SPATIOTEMPORAL ENTROPY MODEL IS ALL YOU NEED FOR LEARNED VIDEO COMPRESSION

Zhenhong Sun, Zhiyu Tan, Xiuyu Sun, Fangyi Zhang, Dongyang Li, Yichen Qian, Hao Li
Alibaba Group, China
{zhenhong.szh, zhiyu.tzy, xiuyu.sxy, zhiyuan.zfy, yingtian.ldy, yichen.qyc, lihao.lh}@alibaba-inc.com

ABSTRACT

The framework of dominant learned video compression methods is usually composed of motion prediction modules as well as motion vector and residual image compression modules, suffering from its complex structure and error propagation problem. Approaches have been proposed to reduce the complexity by replacing motion prediction modules with implicit flow networks. Error propagation aware training strategy is also proposed to alleviate incremental reconstruction errors from previously decoded frames. Although these methods have brought some improvement, little attention has been paid to the framework itself. Inspired by the success of learned image compression through simplifying the framework with a single deep neural network, it is natural to expect a better performance in video compression via a simple yet appropriate framework. Therefore, we propose a framework to directly compress raw-pixel frames (rather than residual images), where no extra motion prediction module is required. Instead, an entropy model is used to estimate the spatiotemporal redundancy in a latent space rather than pixel level, which significantly reduces the complexity of the framework. Specifically, the whole framework is a compression module, consisting of a unified auto-encoder which produces identically distributed latents for all frames, and a spatiotemporal entropy estimation model to minimize the entropy of these latents. Experiments showed that the proposed method outperforms state-of-the-art (SOTA) performance under the metric of multiscale structural similarity (MS-SSIM) and achieves competitive results under the metric of PSNR.

1 Introduction

In the field of video compression, eliminating temporal redundancy between adjacent frames is a key challenge to improve compression performance [38, 30, 23]. Traditional video standards such as H.264 [38], HEVC [30] and VVC [23] use well hand-designed modules (e.g., block motion estimation, motion vector and residual image transform) to reduce the redundancy in video sequences.

Under the traditional video coding schemes, some successful attempts [20, 19, 11, 42, 17, 18, 4] for learned video compression replace each module with a learning-based one. Specially, the recursive block-based motion prediction is replaced by optical flow with warping operation to generate predictive frames, and the modules to compress residual errors and motion vector are replaced by two auto-encoders styled compression networks. To further exploit the temporal redundancy, multi-frame prediction [11, 42, 17] and implicit optical flow methods [18, 4] are proposed to improve the performance of motion prediction.

The aforementioned learned video compression methods have shown competitive performance to traditional codecs, but they still suffer from the following three inherent problems caused by their framework:

- Due to the mechanism of using a previous reconstructed frame as reference, error propagation problem is common in the methods based on the traditional framework, for no matter codecs or learned methods [19];

*Corresponding author.
• The accuracy of a predicted frame may decrease due to errors from motion prediction or propagated from earlier frames, resulting in residual errors which will then increase the difficulty of residual compression;
• Training and inference with the current framework are computationally expensive due to its complexity.

Therefore, it comes to an open question: Does learned video compression method really need such a complex framework? At first glance, the answer seems to be of course YES. Obviously, the advance of this framework has been proved over the last decades. However, we have a different view in this paper. Our inspiration comes from the development of learned image compression: some learning-based methods achieved more remarkable performance with only a simple yet powerful entropy model [5, 32, 6, 22] without following the advanced image codec schemes. Following this phenomenon, it is natural to expect that a similar simple framework (e.g., an appropriate spatiotemporal entropy model) may be also good enough for learned video compression.

To verify this intuition, latent representations from a unified image compression model are visualized for adjacent frames, as shown in Fig. 1 from which we can observe obvious differences highly correlated to temporal redundancy. Motivated by this, we propose a motion-free video compression (MFVC) framework. To be specific, we view frames in a video sequence as independent pictures in the encoding process and use an encoder network to project each frame into quantized latent representations respectively, similar to that in learned image compression methods. Based on these latent representations, a spatial prior module (SPM) is then used inside frames, and a temporal prior module (TPM) is used over adjacent frames, jointly providing accurate entropy for a common arithmetic encoding. In the decoding process, after arithmetic decoding, the lossless quantized latents are projected back to image pixels. There exists no lossy reference-frame dependence, no motion prediction, and no motion and residual compression in the whole scheme.

In particular, this paper has three major contributions:

Figure 1: Correlation between latent representations. The first two rows show the heatmaps of the key channel (channel 44) in latent representations of different frames. The last row shows the differential results of the front two rows. Two adjacent frames in (a) are from the ShakeNDry in UVG Dataset, showing high correlations. Two separated frames in (b) are from the Beauty in UVG Dataset, showing fewer correlations.
• To the best of our knowledge, the motion-free video compression framework is the first learned video compression framework without motion prediction modules while achieving SOTA performance.

• We propose the spatiotemporal entropy model for video frame latent representation compression, which consists of joint hyper-prior encoder-decoder (P-HE/P-HD), spatial prior module, and temporal prior module.

• Experimental results reveal that the motion-free video compression framework achieves SOTA performance under the MS-SSIM and firstly realizes the variable-rate control in a single video compression model.

2 Related Work

2.1 Learned Image Compression

Various handcrafted-based image compression standards have been proposed over the past decades, such as [36, 26, 7]. The newest hand-designed image compression methods are developed from video standards. With block partitioning, intra prediction, residual compression, and other modules, they require about half the storage space as the equivalent quality JPEG (including fixed block partition, DCT/IDCT transform, and Huffman Coding).

