MMVC: Learned Multi-Mode Video Compression with Block-based Prediction Mode Selection and Density-Adaptive Entropy Coding

Bowen Liu, Yu Chen*, Rakesh Chowdary Machineni*, Shiyu Liu, Hun-Seok Kim
University of Michigan, Ann Arbor
{bowenliu, unchenyu, mrakeshc, shiyuliu, hunseok}@umich.edu

This supplementary material provides additional information for our proposed MMVC scheme, including the network architectures, experimental settings, additional qualitative results, and an additional study on the impact of block partitioning. To help understanding the overall datapath of MMVC, a simpler overview diagram is provided to supplement the detailed datapath figure available in the main manuscript.

1. Model Architecture

The detailed architecture of our encoder $E(\cdot)$, decoder $D(\cdot)$ and predictor networks $P^{\text{OFC}}(\cdot)$ and $P^{\text{FP}}(\cdot)$ are presented in Figure 1, 2, 3, and 4.

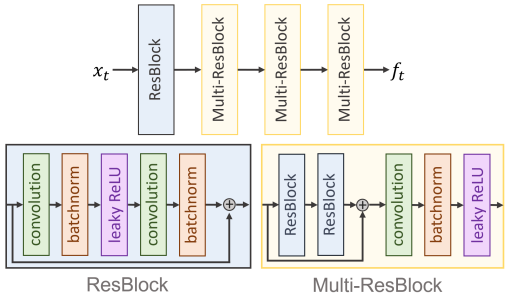

Figure 1. Encoder network structure.

1.1. Feature Encoder

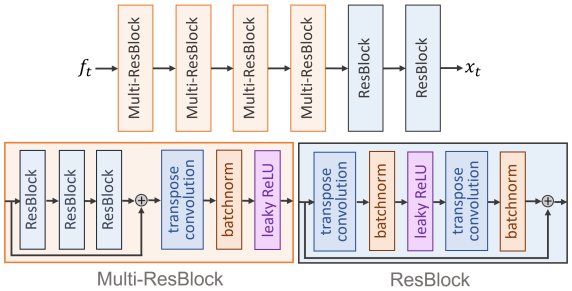

Figure 2. Decoder network structure.

*Equally contributed authors.

The architecture of our encoder $E(\cdot)$ is shown in Figure 1. We use 5×5 2D convolutional layers in ResBlock at the first few layers, and we use 3×3 conv layers for Multi-ResBlocks layers. The first ResBlock has 64 output channels, while the others have 96 output channels. The conv layer in Multi-ResBlock outside of the residual connection has a stride of 2.

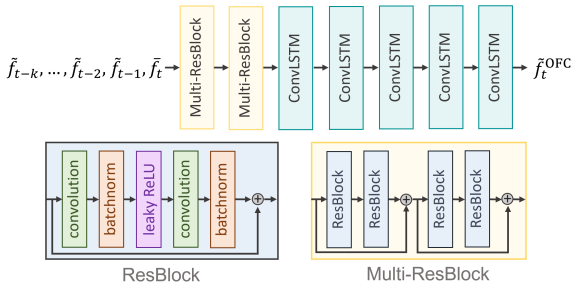

Figure 3. *Optical Flow Conditioned Prediction* network structure.

1.2. Feature Decoder

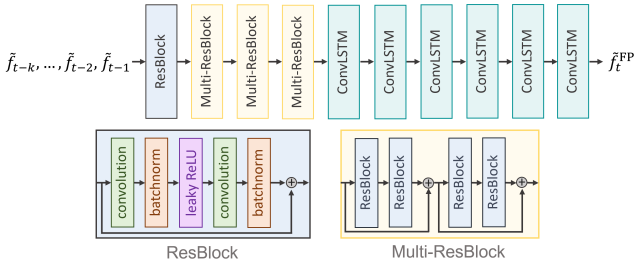

Figure 4. *Feature Prediction* network structure.

Figure 2 depicts the architecture of the decoder network $D(\cdot)$. All transpose conv layers have 3×3 kernels except for the last two ResBlocks. The first three Multi-ResBlocks have 96 output channels, and we reduce this number to 64, 32, and 3 respectively for the remaining Multi-ResBlocks and ResBlocks. The transpose conv layer in Multi-ResBlock outside of the residual connection has a stride of 2.

