# Learned Low Bitrate Video Compression with Space-Time Super-Resolution

Jiayu Yang, Chunhui Yang, Fei Xiong, Feng Wang, Ronggang Wang*
Peking University Shenzhen Graduate School
{jiayuyang}@pku.edu.cn {rgwang}@pkusz.edu.cn

## Abstract

*This paper presents a learned low bitrate video compression framework that consists of pre-processing, compression and post-processing. In pre-processing stage, the source videos are optionally reduced to low-resolution or low-frame-rate ones to better meet with the limited bandwidth. In compression stage, inter-frame prediction is performed by deformable convolution (DCN). The predicted frame is then used as temporal conditions to compress the current frame. In post-processing stage, the decoded videos are fed into a Space-Time Super-Resolution module, in which the videos are restored to original spatial and temporal resolutions. Experimental results on CLIC22 video test conditions demonstrate that the proposed method shows better performance on both objective and subjective quality at low bitrate. Our team name is PKUSZ-LVC.*

## 1. Introduction

Deep learning techniques have been widely applied in image [2, 3, 7, 18, 19] and video [1, 11, 12, 14–17, 22] compression system recently. Compared with classical hybrid compression frameworks such as H.264 [21], H.265 [20] and H.266 [5] that need manual design of complex mode decision methods and can hardly be optimized as a whole to improve overall performance, learning-based compression methods learn to extract representative features by themselves under the guidance of loss function and can be jointly optimized in an end-to-end manner. Recently, learned video compression codec ENVC [11] has shown comparable rate-distortion (RD) performance with VTM-12.0 in the setting of single-reference prediction regarding sRGB PSNR.

In this paper, we present a learned video compression framework towards low bitrate compression by introducing space-time down-sampling and super-resolution in pre- and post-processing stages, respectively. Specifically, in pre-processing stage, the source videos are optionally down-sampled across spatial or temporal dimension according to input resolution and framerate. The down-sampled videos are then compressed by a learned video codec. For intra (I) frames, we compress features extracted from CNN, and model the entropy parameters by Hyperprior [3] entropy model. For predictive (P) frames, we perform inter-frame prediction with DCN and leverage the predicted frame as temporal contexts, with which contextual encoder can automatically capture temporal correlations. In compression stage, the down-sampling flags, latent features from I frames, motion features and contextual features from P frames are encoded and transmitted to decoder side. In post-processing stage, learned Video Frame Interpolation (VFI) sub-module and Video Super-Resolution (VSR) sub-module restore the videos to original space-time resolutions according to the flags. The pipeline is shown in Fig. 1.

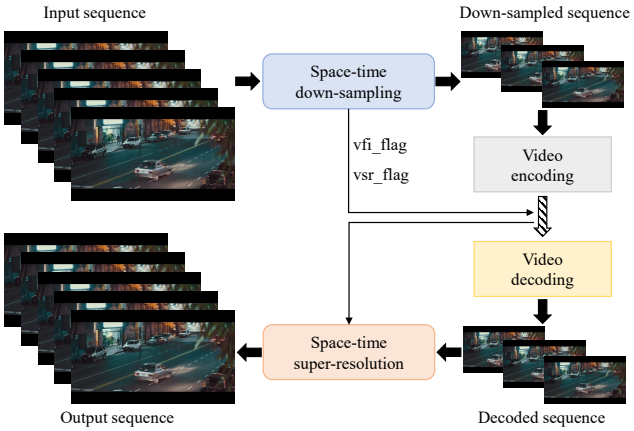


Figure 1. Overview of proposed framework.

## 2. Method

### 2.1. Video Compression

Fig. 2 shows an overview of video compression module. A video sequence is divided into groups of pictures (GoP) and compressed separately. The first frame in a GoP is defined as Intra (I) frame and compressed with an image compression sub-module, while the others are Predictive (P) frames and compressed in a sequential manner with previous aligned frames as conditions. The details of each sub-module are explained as follows.