Recently, some learning-based lossy image compression approaches have been proposed to achieve competitive performance [33, 34, 32, 5, 6, 13, 22, 15, 9, 16]. They all utilize auto-encoder style networks to transform pixels to quantized latent representations and then project back to the pixel space. Indeed, a jointly optimized entropy model is verified to make the latent presentations more suitable to be losslessly compressed to a bitstream based on entropy coding. Some works [5, 34, 16] try to improve the representation ability of the auto-encoder architecture. Ballé et al. [5] bring in the generalized divisive normalization (GDN) transform with optimized parameters to efficiently Gaussianize the local joint statistics of natural images. Toderci et al. [34] introduce a new gated recurrent unit (GRU) inspired by the residual network. Cheng et al. [9] introduce deep residual attention modules to improve the performance.

More works [5, 6, 34, 22, 15, 25] pay attention to the entropy network and obtain obvious improvements. Ballé et al. [5] use a fully factorized prior to minimize the entropy of the elements of the whole latent representation. An improved version [6] is proposed to use a hierarchical learned prior network to occupy the fact that spatially neighboring elements of the latent representation tend to vary together in their scales. Furthermore, context-adaptive based models [34, 22, 15, 16] are introduced to utilize a neural network like PixelCNN [35] to incorporate predictions from neighboring symbols, avoiding storing additional bits. This kind of method outperforms the top standard image codec (BPG) on both the PSNR and MS-SSIM distortion metrics. After that, Qian [25] builds up a global relevance throughout latent features to further explore the relationship in the whole picture. All these works show that designing an accurate entropy model is crucial or even the only thing for learned image compression.

Last but not the least, rate control in learning-based methods is different from the codec schemes too. Early learning-based methods always need to train different models for different reconstruction quality. Toderci et al. [34] use recurrent neural networks to recursively compress residual information similar to the recursive search block in video codecs, encoding the latents of each iteration by a binary representation. Choi et al. [10] propose a conditional convolution to make rate control easier in auto-encoder style networks without incremental inferences.

2.2 Learned Video Compression

Thanks to the success of learned image compression, more attempts are verified to see where learning can obtain gain in video compression [40, 12, 24, 20, 19, 11, 13, 42, 17, 4]. DVC [20] is proposed by replacing each module for the traditional codec as a CNN network. All of these networks are trained in an end-to-end manner and optimized by PSNR or MS-SSIM loss functions. It can be seen as a “deep” version of traditional video coding scheme: an optical flow network (pre-trained spynet [27]) is utilized to predict motion between a frame and its previous compressed frame; a motion compensation network is applied over warped frames as a learning-based loop filter; finally, motion and residual information are compressed by two auto-encoder style compression networks. Bidirectional motion prediction method [11] and multi-frames based unidirectional one [17] obtain advantages over DVC in general. Hierarchical learned video compression (HLVC) [42] extends these methods by combining unidirectional and bi-directional compression with a weighted recurrent quality enhancement network. These methods are all based on the traditional video coding scheme which has some inherent problems: error propagation owing to the use of lossy frames (from lossy reconstruction) as references, inefficient residual compression caused by inaccurate motion prediction, too complicated networks to convergence due to the extra calculations and complicated modules in pre-trained optical flow models.

To tackle these problems, some incremental solutions are introduced. Lu et al. [19] firstly point out the error propagation problem in learned video compression. They design a joint training strategy to train the video codec by using the information from different time steps in one video clip and combines all the information to optimize the learned codec
Figure 2: Operational diagrams of motion-based video compression framework (a) and the proposed framework (b). MEM: Motion Entropy Model, ME: Motion Estimation (e.g., optical flow), Em: Motion Encoder, Dm: Motion Decoder, MC: Motion Compensation (e.g., wrapping operation and post-processing network), REM: Residual Entropy Model, Er: Residual Encoder, Dr: Residual Decoder, E: Encoder, D: Decoder, STEM: Spatiotemporal Entropy Model. The motion prediction consists of ME and MC. x\text{t}, \hat{x}_t, \hat{x}_{t-1}, and \hat{x}_t represent a current frame, predictive frame, previous reconstructive frame, current reconstructive frame, respectively. \hat{y}_t and p(\hat{y}_t) represent the quantitative latent and the probability distribution, respectively.

for better video compression performance. This method can alleviate but not avoid the error propagation problem. In order to reduce the failure cases (e.g., the disocclusion and fast motion cases) derived from the optical flow and bilinear warping operation, a scale-space flow with scale-space warping operation is introduced in [4], which achieves the SOTA performance among other learned methods. Implicit optical flow based methods [18, 4] use an encoder network to directly aggregate motion information instead of a pre-trained flow network, reducing the cost of model storage, computation, and training.

Some researchers try to not use the "traditional" learned video compression framework. Wu et al. [40] view video compression as an image interpolation problem and propose to interpolate target frame from references in a hierarchical manner, where the residual of the interpolation is then compressed via a RNN-based image compression network. Newest works [12, 24] try to directly capture spatiotemporal relationships using a 3D auto-encoder and autoregressive prior model. These methods outperform H264, but worse than HEVC and other learned methods.

The 3D convolution idea provides new thinking on video compression, i.e., a simple framework like that for learned image compression might be good enough for learned video compression, but building a unified spatiotemporal transform is much more complex and challenging than the spatial one, particularly with the 3D convolution which requires multiple frames as inputs. In addition, too computational expensive models like the 3D auto-encoder are also not suitable for low-delay scenarios. To this end, in this paper, we propose a motion-free video compression with a spatial transform and a spatiotemporal entropy model. The spatial transform keeps the advantages coming from learned image compression, and the spatiotemporal entropy model eliminates temporal redundancy in a lossless manner between frames without breaking the transformed representation.

