HYBRID INTRA-PREDICTION IN LOSSLESS VIDEO CODING USING OVERFITTED NEURAL NETWORKS

Victor Sanchez[1], Miguel Hernández-Cabronero[2], and Joan Serra-Sagristà[2]

[1]Department of Computer Science, University of Warwick, United Kingdom

[2]Department of Information & Communications Engineering, Universitat Autònoma de Barcelona, Spain

V.F.Sanchez-Silva@warwick.ac.uk

ML-based Intra-prediction

We propose six machine learning (ML)-based intra-prediction modes to increase the granularity of intra-prediction in modern video codecs for lossless compression.

Each mode is based on a 1-layer overfitted fully-connected neural network (FC-NN – see Fig. 1) and predicts a block in column-wise or row-wise manner (see Table 1 and Fig. 2).

✓ No need for an offline training process.
✓ No need to signal any learned parameters to the decoder.

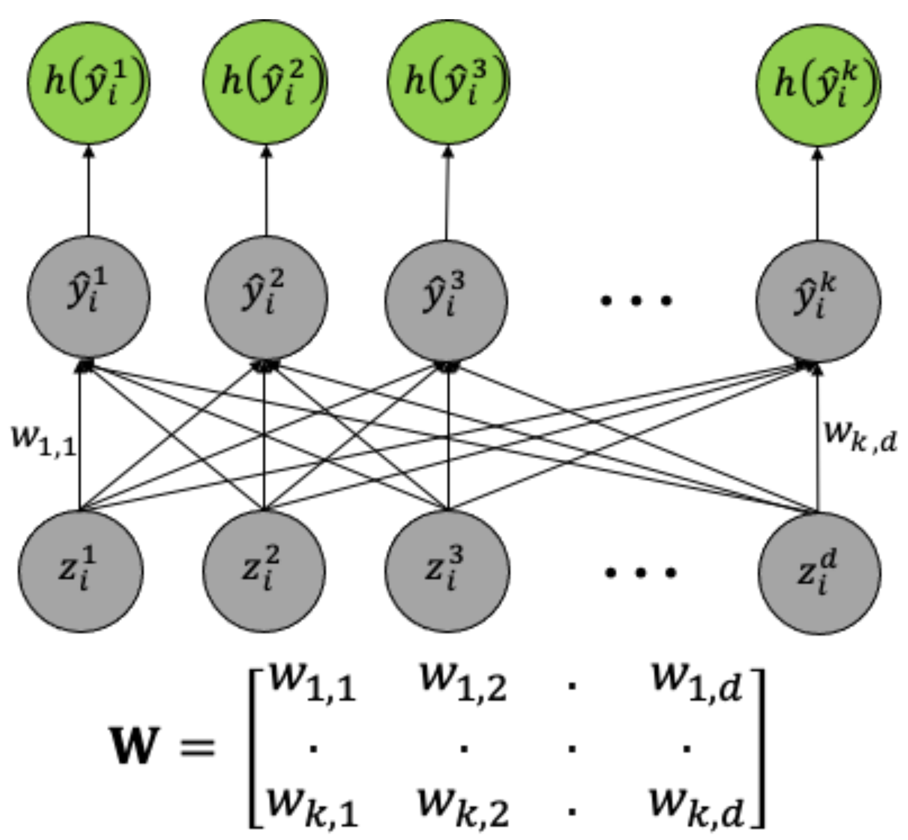

Fig. 1: A d-dimensional feature vector (FV), $\mathbf{z}_i$, is used to predict a k-dimensional vector, $\mathbf{p}_i = h(\hat{\mathbf{y}}_i)$, where $h()$ is the Sigmoid activation function. Matrix $\mathbf{W}$ contains the weights to compute vector $\hat{\mathbf{y}}_i$ from $\mathbf{z}_i$.

Table 1: Proposed ML-based modes that emulate several directions.