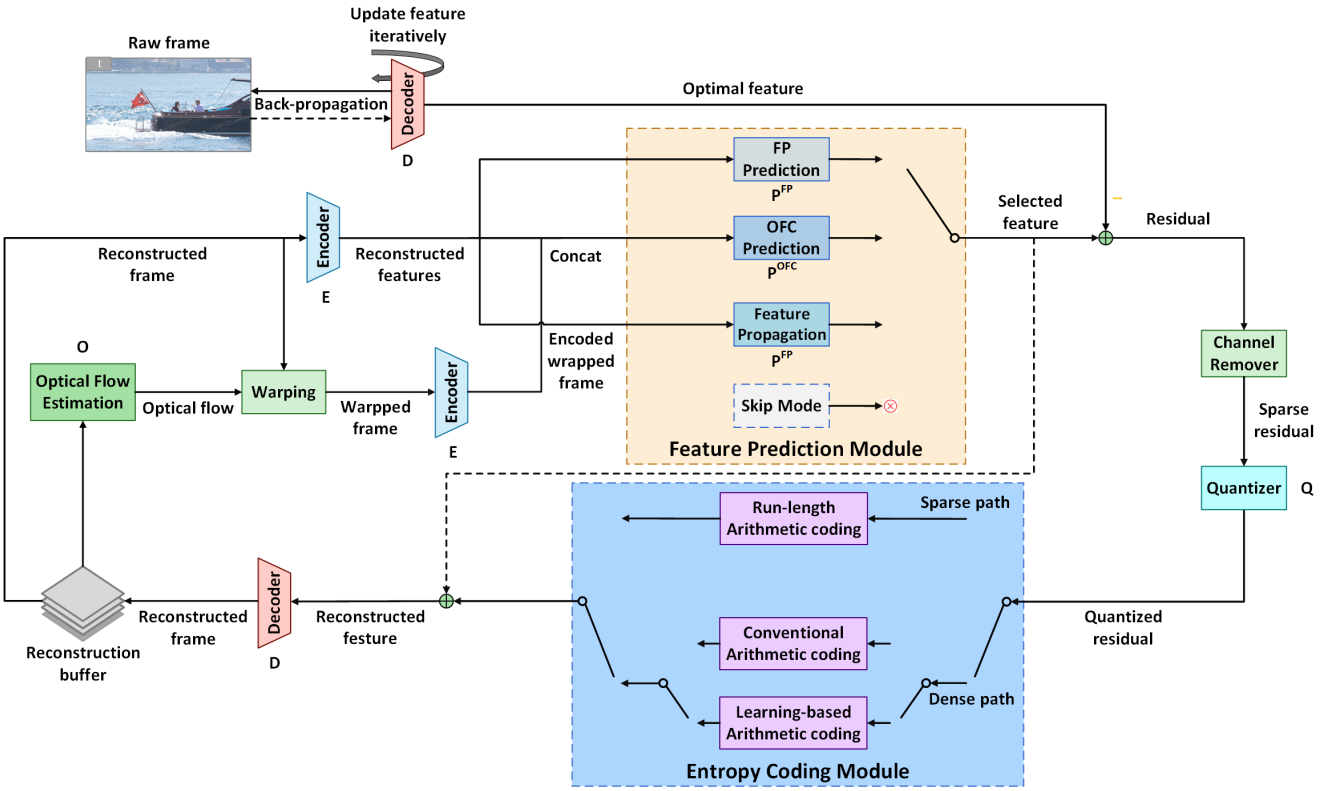

Figure 5. Simplified workflow of our proposed multi-mode video coding method. The current and previous frames are fed into the feature extractor and then go through branches of prediction modes followed by residual channel removal, quantization, and entropy coding process.

1.3. Prediction Network

The predictors $P^{\mathrm{OFC}}(\cdot)$ and $P^{\mathrm{FP}}(\cdot)$ in MMVC have the architectures demonstrated in Figure 3, 4. All conv layers have the same kernel size of 3×3. The output channel size is 96 in ResBlocks and 128 for those in Multi-ResBlocks.

2. HEVC and VVC Configurations

2.1. HEVC Settings

We follow the experiment settings of [2] for HEVC encoding on ffmpeg under *very-slow* mode. The detailed command is presented as follows:

```
ffmpeg -pix fmt yuv420p -s WxH -r FR
-i Video.yuv -vframes N -c:v libx265
-preset veryslow -tune zerolatency
-x265-params "qp=QP:keyint=GOP"
output.mkv
```

2.2. VVC Settings

For rate-distortion curve plotting and visualization generation, we perform experiments with the VTM 11.2 version, under the *low delay P* setting. Following the prior work [1], we choose the QP values to be 22, 25, 28, 31 to achieve various rate-distortion trade-offs.

3. Inference Speed

The latency of our multi-mode approach is (roughly) linearly proportional to the number of modes, and it is higher than a single-mode counterpart. Given the latent representation of each frame (pre-processing step), the encoding and decoding speed is 1852ms and 536ms per 1080p frame, respectively, with A40 single precision GPUs when one specific mode has a dedicated GPU (total 4 GPUs for 4 modes).

