Structure-Preserving Motion Estimation for Learned Video Compression

Han Gao
han.gao@std.uestc.edu.cn
School of CSE, University of Electronic Science and Technology of China
Chengdu, China

Jinzhong Cui
jzcui@uestc.edu.cn
School of CSE, University of Electronic Science and Technology of China
Chengdu, China

Mao Ye*
cvlab.uestc@gmail.com
School of CSE, University of Electronic Science and Technology of China
Chengdu, China

Shuai Li*
shuaili@sdu.edu.cn
School of Control Science and Engineering, Shandong University
Jinan, China

Yu Zhao
School of CSE, University of Electronic Science and Technology of China
Chengdu, China

Xiatian Zhu
Surrey Institute for People-Centred Artificial Intelligence, CVSSP, University of Surrey
Guildford, UK

ABSTRACT

Following the conventional hybrid video coding framework, existing learned video compression methods rely on the decoded previous frame as the reference for motion estimation considering that it is available to the decoder. Diving into its essential advantage of strong representation capability with CNNs, however, we find this strategy is suboptimal due to two reasons: (1) Motion estimation based on the decoded (often distorted) frame would damage both the spatial structure of motion information inferred and the corresponding residual for each frame, making it difficult to be spatially encoded on the whole image basis using CNNs; (2) Typically, it would break the consistent nature across frames since the estimated motion information is no longer consistent with the movement in the original video due to the distortion in the decoded video, lowering the overall temporal coding efficiency. To overcome these problems, a novel asymmetric Structure-Preserving Motion Estimation (SPME) method is proposed, with the aim to fully explore the ignored original previous frame at the encoder side while complying with the decoded previous frame at the decoder side. Concretely, SPME estimates superior spatially structure-preserving and temporally consistent motion field by aggregating the motion prediction of both the original and the decoded reference frames w.r.t the current frame. Critically, our method can be universally applied to the existing feature prediction based video compression methods. Extensive experiments on several standard test datasets show that our SPME can significantly enhance the state-of-the-art methods.

*Mao Ye and Shuai Li are corresponding authors. This work was supported by the National Key R&D Program of China (2018YFE0203900) and Sichuan Science and Technology Program (2020YFG0476).

MM '22, October 10–14, 2022, Lisbon, Portugal.

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548156

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Machine learning**; **Artificial intelligence**.

KEYWORDS

Video compression, deep learning, structure-preserving

ACM Reference Format:
Han Gao, Jinzhong Cui, Mao Ye, Shuai Li, Yu Zhao, and Xiatian Zhu. 2022. Structure-Preserving Motion Estimation for Learned Video Compression. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisbon, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548156

1 INTRODUCTION

The transmission of video data is one of the important reasons leading to Internet congestion. Therefore, efficient compression schemes have always been a high demand to reduce transmission and storage costs. In the past decades, researchers have successively developed a few video coding standards, including H.264/AVC [45], H.265/HEVC [36] and H.266/VVC [9]. These schemes use manually designed modules to reduce spatio-temporal redundancy and achieve good performance, but these modules can not be optimized in an end-to-end manner based on large-scale video data.

At present, deep neural network has been applied to video compression schemes. Resembling the conventional hybrid video coding framework, these schemes replaced the traditional modules, such as motion estimation, motion compensation, information compression, etc., with deep neural networks to achieve an end-to-end optimization. From the perspective of encoding space, they can be roughly divided into three categories: 3D encoder based methods [18, 27], frame prediction based methods [1, 5, 11, 14–17, 21, 25, 26, 28, 34, 46, 48] and feature prediction based methods [19, 23, 35]. Although the first two categories achieve great success, the performance and flexibility are limited. Recently, the route of the last category attracts attentions. As for feature prediction, it is mainly applied to the prediction-then-residual and conditional coding approaches. The first one is to compress the residual between the predicted current frame feature and the original current frame feature [19]; another one regards the predicted

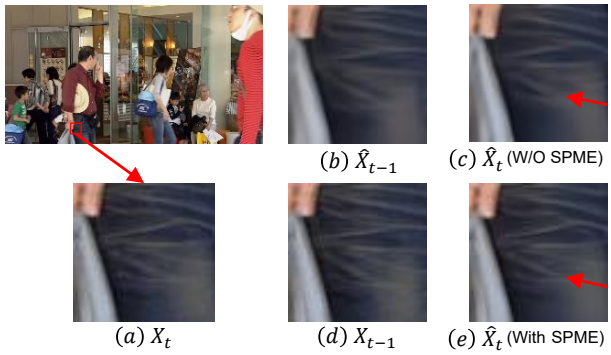

Figure 1: Visual quality of different frames. (a) is the current frame to be encoded. (b) is the decoded previous frame at decoder side. (c) is the reconstructed current frame using only the decoded previous frame as reference. (d) is the original previous frame at encoder side, which has richer information than (b). (e) is the reconstructed current frame using our method, where both the decoded and original previous frames are references. *Zoom in for best view.*

feature as context, which will be used to guide the contextual coding and entropy parameters learning [23].

The above schemes based on feature prediction use the previous decoded frame as the reference frame since it is available at the decoder side. This is expected for the conventional block based video coding, because the residual of each block after motion estimation is independently coded without considering its structure and its effect on the neighboring blocks, and the motion estimation based on the decoded frame provides the smallest residual for each block. Although it seems natural to be extended to the learned video compression, we find that this strategy is actually suboptimal by diving into the difference between the conventional block based video coding and the learned video compression, especially its essential advantage of strong representation capability within CNNs. Concretely, for learned video compression, the motion information and the residual are coded for the whole frame, and the reason it brings better coding performance is the strong representation capability within CNNs. By using the previous decoded frame with inferior quality, the estimated motion field deviates from the real movement of the video, thus damaging the spatial structure of the motion information and the corresponding residual. Moreover, due to the quantization in the learned video coding and the processing of deep CNNs, the distortion in the decoded frame is of random nature. The estimated motion information based on the distorted frames also shows certain randomness around the real motion across frames, breaking the temporally consistent nature of a video, which further makes the exploration of temporal information difficult and lowers the temporal coding efficiency. It is also worth mentioning that such deviated motion information may further aggregate the temporal error propagation since the distortion changes the characteristics of the decoded frames. All in all, the motion estimation completely based on the decoded frame damages the spatial structure and the temporal coherency, which in turn lowers the coding performance.