Mode	Prediction	Flip applied to block before prediction	Direction emulated
H	column-wise	none	horizontal - left to right
V	row-wise	none	vertical - top to bottom
H2	column-wise	left-right	horizontal - right to left
V2	row-wise	up-down	vertical - bottom to top
D	column-wise	up-down first and then left-right	diagonal direction
AD	row-wise	up-down first and then left-right	anti-diagonal

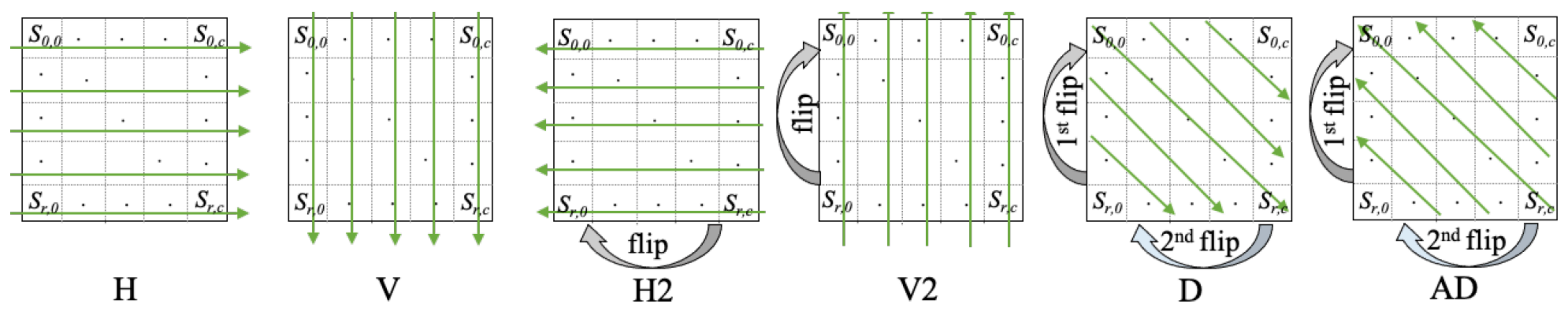

Fig. 2: Modes H2 and V2 require flipping the block before prediction. Modes D and AD require flipping the block twice before prediction. $S_{r,c}$ is the pixel location at row r and column c of the current block.

The feature vector (FV) for each FC-NN is computed by averaging $k/2$ subsets of three reference samples (see Fig. 3).

Each FC-NN is allowed to overfit on the current frame by predicting all blocks (see Fig. 4). Matrix $\mathbf{W}$ is initialized to zeros at the encoder and decoder, which allows to replicate the optimization process at the decoder. The loss function used is:

$$\mathcal{L}_i = \| \mathbf{p}_i - \mathbf{g}_i \|^2$$

i^{th} predicted column (or row) — i^{th} original column (or row)

