AlphaVC: High-Performance and Efficient Learned Video Compression

Yibo Shi, Yunying Ge, Jing Wang✉, and Jue Mao

Huawei Technologies, Beijing China
wangjing215@huawei.com

Abstract. Recently, learned video compression has drawn lots of attention and show a rapid development trend with promising results. However, the previous works still suffer from some criticial issues and have a performance gap with traditional compression standards in terms of widely used PSNR metric. In this paper, we propose several techniques to effectively improve the performance. First, to address the problem of accumulative error, we introduce a conditional-I-frame as the first frame in the GoP, which stabilizes the reconstructed quality and saves the bit-rate. Second, to efficiently improve the accuracy of inter prediction without increasing the complexity of decoder, we propose a pixel-to-feature motion prediction method at encoder side that helps us to obtain high-quality motion information. Third, we propose a probability-based entropy skipping method, which not only brings performance gain, but also greatly reduces the runtime of entropy coding. With these powerful techniques, this paper proposes AlphaVC, a high-performance and efficient learned video compression scheme. To the best of our knowledge, AlphaVC is the first E2E AI codec that exceeds the latest compression standard VVC on all common test datasets for both PSNR (-28.2% BD-rate saving) and MSSSIM (-52.2% BD-rate saving), and has very fast encoding (0.001x VVC) and decoding (1.69x VVC) speeds.

1 Introduction

Video data is reported to occupy more than 82% of all consumer Internet traffic [10], and is expected to see the rapid rate of growth in the next few years, especially the high-definition videos and ultra high-definition videos. Therefore, video compression is a key requirement for the bandwidth-limited Internet. During the past decades, several video coding standards were developed, such as H.264 [35], H.265 [29], and H.266 [7]. These methods are based on hand-designed modules such as block partition, inter prediction and transform [2], etc. While these traditional video compression methods have made a promising performance, their performance are limited since the modules are artificially designed and optimized separately.

Recently, learned image compression [8, 11, 15, 26] based on variational autoencoder [20] has shown great potential, achieving better performance than traditional image codecs [5, 7, 32]. Inspired by the learned image compression, and

combined with the idea of traditional video codecs, many learning-based video compression approaches [1, 14, 16, 17, 19, 21, 24, 27] were proposed.

Given the reference frame, variant kinds of motion compensation (alignment) methods were proposed like scale-space alignment [1], feature-based alignment [19], multi-scale feature-based alignment [28]. These methods aim to improve the diversity of motion compensation and result in more compression-friendly predictions. However, such methods increase the complexity on both encoder and decoder side. Inspired by AMVP (Advanced Motion Vector Prediction) on traditional video compression methods [29], we expect the encoder side to predict a more accurate motion information. Further, at the encoder side of AlphaVC, we propose a pixel-to-feature motion prediction method that can obtain high-quality motion information without increasing the complexity of the decoder.

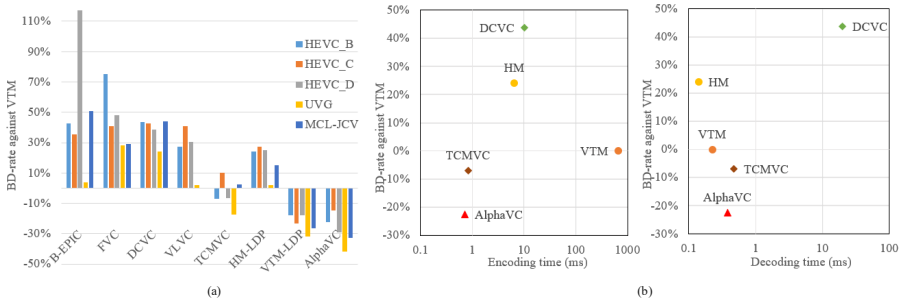

Fig. 1: (a): BD-rate against VTM in terms of PSNR (Lower is better). (b): BD-rate against VTM as a function of encoding/decoding time on 1080p videos.

Existing learned video compression can be divided into two categories: Low-Delay P mode and Low-Delay B/Random-Access mode. For the Low-Delay P mode, the methods [1, 16, 19, 28] only include the P(predictive)-frames and I(image)-frames. For the Low-Delay B or Random-Access mode, the methods [14, 27] insert the B(bidirectional predictive) frames into the GoP to improve compression performance. AlphaVC focuses on the Low-Delay P mode. In this mode, due to the accumulation error in P-frame [23], most existing methods have to use the inefficient I-frame as the first frame in limited length GoP. Unlike the existing methods, we overcome this issue by introducing a conditional I-frame (cI-frame) as the first frame in the GoP, which stabilizes the reconstructed quality and achieves better performance.

In addition, we all know that the entropy coding [13, 18] can only run serially will increase the runtime. Moreover, the auto-regressive entropy module [26], which significantly increase the decoding time, is always used on learned image codecs for a higher compression ratio. We found that most elements of the latents usually have very low information entropy, which means the probability distributions of these elements estimated by entropy module always is highly

concentrated. Inspired by this, we propose an efficient probability-based entropy skipping method (Skip) which can significantly save runtime in entropy coding, and achieve higher performance without auto-regressive.

