Learned Video Compression with Intra-Guided Enhancement and Implicit Motion Information

Nannan Zou[1, 2], Honglei Zhang[1], Francesco Cricri[1], Hamed R. Tavakoli[1]
Jani Lainema[1], Emre Aksu[1], Miska Hannuksela[1], Esa Rahtu[2]
[1]Nokia Technologies, [2]Tampere University, Tampere, Finland
nannan.zou@nokia.com

Abstract

Although learned approaches to video compression have been proposed with promising results, hand-engineered video codecs are still unbeaten. On the other hand, learned image compression has already surpassed traditional image codecs. In this paper, we propose a learned video compression system that mimics part of the pipeline of traditional codecs while leveraging learned image compression. It comprises two main modules: a learned intra-frame compression module, and a learned inter-frame compression module that is conditioned on intra-coded frames. These modules use separate learned probability models for entropy coding. The intra-frame codec uses a variant of non-local attention layers. Regarding the inter-frame codec, we propose an implicit motion information mechanism, and an enhancement of the inter-frame predictions by leveraging the high quality information of intra-coded frames. On the learned probability model side, we propose to use the reference frames as additional conditioning information. We used this system as our submitted entry for the 2021 Challenge on Learned Image Compression (CLIC). In our experiments, we show the effectiveness of our system and its components via a set of ablation studies.

1. Introduction

Recently, neural networks have been applied to image and video compression with promising results. Learned image codecs represent now the state-of-the-art when considering both MS-SSIM [3] and PSNR [6] as the visual quality metric. They typically follow the auto-encoder paradigm, where the encoder and decoder networks operate as non-linear transform and inverse transform, respectively. In the area of video compression instead, hand-engineered video compression systems, such as VVC/H.266 [1] HEVC/H.265 standard [14], are still the state-of-the-art. These systems are mostly based on intra-frame and inter-frame prediction, followed by transform-coding of prediction residuals. There are mainly three general approaches for using machine learning techniques for video compression. In a first approach, one or more components of a traditional codec are replaced or augmented by a neural network. For example, in [4], inter-frame prediction of HEVC is improved by using a deep CNN to produce spatially-varying filters from the decoded frames to synthesize the predicted patch. A second approach considers an architecture that is similar to learned image codecs, where a block of frames is provided to an auto-encoder [7]. A third approach follows a similar pipeline as the one in state-of-the-art traditional codecs, such as [15, 16] where an image compression model compresses key frames and a conditional interpolation model interpolates the other frames.

Various probability models have been proposed for lossless and lossy compression systems based on entropy coding. For lossless image compression, PixelCNN [13] and PixelCNN++ [12] model the pixels in an autoregressive manner. The pixels that have been already decoded are used as context to derive the probability distribution of the next pixel. The decoding is performed sequentially in a raster scan order. For lossy compression, [2] proposed a Hyperprior model where global context information is first derived and encoded separately. The probability distribution is calculated on the condition of this global context information. In [3, 11], the Hyperprior model is combined with the PixelCNN style of autoregressive model, where both global context and local context from already decoded pixels are used. [17] proposed a multi-scale probability model where the pixels are processed in multiple scales and contexts are built from low-resolution scales. Although the autoregressive model and the multi-scale model can achieve a better compression performance, the decoding time can not meet the requirement of the CLIC challenge. In our system, the Hyperprior probability model [2] is used to achieve a balance of decoding speed and estimation accuracy.

In this paper, we describe our end-to-end learned video compression system that we submitted to the 2021 Chal-

lenge on Learned Image Compression (CLIC), video compression track. Our team name was nvc. Our system comprises an intra-frame compression sub-system and an inter-frame compression sub-system, both end-to-end learned. The intra-frame codec is based on an auto-encoder architecture using a variant of the non-local attention blocks [9] and a learned probability model. The inter-frame codec includes a prediction of a frame given two reference frames. We propose to: *(i)* enhance the frame prediction by using intra-codec frames, *(ii)* use only implicit motion information, *(iii)* provide the probability model with additional information from the reference frames. In [16], quality features are signalled from encoder to decoder to control the contribution of previously decoded frames on the current frame, by adapting the forget and update gates of an LSTM module. However, we do not use any recurrent neural network and our encoder does not need to encode any information about quality.