As shown in Fig 1, due to the lossy nature of video codec, some information has been lost in the previous decoded frame (b) compared with the original previous frame (d), and thus leads to inaccurate motion information from the real motion in the video. Fortunately, the motion estimation process is operated at the encoder side. Therefore, motion estimation based on the original previous frame is feasible, and expected to achieve accurate motion information. However, the following motion compensation is still based on the previous decoded frame, affecting the corresponding residual. Therefore, the decoded reference frame still needs to be considered in the motion estimation process to enhance the overall prediction efficiency.

Based on the above analysis, we propose a plug-and-play motion estimation method, named as SPME (Structure-Preserving Motion Estimation), to explore both the original and decoded reference frame information for estimating spatially structure-preserving and temporally consistent motion field. First a Structure-Preserving Motion estimation Network (SPM-Net) is proposed to get a highly-fitting motion field from the perspective of accurate motion and efficient prediction. It extracts the main and auxiliary motion fields between the decoded and original previous frame features w.r.t the current frame feature respectively. Then a fusion module is proposed to fuse these two motion fields and form the final structure-preserving motion field to be transmitted. Further, a Motion Compensation and Prediction Enhancement Network (MCPE-Net) is proposed to predict the current frame feature. In this network, the predicted current frame feature is also enhanced by extracting and mining the useful information in the decoded previous frame feature.

Our contributions can be summarized as follows: (1) We identify a generic limitation of the motion estimation in the feature prediction based video compression methods. To address this, we propose a plug-and-play module, called SPME. Specifically, we propose to use the original previous frame as auxiliary data to enhance the motion estimation between the current frame and the decoded previous frame to form a structure-preserving motion field. In this way, the spatial and temporal compression performance are improved, and the phenomenon of error propagation is restrained to some extent. (2) In order to further enhance the quality of the predicted current frame feature, an motion compensation and prediction enhancement module is developed to extract and mine useful information from the decoded previous frame. (3) On the standard test datasets, the proposed method achieves a gain of 9.99% and 3.59% in terms of bit-rate saving on top of the state-of-the-art methods FVC [19] and DCVC [23], respectively, validating that our method can be widely used as a plug-in unit.

2 RELATED WORKS

Image compression. The past decades have witnessed the vigorous development of image compression industry, a series of image compression standards such as JPEG [42], JPEG2000 [38] and BPG [7] were proposed to reduce spatial redundancy as much as possible through hand-made modules. And great compression performance and operation stability were achieved.

After the rise of deep learning, deep learning based image compression has attracted attentions. Several methods based on recurrent neural networks (RNNs) proposed earlier [20, 40, 41] gradually encode the residual information in the previous step to compress the image. Recent methods [2, 3, 10, 22, 39] use variational auto-encoder (VAE) based on convolutional neural networks (CNNs).

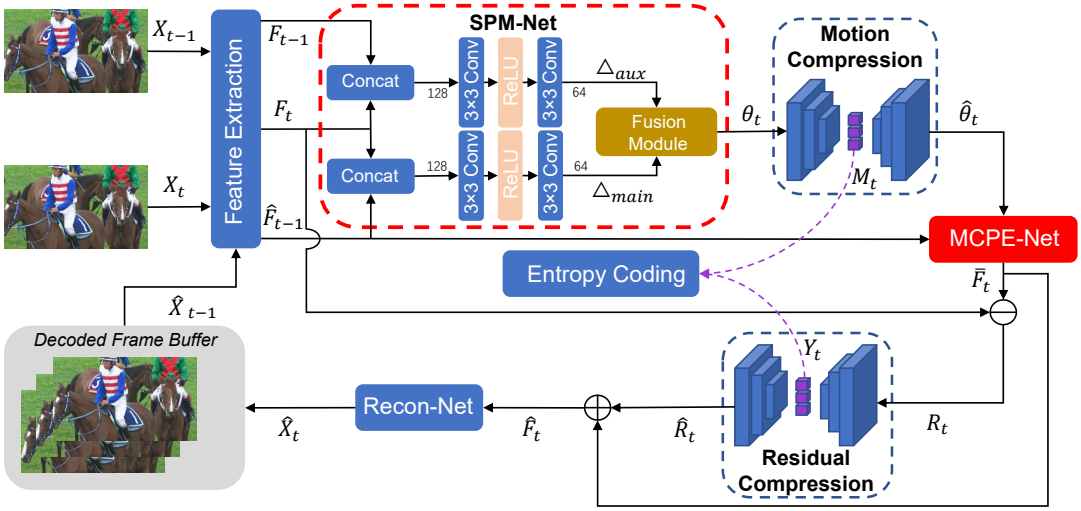

Figure 2: The framework of our proposed SPME based upon the baseline FVC [19] (without multi-frame feature fusion module). For a frame X_t and its references X_{t-1}, $\hat{X}_{t-1}$, the Feature Extraction module extracts their features named F_t, F_{t-1} and $\hat{F}_{t-1}$ respectively. Then the structure-preserving motion field θ_t is calculated by the Structure-Preserving Motion estimation Network (SPM-Net). The Motion Compensation and Prediction Enhancement Network (MCPE-Net) is proposed to predict the current frame feature and enhance it. The motion field θ_t and the feature residual R_t are compressed and transmitted. The number besides the solid line represents the number of channels.

Specifically, pixels are mapped to latent feature space for encoding and decoding. A hyper prior entropy model was proposed firstly to capture the distribution of latent features using additional bits in [4] for better compression performance. Some follow-up methods [24, 30, 32] integrate context factors to further reduce the spatial redundancy in the latent features by serial operation. These methods have achieved inspiring effect compared with the traditional codecs.