With the help of the above technologies, AlphaVC achieves the highest E2E compression performance while being very efficient. As shown in Fig. 1, the proposed AlphaVC outperforms VTM-IPP/VTM-LDP by 28.2%/6.59% , where the VTM is the official software of H.266/VVC, the IPP denotes the configuration using one reference frame and flat QP, and the LDP denotes the better configuration using multiple references and dynamic QP. Note the configuration of AlphaVC is the same as IPP. To the best of our knowledge, AlphaVC is the only learning-based video codec that can consistently achieve comparable or better performance with VTM-LDP in terms of PSNR on all common test datasets. Comparing with the state-of-the-art learning-based video codecs [28], AlphaVC reduces the BD-rate by about 25% while faster encoding and decoding.

Our contributions are summarized as follows:

1. We introduce a new type of frame named conditional-I frame (cI-frame) and propose a new coding mode for learned video compression. It can effectively save the bit rate of I-frame and alleviate the problem of accumulated error.
2. The proposed motion prediction method, utilizing the idea of pixel-to-feature and global-to-local, can significantly improve the accuracy of inter-frame prediction without increasing decoding complexity.
3. An efficient method in entropy estiamtion module and entropy coding have higher performance and faster encoding and decoding time.

2 Related Work

2.1 Image Compression

In the past decades, the traditional image compression methods like JPEG [32], JPEG2000 [9] and BPG [5] can efficiently reduce the image size. Those methods have achieved a high performance by exploiting the hand-crafted techniques, such as DCT [2]. Recently, thanks to variational autoencoder (VAE) [20] and scalar quantization assumption [3], the learning-based image compression methods have achieved great progress. With the optimization of entropy estimation modules [4, 26] and network structure [8, 11], the learning-based image compression methods have achieved better performance than the traditional image compression codecs on common metrics, such as PSNR and MS-SSIM [34].

2.2 Video Compression

Video compression is a more challenging problem compared to image compression. There is a long history of progress for hand-designed video compression methods, and several video coding standards have been proposed, such as H.264(JM) [35], H.265(HM) [29] and more recently H.266(VTM) [7]. With the development of video coding standards, the traditional video compression

methods made significant improvements and provided a strong baseline. Even they have shown a good performance, these algorithms are limited by the hand-designed strategy and the difficult to optimize jointly.

Recently, learning-based video compression has become a new direction. Following the traditional video compression framework, Lu et al. proposed the end-to-end optimized video compression framework DVC [24], in which the neural networks are used to replace all the critical components in traditional video compression codec. Then, the exploration direction of existing approaches can be classified into three categories. One category of approaches focuses on the motion compensation (alignment) method to improve the accuracy of inter prediction. For example, SSF [1] designed a scale-space flow to replace the bilinear warping operation. Hu et al. [19] propose the FVC framework, which apply transformation in feature space with deformable convolution [12]. Later Sheng et al. introduce multi-scale in feature space transformation [28]. Another popular direction is the design of auto-encoder module. Such as Habibian et al. [17] use a 3D spatio-temporal autoencoder network to directly compress multiple frames. Li et al. [21] use the predicted frame as the input of encoder, decoder, instead of explicitly computing the residual. The third category extends the learned video compression to more codec functions, like B-frame [14, 27], utilizing multiple reference frames [19].

3 Method

3.1 Overview

Let $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots\}$ denote a video sequence, video codecs usually break the full sequence into groups of pictures (GoP). Due to the accumulative error of P-frames, in low delay P mode, which is AlphaVC adopted, each group needs to start with an I-frame and then follow P-frames. In AlphaVC, we propose a new codecing mode in GoP, including three types of frames. As shown in Fig. 2(a), the I-frame is only used for the first frame. For other groups, we propose to start with conditional-I-frame instead of I-frame. The Conditional-I-frame (named cI-frame), which uses the reference frame as condition of entropy to reduce the bit-rate, stabilises the reconstructed quality like I-frame, and meanwhile has a high compression rate. The details of each type of our P-frame and cI-frame are summarized as follows:

P-Frame First of all, we define the P-Frame in learned video compression as a class of methods that has the following form on decoder side:

$$\hat{\mathbf{X}}_t = D_p(H_{\text{align}}(\hat{\mathbf{X}}_{t-1}, \hat{\mathbf{m}}_t), \hat{\mathbf{r}}_t) \tag{1}$$

where $D_p(\cdot), H_{\text{align}}(\cdot)$ denote the method of reconstruction and alignment, $\hat{\mathbf{m}}_t, \hat{\mathbf{r}}_t$ are the quantized latent representation of motion, residual. Note that the quantized latent representation is the features to be encoded after the encoder and