2. Proposed methods

Our video codec can be divided into two sub-systems: the intra-frame codec and the inter-frame codec. The intra-frame codec is used to process one frame every nine frames and it does not use any information from other frames. The seven frames between two consecutive intra-frames are processed by the inter-frame codec in a hierarchical sequential manner, i.e., first the frame with index 4 (relative to the start of the intra-frame period) is predicted from intra-coded frames $\{0, 8\}$, then frame 2 and frame 6 are predicted from intra or inter-coded frames $\{0, 4\}$ and $\{4, 8\}$, respectively, finally frames 1, 3, 5 and 7 are predicted from intra or inter-coded frames $\{0, 2\}$, $\{2, 4\}$, $\{4, 6\}$ and $\{6, 8\}$, respectively.

2.1. Intra-frame codec

As shown in Figure 1, our intra-frame codec has an autoencoder architecture similar to other end-to-end learned image coding systems [3, 9, 11]. Light gray boxes are 2D convolution operators where the texts in the boxes indicate kernel size, input channels, output channels, and an optional stride value if it is not 1. Yellow boxes are transposed 2D convolution operators with the same format of texts in them. An input image is first transformed by an input 2D convolution operator and a ReLU activation function to a feature. Then, a sequence of Trunk components and 2D convolution operators with a stride value of 2 convert and downsample the feature to a latent representation of the input frame. The Entropy Model component quantizes and compresses the latent representation into a bitstream. In the Entropy Model, the Hyperprior probability model [2, 11] is used to estimate the probability distribution of the quantized latent representation. The decoder has a mirrored architecture as the encoder, where transposed convolution operators with stride of 2 are used for upsampling.

Our Trunk component incorporates non-local attention and channel attention mechanisms. Non-local attention, which increases the receptive field to the whole input image, has been shown to significantly improve the quality of compressed images in the literature [6, 9]. However, since the memory consumption of the non-local attention block is proportional to the area of the input tensor, we only apply the non-local attention mechanism to the Trunk components operated on the downsampled features. The channel attention technique helps image codecs to improve the compression performance [3, 18]. In our intra-codec, the final attention signals are generated by a sigmoid function applied to the sum of the non-local attention branch and channel attention branch. Features generated by the attention mechanism are used as residuals and added to the input features to generate the final output of the Trunk component. The ResBlocks component in the intra-frame codec comprises a sequence of basic Resnet blocks [8].

2.2. Inter-frame codec

An overview of the inter-frame codec is provided in Figure 2. Components that share weights are shown with the same color. The inputs to the encoder are two reference frames $\hat{x}_{t-1}$, $\hat{x}_{t+1}$ and the target frame x_t. The reference frame may be reconstructed intra-frames or B frames. First, the Encoder Feature Pyramid Net extracts multi-scale features from the input frames. Next, the multi-scale features are aggregated by the Bridge Net and transformed to a latent representation of the target frame. Then, the Entropy Model quantizes and compresses the latent representation into a bitstream. We adopted the Hyperprior probability model [2, 11] as the probability model in our Entropy Model. The Hyperprior probability model uses a context built from the input tensor to derive the probability distribution of the elements in this tensor. To improve compression performance, we enrich the context with extra information derived from the two input reference frames. The two reference frames are transferred to two intermediate tensors by the Entropy Feature Pyramid Net. Then, these two intermediate tensors are concatenated and given to the probability model as an extra context.

Figure 3 shows the architecture of the Bridge Net, where $f_t^{(1/2/3)}$ are features in 3 scales generated from the target frame x_t by the Encoder Feature Pyramid Net; $f_{t-1}^{(1/2/3)}$ and $f_{t+1}^{(1/2/3)}$ are the corresponding features generated from the reference frames. The gray boxes in the figure are 2D convolution operators with the kernel size, input channels and output channels illustrated in the boxes. At each scale, the features from the target frame and reference frames are first transformed by a 2D convolution operator to generate intermediate features. The intermediate features and the downsampled output from the previous scale are concatenated and given to the Aggregator component to generate the out-