3 Proposed Method

3.1 Motion-based Video Compression Framework

3.1.1 Complexity

The mainstream framework of learned video compression consists of many networks: motion prediction network, motion vector compression network, and residual image compression network, as shown in Fig.2. Through the motion estimation network (generally, optical flow network), motion vector mv is predicted between previous reconstructed frame \hat{x}_{t-1} and current frame \hat{x}_t, and then it will be compressed with motion vector compression network. After that,
the predictive frame $\tilde{x}_t$ is generated based on the reconstructed motion vector $\hat{m}uv$ with warping operation from previous reconstructed frame $\hat{x}_{t-1}$. Finally, residual error $x_r$ between $\tilde{x}_t$ and $x_t$ is compressed with residual image compression network. Additionally, each compression network is along with the entropy model networks to efficiently model the latent representations $\hat{y}$ for more accurate entropy minimization.

In the encoding phase, motion decoding, motion compensating, and residual decoding are required to compute the previous reconstructed frame $\hat{x}_{t-1}$, which results in a very high computational cost. Although the explicit optical flow network is omitted in [18, 4] to reduce the model storage and computation, the framework with two compression networks and entropy models networks is still complicated for practical applications, considering the model storage, computation, and variable-rate control.

3.1.2 Error Propagation problem

Error propagation between the reconstructed frames is a common problem in both traditional and learned video compression methods [19]. To investigate details of the error propagation, multiple frames are compressed by two methods (i.e., H.264 and flow-based learned video compression). The PSNR values of the reconstructed frames are demonstrated in Fig. 8, and the values of these methods are high at the beginning of the group of picture [39] (GOP). While, as the GOP size increases, the PSNR values begin to drop from 35.24 dB to 31.05 dB for H.264 and from 34.41 dB to 31.324 dB for the flow-based method, which indicates the error propagation problem causes frames quality to degrade.

Due to the distortion of the predictive frame, which is caused by warped from the previous reconstructed frame, it will generate high-frequency residual regions. Thus, the compression of high-frequency regions is difficult for a residual compression network. Therefore, if the codec does not increase the bitstream for the residual, the error will propagate to the subsequent frames, and PSNR values will drop continuously with the increasing time steps. If increasing the bitstream continuously, the RD performance will degrade and the rate control will lose efficacy. Although inserting high-quality frames in traditional video compression and utilizing the error propagation aware strategy in learned video compression [19] help to alleviate the distortion, the error propagation cannot be completely evitable for the motion-based video compression framework. Instead, the proposed motion-free video compression framework maintains the same quality level along with the frames (i.e., Proposed in Fig. 8), and the new framework will be explained in the next subsection.

3.2 Motion-free Video Compression Framework

As shown in Fig. 1 the ShakeNDry sequences show that a dog throwing water on its body in Fig. 1(a) showing high correlations between two adjacent frames, and the Beauty sequences show that a lady slowly turns her head and wriggles her hair in Fig. 1(b) showing fewer correlations between two separated frames. Base on the observation, the temporal redundancy between frames can be reduced in latent space by minimizing the entropy. To this end, the motion-free video compression framework is proposed. A unified auto-encoder model is used for mapping intra-frames and inter-frames...
A practical diagram of the MFVC framework is displayed in Fig. 4, and this structure is mainly designed for low-delay scenarios. The latent distribution: $\hat{y}_{t-1}$ is included into the current P-frame ($x_t$), and the hyper latent $\hat{y}_t$ is obtained from the previous frame ($x_{t-1}$) compression, which puts the $\hat{y}_{t-1}$ in the buffer. In the decoding of $x_t$, $\hat{y}_{t-1}$ is decoded from $Bits_{y_{t-1}}$ by AD.

The probability distributions of other frames are all estimated by the spatiotemporal entropy model. In this work, the loss function of the spatiotemporal entropy model:

$P-HD$, $SPM$ and $TPM$. Similar to [43], the probability distribution $p(y)$ is obtained from the previous frame ($x_{t-1}$) as inputs. The P-HE network takes $\hat{y}_t$ and $\hat{y}_{t-1}$ as inputs to generate the hyper latent $\hat{z}_t$, the SPM network uses PixelCNN structure [35] to capture the spatial priors between the neighbors in $\hat{y}_{t-1}$, and the TPM network uses three convolutions with Leaky ReLU to extract temporal priors from the previous latent $\hat{y}_{t-1}$. To generate the probability distribution parameters $(\mu_t, \sigma_t)$, the EPM module is introduced to fuse the outputs of P-HD, SPM and TPM. Similar to [43], the probability $p(y)$ of quantized latent $\hat{y}$ is modeled as Laplacian distribution:

$p(y) = \prod_{i=1}^{L} \left( \frac{1}{\sqrt{2\sigma_i}} \right) e^{-\frac{|y_i|}{\sigma_i}}$.

For the hyper latent $\hat{z}_t$, a set of channel-wise trainable parameters $(\mu_{z_t}, \sigma_{z_t})$ are defined to represent the Laplacian distribution:

$p(z_t) = \prod_{i=1}^{L} \left( \frac{1}{\sqrt{2\sigma_{z_t}}} \right) e^{-\frac{|z_{t,i}|}{\sigma_{z_t}}}$.