Video compression. In order to remove spatio-temporal redundancy effectively, a host of video compression standards, such as H.264/AVC [45], H.265/HEVC [36] and H.266/VVC [9] were proposed. These standards based on hybrid compression framework achieved efficient compression performance through hand-made modules such as prediction, transformation, quantization, entropy coding and in-loop filtering, etc.

In recent years, video compression based on deep learning has become a new research hotspot. One of the representative method is based on *frame prediction*, for example, in the classic scheme DVC [28], optical flow estimation is used by Lu *et al.* to replace the motion estimation in the traditional coding method, and neural networks are employed to replace each module in the traditional methods to realize an end-to-end optimization; In M-LVC [25], multi-reference frames are applied to promote deep video compression; RLVC [48] uses recurrent auto-encoder and recurrent probability model to improve motion and residual compression; In SSF [1], optical flow is introduced from two dimensions to multi-dimensional space by Gaussian blurring, and motion compensation is carried out in the high-dimensional space. *Feature prediction* based method is another mainstream, which can use the rich information in feature space to achieve better reconstruction effect. It has two technical routes, for example, in FVC [19], deformable convolution [13] is firstly applied to realize feature-level prediction, and the residual is calculated in the feature space; in DCVC [23], video compression is regarded as a conditional coding problem, which aims at generating condition in the feature space based on feature prediction to remove the temporal redundancy.

The above feature prediction based methods use the decoded previous frame at the decoder side as reference resembling the conventional hybrid video coding framework. However, this damages the structure of the obtained motion vector and the corresponding residual for each frame, making it hard to be spatially encoded on the image basis. Luckily, we have the original previous frame at the encoder side, which contains all the information lost in the corresponding decoded frame. Thus our work aims at the two technical routes of feature prediction based methods, pays attention to the role of the original previous frame at the encoder side, and uses its rich information in an adaptive manner.

3 PROPOSED METHOD

Problem statements. Let $X = \{X_1, X_2, \cdots, X_{t-1}, X_t, \cdots,\}$ be a GOP in a video sequence. At Low Delay P (LDP) encoding mode, X_1 is a key frame, and other frames are forward predicted frames. X_t is the frame to be encoded at the current time. Our goal is to use the previous frame as reference and less bits as much as possible to get a higher quality reconstructed frame $\hat{X}_t$. In our method, both $\hat{X}_{t-1}$ and X_{t-1} are used as references for motion estimation.

Overview. Without loss of generality and in consideration of complexity, we firstly take FVC* [19] (without multi-frame feature fusion module) as the baseline and apply our method on this framework, and the extension to DCVC [23] is described at the experiment part. As shown in Fig. 2, the framework consists of 7 modules:

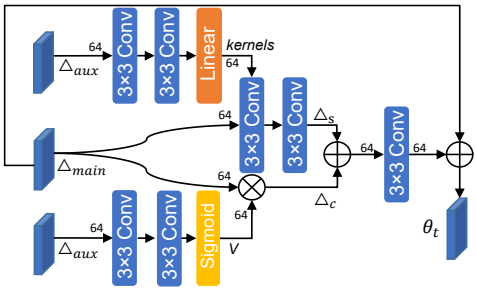

Figure 3: The network structure of the fusion module in the proposed SPM-Net, which consists of two branches: the upper is at spatial level, and the lower is at feature channel level.

Feature Extraction, SPM-Net, Motion Compression, MCPE-Net, Residual Compression, Recon-Net and Entropy Coding.

First, the current frame X_t, the decoded previous frame $\hat{X}_{t-1}$ and the original previous frame X_{t-1} are mapped to the feature space through the **Feature Extraction** module. Then the corresponding features $F_t, \hat{F}_{t-1}, F_{t-1}$ are input to the Structure-Preserving Motion estimation Network (**SPM-Net**) to calculate the highly-fitting motion field θ_t. Then θ_t is compressed by a **Motion Compression** module and sent to the decoder side to form the corresponding reconstructed motion field $\hat{\theta}_t$. After that, the Motion Compensation and Prediction Enhancement Network (**MCPE-Net**) is used to generate the predicted current frame feature $\bar{F}_t$ by using $\hat{\theta}_t$ and $\hat{F}_{t-1}$. The residual R_t between F_t and $\bar{F}_t$ is compressed through a **Residual Compression** module and sent to the decoder side to form the corresponding reconstructed residual $\hat{R}_t$. Then $\hat{R}_t$ is added to $\bar{F}_t$ to form the reconstructed feature $\hat{F}_t$. Finally, we use a reconstruction network (**Recon-Net**) to reconstruct the current frame $\hat{X}_t$. The **Entropy Coding** module provided by CompressAI [6] is used for transforming the quantized features into the bit-streams and the probability distribution of transmitted information is obtained by a CNN network at the training stage.

Next, we will mainly introduce the modules **SPM-Net** and **MCPE-Net**. Other modules are the same as FVC [19].

3.1 Structure-Preserving Motion estimation Network (SPM-Net)

As shown in Fig. 2, the SPM-Net includes three parts: main motion field Δ_{main} computation; auxiliary motion field Δ_{aux} computation; and motion field fusion.

Two motion field computation. The two branches in SPM-Net, as shown in Fig. 2, are used for generating the main motion field Δ_{main} and the auxiliary motion field Δ_{aux}. A lightweight network is employed as the following,

$$\Delta_{main} = Conv_{3\times3} \circ ReLU \circ Conv_{3\times3} \circ C(F_t, \hat{F}_{t-1}),$$

$$\Delta_{aux} = Conv_{3\times3} \circ ReLU \circ Conv_{3\times3} \circ C(F_t, F_{t-1}),$$

where $C(\cdot,\cdot)$ represents the concatenation operation, $Conv_{3\times3}$ represents the convolution operation with the kernel size of 3×3, and $ReLU$ is the activation function.