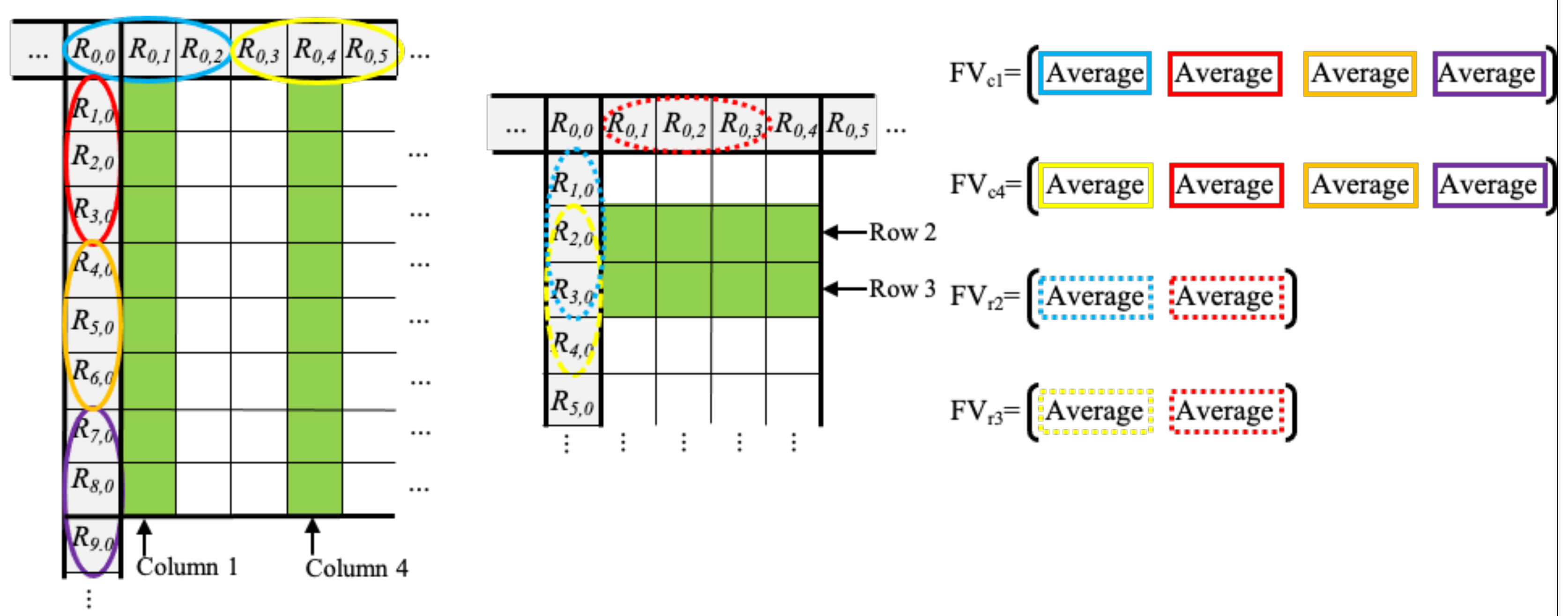

Fig. 3: Example FVs to predict (left) columns of size $k = 8$ and (middle) rows of size $k = 4$. $R_{r,c}$ denotes the reference at row r and column c. (Right) Each element in a FV is the average of three reference samples. FV_{ci} and FV_{ri} denote, respectively, the FV for the i^{th} column and row.