4. Details on Mode Selection Strategies

As shown in the workflow in Figure 5, all residuals under different modes are obtained from the same target feature representation by calculating the difference from the prediction. When (quantized) residuals are added to the prediction, all reconstructed frames from different modes have similar quality. Hence, choosing the prediction mode that generates the lowest entropy residual is naturally the best strategy.

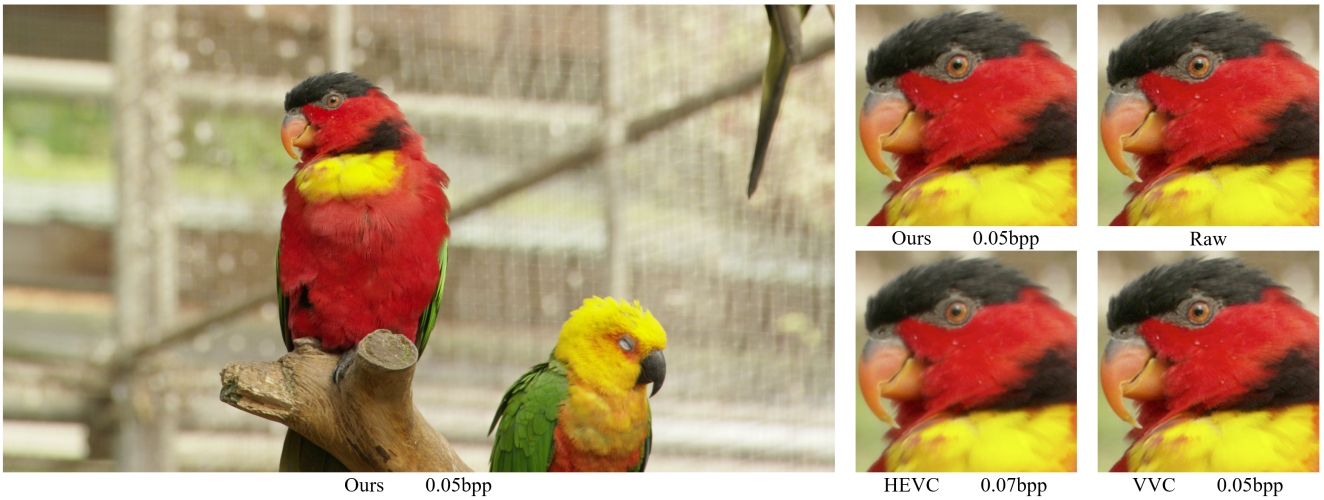

Figure 6. Visualization of a frame from the MCL-JCV dataset. Here we compare our reconstruction with results obtained from the conventional standards, and also with the raw frame.

5. Subjective Quality Study

In Figure 6 we visualize a compressed frame from the MCL-JCV dataset. For reference, we compare our result with reconstructions from HEVC and VVC, along with the raw frame. Compared with these conventional codecs, our proposed MMVC can better preserve details around the parrot's eye and texture of the feathers in our reconstruction with similar or lower bitrate.

Table 1. The block size vs. bitrate comparison for different block partition strategies. We set $k = 16$ as the anchor baseline. Numbers indicate the increased bitrate percentage under similar quality.

Partitioning	UVG	MCL-JCV	Kinetics
k = 8	21.5%	25.6%	76.9%
k = 16	0%	0%	0%
k = 32	11.8%	13.9%	50.2%
k = 64	39.7%	46.9%	138.6%

We visualize the (pixel domain) residuals between previous / warped / (decoded) predicted / reconstructed frame and the current frame at time t in Figure 7. All sub-figures share the same color bar and scale, where darker color implies lower magnitude, and lighter color indicates higher magnitude. By this example, we observe that the optical flow predicts static motions reasonably well, serving as a useful condition in the following prediction stage. Moreover, with the ensemble of the best prediction path in each block, we are able to get sparse residuals and small reconstruction (quantization) errors.

6. Study on Block Partitioning

In our implementation, we set $k = 16$ to partition each frame or feature representation to 16×16 blocks. Notice that k does not change with the input frame size. It implies that an input of higher resolution results in larger blocks. To show the impact of different block partitioning method, we experimented with $k = 8, 16, 32, 64$ on different datasets and measured the k vs. bitrate performance as summarized in Table 1. All tests with $k = 16$ achieve the best performance (lowest bitrate), thus we use $k = 16$ as the baseline for this evaluation. For smaller k values (larger blocks), the likelihood of observing unchanged blocks deceases, reducing the utilization of the *Skip Mode* and therefore degrading the overall performance. Larger k values (smaller blocks) leads to higher chances of activating the *Skip Mode*. However, this strategy needs more bits to store the mode selection map, diminishing the benefits of using the sparse entropy coding path more often given that run-length coding has to operate with smaller windows. Therefore, having a proper partitioning strategy in our scheme is a key to reach to a balanced position in this trade-off.