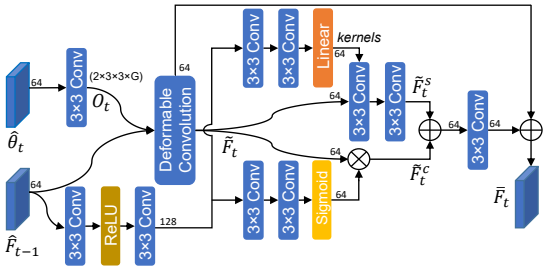

Figure 4: The network structure of MCPE-Net.

Motion field fusion. The fusion module is shown in Fig. 3. The main motion field is adaptively fused with the auxiliary motion field at spatial and channel levels respectively. At the spatial level, by extracting spatial information in Δ_{aux}, a kernel-adaptive network is used to predict convolution kernels as follows,

$$kernels = Linear \circ (Conv_{3\times3})^2(\Delta_{aux}), \tag{1}$$

where $kernels \in \mathbb{R}^{64\times3\times3}$ represents 3×3 convolution kernels with the same number of channels as Δ_{main}, $Linear$ represents a linear layer and $(\cdot)^n$ represents the serial cascade of n modules. Then, these convolution kernels are applied to the feature maps of Δ_{main} respectively as follows,

$$\Delta_s = Conv_{3\times3} \circ Conv_k(\Delta_{main}), \tag{2}$$

where $Conv_k$ represents the convolution with the predicted $kernels$. The spatial correlation between Δ_{main} and Δ_{aux} is captured by the agile kernels.

At the channel level, attention mechanism is also applied. A lightweight network is used to obtain the channel attention weights denoted by

$$V = Sigmoid \circ (Conv_{3\times3})^2(\Delta_{aux}), \tag{3}$$

where $Sigmoid$ means the standard Sigmoid function layer. Then the weight is applied to the main motion field as $\Delta_c = V \otimes \Delta_{main}$.

At last, we fuse Δ_s and Δ_c to achieve the goal of absorbing the native motion information Δ_{aux}, which is denoted by

$$\theta_t = \Delta_{main} + Conv_{3\times3}(\Delta_s + \Delta_c). \tag{4}$$

After the highly-fitting motion field θ_t is generated, it will be input into the **Motion Compression** module and then the code stream is transmitted to the decoder side.

Remark. Because we fuse Δ_{aux} based on Δ_{main}, the information in the decoded previous frame is considered, thus the estimated motion field is fitting the decoded previous frame. At the same time, the auxiliary motion field Δ_{aux} is beneficial to remedy the inaccurate features in $\hat{F}_{t-1}$ for structure-preserving motion estimation.

3.2 Motion Compensation and Prediction Enhancement Network (MCPE-Net)

As denoted in Fig. 2, after the **Motion Compression** module decompressed the motion field $\hat{\theta}_t$, this motion field will be employed to predict the feature of the current frame X_t. But at the decoder side, only the decoded previous frame $\hat{X}_{t-1}$ and its corresponding feature $\hat{F}_{t-1}$ can be accessed. As stated before, the decompressed motion field cannot retrieve the missing information in the feature

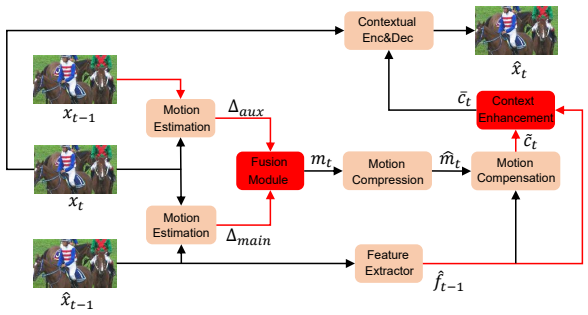

Figure 5: The illustration of applying our method to DCVC [23]. The red blocks and red lines represent our proposed modules.

of the decoded frame. The predicted current frame feature needs to be enhanced based on the decoded previous frame feature. As shown in Fig. 4, the proposed MCPE-Net consists of two parts: deformable convolution based feature prediction and enhancement.

At first, deformable convolution is used to predict $\tilde{F}_t$ based on $\hat{F}_{t-1}$ and the motion field $\hat{\theta}_t$ [19]. The deformable offsets are calculated as $O_t = Conv_{3\times3}(\hat{\theta}_t)$ where $O_t \in \mathbb{R}^{G\times2\times3\times3\times H\times W}$, where "2" represents horizontal and vertical directions; "3×3" represents that each point has have 9 directions; "H" and "W" represent the height and width of the feature map respectively; while "G" is the channel group number which is consistent with FVC [19], set as 8. Finally, the initial predicted feature $\tilde{F}_t$ is calculated as

$$\tilde{F}_t = DCN(O_t, \hat{F}_{t-1}), \tag{5}$$

where $DCN(\cdot,\cdot)$ represents the deformable convolution.

The enhancement part uses the decoded previous frame feature $\hat{F}_{t-1}$ to enhance the predicted current frame feature. The spatial and channel attention mechanisms are applied to obtain useful information in the decoded previous frame feature. At the spatial attention level, $\tilde{F}_t^s = Conv_{3\times3} \circ Conv_k(\tilde{F}_t)$; while at the channel attention level, $\tilde{F}_t^c = (Sigmoid \circ (Conv_{3\times3})^2 \circ Convs(\hat{F}_{t-1})) \otimes \tilde{F}_t$. Finally, the final enhanced feature $\bar{F}_t$ is calculated as

$$\bar{F}_t = \tilde{F}_t + Conv_{3\times3}(\tilde{F}_t^c + \tilde{F}_t^s). \tag{6}$$

Remark. MCPE-Net uses the similar idea in Section 3.1. The overlapping or similar features existed in $\hat{F}_{t-1}$ are combined in the the predicted feature $\tilde{F}_t$, and it compensates the missing information in $\tilde{F}_t$ by $\hat{F}_{t-1}$. As shown in Fig. 1(e), the enhanced feature can reconstruct more details.

