6. Supplementary material

6.1. Commands for standard codecs

We use the following commands to obtain compressed videos for standard codecs. For ffmpeg, we use the medium preset, disable B-frames and otherwise use default settings. For HM, we use HM-16.25 with the LowDelay-P config, which can be obtained from https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tags/HM-16.25.

```
# For ffmpeg x264
ffmpeg -y -f rawvideo \
  -pix_fmt yuv420p \
  -s:v <width>x<height> \
  -r <framerate> \
  -i <input.yuv> \
  -c:v libx264 \
  -preset medium \
  -crf <crf> \
  -x264-params bframes=0 \
  <output>

# For ffmpeg x265
ffmpeg -y -f rawvideo \
  -pix_fmt yuv420p \
  -s:v <width>x<height> \
  -r <framerate> \
  -i <input.yuv> \
  -c:v libx265 \
  -preset medium \
  -crf <crf> \
  -x265-params bframes=0 \
  <output>

# For HM-16.25 LowDelayP
./bin/TAppEncoderStatic -c \
  ./cfg/encoder_lowdelay_P_main.cfg \
  -i <input.yuv> \
  --InputBitDepth=8 \
  -wdt <width> \
  -hgt <height> \
  -fr <framerate> \
  -f <numframes> \
  -q <qp> \
  -o <output>
```

6.2. Detailed computational complexity

We report the number of model parameters and multiply-and-accumulate (MAC) operations for the RGB and YUV codec in Table 6, separating the numbers for the sender ("send") and receiver ("recv") side. MACs are normalized with respect to the number of input pixels to enable comparisons across input resolutions and input modalities. For each group of five rows, the shown percentage is computed with respect to that group. All percentages are rounded, and some columns do not sum to 100% as a result.

	Module	Parameters		KMACs / pixel	
RGB send	Image-AE	9.414M	30.4%	198.2	30.6%
	Flow-AE	10.038M	32.4%	210.2	32.4%
	Res-AE	10.028M	32.5%	207.8	32.0%
	Flow-Pred	0.748M	2.4%	16.2	2.5%
	Res-Pred	0.748M	2.4%	16.2	2.5%
RGB recv	Image-AE	5.790M	28.8%	158.9	30.1%
	Flow-AE	6.410M	31.9%	168.6	31.9%
	Res-AE	6.410M	31.9%	168.5	31.9%
	Flow-Pred	0.748M	3.7%	16.2	3.1%
	Res-Pred	0.748M	3.7%	16.2	3.1%
YUV420 send	Image-AE	9.496M	30.4%	208.0	30.5%
	Flow-AE	10.124M	32.4%	225.7	33.0%
	Res-AE	10.111M	32.3%	217.6	31.9%
	Flow-Pred	0.751M	2.4%	16.3	2.4%
	Res-Pred	0.750M	2.4%	15.3	2.2%
YUV420 recv	Image-AE	5.880M	28.9%	171.1	29.9%
	Flow-AE	6.500M	31.9%	188.8	33.0%
	Res-AE	6.490M	31.9%	180.7	31.6%
	Flow-Pred	0.751M	3.7%	16.3	2.9%
	Res-Pred	0.750M	3.7%	15.3	2.7%

Table 6: Parameters and MACs per pixel of our RGB and YUV420 codecs.

Note that in order to transmit data, the modules are run in their entirety due to the design of the SSF codec. Compute for the sender side is therefore identical to the total compute as reported in Table 2 in the main text, and we repeat those numbers here for completeness.

On the receiver side, the predictors are run in their entirety, but only the decoder and hyper-decoder of each of the hyperprior modules has to be run. We see that the computational complexity of our predictors is low on the receiver side still: 7.4% of total parameters, 6.2% of total MACs for the RGB receiver, and only 5.6% for the YUV420 codec.

As a minor detail, for the SSF model, the residual decoder does not have to be run on the sender side for the very last frame in a Group of Pictures. As this has a negligible effect on compute when coding with a large GoP size we do not take this into account here.