

# **[AHG11] Autoencoder-based intra prediction with auxiliary feature**

**JVET-X0125**

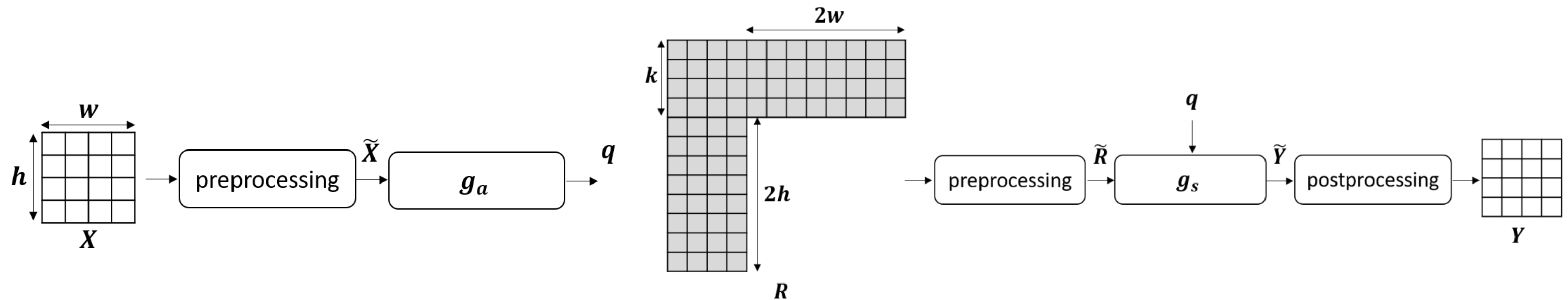
**Luhang Xu, Yue Yu, Haoping Yu, Kazushi Sato, Zhenyu Dai, Zhihuang Xie, Dong Wang**

# Introduction

- Autoencoder-based intra prediction mode (also called autoencoder mode) is proposed as an additional intra mode;
- The proposed autoencoder mode generates a prediction signal by using the auxiliary feature information and neighboring reconstructed samples of current CU;
- The residual generated by autoencoder mode is transformed by LFNST kernel of planar mode when LFNST mode is selected.

## Architecture of autoencoder (1/2)

- The autoencoder consists of a pair of jointly trained encoder network  $g_a(\cdot, \phi_{w,h})$  and decoder network  $g_s(\cdot, \theta_{w,h})$ ;



**Encoder network extracts auxiliary feature  $q$  from a  $w \times h$  original block  $X$**

**Decoder network predicts block  $Y$  by using reference samples  $R$  and auxiliary feature  $q$ ;  $k$ , number of reference lines, is set to 4 for block size  $\min(w, h) < 8$ , and 8 for other block size.**

# Architecture of autoencoder (2/2)

- Encoder network  $g_a(\cdot, \phi_{w,h})$  and decoder network  $g_s(\cdot, \theta_{w,h})$  are all fully connected neural networks

layer index	input	layer type	number of neurons	activation
0	$\tilde{X}$	Fully connection	512	Leaky ReLU
1	0	Fully connection	512	Leaky ReLU
2	1	Fully connection	2	Sigmoid

Architecture of the encoder network,  
 $\tilde{X}$  is preprocessed version of original block  $X$

layer index	input	layer type	number of neurons	activation
0	$\tilde{R}, q$	Fully connection	512	Leaky ReLU
1	0	Fully connection	512	Leaky ReLU
2	1	Fully connection	$w \times h$	None

Architecture of the decoder network,  
 $\tilde{R}$  is preprocessed version of reference samples  $R$

# Preprocessing and postprocessing

- Preprocessing for input of encoder network

- Step-1. Mean value subtraction for the current original block  $\mathbf{X}$ .
- Step-2. The “zero-mean” samples in the current block are divided by  $2^b$ ,  $b$  being the internal bit-depth.
- Step-3. Then the normalized block is flattened into vector  $\tilde{\mathbf{X}}$ .