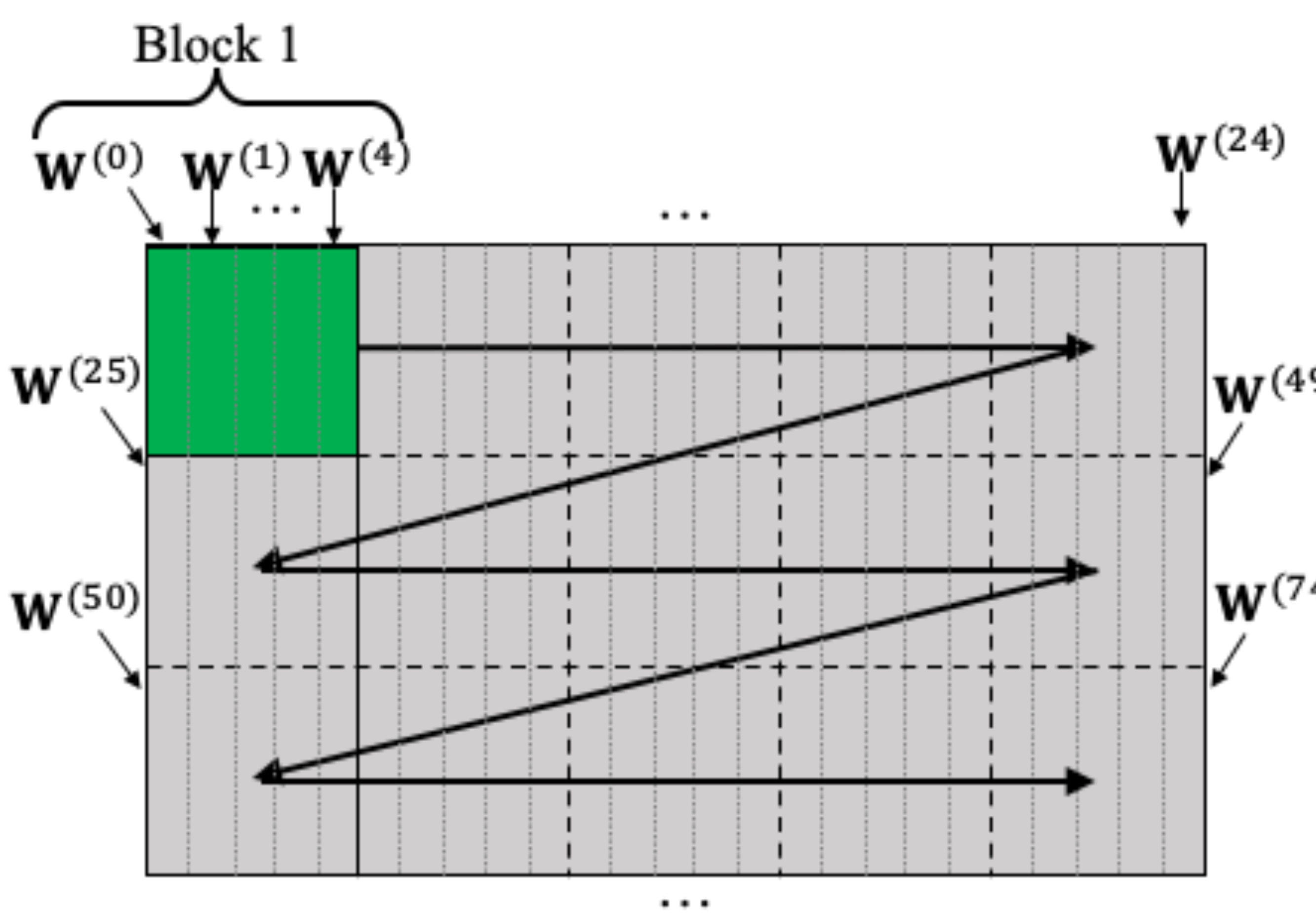

Fig. 4. Intra-prediction of a frame with 15 blocks using ML-based mode H. After predicting the i^{th} column with $\mathbf{W}^{(i)}$, this matrix is updated using gradient descent to produce $\mathbf{W}^{(i+1)}$ to predict next column. In this example, $\mathbf{W}$ is updated 5 times per block (75 times in total) regardless of how often mode H is selected as the best mode by the encoder.

Performance Evaluation

- We use 1800 frames (Y-component) of several sequences.
- Three mode selection processes with $k \times k$ blocks, $k \in \{2, 4, 8, 16, 32, 64\}$, that select the mode that produces the residual block with the smallest energy (see Table 3 and Fig. 5)

1. **Process A**: 35 HEVC modes + 6 ML-based modes.
2. **Process B**: 29 most-frequently used HEVC modes + 6 ML-based modes.
3. **Process C**: only 35 HEVC modes.

✓ Our approach yields improved prediction for all block sizes.
✓ The larger the block size, the higher the performance.

Table 3: Average prediction differences (PSNR - dB) for block size $k \times k$.

	Process A vs. *Process C* (k)						*Process B* vs. *Process C* (k)					
Class	**2**	**4**	**8**	**16**	**32**	**64**	**2**	**4**	**8**	**16**	**32**	**64**
Class A	0.40	0.42	0.89	2.28	5.07	8.05	0.35	0.35	0.81	2.23	5.06	8.05
Class B	0.40	0.40	0.80	1.86	3.86	5.87	0.36	0.35	0.75	1.83	3.86	5.87
Class C	0.95	0.81	1.23	2.41	4.57	6.72	0.90	0.76	1.18	2.38	4.56	6.72
Class D	0.44	0.34	0.82	2.20	4.55	6.68	0.38	0.27	0.75	2.16	4.54	6.68
Class E	0.41	0.51	1.34	3.22	6.63	9.56	0.32	0.43	1.24	3.16	6.62	9.56
Class F	0.74	0.41	0.79	2.03	3.90	5.17	0.70	0.34	0.72	2.01	3.90	5.17
All classes	0.56	0.48	0.98	2.33	4.76	7.01	0.50	0.41	0.91	2.30	4.76	7.01