The latent $\hat{y}_{t-1}$ and the hyper latent $\hat{z}_t$ are both compressed by AE and AD, the cost of transmitting $\hat{z}_t$ is included into the loss function of the spatiotemporal entropy model:

$L_P = R_{\hat{y}_{t-1}} + R_{\hat{z}_t}$

$= \mathbb{E}_{x \sim p_x} \left[ -\log_2 p_{\hat{y}_{t-1}}(\hat{y}_{t-1}) \right] + \mathbb{E}_{x \sim p_x} \left[ -\log_2 p_{z_t}(\hat{z}_t) \right]$.

Figure 4: MFVC Framework for low-delay scenarios. Q: Quantization, SPM: Spatial Prior Module, TPM: Temporal Prior Module, EPM: Entropy Parameter Module, P-HE: P-frame Hyper-prior Encoder, P-HD: P-frame Hyper-prior Decoder, AE/AD: Arithmetic Encoding/Decoding. The dashed line indicates the AE/AD. The blue line indicates the compaction is conducted using three prior modules (a set of channel-wise trainable parameters), the SPM network uses PixelCNN structure [35] to capture the spatial priors between the neighbors in $\hat{y}_{t-1}$, and the TPM network uses three convolutions with Leaky ReLU to extract temporal priors from the previous latent $\hat{y}_{t-1}$. To generate the probability distribution parameters $(\mu_t, \sigma_t)$, the EPM module is introduced to fuse the outputs of P-HD, SPM and TPM. Similar to [43], the probability $p(y)$ of quantized latent $\hat{y}$ is modeled as Laplacian distribution:

$p(y) = \prod_{i=1}^{L} \left( \frac{1}{\sqrt{2\sigma_i}} \right) e^{-\frac{|y_i|}{\sigma_i}}$.

For the hyper latent $\hat{z}_t$, a set of channel-wise trainable parameters $(\mu_{z_t}, \sigma_{z_t})$ are defined to represent the Laplacian distribution:

$p(z_t) = \prod_{i=1}^{L} \left( \frac{1}{\sqrt{2\sigma_{z_t}}} \right) e^{-\frac{|z_{t,i}|}{\sigma_{z_t}}}$.

The latent $\hat{y}_{t-1}$ and the hyper latent $\hat{z}_t$ are both compressed by AE and AD, the cost of transmitting $\hat{z}_t$ is included into the loss function of the spatiotemporal entropy model:

$L_P = R_{\hat{y}_{t-1}} + R_{\hat{z}_t}$

$= \mathbb{E}_{x \sim p_x} \left[ -\log_2 p_{\hat{y}_{t-1}}(\hat{y}_{t-1}) \right] + \mathbb{E}_{x \sim p_x} \left[ -\log_2 p_{z_t}(\hat{z}_t) \right]$.

(3)
Table 1: Details of each module in the P-frame compression network. The "Conv" prefix corresponds to convolutional layers followed by the kernel size, the number of output channels, and downsampling stride (e.g., the first layer of the P-HE uses $5 \times 5$ kernels with 256 channels and a stride of one). The "Deconv" prefix corresponds to upsampled convolutions (i.e., in TensorFlow, tf.conv2d_transpose). The "Masked" prefix corresponds to a masked convolution as in [35].

Since the quality of the reconstructed frame is only determined by the unified auto-encoder model, the reconstructed distortion of the P-frame is left out of the loss function $L_{P}$, simplifying the training process. Based on the MFVC framework, it does not require motion compensating, motion and residual decoding in the encoding process, which reduces the storage and computation of the model.

### 3.3 Variable Rate Control

In learned image compression methods, optimizing the network with different Lagrange multiplier ($\lambda$) can trade off the distortion against the rate $L_{I}$ and the learning goal of these methods can be formulated as:

\[
L_{I} = R + \lambda D
\]

with $\lambda \in \{\lambda_{1}, \lambda_{2}, ..., \lambda_{n-1}, \lambda_{n}\}$, \hspace{1cm} (4)

where $\lambda$ corresponds to $n$ different compression rates, $R$ is referred to as the expected length of the compressed bitstream, and $D$ is the expected distortion of the reconstructed image concerning the original image measured by either mean squared error (MSE) or MS-SSIM. Drawing lessons from [10], conditional convolutions are added into the autoencoder network [22] for the variable-rate control in the unified auto-encoder model. Apart from the input frame, the unified auto-encoder model also takes $\lambda$ as input to enable the conditional convolutions and produces a compressed image with varying rates.

Benefitting from transferring the temporal compaction from the pixel space to the latent space, the loss function $L_{P}$ only concentrates on the compressed bitstream of the latent $\hat{y}_{t \rightarrow e}$ and the hyper latent $\hat{z}_{t}$. When training the spatiotemporal entropy model, the latents $\hat{y}_{t}$ and $\hat{y}_{t-1}$ are produced by the fixed unified auto-encoder model using randomly selected $\lambda$ from the set $\{\lambda_{1}, \lambda_{2}, ..., \lambda_{n-1}, \lambda_{n}\}$. Based on this strategy, the P-frame compression can achieve the variable-rate control directly without any other operations.

### 4 Experiments

#### 4.1 Implementation Details

**Details For Datasets.** The unified auto-encoder model was trained as I-frame compression on 30K color PNG images with high resolution, scraped from Flickr [3], and CLIC training dataset [1]. The Vimeo-90k dataset [41] is a widely used dataset for low-level vision tasks and has been applied to learned video compression tasks recently, so we used it for the P-frame training. To evaluate the compression performance and compare it with other methods, we employed three datasets with the resolution of 1080P (1920x1080). HEVC common test sequences [30] are the most popular test sequences for evaluating video compression performance, in which the contents are diversified and challenging. For a regular comparison with [20], we used Class B (1080P) in experiments. Ultra Video Group (UVG) dataset [21] is composed of 16 versatile test video sequences captured at 120 fps, in which the motion between adjacent frames is small. MCL-JVC dataset [37] consists of 30 movie clips with a variety of scenarios, which contains less noise and has been used for video quality assessment [38].

