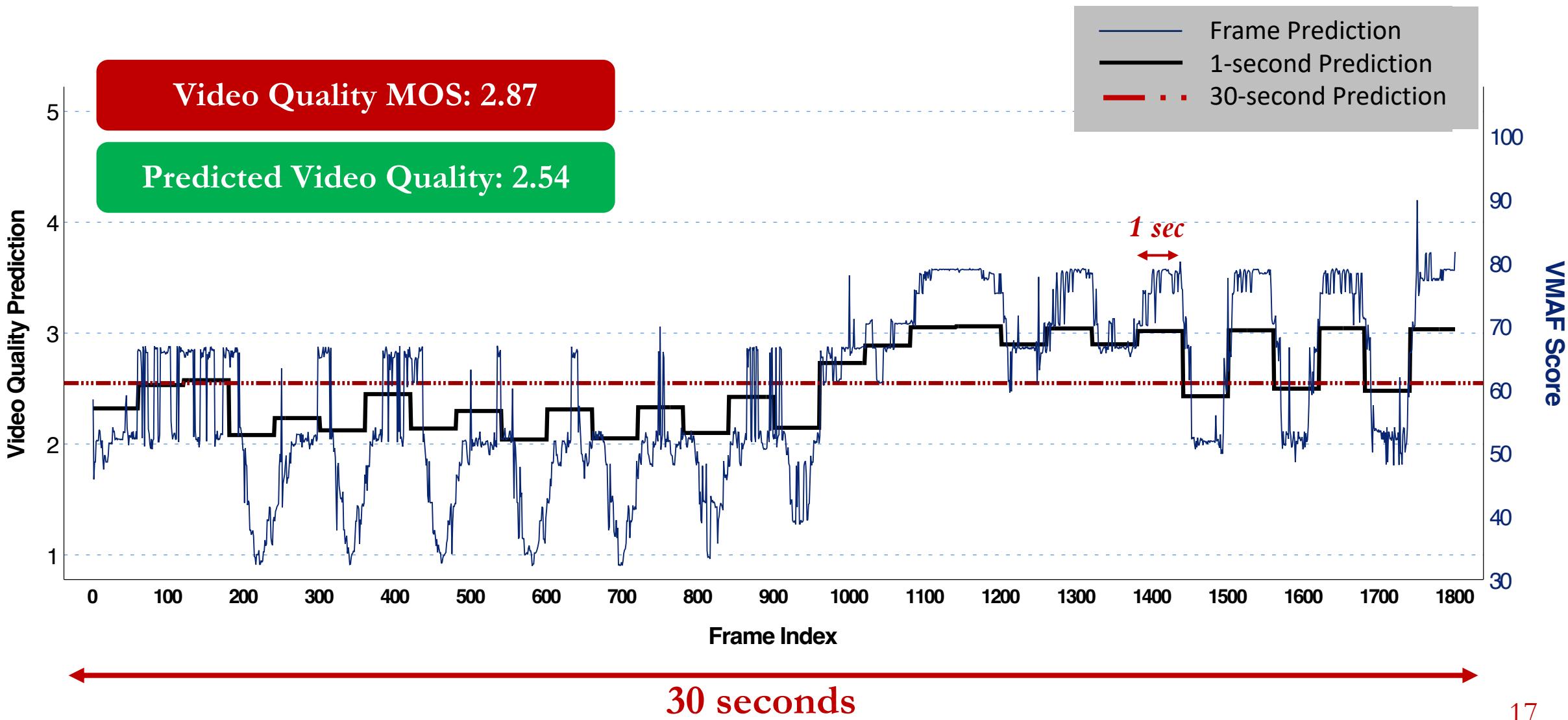
KUGVD Dataset

## *Performance of bitstream-based models*

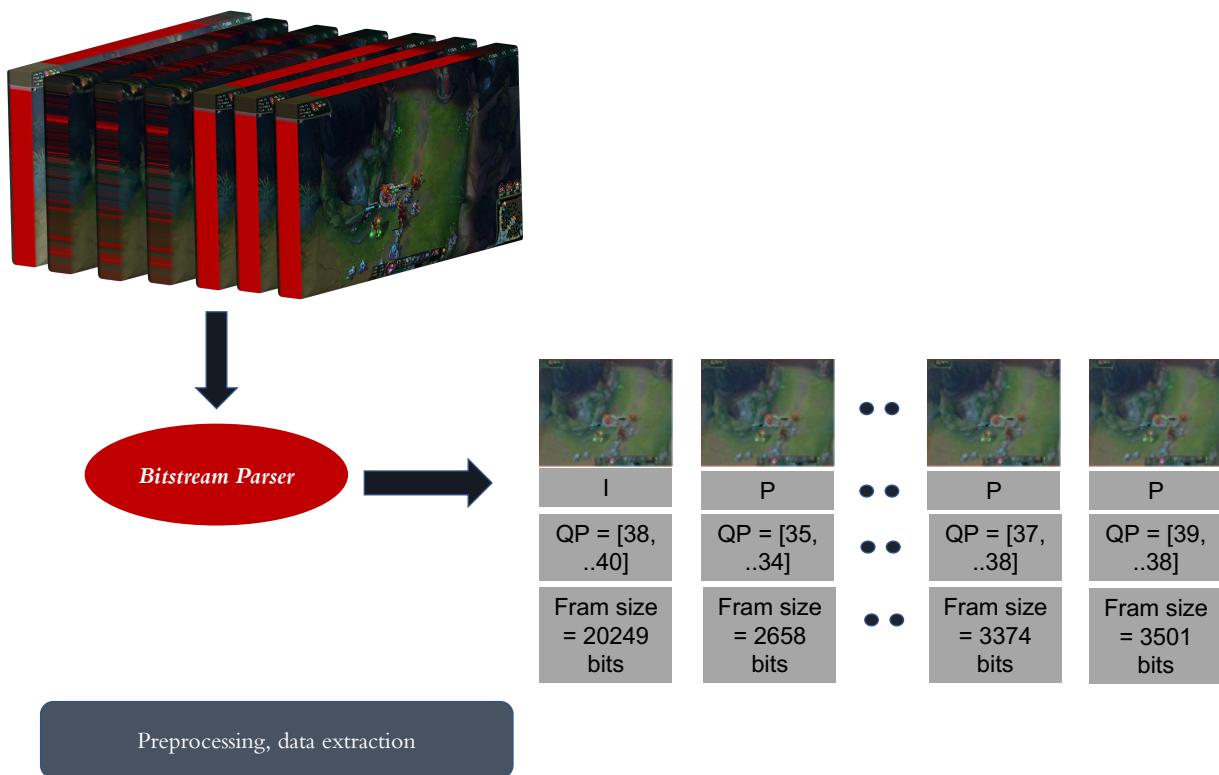
<i>Dataset</i>		<i>P.1203 m1</i>	<i>P.1203 m3</i>	<i>P.1204.3</i>	<i>BQGV</i>	<i>Proposed Model</i>
GVSET	PLCC	0.67	0.91	0.92	0.68	<b>0.95</b>
	RMSE	1.53	0.86	0.53	0.98	<b>0.36</b>
KUGVD	PLCC	0.65	0.92	0.96	0.74	<b>0.96</b>
	RMSE	1.45	0.59	0.41	0.87	<b>0.32</b>
CGVDS	PLCC	0.80	0.86	0.83	0.88	<b>0.96</b>
	RMSE	0.70	0.49	0.52	0.40	<b>0.24</b>

Performance of bitstream-based models on all three gaming dataset, only at 1080p resolution.