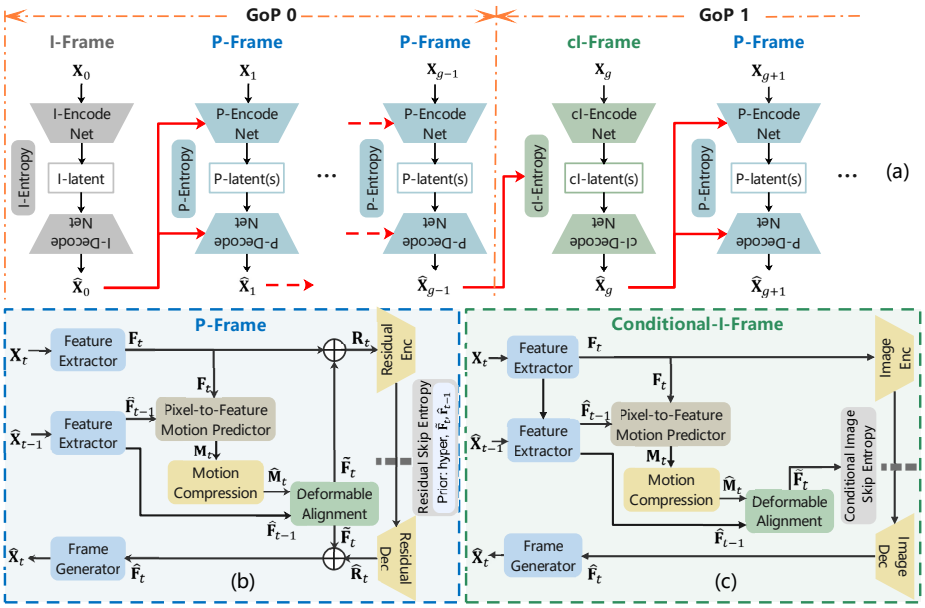

Fig. 2: Overview of our proposed video compression scheme. (a): Two kinds of GoP. (b): The framework of P-frame. (c): The framework of cI-frame.

quantization. That is, the reference frame $\hat{\mathbf{X}}_{t-1}$ will participate in and affect the reconstruction of current frame, which means that the consecutive P-frame will generate cumulative errors.

In this paper, we use the feature-align based P-frame framework, Fig. 2(b) sketches our P-frame compression framework. We first transform $\hat{\mathbf{X}}_{t-1}, \mathbf{X_t}$ into feature space $\hat{\mathbf{F}}_{t-1}$, $\mathbf{F_t}$. Then motion predictor will generate the predicted motion $\mathbf{M}_t$ and the predicted motion will be compressed by motion compression model. The predicted feature $\tilde{\mathbf{F}}_t$ is generated by deformable alignment [12] with the reconstructed motion $\hat{\mathbf{M}}_t$ and reference feature $\hat{\mathbf{F}}_{t-1}$. Finally, the residual in feature-based $\mathbf{R_t} = \mathbf{F_t} - \tilde{\mathbf{F}}_\mathbf{t}$ will be compressed by residual compression model. The reconstructed feature $\hat{\mathbf{F}}_t = \hat{\mathbf{R}}_t + \tilde{\mathbf{F}}_t$ is transformed into the current reconstruct frame $\hat{\mathbf{X}}_t$ with frame generator.

Both the motion compression model and residual compression model are implemented by auto-encoder structure [4], including an encoder module, decoder module and the proposed entropy estiamtion module. The newtork structure of auto-encoder part is the same as FVC [19]. To further reduce redundant information, we introduce the temporal and structure prior for the entropy estimation module in both motion and residual compression models:

$$\begin{aligned} &\mathbb{E}_{\hat{\mathbf{m}}_t \sim p_t}[-\log_2 q_t(\hat{\mathbf{m}}_t|\hat{\mathbf{F}}_{t-1}, \hat{\mathbf{m}}_{t-1})] \\ &\mathbb{E}_{\hat{\mathbf{r}}_t \sim p_t}[-\log_2 q_t(\hat{\mathbf{r}}_t|\tilde{\mathbf{F}}_t, \hat{\mathbf{r}}_{t-1})] \end{aligned} \tag{2}$$

the reference feature $\hat{\mathbf{F}}_{t-1}$ and previous quantized motion latent representation $\hat{\mathbf{m}}_{t-1}$ are structure and temporal priors of $\hat{\mathbf{m}}_t$ respectively, and the predicted feature $\tilde{\mathbf{F}}_t$ and previous quantized residual latent representation $\hat{\mathbf{r}}_{t-1}$ are structure and temporal priors of $\hat{\mathbf{r}}_t$ respectively.

Conditional-I-Frame (cI-frame) We introduce a new type of frame called the cI-frame like [22], which can be formulated as:

$$\begin{aligned} &\text{Auto-Encoder} : \hat{\mathbf{y}}_t = Q(E_{cI}(\mathbf{X_t})), \hat{\mathbf{X}}_t = D_{cI}(\hat{\mathbf{y}}_t), \\ &\text{Entropy} : R(\hat{\mathbf{y}}_t|\hat{\mathbf{X}}_{t-1}) = \mathbb{E}_{\hat{\mathbf{y}}_t \sim p_t}[-\log_2 q_t(\hat{\mathbf{y}}_t|H_{\text{align}}(\hat{\mathbf{X}}_{t-1}, \hat{\mathbf{m}}_t))], \end{aligned} \tag{3}$$

where $\hat{\mathbf{y}}_t$ is the quantized latent representation of $\mathbf{X_t}$, $E_{cI}(\cdot), Q(\cdot), D_{cI}(\cdot)$ denote the function of cI encoder module, quantization and reconstruction. That is, cI-frame reduces the inter redundant information through the entropy conditioned on $\hat{\mathbf{X}}_{t-1}$. For cI-frame, the input of the autoencoder does not use the reference frames, thus make the reconstructed quality stable. Further, we use cI-frame as the first frame in the GoP excluding the first GoP, which not only stabilizes the sequence quality like I-frame, but also improves the compression ratio, thereby alleviating the problem of accumulated errors.

The framework for cI-frame is shown in Fig. 2(c). The feature extractor, motion prediction and motion compression part share the same structure with P-frame framework. $\tilde{\mathbf{F}}_t$ is only used as the prior, the current feature $\mathbf{F_t}$ will be the only input of the encoder.