7. Future Work

The proposed MMVC framework demonstrates the significant benefit of having multiple specialized models and datapath modes that are fine-tuned for certain contexts in the video. A possible future research is to learn a 'unified adaptive mode' that can automatically apply different (implicit) modes to each block of the frame. An automatic (learning-based) mode selection / adaptation is a future research topic to eliminate the overhead of evaluating all modes as in our MMVC framework. Replacing the arithmetic coding in

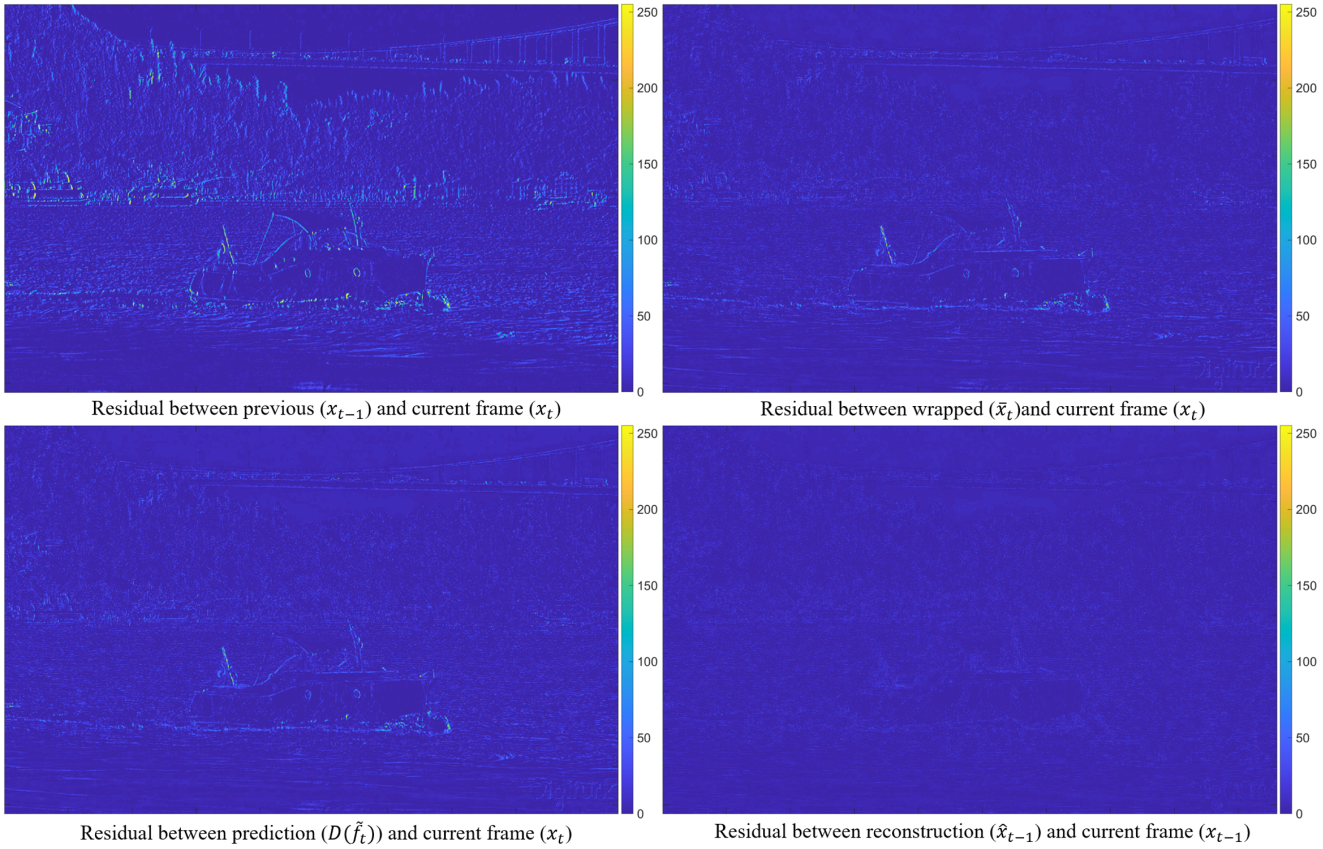

Figure 7. Visualization of residuals between previous / optical flow warped / predicted / reconstructed frame and current frame in the pixel domain. The result is presented as a heatmap, where the blue color indicates a small difference between the prediction and raw frame.

MMVC (for FP mode) to a 'learned' entropy coding model is another promising future work.

References

[1] Zhihao Hu, Guo Lu, and Dong Xu. Fvc: A new framework towards deep video compression in feature space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1502–1511, June 2021. 2

[2] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 2