**Details For Training.** In the I-frame training of the unified auto-encoder model, the network was optimized with a batch size of 8 and a patch size of 256 $\times$ 256 randomly extracted from the training dataset. Two quality metrics (i.e., MSE and MS-SSIM) were used in the unified auto-encoder model. In particular, Adam [14] was used with multistage learning rates ($\{1e-4, 5e-5, 1e-5, 5e-6, 1e-6\}$) that changed with iteration boundaries ($\{1600000, 2100000, 2300000, 2400000, 2500000\}$). The value of Lagrange multiplier $\lambda$ was chosen from $\{50, 105, 160, 300, 480, 710, 1000, 1780, 2915\}$ for MSE and $\{3, 5, 8, 14, 20, 35, 52, 98, 145\}$ for MS-SSIM.

In the P-frame training period, we extracted 5 interval fragments from each video sequence of Vimeo-90k, with 7 consecutive frames in each fragment. To search for different motions, the first frame was fixed as the reference frame.
and the current frame was randomly selected from six subsequent frames. The spatiotemporal entropy model was optimized with a batch size of 16 and a patch size of $256 \times 512$ randomly extracted 2 frames from the 7-frame fragment. Adam was also used with multistage learning rates ($\{1e^{-4}, 5e^{-5}, 1e^{-5}, 5e^{-6}, 1e^{-6}\}$) that changed with iteration boundaries ($\{1200000, 1600000, 1800000, 1900000, 1950000\}$). The training was conducted in a distributed way with 4 Tesla V100, taking 42 hours for the unified auto-encoder model and 32 hours for the spatiotemporal entropy model.

**Details For Inference.** We evaluated the compression performance on three video test datasets. To evaluate the RD performance, the rate was measured by bits per pixel (BPP), and the quality was measured by either PSNR or MS-SSIM. The RD curves are drawn to demonstrate their coding efficiency in Fig[5]. To compare the compression performance fairly, we followed the settings in [40] [20] [42] [4]. 100 frames were compressed with a GOP size of 10 for HEVC Class B sequences, and all frames were compressed with a GOP size of 12 for the UVG dataset and MCL-JCV dataset. **FFmpeg** [4] was used to evaluate the performance of H. 264 and H. 265 in ‘very fast’ mode with exact settings exhibited in Appendix.

### 4.2 Performance Comparison

**RD Performance.** Fig.[5] demonstrates the RD performance of different methods on three datasets, using PSNR and MS-SSIM metrics respectively. The proposed model with 9 variable rates is compared with well-known compression standards (e.g., H. 264 [38] and H. 265 [30]) and learned video compression methods (e.g., Wu’s [40], DVC [20], HLVC [42], Agustsson’s [4]). RD curves of Wu’s, DVC, and HLVC are from the release in their GitHub, and RD curves of Agustsson’s and Habibian’s are obtained from the authors. Methods of Wu’s and HLVC are designed for B-frame video compression, which is generally better than the P-frame compression because more reference frames are used. Fig.[5] illustrates that the proposed model performs significantly better than other methods under the metric of MS-SSIM, saving 11.10%, 24.52%, 31.43%, 37.02%, 55.86% bits (bits-saving is all measured by BDBR [8]) compared with Agustsson’s, HLVC, Habibian’s, DVC and Wu’s on the UVG dataset, respectively.

In terms of PSNR, the RD performance of the proposed model is better than that of the other learned methods on UVG/MCL-JCV datasets and is competitive to that of H.265. The lower-right figure in Fig.[5] reveals that the RD performance of the proposed model is slightly worse than that of other learned methods on HEVC Class B dataset in terms of PSNR.

The performance of the proposed model under the metric of MS-SSIM is better than that of PSNR, which might be explained that the temporal compaction is conducted in the latent space instead of the pixel space, and more attentions
are paid to structure information rather than pixel information. Meanwhile, because HEVC Class B dataset has more background noise than the other datasets, the proposed model could not recover the noise to meet the metric PSNR. Although there is some performance deficiency on the HEVC Class B in terms of PSNR, the proposed model is just a feasible attempt to the MFVC framework, which illustrates that the MFVC framework has more potential for optimization.

| Model       | Encoder FLOPs | Encoder Latency | Decoder FLOPs | Decoder Latency | UVG(BDBR) MS-SSIM | Class B (BDBR) MS-SSIM |
|-------------|---------------|-----------------|---------------|-----------------|--------------------|------------------------|
| DVC [20]    | 3074G         | 426ms           | 1434G         | 189ms           | -25.24%            | -56.87%                |
| HLVC [42]   | 2831G         | 376ms           | 1479G         | 184ms           | -49.00%            | -61.07%                |
| Proposed w/o SPM | 613G         | 104ms           | 1479G         | 121ms           | -55.70%            | -61.07%                |
| Proposed    | 643G         | 114ms           | 1509G         | 23s             | -60.29%            | -62.73%                |

Table 2: Floating-point performance on a Tesla V100 with an input p-frame of 1080P (i.e., resolution: 1920 × 1080 × 3) except for the decoding of the proposed. Due to the serial characteristic of the PixelCNN, the decoding of the proposed is conducted on an 8-core CPU. BDBR is measured by setting H.264 as the anchor. Note that the proposed method is in redundancy design, and the channels are more than 256. The channels of DVC and HLVC are 128 or less.