3.3 Loss Function

The purpose of video compression is to use less bits to obtain better reconstruction quality. Therefore, the same as the baseline [19], our scheme also optimizes the rate-distortion (R-D) tradeoff as follows,

$$L = \lambda D + R = \lambda d(x_t, \hat{x}_t) + R(\hat{M}_t) + R(\hat{Y}_t), \tag{7}$$

where $d(x_t, \hat{x}_t)$ represents the distortion between the input frame and the reconstructed frame, and $d(\cdot,\cdot)$ represents the mean square error MSE or the multi-scale structural similarity MS-SSIM. $R(\hat{M}_t)$ represents the number of bits required for the latent representation of encoded motion information and the corresponding hyper prior information, and $R(\hat{Y}_t)$ represents the number of bits required for the latent representation of encoded residual information and the corresponding hyper prior information. λ is the Lagrange multiplier used to control the trade-off between bit-rate and distortion.

Table 1: BDBR(%) results of FVC*, DCVC and their respective enhanced versions based on our modules when compared with H.265 in terms of PSNR.

	FVC*	SPME(FVC*)	DCVC	SPME(DCVC)
HEVC B	-21.45	**-26.78**	-35.59	**-39.53**
HEVC C	-2.14	**-9.06**	-14.88	**-18.93**
HEVC D	-16.55	**-21.60**	-26.26	**-29.98**
HEVC E	8.31	**-11.08**	-17.69	**-21.04**
MCL-JCV	16.12	**-3.16**	-28.78	**-31.70**
UVG	-12.82	**-16.81**	-37.74	**-41.30**

4 EXPERIMENTS

4.1 Experimental setup

Training dataset. We use Vimeo-90k [47] as the training dataset, which contains 89,800 video clips with each video having 7 frames. And the resolution of the images is randomly cut from 448x256 to 256x256.

Testing dataset. Similar to the baseline FVC [19], we use HEVC [36], UVG [31], MCL-JCV [43] as the testing dataset. The HEVC dataset (Class B, Class C, Class D, Class E) contains sixteen videos with the resolutions from 416 × 240 to 1920 × 1080. The UVG dadaset contains seven videos with the resolution of 1920x1080. And the MCL-JCV dataset consists of thirty 1080p video sequences.

Evaluation metrics. We use PSNR and MS-SSIM [44] to measure the distortion between the reconstructed frame and the original frame. BPP (bit per pixel) is used to measure the number of bits required for motion information, feature residual information and their respective hyper prior information of each specific pixel in the encoded frame. BDBR [8] value is also calculated, which represents the average bit delta using the same PSNR or MS-SSIM.

Implementation details. Due to space limitations, more details can be obtained from the Appendix. The code and appendix materials are available at *https://github.com/gaohan-12/SPME*.

4.2 Transplanting to DCVC

Motivation. To verify that our method can be widely used as a plug-in unit in the learned video compression frameworks, we transplant our method from FVC* [19] to DCVC [23]. DCVC is a framework based on conditional coding, which aims at generating accurate context information based on motion estimation and compensation. The context is regarded as temporal prior, which is used to guide the entropy coding together with the hyper prior and spatial prior. As shown in Fig. 5, the red blocks and lines represent our proposed method, and the other modules have the same structure with DCVC. Next, we will mainly describe how to transplant our proposed method to DCVC framework, and other modules same with DCVC will be skipped.

Structure-preserving motion estimation. At the encoder side, because DCVC uses optical flow as the motion field, we fuse the