Furthermore, we propose two novel strategies in both P-frame and cI-frame, named pixel-to-feature motion prediction (P2F MP) and probability-based entropy skipping method (Skip), to improve the accuracy of inter prediction and coding efficiency.

3.2 Pixel-to-Feature Motion Prediction

Inter-frame prediction is a critical module to improve the efficiency of inter-frame coding, since it determines the accuracy of the predicted frame. We propose pixel-to-feature motion prediction to fully exploit the diversity of feature-based alignment and the state-of-the-art optical flow network. The illustration is shown in Fig. 3.

Given the previous reconstructed frame $\hat{\mathbf{X}}_{t-1}$ and the current frame $\mathbf{X}_t$, the optical flow in pixel space $\mathbf{M}_\mathbf{t}^{\text{pixel}}$ will be generated by a state-of-the-art optical flow network [30, 31]. The pixel space motion $\mathbf{M}_\mathbf{t}^{\text{pixel}}$ is then used to initialize a motion in feature space $\mathbf{M}_\mathbf{t}^{\text{init}}$. Then, we apply the deformable alignment $D(\cdot,\cdot)$ to the reference feature $\hat{\mathbf{F}}_{t-1}$ by $\mathbf{M}_\mathbf{t}^{\text{init}}$:

$$\bar{\mathbf{F}}_t = D(\hat{\mathbf{F}}_{t-1}, \mathbf{M}_\mathbf{t}^{\text{init}}) \tag{4}$$

After initial alignment, the motion local refinement network will refine the initial motion locally according to the initially aligned feature $\bar{\mathbf{F}}_t$ and the target feature $\mathbf{F_t}$, and then generate the final predicted motion $\mathbf{M_t}$.

$$\mathbf{M_t} = \text{Refine}(\bar{\mathbf{F}}_\mathbf{t}, \mathbf{F_t}) + \mathbf{M}_\mathbf{t}^{\text{init}} \tag{5}$$

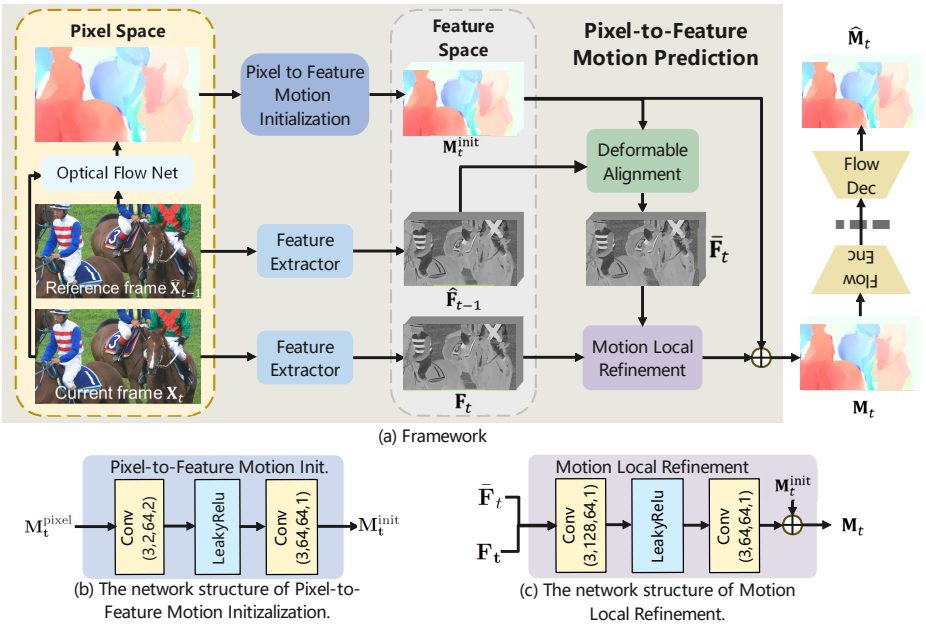

Fig. 3: Illustration of our proposed pixel-to-feature motion prediction module.

Finally, the predicted motion will be compressed to reconstruct motion $\hat{\mathbf{M}}_t$ through motion compression model.

Unlike existing methods, AlphaVC neither learn motion directly from features [19] that are difficult to fit through convolutions nor compress the generated optical flow directly [24]. We follow pixel-to-feature and global-to-local principles, first generate the feature space motion before coding with optical flow, then performing further fine-tuning through alignment feedback. Experiments show that this method greatly improves the accuracy of inter-frame prediction without affecting the decoding complexity and running time.

3.3 Probability-base Entropy Skipping Method

For a latent representation variable $\mathbf{v}$ in learned image or video compression, we first quantize it with round-based quantization $\hat{\mathbf{v}} = [\mathbf{v}]$, and estimate the probability distribution of $\mathbf{v}$ by an entropy estimation module with some priors, such as hyper [4], context [26], etc. Then $\hat{\mathbf{v}}$ is compressed into the bitstream by entropy coding like arithmetic coding [18], asymmetric numeral system [13]. In video compression, due to the introduction of the reference frame, the entropy of quantized latent representation variables like $\hat{\mathbf{m}}_t, \hat{\mathbf{r}}_t$ in P-frame is very small, especially in low bit-rate. That means the probability distributions of most elements in the latent variable are concentrated. If it is slightly off-center for such an element, we will encode it to bitstream with a high cost. In other words, if we skip these elements without encoding/decoding and replace them with the

peak of probability distribution, we can save both bit-rate and runtime of entropy coding with little error expectations. Inspired by this idea, we propose an efficient probability-based entropy skipping method (Skip).