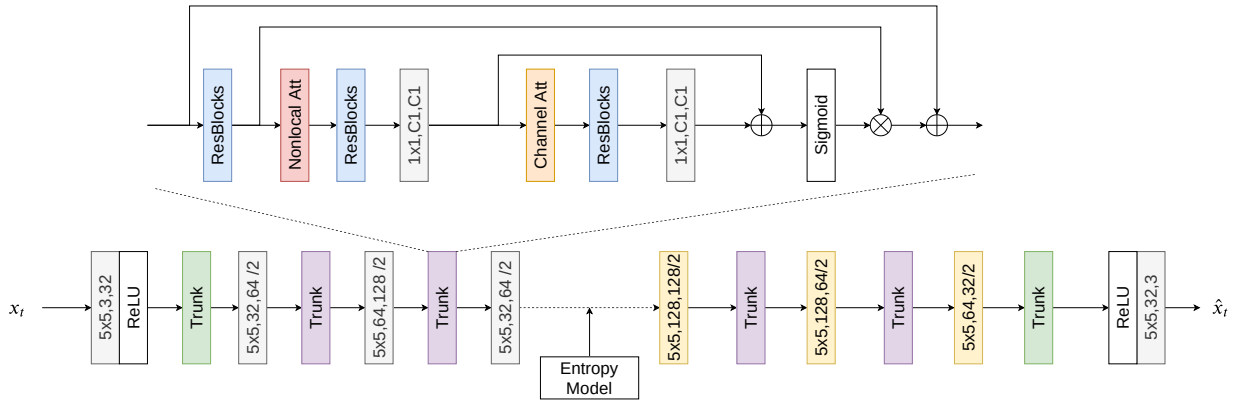

Figure 1. Architecture of the intra-frame codec

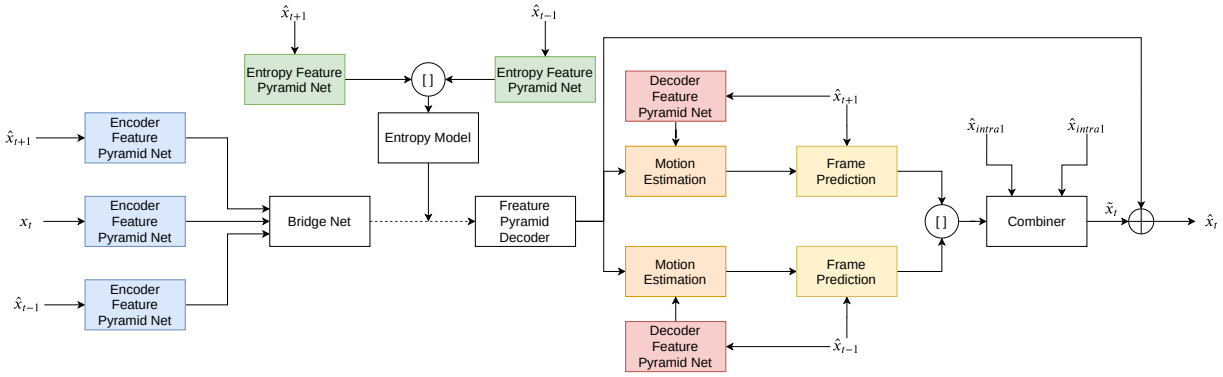

Figure 2. Architecture of the inter-frame codec

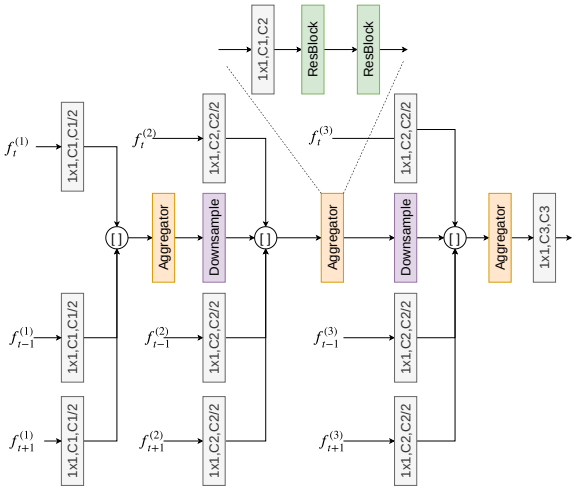

Figure 3. The Bridge component, part of the inter-frame encoder.

put of this scale. The Aggregator component consists of one 2D convolution operator and two basic residual blocks [8]. The output from the Aggregator of the last scale is further transformed by a 1x1 2D convolution operator to generate the latent representation of the target frame.