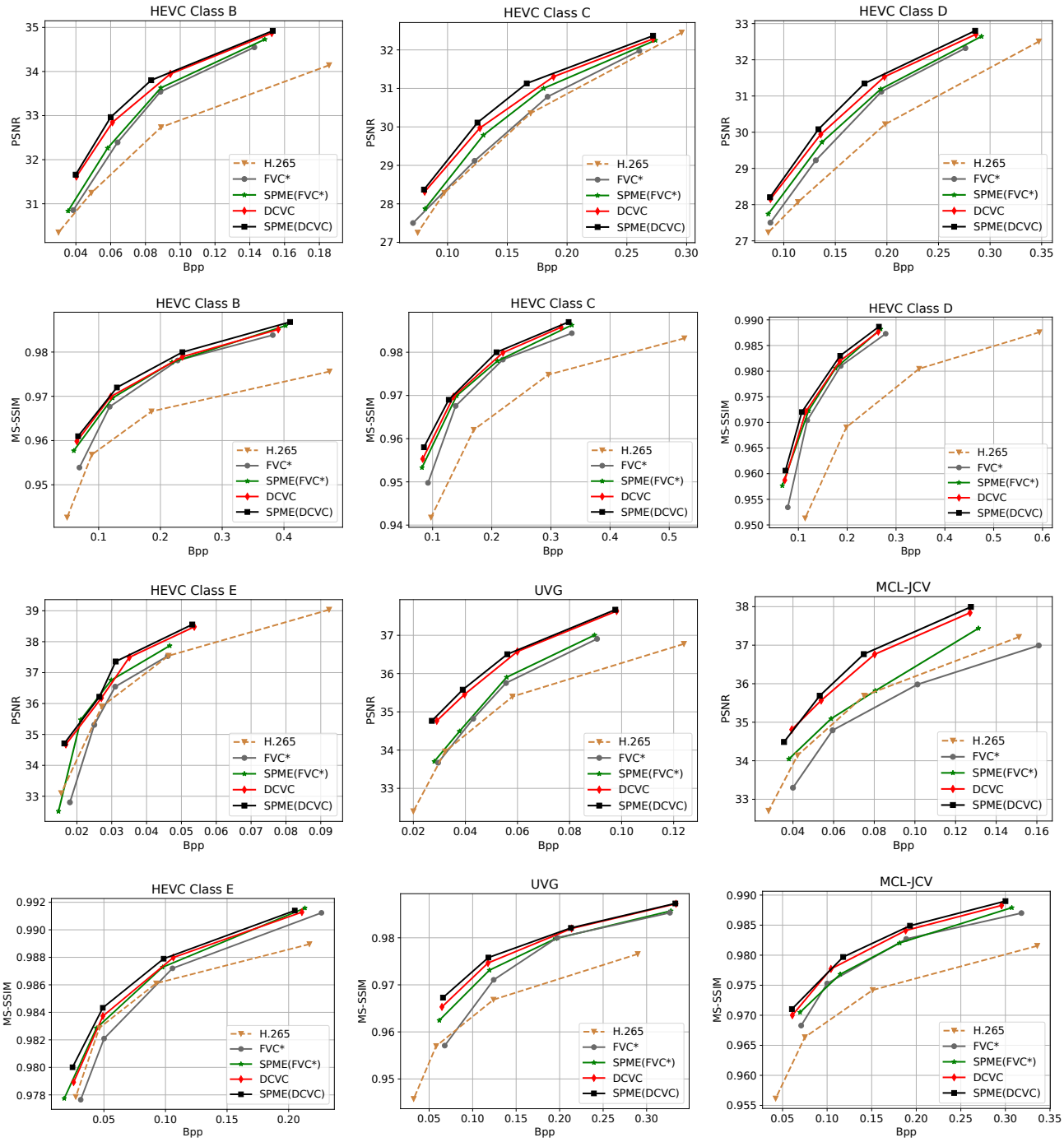

Figure 6: The experimental results on the HEVC Class B, Class C, Class D, Class E, UVG and MCL-JCV datasets.

optical flows with channel = 2. Firstly, two optical flows named Δ_{main} and Δ_{aux} are calculated by SpyNet [33], in which Δ_{main} is calculated between the current frame x_t and the decoded previous frame $\hat{x}_{t-1}$, and Δ_{aux} is calculated between the current frame x_t and the original previous frame x_{t-1}. Then we fuse Δ_{aux} based on Δ_{main} to make up for the missing structure information in Δ_{main} due to the unclear pixel in $\hat{x}_{t-1}$. The fusion module has the same structure as Fig. 3 but with the different number of channels to be consistent with the channel of optical flow, setting to 2.

Context enhancement. At the decoder side, we propose to use the previous frame feature $\hat{f}_{t-1}$ to enhance the initial context. Similarly,

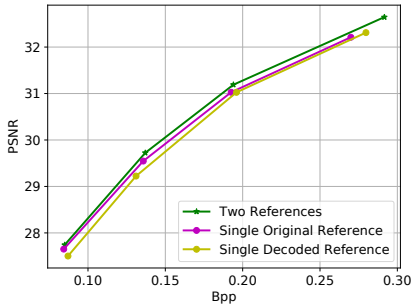

Figure 7: Effectiveness of different reference frames on HEVC Class D dataset. *Zoom in for best view.*

we also apply the same structure as shown in Fig. 3 to enhance the initial context $\tilde{c}_t$ at both the spatial and channel levels to form the enhanced context $\bar{c}_t$. Then $\bar{c}_t$ will be used to guide the contextual encoder, contextual decoder and entropy coding.

Other details. More details can be obtained from the Appendix, as Sec. 4.1 mentioned.

4.3 Experimental results

Settings of competitors. In order to verify the effectiveness of our proposed method, we compare the test results of our model with the traditional method H.265 [36] and the baselines FVC [19] and DCVC [23]. For H.265, we use the same instructions in [46] to execute the FFmpeg X265 compression software with *medium* mode. For FVC, for fair comparison and in consideration of complexity, we implemented their model without multi-frame feature fusion module, termed as FVC*. As for intra-frame coding, we directly use the existing deep image compression model cheng-2020-anchor [12] for MSE loss and hyperprior [4] for MS-SSIM loss provided by CompressAI [6], and the quality levels corresponding to four λ values are set to 3, 4, 5 and 6 respectively. Following the previous methods [19, 29], we set the GOP size of HEVC, UVG and MCL-JCV datasets to 10, 12 and 12 respectively. For consistency, the tested frame number of HEVC datasets is 100 (10 GOPs), and the tested frame number of MCL-JCV and UVG datasets are equal to the total number of frames in the whole sequence respectively.

Results. Fig. 6 shows the rate-distortion curves of these methods on the HEVC, UVG and MCL-JCV datasets. We can find that the baselines FVC* and DCVC can get further notable improvement through adding our proposed method for all bit-rate ranges in terms of PSNR and MS-SSIM, particular on the HEVC Class E dataset. Table 1 gives the BDBR results when compared with H.265 [36] in terms of PSNR. Result shows that FVC* can only achieve about 4.76% gains in PSNR metrics on these datasets, but if it is incorporated with our proposed method, about 14.75% gains can be obtained. As for DCVC, it can get 26.82% gains without our method, but 30.41% when our method is incorporated. It means that our proposed method is beneficial to the compression scheme in feature space as an agile plug-in unit. It should be noted that the experimental results show that the gain of our proposed method on FVC* is greater than that on DCVC. This is because that the compression process of FVC* is based on prediction-then-residual, which leads to a strong dependence on the accuracy of predicted features, but DCVC is based on prediction-then-condition, which is not particularly dependent on the predicted features. It is also shown in [37], the temporal context comes directly from the feature of the decoded previous frame without any prediction operation, but the performance is still satisfied.

Table 2: Ablation study of the attention mechanism on the HEVC Class D dataset in terms of PSNR. *1rt* and *2nd* represent the highly-fitting motion field fusion module and prediction enhancement module respectively. *Spatial* and *Channel* represent the different branches in the modules.

Spatial (1rt)	Channel (1rt)	Spatial (2nd)	Channel (2nd)	Bit-rate increase
✓	✓	✓	✓	0.0%
✓	✓	✗	✗	4.33%
✓	✗	✓	✗	3.58%
✗	✓	✗	✓	4.29%
✗	✗	✗	✗	6.34%

4.4 Ablation study

Unless otherwise specified, the following ablation experiments are performed on FVC*.