For a latent representation variable $\mathbf{v}$, we define $\mathcal{Q}$ as the probability density set of $\mathbf{v}$ estimated by its entropy module. The value which has the maximum probability density of the i-th element is calculated as:

$$\theta_i = \arg\max_{\theta_i} q_i(\theta_i) \tag{6}$$

The probability that the element v_i is close to θ_i can be computed by:

$$q_i^{\max} = \int_{\theta_i - 0.5}^{\theta_i + 0.5} q_i(x)\, dx \tag{7}$$

If the probability $q_i^{\max}$ is high enough, we will not encode/decode the element to/from the bitstream, and replace the value with θ_i. After this operation, the quantized latent representation will become $\hat{\mathbf{v}}^{\mathrm{s}}$:

$$\hat{v_i}^{\mathrm{s}} = \begin{cases} \theta_i, & q_i^{\max} >= \tau \\ [v_i], & q_i^{\max} < \tau \end{cases} \tag{8}$$

where τ is a threshold to determine whether to skip.

In our paper, we use gaussian distribution as the estimated probability density of all the quantized latent representations. Hence the Eq. 6 and Eq. 7 can be easily solved as:

$$\theta_i = \mu_i, q_i^{\max} = \mathrm{erf}(\frac{1}{2\sqrt{2}\sigma_i}). \tag{9}$$

It can be seen that $q_i^{\max}$ is the monotone function of σ_i, we use σ_i as the condition of Eq. 8 to further reduce the computational complexity:

$$\hat{v_i}^{\mathrm{s}} = \begin{cases} \mu_i, & \sigma_i < \tau_\sigma \\ [v_i], & \sigma_i >= \tau_\sigma \end{cases} \tag{10}$$

There are two benefits of Skip. First, it can dynamically reduce the number of elements that need to be entropy encoded, significantly reducing the serial CPU runtime. Second, we can better trade-off errors and bit rates for elements with high determinism, thereby achieving high compression performance.

3.4 Loss Function

Our proposed AlphaVC targets to jointly optimize the rate-distortion (R-D) cost.

$$L = R + \lambda \cdot D = (R_0^{\mathrm{I}} + \lambda \cdot D_0^{\mathrm{I}}) + \sum_{t=1}^{\mathrm{T}-1} (R_t^{\mathrm{p}} + \lambda \cdot D_t^{\mathrm{p}}) + (R_{\mathrm{T}}^{\mathrm{cI}} + \lambda \cdot D_{\mathrm{T}}^{\mathrm{cI}}) \tag{11}$$

where the training GoP size is T, λ controls the trade-off, $R_0^{\mathrm{I}} - D_0^{\mathrm{I}}$, $R_t^{\mathrm{p}} - D_t^{\mathrm{p}}$ and $R_{\mathrm{T}}^{\mathrm{cI}} - D_{\mathrm{T}}^{\mathrm{cI}}$ represent the rate-distortion of the 0-th I-frame, the t-th P-frame and the T-th cI-frame, respectively.

4 Experiments

4.1 Setup

Training. We train our model on the Vimeo-90k dataset. This dataset consists of 4278 videos with 89800 independent shots that are different from each other in content. We randomly crop the frames to patches of size 256×256, and start training from scratch. We train the models with Adam optimizer for 60 epochs, where the batchsize was set to 8 and learning rate was initially set to $1e-4$ and reduced to half for 30 epochs. The skip operation will been enabled during training. The loss function is the joint rate-distortion loss as shown in Eq. 11, where the multiplier λ is chosen from (0.07, 0.05, 0.01, 0.005, 0.001, 0.0007) for the MSE optimization. The the MS-SSIM optimized models are finetuned from MSE-optimized model with $\lambda = 0.03$, 0.01, 0.007, 0.005, 0.001.

Testing. We evaluate our proposed algorithm on the HEVC datasets [6] (Class B,C,D,E), the UVG datasets [25], and the MCL-JCV datasets [33]. The HEVC datasets contain 16 videos with different resolution 416×240, 832×480 and 1920×1080. The UVG and MCL-JVC datasets contain 7 and 30 1080p videos, respectively. The GoP size in AlphaVC is set to 20 for all testing datasets.

Camparision. Both IPP and LDP configuration of VTM-10.0 and HM-16.20 are used for comparision. The IPP only references the previous frame, and each P-frame has the flat QP, which is the same configuration with AlphaVC. The LDP is the default low-delay P configuration that references multiple previous frames and has dynamic QP for each P-frame. In addition, state-of-the-art learning-based video compression methods, i.e., FVC (CVPR'21) [19], DCVC (NIPS'21) [21], B-EPIC (ICCV'21) [27], VLVC (2021) [14], TCMVC (2021) [28]. Note that, B-EPIC and VLVC don't belong to IPPP mode, due to the introduction of B-frame.

4.2 Experiment results

Performance Fig. 4, 5 shows the experimental results on all testing datasets. It is obvious that AlphaVC achieves the bset performance of all methods. In terms of MS-SSIM, AlphaVC significantly outperforms all the other methods over the entire bitrate range and on all the datasets. In terms of PSNR, AlphaVC significantly outperforms all the learning-based codecs and VTM-IPP, and even outperforms VTM-LDP in most situations. As mentioned before, VTM-LDP references multiple previous frames and has dynamic QP for each P-frame. which is not adopted by AlphaVC.