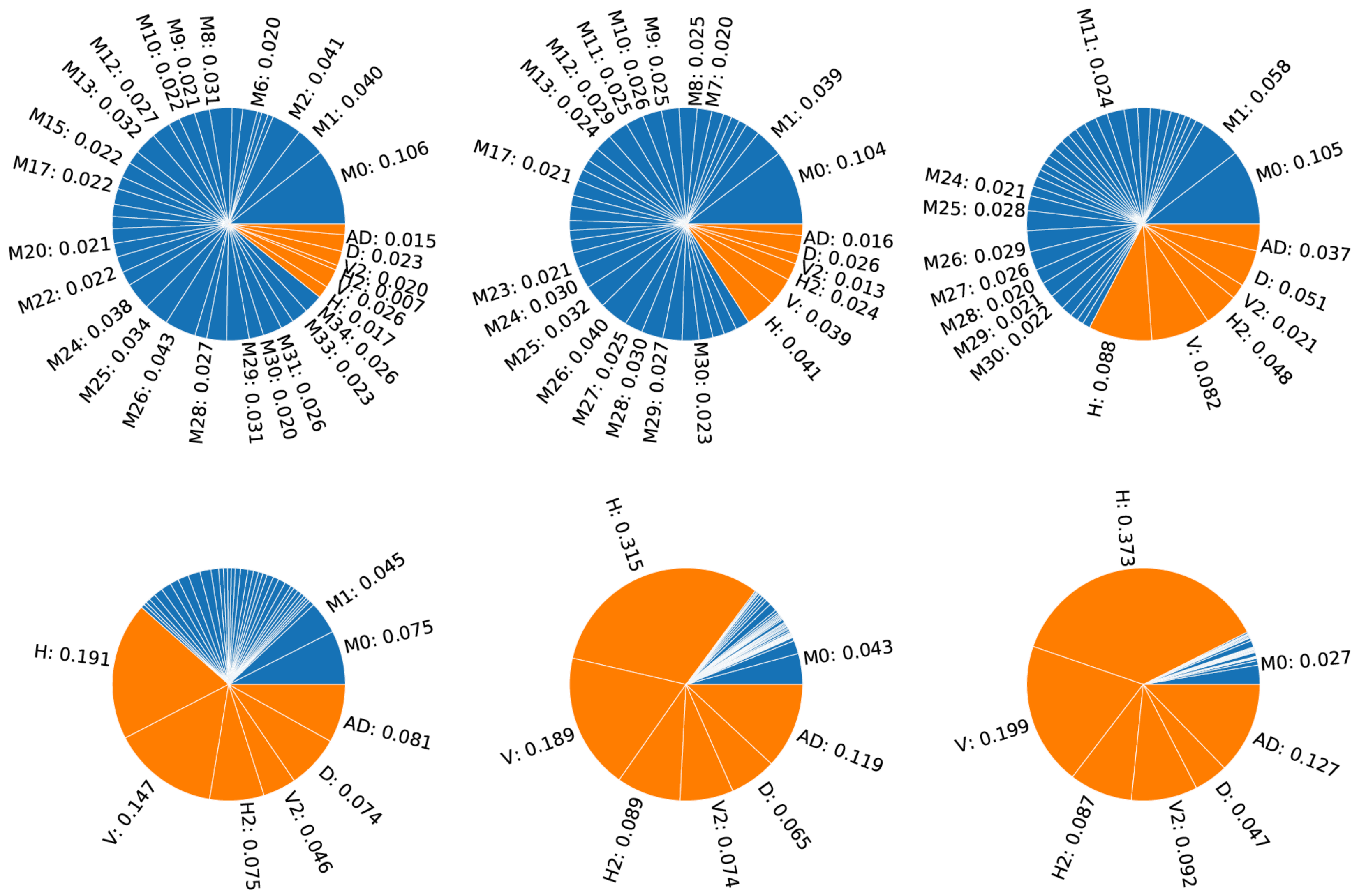

Fig. 5: Process A's distributions of modes for all classes for different block sizes $k \times k$, sorted by mode index. From top to bottom and left to right: $k = \{2, 4, 8, 16, 32, 64\}$. Orange is used for the proposed ML-based modes, and Mi denotes the i^{th} HEVC mode. HEVC modes with a choice fraction below 0.02 are not labeled to improve readability.