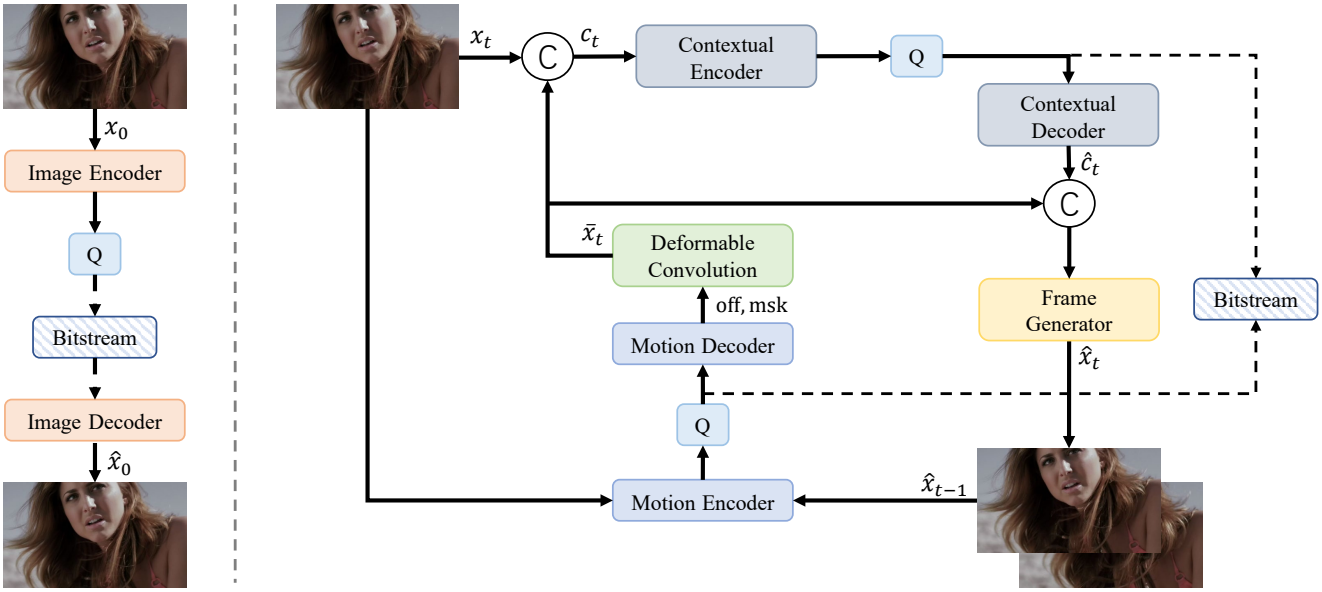


Figure 2. Overview of video compression module. The first frame in a group of pictures (GoP) is compressed as Intra frame (left), while the others are compressed as Predictive (P) frames (right). *Q* stands for quantization and *C* denotes concatenation operation.

#### 2.1.1 Backbones

We perform Intra compression, motion compression and context compression with similar architectures but independent parameters. Encoder and Decoder are built as that in Cheng *et al*. [7], where residual blocks increase receptive filed, attention module captures challenging parts and subpixel convolution helps better reconstruction. Entropy model is based on mean-scale hyperprior [19].

#### 2.1.2 Inter-frame prediction

Different from most existing works that employ flows to express the motions and perform warp operation for alignment, we propose to compensate the motions by deformable convolutions (DCN) [10]. Benefit from its diverse sampling positions, DCN can better deal with complex motions. In DCN, a spatially varying offset field is learned to deform sampling positions of basic convolution. DCNv2 [24] further introduces a modulation mask to evaluate the relative influence of corresponding locations. Given current position $p$ and kernel size $n$, DCNv2 can be described as:

$$y(p) = \sum_{k=1}^{n^2} w\left(p_k\right) \cdot x\left(p + p_k + o_k(p)\right) \cdot m_k(p) \qquad (1)$$

where $k$ represents the index of sampling positions, $w$, $o$ and $m$ denote weight, offset and modulation mask, respectively.