Effectiveness of different reference frames. As shown in Fig. 7, we test the effectiveness of using different reference frames on the HEVC Class D dataset. The first case, termed as *Single Original Reference*, uses the original previous frame X_{t-1} for motion estimation at the encoder side, but uses the decoded previous frame $\hat{X}_{t-1}$ for motion compensation at the decoder side. In the second case, termed as *Single Decoded Reference*, $\hat{X}_{t-1}$ is used as reference at both of encoder and decoder sides. Our scheme is termed as *Two References*, where both $\hat{X}_{t-1}$ and X_{t-1} are used for motion estimation at the encoder side but only $\hat{X}_{t-1}$ is used for motion compensation at the decoder side. It can be seen from Fig. 7, the performance of *Two References* outperforms the other two methods. This point proves that because the ideal motion field (Single Original Reference) is not consistent with the decoded previous frame, the prediction of the current frame feature is not the best. So we call the fused motion field in Eq. (4) as the highly-fitting motion field which combines two frames and also fits the decoded previous frame. From the experiment, it can also be observed that *Single Original Reference* is indeed better than *Single Decoded Reference* because it uses more information in the original frame X_{t-1}.

Effectiveness of different components and branches. As discussed in Sec. 3.1 and Sec. 3.2, the similar attention mechanism is applied to motion field fusion module and prediction enhancement module respectively. To verify the role of spatial and channel level attention branches in the two modules, we test the effect of using different branches on the results. As shown in Table. 2, the "$1rt$" and "$2nd$" represent the motion field fusion module and prediction enhancement module respectively. The symbols "✓" and "✗" mean the corresponding branch is used or not used respectively. The second row indicates that the performance will drop if we only fuse the motion field but do not enhance the predicted feature. It is because that the highly-fitting motion field although can shift the feature to the exact location, but it dose not rescue the missing

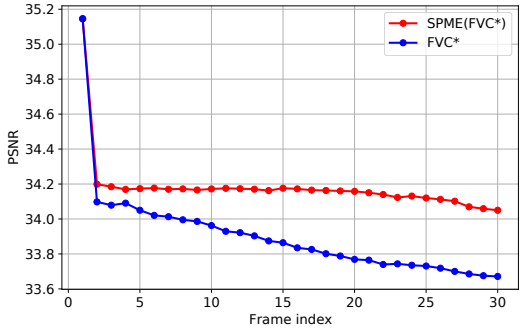

Figure 8: Analysis of error propagation on UVG dataset with $\lambda = 2048$. The horizontal and longitudinal axises represent the frame index in a GOP and the PSNR of the corresponding frame respectively. The first frame is the I frame, and the other frames are P frames.

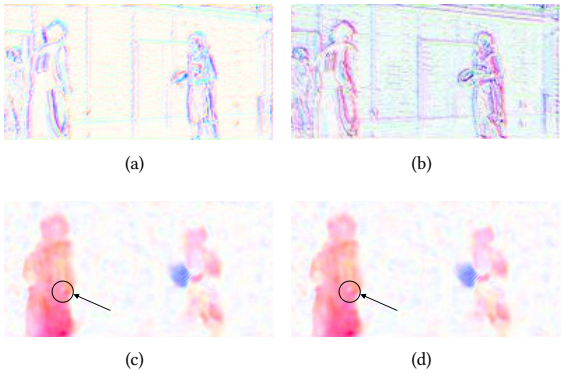

Figure 9: Visualization of different motion fields. (b) has more details than (a) at the edges of the objects. (d) is clearer than (c) in the marked area. *Zoom in for best view.*

information in the decoded previous frame, an enhancement module is necessary to address this. The third and fourth rows indicate that 3.58% bit-rate will increase if the channel level branches in the two modules are disabled, 4.29% bit-rate will increase if the spatial level branches are disabled. This means that the spatial attention is more important than channel attention. This is reasonable because we rely on spatial attention mechanism to obtain complementary information from the decoded previous frame to enhance the predicted current frame feature. The last row demonstrates that the optimal effect can be achieved only by the joint all branches.

Analysis of error propagation. As discussed above, the previous feature prediction based compression method used the decoded previous frame as a reference. Due to the phenomenon of error propagation, the frames in the rear of a GOP will suffer from serious distortion. If we continue to use the frame with severe distortion as reference, this situation emerges that no clear corresponding pixel or feature can be found in the reference frame for motion estimation, so that the motion information is with structural missing. For verifying the role of the original previous frame when compressing the current frame, we compare the quality of each frame in a random GOP in the UVG dataset with FVC* [19]. In order to fully prove that our proposed method is more useful for slowing down error propagation, we enlarge the GOP size to 30. As shown in Fig. 8, we can find that our proposed method can restrain the phenomenon of error propagation to a certain extent, particularly for the frames in the rear of a GOP. While FVC* has constant attenuation when the GOP size is increasing.

Visualization of different motion fields. As shown in Fig. 9, in order to further verify the effectiveness of our proposed method, we visualize the motion field to be transmitted (named θ_t in FVC* based model and m_t in DCVC based model) between the ninth frame and the tenth frame in the *BasketballPass* sequence from the HEVC Class D dataset. (a) is the motion field from FVC* [19], and (b) is the corresponding highly-fitting motion field. After motion fusion, highly-fitting motion field contains more detailed information, especially at the edges of the objects. It is well known that the edges of objects are prone to pixel details losing, which is difficult to exactly estimate the motion at these areas. However, our proposed method can well inhibit the occurrence and spread of this situation because the useful and clear information in x_{t-1} is considered. (c) is the motion field from DCVC, also called optical flow, and (d) is the corresponding highly-fitting motion field after motion fusion. It can be found that the quality of optical flow has also been improved after fusion, but the amount of change is not as much as FVC* based model. This dues to two reasons. DCVC uses only two channels to represent the motion information, and to be consistent with DCVC, our SPME does not change this, making the representation of motion information less rich than FVC. Thus the enhanced motion estimation is difficult to keep both the real motion information and the motion information based on the decoded frames. Another point is that as shown in Table 1, the quality of the decoded previous frame by DCVC is better than that of FVC*. However, it is worth mentioning that even with such limited conditions, our method still improves the performance of DCVC.