- Preprocessing for input of decoder network

- Step-1. The reference samples  $\mathbf{R}$  of the current block are divided by  $2^b$ ,  $b$  being the internal bit-depth
- Step-2. Mean value subtraction for the available reference samples in  $\mathbf{R}$ , same padding strategy for unavailable reference samples in VVC

- Postprocessing for output of decoder network

- The postprocessing in the decoder network de-normalizes  $\tilde{\mathbf{Y}}$  back to internal bit-depth. The whole process can be described in the following equation,

$$\mathbf{Y} = \min(\max(2^b \times (\text{reshape}(\tilde{\mathbf{Y}}) + \mu), 0), 2^b)$$

# Integration into VTM

- Decoding process in VTM decoder
  - Parsing auxiliary feature  $\mathbf{q}$  from bitstream if autoencoder mode is selected
  - Preprocessing the neighboring reference samples to obtain  $\tilde{\mathbf{R}}$
  - Feeding  $\tilde{\mathbf{R}}$  and feature  $\mathbf{q}$  into the decoder network
  - Postprocessing the output of decoder network.
- Selection of prediction mode in VTM encoder
  - All the possible combinations of feature  $\mathbf{q}$  and their corresponding autoencoder modes are evaluated and compared, using rate distortion (RD) full search, with other conventional intra prediction modes. The mode with the least RD cost is selected for the current CU.



# Training and inference information

Network Information in Training Stage		
Mandatory	HW environment:	CPU: Intel(R) Xeon(R) Gold 6142 CPU @ 2.60GHz GPU: Tesla V100 (32GB)
	SW environment:	OS: Ubuntu 16.04.4
	Framework:	Pytorch v1.6.0
	Epoch:	300
	Batch size:	20Kx100
	Training time:	
	Training data information:	Training images that converted into YCbCr in DIV2K First frame of each sequence in BVI-DVC
	Configurations for generating compressed training data (if different to VTM CTC):	VTM-13.0, all intra configuration and with qp {22, 27, 32, 37, 42}
	Patch size	$g_a(\cdot, \phi_{w,h})$ and $g_s(\cdot, \theta_{w,h})$ are trained on pairs of a $w \times h$ block and its reference samples.
	Learning rate:	1e-4
	Optimizer:	Adam
	Loss function:	L1
	Preprocessing:	See 1.3.1 and 1.3.2
	Other information:	

Network Information in Inference Stage		
Mandatory	HW environment:	
	SW environment:	CPU: Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz GPU: None
	Framework:	OS: Ubuntu 16.04.4 Libtorch v1.9.0
	Total Conv. Layers	None
	Total FC Layers	3 for encoder network, 3 for decoder network.
	Total Parameter Number	Decoder network { $\theta_{4,4}$ : 313360, $\theta_{8,4}$ : 337952, $\theta_{8,8}$ : 460864, $\theta_{16,4}$ : 387136, $\theta_{16,8}$ : 559232, $\theta_{16,16}$ : 690432, $\theta_{32,4}$ : 485514, $\theta_{32,16}$ : 952832, $\theta_{32,32}$ : 1346560 } Encoder network (encoder network does not perform at VTM decoder side) { $\phi_{4,4}$ : 272386, $\phi_{8,4}$ : 280578, $\phi_{8,8}$ : 296962, $\phi_{16,4}$ : 29696, $\phi_{16,8}$ : 329730, $\phi_{16,16}$ : 395266, $\phi_{32,4}$ : 329730, $\phi_{32,16}$ : 526388, $\phi_{32,32}$ : 788482 }
	Parameter Precision	float32
	Memory Parameter (MB)	34.5MB
	Memory Temp (MB)	
	Kmac/pxl	Decoder network { $\theta_{4,4}$ : 19.38Kmac, $\theta_{8,4}$ : 10.63Kmac, $\theta_{8,8}$ : 7.19Kmac, $\theta_{16,4}$ : 6.09Kmac, $\theta_{16,8}$ : 4.38Kmac, $\theta_{16,16}$ : 2.70Kmac, $\theta_{32,4}$ : 3.83Kmac, $\theta_{32,16}$ : 1.86Kmac, $\theta_{32,32}$ : 1.32Kmac } Encoder network (encoder network does not perform at VTM decoder side) { $\phi_{4,4}$ : 16.88Kmac, $\phi_{8,4}$ : 8.75Kmac, $\phi_{8,8}$ : 4.69Kmac, $\phi_{16,4}$ : 4.69Kmac, $\phi_{16,8}$ : 2.58Kmac, $\phi_{16,16}$ : 1.56Kmac, $\phi_{32,4}$ : 2.58Kmac, $\phi_{32,16}$ : 1.04Kmac, $\phi_{32,32}$ : 0.77Kmac }
	Preprocessing:	See 1.3.2
	Other information:	
Optional		