Table 1 and Table 2 show the BD-rate savings in PSNR and MS-SSIM that anchored by VTM-IPP. In terms of PSNR, AlphaVC achieves an average 28.2% bitrate saving compared to VTM-IPP, outperforming all the reported methods, including the stronger VTM-LDP (23.5% bitrate saving). In the worst case,

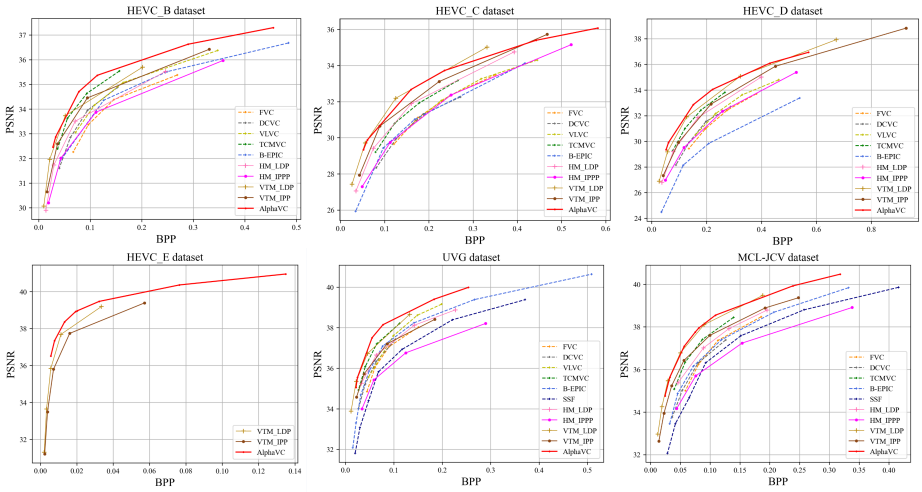

Fig. 4: PSNR based R-D Curves of traditional codecs and state-of-the-art learning-based codecs on each datasets. The red solid line is AlphaVC. Traditional codecs are all represented by solid lines, and other learning-based codecs are represented by dotted lines.

AlphaVC also achieves a BD-rate saving of 14.9% showing a good stability. In terms of MS-SSIM, learning-based codecs generally have better performances than traditional codecs, among with AlphaVC performing the best, by saving an additional 8% bitrate over the best SOTA TCMVC.

Table 1: BD-rate calculated by PSNR with the anchor of VTM-IPP. Red means more bits ($> 3\%$) required. Green means fewer bits ($< -3\%$) required.

	VTM-IPP	VTM-LDP	HM-IPP	HM-LDP	SSF	FVC	DCVC	VLVC	TCMVC	B-EPIC	AlphaVC
HEVC_B	0	-17.9%	55.2%	24.0%	-	75.4%	43.7%	27.1%	-6.92%	42.5%	**-22.5%**
HEVC_C	0	**-23.1%**	38.6%	27.1%	-	40.9%	42.8%	40.8%	10.2%	35.6%	-14.9%
HEVC_D	0	-17.9%	35.7%	24.9%	-	47.9%	38.6%	30.5%	-6.61%	117.%	**-29.0%**
UVG	0	-31.9%	18.5%	1.99%	57.7%	28.4%	24.0%	2.15%	-17.3%	3.78%	**-41.7%**
MCL-JCV	0	-26.6%	26.3%	15.2%	50.6%	29.3%	43.8%	-	2.32%	50.6%	**-32.9%**
Avg	0	-23.5%	35.6%	19.7%	54.2%	44.4%	38.6%	25.1%	-3.66%	49.9%	**-28.2%**

Complexity The MAC(Multiply Accumulate) of the P-frame at the decoding side is about 1.13M/pixel, and the cI-frame is about 0.98M/pixel. We use arithmetic coding for the complete entropy encoding and decoding process, and 1080p videos to evaluate the runtime. The runtime of the encoding side includes model inference, data transmission from GPU to CPU and entropy encoding, and the

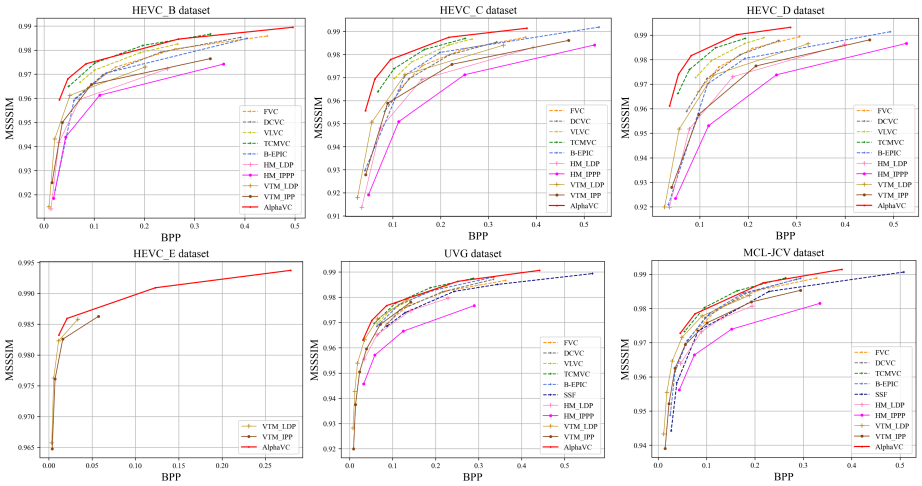

Fig. 5: MS-SSIM based R-D Curves.