**Floating Point Performance.** We compared the floating-point performance between the proposed methods and the open-source methods (i.e., DVC, and HLVC-layer2-P). We tested the first P-frame compression for all three methods and the values are summarized in Table 2. The results reveal that the proposed w/o SPM saves 3-4 times of inference time in the encoding period and saves 30% inference time in the decoding period. Meanwhile, comparing the RD performance on the UVG dataset, the proposed w/o SPM saves 30.46% (DVC) and 6.7% (HLVC) bits in terms of MS-SSIM. Due to the redundancy design, the complexity of the network can be reduced to further speed up the inference in the future.

### 4.3 Ablation Study

![Figure 6: Ablation study on the UVG dataset. All variants are optimized by MSE and the test GOP size is set as 12 except for the Proposed (GOP1). w/o Residual: the proposed method removes the residual function and \( \hat{y}_t \) is directly used for the probability prediction in the spatiotemporal entropy model.](image)

To verify the effectiveness of each modular component in the proposed model, a set of ablation experiments were conducted on the UVG dataset and RD curves are drawn in Fig. 6. By setting the I-frame compression (GOP size of 1) as the anchor, the proposed model saves 37.71% bits while maintaining the same reconstructive quality. If \( \hat{y}_t \) is directly used for the probability prediction, the proposed model (w/o Residual) saves 36.43% bits. If the SPM and the TPM are not used respectively, the proposed model saves 30.76% bits (w/o SPM) and 21.32% bits (w/o TPM) accordingly. If SPM and TPM are both removed, the proposed model degrades to the I-frame compression and only saves 2.56% bits (w/o SPM & TPM). This reveals that SPM and TPM both contribute to accurate probability prediction, leading to a bits-saving of 6.95% (SPM), 16.39% (TPM), and 35.15% (Both) on the UVG dataset.

To verify the effectiveness of each modular component in the proposed model, a set of ablation experiments were conducted on the UVG dataset and RD curves are drawn in Fig. 6. By setting the I-frame compression (GOP size of 1) as the anchor, the proposed model saves 37.71% bits while maintaining the same reconstructive quality. If \( \hat{y}_t \) is directly used for the probability prediction, the proposed model (w/o Residual) saves 36.43% bits. If the SPM and the TPM are not used respectively, the proposed model saves 30.76% bits (w/o SPM) and 21.32% bits (w/o TPM) accordingly. If SPM and TPM are both removed, the proposed model degrades to the I-frame compression and only saves 2.56% bits (w/o SPM & TPM). This reveals that SPM and TPM both contribute to accurate probability prediction, leading to a bits-saving of 6.95% (SPM), 16.39% (TPM), and 35.15% (Both) on the UVG dataset.

Additionally, we visualized the entropy values to further distinguish the difference between the variants, and visualizations are displayed in Fig. 7. The motion information between adjacent frames is shown in Fig. 7(a). And the entropy values of the proposed w/o SPM & TPM, w/o SPM, and the proposed are displayed in Fig. 7(b), (c), and (d), respectively. Through observing Fig. 7(b), (c), and their differential results Fig. 7(e), the TPM can reduce more entropy.
in motion area (mainly on fluttering hair) due to exploiting the temporal redundancy. The differential results (Fig. 7(f)) between Fig. 7(d) and (c) demonstrate the SPM can save some bits on the outline and background by referring to neighborhood points in front of them.

5 Conclusion

In this paper, we mainly discuss an open challenge that whether we could achieve competitive performance with dominant learned video compression methods in a simple framework. Firstly, we compare the dominant framework of learned based methods and that of traditional video coding schemes. It is shown that the current learned one suffers from its complex structure and error propagation problems. Secondly, we discuss some efforts, e.g., error propagation aware training, and implicit flow network, to solve these inherent issues. Although the methods have brought some improvement, little attention has been paid to the framework itself. Finally, a learned image compression style framework is proposed in this paper, called motion-free video compression (MFVC). It utilizes a variable-rate auto-encoder and a spatiotemporal entropy model to outperform SOTA performance under the metric of MS-SSIM. Last but not the least, the MFVC indicates that we may attempt more advanced sequence technologies, e.g., ConvLSTMs [29], 3D Convolution, explicit/implicit optical flow, and transformers, on entropy model rather than on raw pixels. More experiments results are present in the Appendix.

References

[1] Challenge on learned image compression 2018. http://challenge.compression.cc/
[2] Ffmpeg. http://www.ffmpeg.org/
[3] Flickr. https://www.flickr.com/
[4] 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 Conference on Computer Vision and Pattern Recognition, pages 8503–8512, 2020.

[5] Johannes Balle, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In International Conference on Learning Representations, 2017.

[6] Johannes Balle, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In International Conference on Learning Representations, 2018.

[7] Fabrice Bellard. Bpg image format. https://bellard.org/bpg, 1, 2015.

[8] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. VCEG-M33, 2001.

[9] 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/CVF Conference on Computer Vision and Pattern Recognition, pages 7939–7948, 2020.

[10] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Variable rate deep image compression with a conditional autoencoder. In Proceedings of the IEEE International Conference on Computer Vision, pages 3146–3154, 2019.

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

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

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

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

[15] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. In International Conference on Learning Representations, 2019.

[16] Jooyoung Lee, Seunghyun Cho, and Munchurl Kim. A hybrid architecture of jointly learning image compression and quality enhancement with improved entropy minimization. arXiv preprint arXiv:1912.12817, 2019.

[17] 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.

[18] Haojie Liu, Lichao Huang, Ming Lu, Tong Chen, Zhan Ma, et al. Learned video compression via joint spatial-temporal correlation exploration. pages 11580–11587, 2020.