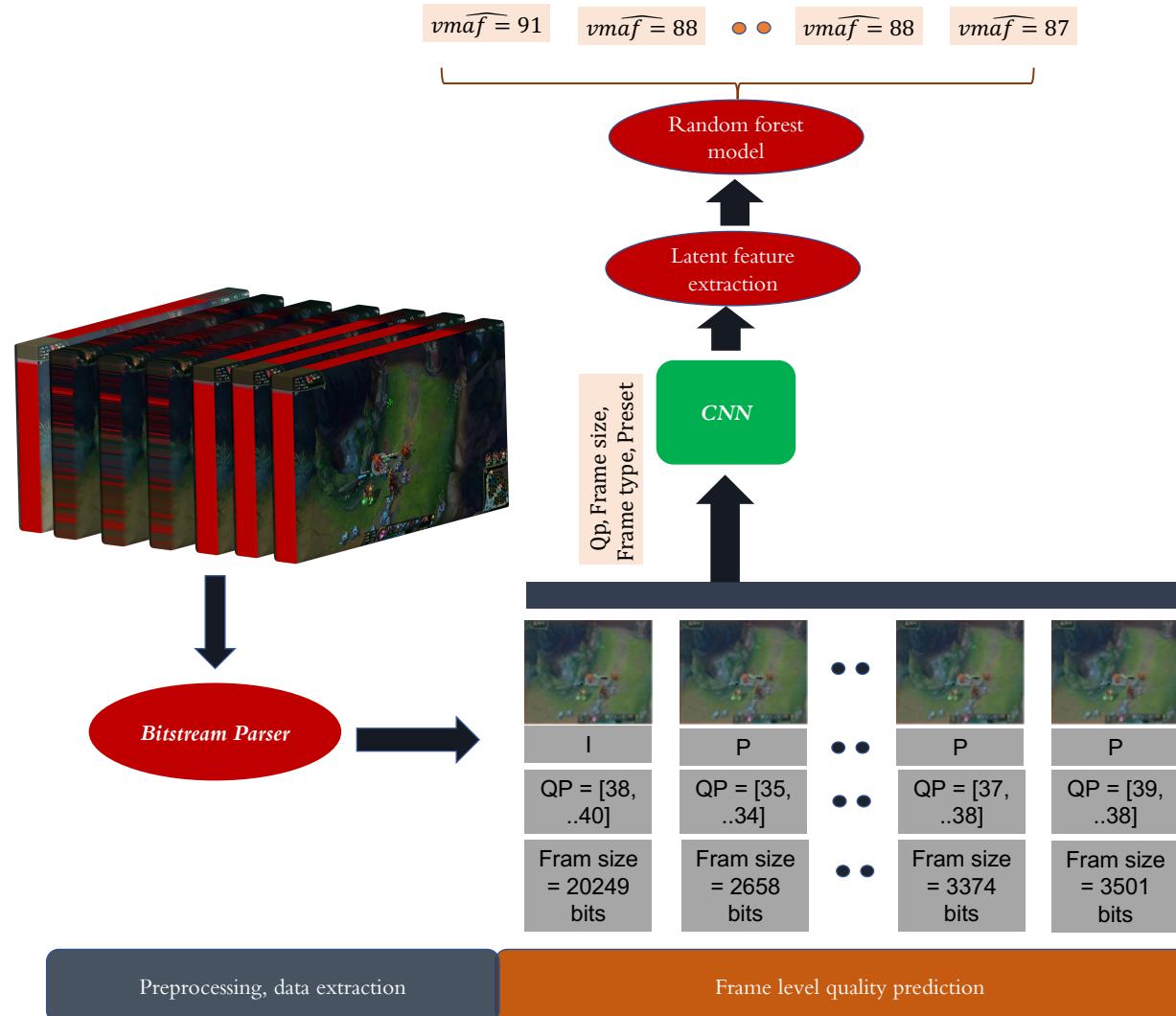
# *Model Prediction*



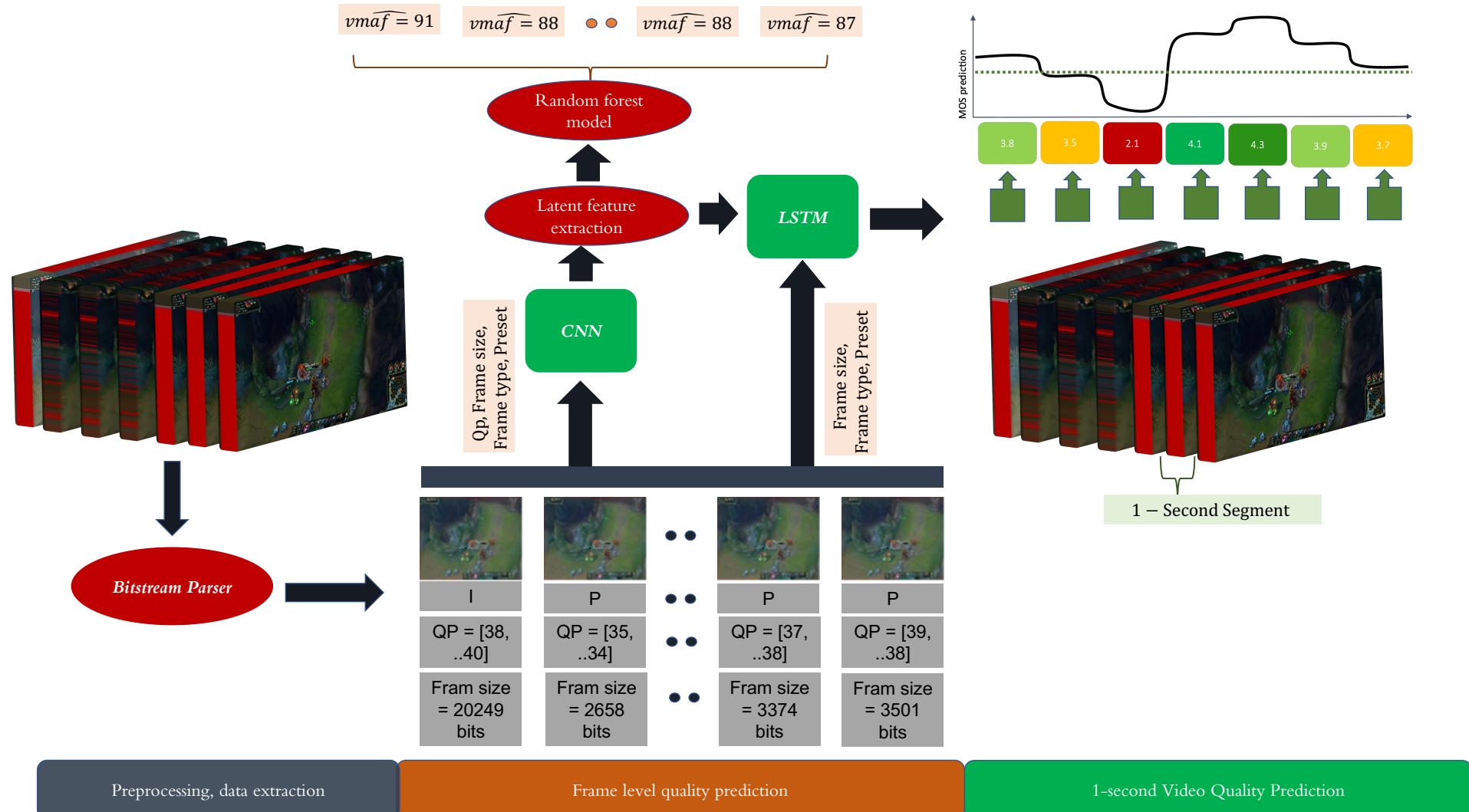
# *Video Quality Prediction Pipeline (Pre-processing, data extraction)*