Table 2: BD-rate calculated by MS-SSIM with the anchor of VTM-PVC-IPP. Red means more bits ($> 3\%$) required. Green means fewer bits ($< -3\%$) required.

	VTM-IPP	VTM-LDP	HM-IPP	HM-LDP	SSF	FVC	DCVC	VLVC	TCMVC	B-EPIC	AlphaVC
HEVC_B	0	-20.5%	54.6%	17.4%	-	-21.3%	-16.0%	-42.5%	-53.5%	-7.1%	**-61.6%**
HEVC_C	0	-20.7%	53.6%	12.8%	-	-22.2%	-12.8%	-41.6%	-47.6%	-15.4%	**-58.9%**
HEVC_D	0	-27.2%	39.3%	-1.5%	-	-34.7%	-33.0%	-49.6%	-60.7%	-21.5%	**-67.2%**
UVG	0	-26.7%	56.3%	20.2%	33.9%	11.5%	10.9%	-12.9%	-22.0%	-1.63%	**-32.9%**
MCL-JCV	0	-26.0%	49.6%	14.5%	-4.5%	-18.8%	-17.9%	-	-38.8%	-19.9%	**-40.5%**
Avg	0	-24.2%	49.9%	11.5%	14.7%	-17.1%	-13.7%	-36.6%	-44.5%	-13.1%	**-52.2%**

runtime of the decoding side includes entropy decoding, data transmission and model inference. The comparison results are shown in Table 3, in which running platform of AlphaVC is Intel(R) Xeon(R) Gold 6278C CPU and NVIDIA V100 GPU. The encoding and decoding times of AlphaVC on a 1080p frame average about 715ms and 379ms. The encoding time is about 1000x faster than VTM, and the decoding time is similar to VTM (1.69x). Even though AlphaVC uses more parameters than TCMVC, it is still faster. The main reason is the proposed probability-based skip entropy technique, which significantly reduces the running time on CPU. In addition, we can find that the cI-frame is slower than P-frame although the cI-frame has less complexity. This is also because the bit-rate in the cI-frame is higher, and the number of skipping elements in the cI-frame is fewer.

Table 3: Complexity on 1080p video. We compare our AlphaVC including cI-Frame and p-Frame with traditional codecs and TCMVC. The time ratio is calculated with the anchor of VTM.

Method	Params.	Enc-T (s)	Dec-T (s)	Enc-T ratio	Dec-T ratio
VTM-10.0-IPP	-	661.9	0.224	1.0000	1.0000
HM-16.40-IPP	-	26.47	0.140	0.0400	0.6250
TCMVC	10.7M	0.827	0.472	0.0012	2.1071
AlphaVC	63.7M	0.715	0.379	0.0011	1.6920
AlphaVC-cI	29.9M	0.733	0.580	0.0011	2.5893
AlphaVC-P	33.8M	0.685	0.365	0.0010	1.6295

4.3 Ablation Study and Analysis

Frame Analysis We use three types of frame in AlphaVC:I-frame, cI-frame and P-frame. To justify this approach and evaluate each type of frame, we train two additional models AlphaVC-P and AlphaVC-cI. AlphaVC-P only includes I-frame and P-frame, and the GoP size is the same with AlphaVC in the test phase. AlphaVC-cI only includes I-frame and cI-frame, and there is no group in AlphaVC-cI, I-frame is only used in the first frame and all subsequent frames are cI-frames. The R-D performance is shown in Fig. 6(a), AlphaVC-P achieves comparable performance with VTM_IPP, and AlphaVC-cI only achieves comparable performance with HM_IPP. The reason may be that cI-frame utilizes reference frames in a more implicityly way: as the condition of entropy. The reason is that, although the cI-frame is not good enough, it is stable and has no accumulated error as shown in Fig. 6(b). By combining these two types of frame, AlphaVC achieves better R-D performance for the following two reasons:

1. The accumulated error of P-frame in AlphaVC is smaller than the P-frame in AlphaVC-P. (see in Fig. 6(b)).
2. The performance of cI-frame is much better than I-frame (see in Fig. 6, similar distortion with smaller rate).

Effectiveness of Different Components. We demonstrate the effectiveness of our proposed components with AlphaVC-P as the anchor. We gradually remove the P2F MP, Skip in $\hat{\mathbf{m}}$ and Skip in $\hat{\mathbf{r}}$ from AlphaVC-P. Note that, without P2F MP, the current feature and reference feature will be fed to the motion compression module directly. The BD-Rate savings against AlphaVC-P are presented in Table 4(b). Moreover, a more intuitive analysis for the proposed methods is shown in Fig. 7.

As shown in Table 4(b), P2F MP brings 10.4% BD-rate saving. From Fig. 7(b), we can see that the compressed motion with P2F MP is more accurate and with smaller entropy.