[19] Guo Lu, Chunlei Cai, Xiaoyun Zhang, Li Chen, Wanli Ouyang, Dong Xu, and Zhiyong Gao. Content adaptive and error propagation aware deep video compression. In Proceedings of the IEEE European Conference on Computer Vision, pages 1–17, 2020.

[20] 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 Conference on Computer Vision and Pattern Recognition, pages 11006–11015, 2019.

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

[22] David Minnen, Johannes Balle, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. In Advances in Neural Information Processing Systems, pages 10771–10780, 2018.

[23] Jeni Ranner Ohm and Gary J Sullivan. Versatile video coding—towards the next generation of video compression. In Picture Coding Symposium, volume 2018, 2018.

[24] Jorge Pessoa, Helena Aidos, Pedro Tomás, and Márió AT Figueiredo. End-to-end learning of video compression using spatio-temporal autoencoders. In 2020 IEEE Workshop on Signal Processing Systems (SiPS), pages 1–6. IEEE, 2020.

[25] Yichen Qian, Zhiyu Tan, Xiuyu Sun, Ming Lin, Dongyang Li, Zhenhong Sun, Hao Li, and Rong Jin. Learning accurate entropy enhancement with improved entropy minimization. arXiv preprint arXiv:2010.08321, 2020.

[26] Majid Rabbanie and Rajan Joshi. An overview of the jpeg 2000 still image compression standard. Signal processing: Image communication, 17(1):3–48, 2002.

[27] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4161–4170, 2017.

[28] Jorma Rissanen and Glen Langdon. Universal modeling and coding. IEEE Transactions on Information Theory, 27(1):12–23, 1981.

[29] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In Advances in neural information processing systems, pages 802–810, 2015.

[30] 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.

[31] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8934–8943, 2018.

[32] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. In International Conference on Learning Representations, 2017.

[33] George Toderici, Sean M O’Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. arXiv preprint arXiv:1511.06085, 2015.
[34] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5306–5314, 2017.

[35] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In Advances in neural information processing systems, pages 4790–4798, 2016.

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

[37] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C-C Jay Kuo. Mcl-jcv: A jnd-based h.264/avc video quality assessment dataset. In 2016 IEEE International Conference on Image Processing (ICIP), pages 1509–1513. IEEE, 2016.

[38] 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.

[39] Wikipedia. Group of pictures. https://en.wikipedia.org/wiki/Group_of_pictures.

[40] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In Proceedings of the European Conference on Computer Vision, pages 416–431, 2018.

[41] 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.

[42] 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 Conference on Computer Vision and Pattern Recognition, pages 6628–6637, 2020.

[43] Lei Zhou, Zhenhong Sun, Xiangji Wu, and Junmin Wu. End-to-end optimized image compression with attention mechanism. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, June 2019.
Table 3: BD-Rate Gains of Proposed, H. 265 [30], Wu’s [40], DVC [20], HLVC [42], Habibian’s [12], and Agustsson’s [4] against the H.264 [38]. Negative values in BDBR represent the bits saving. “/” represents that the method didn’t evaluate the RD performance on the dataset in that column.

A Appendix

A.1 Distortion of Motion Prediction

In the main body of the paper, we mentioned that dominant learned video compression methods suffer from three inherent problems caused by the motion-based video compression framework, and two problems (i.e., the complexity and error propagation problem) are already present in section 3.1. In this section, we will explain the second problem that the inaccuracy of a predictive frame may result in residual errors and then bring more difficulty to the residual compression in the motion-based video compression method. An accurate optical flow network (i.e., PWCnet [31]) was used as the motion estimation module in motion-based video compression framework. The compression process is evaluated on three different video sequences, e.g., Beauty and YachtRide in UVG, and BasketballDrive in HEVC Class B. The qualitative results are visualized in Fig 9, Fig 10 and Fig 11.

In these figures, (a) is the previous reconstructive frame and (b) is the current frame. (c) is the optical flow between (a) and (b). (d) is the predictive frame after motion compensating (i.e., warped from the previous reconstructive frame with the reconstructive optical flow). (e) is the differential results (5 times higher for visualization) between (a) and (b), showing the motion information between adjacent frames. (f) is the differential results (5 times higher for visualization) between (d) and (b), and (f) is the residual image for residual compression in motion-based video compression. The ellipse regions in (d) are distorted regions caused by inaccurate motion prediction. The corresponding regions in (e) and (f) reveal that the residual image to be compressed will have higher temporal redundancy than the direct differential results between the previous reconstructive frame and the current frame. This phenomenon proves that motion prediction may decrease the accuracy of a predictive frame.

A.2 Details for Error Propagation

Figure 8: PSNR values for multi-frames compression on HEVC Class B dataset. H. 264 is conducted at a fixed compression rate (i.e., quantization parameter of 27). Flow-based method is a typical learned video compression with optical flow and wrapping operation in a single rate model. These two methods are both based on the motion-based framework. Note that the BPP of the 0th frame on the H.265 curve is four times smaller than the actual value for intuitive visualization (from 0.82 to 0.21).
In the main body of the paper, figure 3 only displays the PSNR values of those three methods. In the section, more details about bits per pixel (BPP) are shown in Fig. 8. The BPP curves of H.264 and flow-based method remain stable as the frame number increases, the qualities of the reconstruction begin to drop from 35.24 dB to 31.05 dB for H.264 and from 34.41 dB to 31.324 dB for flow-based method, showing a big error propagation in a single compression rate. Additionally, the proposed motion-free video compression framework maintains the same quality level along with the frames (whether in a higher or lower bitrate).