# *Video Quality Prediction Pipeline (at frame level)*



# Video Quality Prediction Pipeline (in 1-sec duration)



## *Discussion*

- *Future Extension*

- Extend the codec type → select a constant block size.
- Extend the resolution ranges  development of multiple models for different resolutions.  
rescaling the input to  $89 \times 89$  that is used for 1080p resolution.  
upsampling 720p → PLCC = 0.92 , RMSE = 0.49

- *Capability to predict non-gaming content*

- Used a dataset with 2160p (4k)resolution and different encoding setting  
→ PLCC = 0.74, RMSE = 0.67

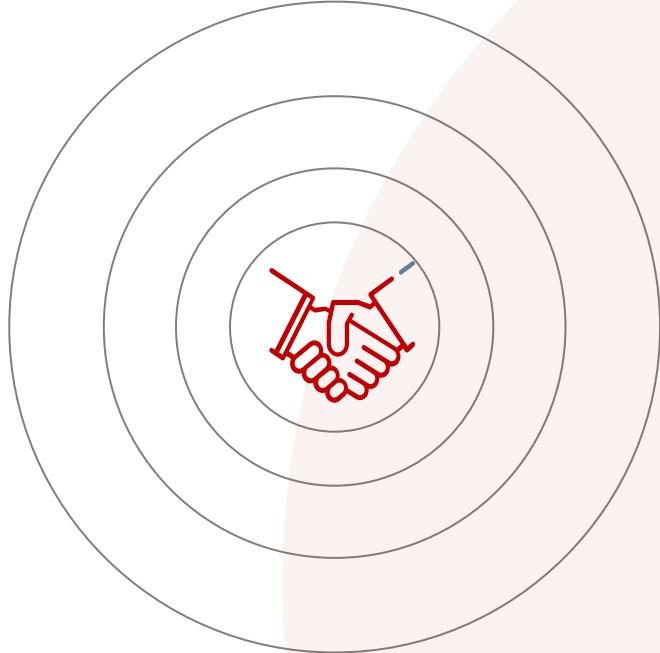
- *Computational Complexity*

- Less than 1 second on GPU

# *Conclusion*

**Deep-BVQM**  
is a bitstream-based model  
which...

- is capable to predict at the frame level.
- is developed for gaming encoding setting.
- simplifies the model extension and expansion.
- is a lightweight prediction method.



# Thanks!

## Any questions?

You can find me at:

[n.jamshidiavanaki@tu-berlin.de](mailto:n.jamshidiavanaki@tu-berlin.de)

[nasim.jamshidi.avanaki@hhi.fraunhofer.de](mailto:nasim.jamshidi.avanaki@hhi.fraunhofer.de)