#### 2.1.3 Conditional coding

It is shown in Li *et al*. [14] that the entropy of residue coding tends to be greater than or equal to that of conditional coding. Inspired by this, we employ a contextual encoder to perform conditional coding. Specifically, we define the condition as predicted frame and concatenate the frame with current frame along channel dimension. Then the temporal correlation is explored by contextual encoder. At decoder side, the contextual features are reconstructed to contextual pixels by contextual decoder. The pixels together with predicted frame are fed into frame generator to get the final decoded frame. The whole process can be formulated as:

$$\hat{x}_t = f_{\text{gen}}\left(f_{\text{dec}}\left(\lfloor f_{\text{enc}}\left(x_t \mid \bar{x}_t\right)\rceil\right) \mid \bar{x}_t\right) \qquad (2)$$

where $f_{enc}$, $f_{dec}$ and $f_{gen}$ are contextual encoder, decoder and frame generator, respectively. $\bar{x}_t$ and $\hat{x}_t$ denote predicted frame and reconstructed frame.

#### 2.1.4 Variable rate

The compression module operates at variable rate by introducing scaling factors before quantization [9]. In training stage, the model is trained with L constraints $\{\lambda_1, \cdots, \lambda_L\}$ simultaneously. Then in inference stage, the rate can be adjusted by performing feature-wise multiplication with different factors. The continuous bit rate adjustment is implemented by interpolating fine-grained scaling factors pairs in a weighted geometric averaging manner.


Figure 3. Framework of Video Frame Interpolation (VFI) sub-module (left) and Video Super-Resolution (VSR) sub-module (right).

#### 2.1.5 Loss function

The compression framework is optimized with total rate-distortion (RD) loss unrolled over N frames of a sequence:

$$\sum_{i=0}^{N-1} \lambda d\left(\mathbf{x}_i, \hat{\mathbf{x}}_i\right) + \left[H\left(\mathbf{I}_0\right) + \sum_{i=1}^{N-1}\left(H\left(\mathbf{m}_i\right) + H\left(\mathbf{c}_i\right)\right)\right] \tag{3}$$

where $H(\cdot)$ denotes estimated entropy of encoded latents, including the side information from hyperprior, and $d$ represents mean squared error (MSE) loss.

### 2.2. Space-Time Video Super-Resolution

We perform space-time video super-resolution on down-sampled decoded frames to restore the original spatial and temporal resolutions. In encoder side, spatial down-sampling is implemented by bicubic interpolation with a factor of 2, and temporal down-sampling is performed by sampling frames with interval of 2.

#### 2.2.1 Video frame interpolation

This sub-module takes two adjacent frames as reference and interpolates the middle frame, whose framework is demonstrated on the left in Fig. 3. The two neighboring reference frames are aligned by DCN and added up to get the interpolated frame, which can be formulated as:

$$x_t = DCN\left(x_{t-1}, o_{t-1}, m_{t-1}\right) + DCN\left(x_{t+1}, o_{t+1}, m_{t+1}\right) \tag{4}$$

#### 2.2.2 Video super-resolution

This sub-module takes three low-resolution frames as input and generates a high-resolution one. The previous frame and following frame are first aligned by DCN, then the aligned frames together with middle frame are concatenated and fed into a convolutional layer. Finally, the high-resolution frame is reconstructed with Pixel Shuffle operation. The framework is shown on the right in Fig. 3 and formulations are as follows,

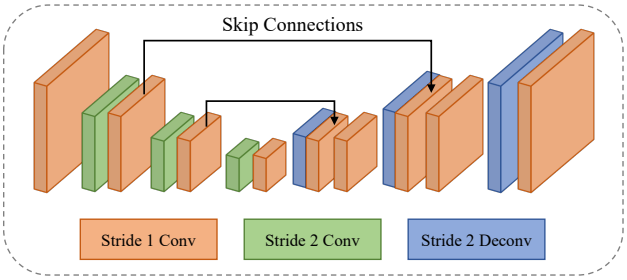


Figure 4. Architecture of offset field estimator (OFE) sub-module.

$$\begin{aligned} \bar{x}_{t-1}^{LR} &= DCN\left(x_{t-1}^{LR}, o_{t-1}, m_{t-1}\right) \\ \bar{x}_{t+1}^{LR} &= DCN\left(x_{t+1}^{LR}, o_{t+1}, m_{t+1}\right) \\ x_t^{HR} &= PixelShuffle(Conv([\bar{x}_{t-1}^{LR}, x_t^{LR}, \bar{x}_{t+1}^{LR}])) \end{aligned} \tag{5}$$

where $\bar{x}$ denotes aligned frame and $[\cdot,\cdot]$ is concatenation.