Besides, the BPP of the P-frame is not many times lower than that of the I-frame in the proposed method due to no error propagation between the reconstructive frames. That means the proposed method does not need to sacrifice the reconstruction qualities to achieve better RD performance. Based on the proposed framework, more efforts will be paid to improve the spatiotemporal entropy model without considering the reconstruction quality in the future.

A.3 Performance Comparison Based on BDBR

To further compare the RD performance between different methods, the BD-rate gains measured by BDBR [8] by setting H.264 as the anchor and the results are summarized in Table 3. As consistent with the results of RD curves, Table 3 also illustrates that the proposed model performs significantly better than other methods under the metric of MS-SSIM, achieving the SOTA RD performance. In terms of PSNR, the RD performance of the proposed model is better than that of the other learned methods on UVG/MCL-JCV datasets and is competitive to that of H.265. The BDBR results also reveal that the RD performance of the proposed model is slightly worse than that of other learned methods on HEVC Class B dataset in terms of PSNR. Meanwhile, the proposed w/o SPM has a slight degradation compared with the proposed method, which is also stated in the ablation study.

A.4 Settings of H.264 and H.265

FFmpeg [2] was used to evaluate the performance of H.264 and H.265 in very fast mode and the exact commands as following:

**H.264 (very fast)**
```
ffmpeg -pix_fmt yuv420p -s WxH -r FPS -i Name.yuv -c:v libx264 -preset very fast -tune zerolatency -crf QP -g GOP -v 1 -bf 0 Name.mkv
```

**H.265 (very fast)**
```
ffmpeg -pix_fmt yuv420p -s WxH -r FPS -i Name.yuv -c:v libx265 -preset very fast -tune zerolatency -x265-params 'crf=QP:keyint=GOP:verbose=1:bframes=0' Name.mkv
```

where 'WxH' represents resolution, 'FPS' represents frames per second, 'Name.yuv' represents the input file, QP represents quantization parameter (i.e., compression rate), GOP represents group of pictures, and 'Name.mkv' represents the out file.

A.5 Future works

Based on the separation of the RD performance, more efforts should be paid to promote the unified auto-encoder network for the quality of the reconstruction and improve the spatiotemporal entropy network for less compressed bitstream. To further enhance the performance of the proposed framework, an interesting topic is to use multiple latents for the spatiotemporal entropy network, such as the ConvLSTMs [29] for temporal relationships, or B-frame compression. To make up the gap with traditional video codecs, such as H.264 and H.265, the input of learned video compression may be consistent with the traditional codecs, exploring the training methods in YUV color space. To be adaptive to the different video contents, some works need to be conducted to explore finetuning the Encoder in the proposed framework.

A.6 Subjective Comparision

In Fig 12 and Fig 13 we visualize the reconstructive frames taken from two different evaluation video sequences Beauty and YachtRide in UVG. Comparisons are conducted between the proposed method, H.264 and H.265. The experimental results verify that our proposed methods save more bits at the same reconstruction quality level.
Figure 9: Visualizations of the motion prediction process on the *Beauty* sequences in UVG dataset. (a) is the previous reconstructive frame. (b) is the current frame. (c) is the optical flow between (a) and (b). (d) is the predictive frame after motion compensating (i.e., warped from the previous reconstructive frame with the reconstructive optical flow). (e) is the differential results (5 times higher for visualization) between (a) and (b), showing the motion information between adjacent frames. (f) is the differential results (5 times higher for visualization) between (d) and (b), and (f) is the residual image for residual compression in the motion-based video compression. The ellipse regions in (d) are distorted regions caused by inaccurate motion prediction. The corresponding regions in (e) and (f) reveal that the residual image to be compressed will have higher temporal redundancy than the direct differential results between the previous reconstructive frame and the current frame.
Figure 10: Visualizations of the motion prediction process on the YachtRide sequences in UVG dataset. (a) is the previous reconstructive frame. (b) is the current frame. (c) is the optical flow between (a) and (b). (d) is the predictive frame after motion compensating (i.e., warped from the previous reconstructive frame with the reconstructive optical flow). (e) is the differential results (5 times higher for visualization) between (a) and (b), showing the motion information between adjacent frames. (f) is the differential results (5 times higher for visualization) between (d) and (b), and (f) is the residual image for residual compression in the motion-based video compression. The ellipse regions in (d) are distorted regions caused by inaccurate motion prediction. The corresponding regions in (e) and (f) reveal that the residual image to be compressed will have higher temporal redundancy than the direct differential results between the previous reconstructive frame and the current frame.
Figure 11: Visualizations of the motion prediction process on the BasketballDrive sequences in HEVC Class B dataset. (a) is the previous reconstructive frame. (b) is the current frame. (c) is the optical flow between (a) and (b). (d) is the predictive frame after motion compensating (i.e., warped from the previous reconstructive frame with the reconstructive optical flow). (e) is the differential results (5 times higher for visualization) between (a) and (b), showing the motion information between adjacent frames. (f) is the differential results (5 times higher for visualization) between (d) and (b), and (f) is the residual image for residual compression in the motion-based video compression. The ellipse regions in (d) are distorted regions caused by inaccurate motion prediction. The corresponding regions in (e) and (f) reveal that the residual image to be compressed will have higher temporal redundancy than the direct differential results between the previous reconstructive frame and the current frame.
Figure 12: Comparison between the proposed method, H.264 and H.265 on the Beauty sequences in UVG dataset. Our proposed methods save more bits at the same reconstruction quality level.

Figure 13: Comparison between the proposed method, H.264 and H.265 on the YachtRide sequences in UVG dataset. Our proposed methods save more bits at the same reconstruction quality level.