# Simulation results over VTM-11.0

Simulation results for qp {22, 27, 32, 37, 42}

	All Intra Main10											Random access Main10									
	BD-rate Over VTM-11.0											BD-rate Over VTM-10.0									
	Y-PSNR	U-PSNR	V-PSNR	Y-MSIM	U-MSIM	V-MSIM	EncT	DecT	bit DIFF			Y-PSNR	U-PSNR	V-PSNR	Y-MSIM	U-MSIM	V-MSIM	EncT	DecT	bit DIFF	
Class A1	-1.11%	-0.88%	-0.91%	-1.05%	-0.69%	-0.71%	307%	381%	1%		Class A1	-0.56%	-0.42%	-0.68%	-0.52%	-0.37%	-0.62%	130%	177%	0%	
Class A2	-0.83%	-0.73%	-0.70%	-0.74%	-0.59%	-0.61%	332%	445%	1%		Class A2	-0.36%	-0.48%	-0.33%	-0.32%	-0.28%	-0.26%	132%	169%	0%	
Class B	-1.04%	-0.80%	-0.84%	-0.90%	-0.64%	-0.71%	349%	545%	1%		Class B	-0.59%	-0.42%	-0.41%	-0.50%	-0.34%	-0.15%	138%	238%	1%	
Class C	-1.07%	-0.81%	-0.70%	-1.03%	-0.73%	-0.64%	339%	777%	1%		Class C	-0.55%	-0.63%	-0.62%	-0.48%	-0.69%	-0.56%	148%	480%	0%	
Class E	-1.55%	-1.30%	-1.54%	-1.35%	-0.91%	-1.52%	358%	694%	1%		Class E										
Overall	-1.11%	-0.89%	-0.91%	-1.00%	-0.70%	-0.81%	338%	559%	1%		Overall	-0.53%	-0.49%	-0.50%	-0.46%	-0.43%	-0.38%	138%	253%	0%	
Class D	-1.08%	-0.78%	-0.51%	-0.84%	-0.36%	-0.11%	324%	1509%	1%		Class D	-0.53%	-0.36%	-0.11%	-0.46%	-0.46%	0.38%	149%	969%	0%	
Class F	-0.77%	-0.56%	-0.51%	-0.68%	-0.53%	-0.44%	216%	459%	1%		Class F	-0.39%	-0.26%	-0.37%	-0.36%	-0.23%	0.04%	143%	347%	0%	
Class H	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#DIV/0!	#DIV/0!	#DIV/0!		Class H	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#DIV/0!	#DIV/0!	#DIV/0!	