#### 2.2.3 Offset field estimator

This sub-module takes reference frames as input and outputs spatially-variant offsets $o$ and masks $m$ of DCN. Specifically, input frames (2 for interpolation and 3 for super-resolution) are concatenated together and fed into a U-Net based network. The stacked convolutional layers with down-sampling in U-Net enlarge the receptive field and capture large temporal motions. Besides, skip connections maintain original fine-grained information. The architecture is shown in Fig. 4 and the definition is:

$$o_{t-1}, m_{t-1}, o_{t+1}, m_{t+1} = OFE([x_{t-1}, x_t, x_{t+1}]) \tag{6}$$

#### 2.2.4 Loss function

The loss function of VFI model and VSR model is defined as Charbonnier penalty function [6] between generated frames $\hat{x}_t$ and raw frames $x_t$ for its global smoothness and robustness to outliers:

$$L\left(x_t, \hat{x}_t\right) = \sqrt{\left(x_t - \hat{x}_t\right)^2 + \epsilon^2} \tag{7}$$

where $\epsilon$ is set to $1 \times 10^{-6}$ empirically. Deformable offsets are learned by optimizing final loss $L$ without extra supervision such as optical flow.

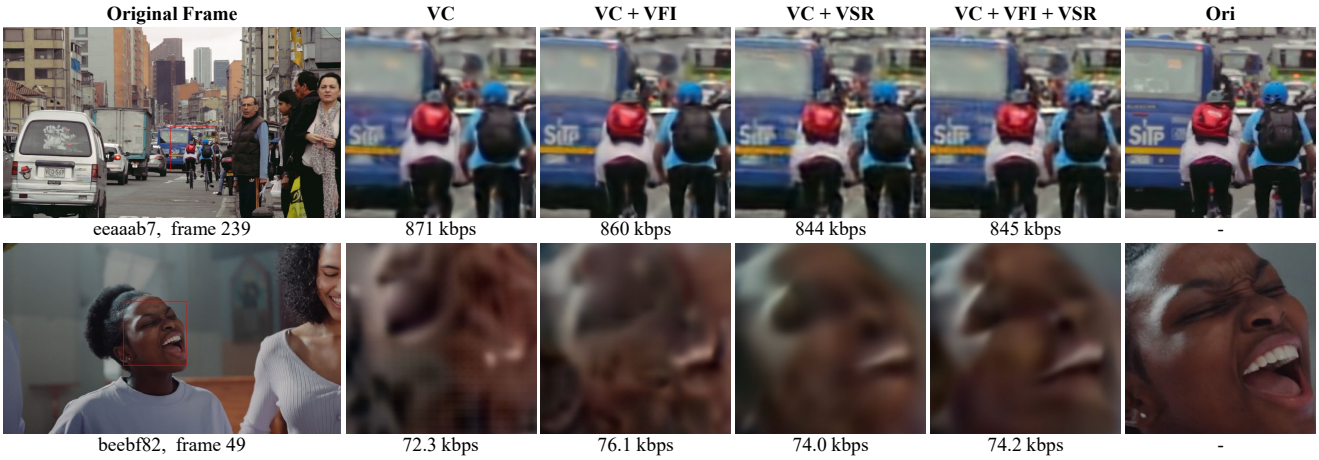


Figure 5. Visual comparison of different methods at two target bitrates. The sequences are from CLIC2022 validation set.

# 3. Experiment

## 3.1. Implementation Details

**Training Data.** We use the Vimeo-90k [23] training split for training. For each video sequence, we randomly select 3 frames (1 as I frame and 2 as P frames), and crop the frames into patches with size of $256 \times 256$. Batch size is set as 16.

**Testing Data.** The 30 video subsets from CLIC2022 validation set are used for testing.

**Settings.** Two compression models are trained with $\lambda$ groups set as $\{0.0005, 0.001, 0.004, 0.008, 0.015\}$ and $\{3e^{-5}, 6e^{-5}, 1e^{-4}, 3e^{-4}, 5e^{-4}\}$ for 1 and 0.1 mbps constraint. VFI model and VSR model are trained with original frames instead of decoded frames from compression models for better performance. Each model is trained up to $5 \times 10^5$ iterations with Adam optimizer [13] and default hyper-parameter settings. Learning rate is initially as $1e^{-4}$ and retained throughout training. For Encoder and Decoder, we use the same parameter settings as that in Cheng [7]. For DCN, the kernel size of deformable conv is 3. For OFE submodule, the filter num of each convolutional layer is 64.