The decoder takes the compressed bitstream of the latent representation as its input. The bitstream is first decompressed and dequantized to generate a reconstruction of the latent representation by the Entropy Model. Like the encoding process, the Entropy Model also uses the intermediate tensors from the reference frames as the auxiliary context to estimate the probability distribution. The latent representation is implicitly embedded with motion information and residuals. The decoder uses the motion information to warp the reference frames to generate a prediction of the target frame. The predicted target frames by the reference frames are combined by the Combiner component to produce the final prediction $\tilde{x}_t$. Then, the residual is added to $\tilde{x}_t$ to generate the final reconstructed target frame $\hat{x}_t$.

At the decoder side, Feature Pyramid Decoder takes the reconstructed latent representation as its input and generates multi-scale motion features and one residual output. The Motion Estimation component is responsible for motion estimation given the two sets of multi-scale features. One set of the multi-scale features is the output from the

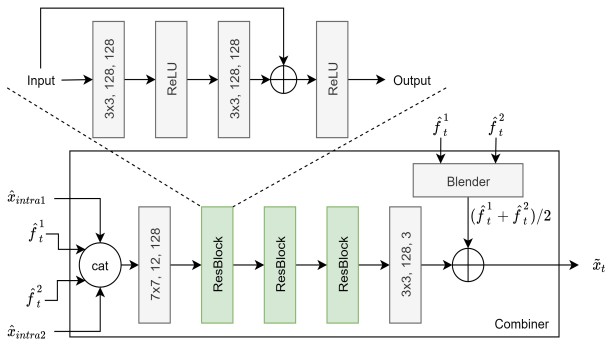

Figure 4. The Combiner, part of the inter-frame decoder.

Feature Pyramid Decoder, and the other set is generated from the reference frame using a Decoder Feature Pyramid Net. The Motion Estimation is a component that has a similar architecture as FlowNet [5, 16]. Different from other systems [10], the Motion Estimation is randomly initialized and end-to-end learned together with other components in the system. After the motion estimation, the system generates temporary predictions of the target frame from the reference frames using a warping operation performed by the Frame Prediction component.

The two temporary predicted reference frames, together with the two closest intra-coded frames are concatenated and given to the Combiner to generate the final prediction of the target frame. Figure 4 illustrates the architecture of the Combiner. It takes the closest intra-coded frames $\hat{x}_{intra1}$ and $\hat{x}_{intra2}$, and the temporary predictions of the target frame $\hat{f}_t^1$ and $\hat{f}_t^2$ as its inputs. The blended version, i.e. arithmetic mean, of the two reference frames are added to the output of the last 2D convolution operator of the Combiner. This design follows the residual paradigm of many deep learning systems [8]. The output of the Combiner is added to the residual output from the Feature Pyramid Decoder to generate the final reconstructed target frame $\hat{x}_t$.

Both the intra-frame codec and the inter-frame codecs were trained on 256x256 patches, by using MS-SSIM and rate loss as the training objectives: $\mathcal{L} = \mathcal{D} + \lambda\mathcal{R}$, where the distortion $\mathcal{D}$ is the negative MS-SSIM, $\mathcal{R}$ is the rate derived from the probability model, and λ is a hyper-parameter. The intra-frame codec was trained only as a stand-alone module. The inter-frame codec was first pretrained as a stand-alone module, by using uncompressed reference frames, and then finetuned by using a similar pipeline as at inference time, i.e., by using intra-coded frames and inter-coded frames as reference frames in hierarchical sequential processing.

3. Experiments

All the training sessions were performed on the CLIC video dataset, using the *Adam* optimizer. We trained our intra-frame codec on all the frames of all videos in the dataset for 20 epochs, with a learning rate of 5e-5 and batch-size of 32 frames. The inter-frame codec was pretrained on uncompressed reference frames with a distance from the target frame randomly chosen from the set $\{1, 2, 4\}$, for 6 epochs, a learning rate of 3e-5 and batch-size of 16 sets of two reference and one target frames. Finetuning of the inter-frame codec is performed on all frames of all videos in the dataset, for 4 epochs, with a learning rate of 8e-5 and batch-size of 54 sets of 9 frames. The first row of Table 1 reports the results on the validation set. In addition, we report the results of an ablation study for the following setups: the full proposed system, the case without intra-frame enhancement, and the case where the Combiner includes only the arithmetic mean of temporary predicted frames. The ablation study is performed by training on only 26 videos from the CLIC training dataset, and evaluating on 10% of the frames in those videos. The "Loss" values were computed on the evaluation set by using $\lambda = 0.31$, similarly as for the training loss. Please note that the loss values are negative due to the negative MS-SSIM loss term. A smaller loss value means better performance. These results clearly show the benefits of our proposed techniques on the codec performance.