Simulation results for qp {22, 27, 32, 37}

	All Intra Main10 (QP22,27,32,37)										Random access Main10 (QP22,27,32,37)									
	BD-rate Over										BD-rate Over									
	Y-PSNR	U-PSNR	V-PSNR	Y-MSIM	U-MSIM	V-MSIM	EncT	DecT	Y-PSNR		U-PSNR	V-PSNR	Y-MSIM	U-MSIM	V-MSIM	EncT	DecT			
Class A1	-1.27%	-1.08%	-1.08%	-1.28%	-1.01%	-0.98%	325%	410%		Class A1	-0.67%	-0.58%	-0.81%	-0.72%	-0.72%	-0.81%	133%	187%		
Class A2	-0.96%	-0.89%	-0.85%	-0.98%	-0.84%	-0.83%	340%	478%		Class A2	-0.43%	-0.45%	-0.39%	-0.41%	-0.42%	-0.39%	135%	177%		
Class B	-1.16%	-0.95%	-0.96%	-1.15%	-0.88%	-0.91%	345%	563%		Class B	-0.67%	-0.58%	-0.57%	-0.65%	-0.52%	-0.28%	142%	247%		
Class C	-1.16%	-0.87%	-0.76%	-1.17%	-0.77%	-0.73%	330%	725%		Class C	-0.57%	-0.57%	-0.54%	-0.53%	-0.54%	-0.48%	152%	472%		
Class E	-1.77%	-1.62%	-1.75%	-1.71%	-1.52%	-1.81%	356%	668%		Class E										
Overall	-1.25%	-1.05%	-1.05%	-1.24%	-0.98%	-1.02%	339%	565%		Overall	-0.59%	-0.55%	-0.57%	-0.59%	-0.54%	-0.46%	141%	260%		
Class D	-1.17%	-0.95%	-0.59%	-1.04%	-0.78%	-0.38%	317%	1409%		Class D	-0.57%	-0.32%	-0.22%	-0.38%	-0.53%	-0.23%	151%	913%		
Class F	-0.81%	-0.55%	-0.61%	-0.86%	-0.41%	-0.72%	216%	437%		Class F	-0.41%	-0.43%	-0.38%	-0.28%	-0.58%	-0.17%	144%	345%		
Class H	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#DIV/0!	#DIV/0!		Class H	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#DIV/0!	#DIV/0!		

Simulation results for qp {27, 32, 37, 42}

	All Intra Main10 (QP27,32,37,42)									Random access Main10 (QP27,32,37,42)									
	BD-rate Over									BD-rate Over									
	Y-PSNR	U-PSNR	V-PSNR	Y-MSIM	U-MSIM	V-MSIM	EncT	DecT		Y-PSNR	U-PSNR	V-PSNR	Y-MSIM	U-MSIM	V-MSIM	EncT	DecT		
Class A1	-1.04%	-0.77%	-0.80%	-1.00%	-0.57%	-0.61%	300%	358%		Class A1	-0.50%	-0.33%	-0.63%	-0.47%	-0.27%	-0.59%	126%	166%	
Class A2	-0.68%	-0.65%	-0.62%	-0.69%	-0.52%	-0.56%	337%	449%		Class A2	-0.33%	-0.47%	-0.31%	-0.30%	-0.27%	-0.26%	128%	164%	
Class B	-0.99%	-0.63%	-0.72%	-0.86%	-0.54%	-0.63%	377%	619%		Class B	-0.55%	-0.33%	-0.22%	-0.48%	-0.27%	-0.07%	134%	236%	
Class C	-1.03%	-0.67%	-0.67%	-1.01%	-0.66%	-0.62%	350%	813%		Class C	-0.52%	-0.67%	-0.69%	-0.46%	-0.73%	-0.59%	144%	496%	
Class E	-1.44%	-1.14%	-1.52%	-1.30%	-0.78%	-1.50%	363%	722%		Class E									
Overall	-1.03%	-0.75%	-0.84%	-0.96%	-0.61%	-0.76%	344%	560%		Overall	-0.49%	-0.45%	-0.45%	-0.44%	-0.39%	-0.35%	134%	250%	
Class D	-1.01%	-0.56%	-0.34%	-0.82%	-0.24%	0.04%	334%	1577%		Class D	-0.53%	-0.45%	-0.17%	-0.45%	-0.54%	0.38%	145%	1019%	
Class F	-0.76%	-0.49%	-0.47%	-0.65%	-0.50%	-0.47%	217%	468%		Class F	-0.38%	-0.34%	-0.41%	-0.37%	-0.23%	0.08%	141%	351%	
Class H	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	377%	695%		Class H	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#DIV/0!	351%	



# Conclusion

- This contribution presents an autoencoder-based intra prediction mode;
- The proposed intra prediction mode reportedly shows coding gains over VTM-11.0;
- It is proposed to further study this topic in the Ad-Hoc group.

Thank you

oppo