**Implementation.** Compression framework is based on CompressAI [4] and DCN module is based on MMCV [8].

## 3.2. Evaluation Results

We compare the basic compression framework with compression and space-time super-resolution framework.

**Model complexity.** We evaluate model complexity by number of parameters and runtime (Table 1). Experiment is conducted on a 1080P, 30fps sequence with duration of 10s. VC_LR denotes compressed video is a low-resolution one.

**Quantitative results.** Rate-distortion performance of two models with different target bitrates is shown in Fig. 6, *e.g.*, $0.1m$ denotes the model that trained towards 0.1 mbps. Our method with space-time super-resolution shows better performance at low bitrate, which benefits from the lower burden of compression after down-sampling source videos.

| | Enc time (s) | Dec time (s) | Params (MB) |
|---|---|---|---|
| VC | 397.26 | 265.70 | 70.11 |
| VC_LR | 80.76 | 38.42 | 70.11 |
| VFI | - | 60.65 | 0.70 |
| VSR | - | 67.33 | 0.70 |

Table 1. Model Complexity between the two frameworks.

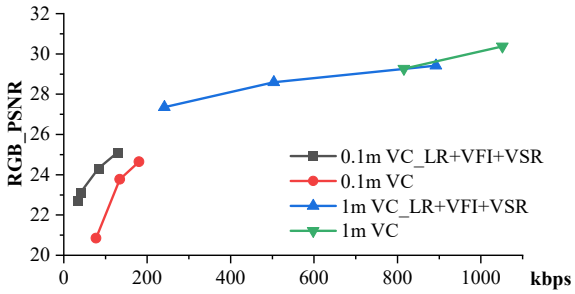


Figure 6. RD performance of different models and methods.

**Qualitative results.** Fig. 5 presents the visual comparison of different combinations. We index frames from 0, which means the frames with odd number, *e.g.*, 239 and 49, are interpolated frames instead of encoded frames. Our method with space-time super-resolution maintains more details and shows better visual quality at low bitrate.

# 4. Conclusion

We propose a learned low bitrate video compression framework with space-time super-resolution and validate its effectiveness. For future work, we suggest to employ more powerful compression and super-resolution models.

## Acknowledgement

This work is supported by National Natural Science Foundation of China U21B2012, 62072013 and 61902008, Shenzhen Cultivation of Excellent Scientific and Technological Innovation Talents RCJC20200714114435057, Shenzhen Research Projects of JCYJ20180503182128089 and 201806080921419290, Shenzhen Fundamental Research Program (GXWD20201231165807007-20200806163656003).

## References

[1] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2020. 1

[2] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 1

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

[4] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020. 4

[5] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 1

[6] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, volume 2, pages 168–172 vol.2, 1994. 3

[7] 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

[8] MMCV Contributors. MMCV: OpenMMLab computer vision foundation. https://github.com/open-mmlab/mmcv, 2018. 4

[9] Ze Cui, Jing Wang, Bo Bai, Tiansheng Guo, and Yihui Feng. G-vae: A continuously variable rate deep image compression framework. *arXiv preprint arXiv:2003.02012*, 2020. 2

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 2

[11] Zongyu Guo, Runsen Feng, Zhizheng Zhang, Xin Jin, and Zhibo Chen. Learning cross-scale prediction for efficient neural video compression. *arXiv preprint arXiv:2112.13309*, 2021. 1

[12] 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*, pages 1502–1511, 2021. 1

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[14] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2

[15] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-lvc: Multiple frames prediction for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3554, 2020. 1

[16] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11006–11015, 2019. 1

[17] Fabian Mentzer, Eirikur Agustsson, Johannes Ballé, David Minnen, Nick Johnston, and George Toderici. Towards generative video compression. *arXiv preprint arXiv:2107.12038*, 2021. 1

[18] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33, 2020. 1

[19] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. *arXiv preprint arXiv:1809.02736*, 2018. 1, 2

[20] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012. 1

[21] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003. 1

[22] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 416–431, 2018. 1

[23] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 4

[24] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 2