Model	Data	BPP	MS-SSIM	Loss
Full	Whole	0.02490	0.96495	−0.95723
Full	Subset	0.01616	0.96381	−0.95880
NoIE	Subset	0.01837	0.96259	−0.95689
NoIECN	Subset	0.00712	0.69398	−0.69177

Table 1. Experimental results. "Full": full model. "NoIE": no intra-guided enhancement. "NoIECN": the Combiner module includes only a linear combination of temporary predicted frames. "Whole" and "Subset" refer to whole CLIC data split and a subset of it, respectively. "Loss" is the evaluation loss.

4. Conclusions

In this paper, we proposed a learned video codec that leverages the good performance of learned image prediction and the successful pipelines of traditional video codecs, by splitting the processing between a learned intra-frame codec and a learned inter-frame codec, where the intra-frames additionally enhance the prediction of other frames. Furthermore, we proposed to condition the probability model of the inter-frame codec also on the reference frames. Another contribution consisted of using only implicit motion information within our inter-frame codec. We evaluated our full model on the CLIC 2021 video dataset. Also, we performed an ablation study for a selected set of components of our codec, showing their benefits.

References

[1] International Organization for Standardization, 2021. ISO/IEC 23090-3:2021 - Information technology Coded representation of immersive media Part 3: Versatile video coding. 1

[2] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv:1802.01436 [cs, eess, math]*, May 2018. arXiv: 1802.01436. 1, 2

[3] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[4] Hyomin Choi and Ivan V. Bajic. Deep Frame Prediction for Video Coding. *arXiv:1901.00062 [cs, eess]*, June 2019. arXiv: 1901.00062. 1

[5] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Husser, Caner Hazrba, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv:1504.06852 [cs]*, May 2015. 4

[6] Zongyu Guo, Zhizheng Zhang, Runsen Feng, and Zhibo Chen. Causal contextual prediction for learned image compression. *arXiv:2011.09704 [cs.CV]*, 2021. 1, 2

[7] A. Habibian, T. V. Rozendaal, J. Tomczak, and T. Cohen. Video compression with rate-distortion autoencoders. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7032–7041, 2019. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv:1512.03385 [cs]*, Dec 2015. 2, 3, 4

[9] Haojie Liu, Tong Chen, Peiyao Guo, Qiu Shen, Xun Cao, Yao Wang, and Zhan Ma. Non-local attention optimized deep image compression. *arXiv:1904.09757 [cs, eess]*, Apr 2019. 2

[10] Haojie Liu, Ming Lu, Zhan Ma, Fan Wang, Zhihuang Xie, Xun Cao, and Yao Wang. Neural video coding using multiscale motion compensation and spatiotemporal context model. *arXiv:2007.04574 [cs, eess]*, Jul 2020. 4

[11] David Minnen, Johannes Ball, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. *arXiv:1809.02736 [cs]*, Sep 2018. arXiv: 1809.02736. 1, 2

[12] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel Recurrent Neural Networks. *arXiv:1601.06759 [cs]*, Aug. 2016. arXiv: 1601.06759. 1

[13] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *ArXiv*, abs/1701.05517, 2017. 1

[14] Gary Sullivan, J.-R Ohm, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22, 12 2012. 1

[15] Chao-Yuan Wu, Nayan Singhal, and Philipp Krähenbühl. Video compression through image interpolation. In *ECCV*, 2018. 1

[16] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with hierarchical quality and recurrent enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 4

[17] Honglei Zhang, Francesco Cricri, Hamed R. Tavakoli, Nannan Zou, Emre Aksu, and Miska M. Hannuksela. Lossless image compression using a multi-scale progressive statistical model. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, Nov 2020. 1

[18] Lei Zhou, Zhenhong Sun, Xiangji Wu, and Junmin Wu. End-to-end optimized image compression with attention mechanism. page 00, 2019. 2