5 CONCLUSION

This paper investigates exploring the rich information in the original previous frame at the encoder side to make up for the disadvantage of using only the decoded previous frame as the reference for motion estimation. Specifically, a structure-preserving motion estimation network is proposed to use the original previous frame as auxiliary data to enhance the motion estimation between the current frame and the decoded previous frame to form a highly-fitting motion field. It keeps the spatial structure and teporal coherency of the motion information and the corresponding prediction residual in order to be better coded with CNNs while considering the prediction efficiency. Then a motion compensation and prediction enhancement module is developed to predict and enhance the current frame feature. Experimental results showed that our method further saves 9.99% bit-rate over the baseline FVC*, and the phenomenon of error propagation has also been alleviated to a certain extent. Furthermore, there are also gains when we transplant our proposed method to the other feature prediction based video compression framework, which indicates that our approach can be widely used as a plug-in unit.

REFERENCES

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

[2] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. 2019. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 221–231.

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

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

[5] Jean Bégaint, Franck Galpin, Philippe Guillotel, and Christine Guillemot. 2019. Deep frame interpolation for video compression. In *DCC 2019-Data Compression Conference*. IEEE, 1–10.

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

[7] Fabrice Bellard. 2016. BPG image format (2014). *URL http://bellard. org/bpg/.[Online, Accessed 2016-08-05]* 1, 2 (2016).

[8] Gisle Bjontegaard. 2001. Calculation of average PSNR differences between RD-curves. *VCEG-M33* (2001).

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

[10] Chunlei Cai, Li Chen, Xiaoyun Zhang, and Zhiyong Gao. 2019. End-to-end optimized ROI image compression. *IEEE Transactions on Image Processing* 29 (2019), 3442–3457.

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. 2019. Learning image and video compression through spatial-temporal energy compaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10071–10080.

[12] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. 2020. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7939–7948.

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

[14] Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers. 2019. Neural inter-frame compression for video coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6421–6429.

[15] Runsen Feng, Zongyu Guo, Zhizheng Zhang, and Zhibo Chen. 2021. Versatile Learned Video Compression. *arXiv preprint arXiv:2111.03386* (2021).

[16] Runsen Feng, Yaojun Wu, Zongyu Guo, Zhizheng Zhang, and Zhibo Chen. 2020. Learned video compression with feature-level residuals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 120–121.

[17] Zongyu Guo, Runsen Feng, Zhizheng Zhang, Xin Jin, and Zhibo Chen. 2021. Learning Cross-Scale Prediction for Efficient Neural Video Compression. *arXiv preprint arXiv:2112.13309* (2021).

[18] Amirhossein Habibian, Ties van Rozendaal, Jakub M Tomczak, and Taco S Cohen. 2019. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7033–7042.

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

[20] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. 2018. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4385–4393.

[21] Théo Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier Déforges. 2020. Optical flow and mode selection for learning-based video coding. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 1–6.

[22] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. 2018. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452* (2018).

[23] Jiahao Li, Bin Li, and Yan Lu. 2021. Deep contextual video compression. *Advances in Neural Information Processing Systems* 34 (2021).

[24] Mu Li, Kede Ma, Jane You, David Zhang, and Wangmeng Zuo. 2020. Efficient and effective context-based convolutional entropy modeling for image compression. *IEEE Transactions on Image Processing* 29 (2020), 5900–5911.

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

[26] Haojie Liu, Ming Lu, Zhan Ma, Fan Wang, Zhihuang Xie, Xun Cao, and Yao Wang. 2020. Neural video coding using multiscale motion compensation and spatiotemporal context model. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 8 (2020), 3182–3196.

[27] Jerry Liu, Shenlong Wang, Wei-Chiu Ma, Meet Shah, Rui Hu, Pranaab Dhawan, and Raquel Urtasun. 2020. Conditional entropy coding for efficient video compression. In *European Conference on Computer Vision*. Springer, 453–468.

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

[29] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. 2020. An end-to-end learning framework for video compression. *IEEE transactions on pattern analysis and machine intelligence* 43, 10 (2020), 3292–3308.

[30] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. 2018. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4394–4402.

[31] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. 2020. UVG dataset: 50/120fps 4K sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*. 297–302.

[32] David Minnen, Johannes Ballé, and George D Toderici. 2018. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems* 31 (2018).

[33] Anurag Ranjan and Michael J Black. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4161–4170.

[34] Oren Rippel, Alexander G Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir Bourdev. 2021. Elf-vc: Efficient learned flexible-rate video coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14479–14488.

[35] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. 2021. Temporal Context Mining for Learned Video Compression. *arXiv preprint arXiv:2111.13850* (2021).

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

[37] Zhenhong Sun, Zhiyu Tan, Xiuyu Sun, Fangyi Zhang, Dongyang Li, Yichen Qian, and Hao Li. 2021. Spatiotemporal Entropy Model is All You Need for Learned Video Compression. *arXiv preprint arXiv:2104.06083* (2021).

[38] David S Taubman and Michael W Marcellin. 2002. JPEG2000: Standard for interactive imaging. *Proc. IEEE* 90, 8 (2002), 1336–1357.

[39] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. 2017. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395* (2017).

[40] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. 2015. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085* (2015).

[41] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. 2017. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 5306–5314.

[42] Gregory K Wallace. 1992. The JPEG still picture compression standard. *IEEE transactions on consumer electronics* 38, 1 (1992), xviii–xxxiv.

[43] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C-C Jay Kuo. 2016. MCL-JCV: a JND-based H. 264/AVC video quality assessment dataset. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1509–1513.

[44] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. Ieee, 1398–1402.

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

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

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

[48] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. 2020. Learning for video compression with recurrent auto-encoder and recurrent probability model. *IEEE Journal of Selected Topics in Signal Processing* 15, 2 (2020), 388–401.