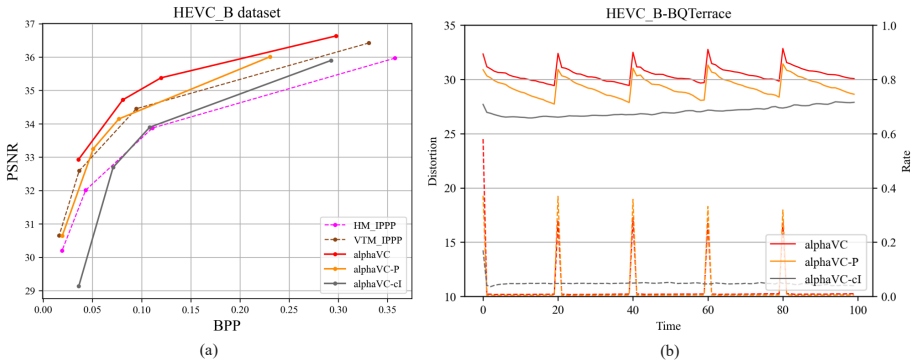

Fig. 6: Comparison with each type of frame in AlphaVC. AlphaVC-P only include P-frame and I-frame, the GoP size is 20 samed as AlphaVC. AlphaVC-cI only include cI-frame and I-frame, only the first frame uses the I-frame. (a): R-D performance of AlphaVC, AlphaVC-P and AlphaVC-cI under PSNR on HEVC class B dataset. (b): Example of performance comparison for each type of frame, the tested sequence is BQTerrace in class B. The solid line indicates the curve of distortion, the dashed line indicates the curve of rate.

Table 4: Effectiveness of our different components. The BD-rate values are computed under PSNR on HEVC class B dataset.

(a)

I-frame	✓	✓	✓
P-frame	✓	✓	
cI-frame	✓		✓
BD-Rate	0%	21.4%	92.7%

(b)

P2F MP	✓			
Skip in M.	✓	✓		
Skip in R.	✓	✓	✓	
BD-Rate	0%	10.4%	18.6%	37.5%

To analyze Skip, we first explore the relationship between the replacement error, and the variance of Gaussian distribution as shown in Fig. 7(c). Notice that the replacement error is highly correlated with variance, and elements with smaller variance have small errors. Therefore, skipping the entropy coding of these elements will not cause any loss, and may even improve performance. Due to the smoothness of motion information, the Skip ratio of motion latents is as high as 90% at each quality level as shown in Fig. 7(d), The Skip ratio of residual latents gradually increases (60% – 90%) with the decrease of quality. With the number of skipped elements increases, we can clearly see in Fig. 7(d) that the runtime of entropy coding on CPU is greatly reduced. In addition, as shown in Table 4(b), the probability-based skip entropy method can also improve performance obviously.

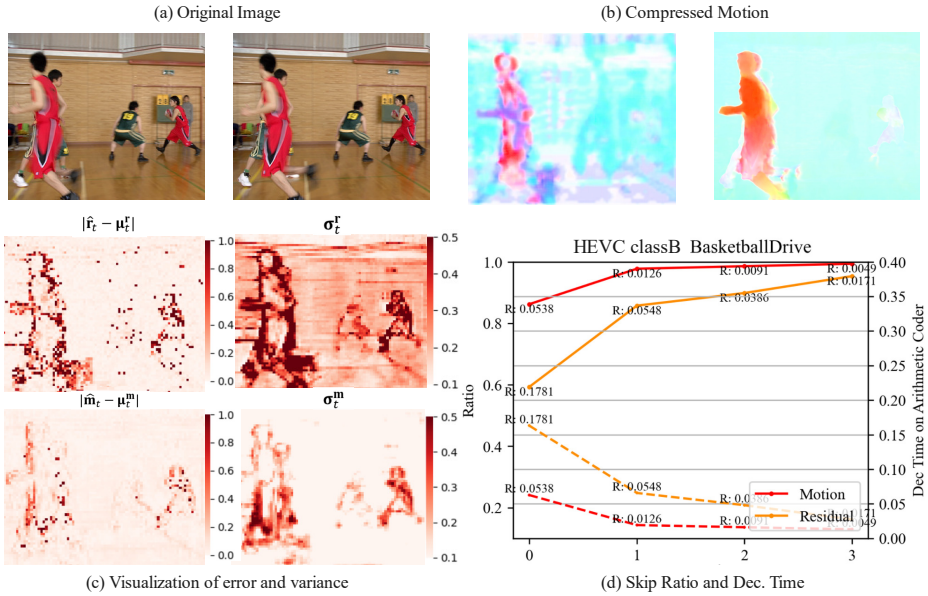

Fig. 7: Analysis of methods. (a): Two adjacent original frames of HEVC classB BasketballDrive. (b): Left/Right: The compressed motion wo/w our motion prediction module. (c): Visualization of variance of gaussian distortion σ and error after replacement. (d): Example result of the average skip ratio and arithmetic decoding time at 4 different bit rates, the ratio is calculated by skipped elements / total elements. The motion and residual latents are shown in the red and yellow curve, respectively. The solid and dotted curves represent ratio and time, respectively. The number on curves indicates bit-rate(BPP).

5 Conclusion

This paper proposed a high-performance and efficient learned video compression approach named AlphaVC. Specifically, we designed a new coding mode including three types of frame: I-frame, P-frame, and cI-frame, to reduce the bit rate of I-frame and mitigate the accumulative error. We then proposed two efficient techniques: P2F MP for improving the accuracy of inter-frame prediction at the encoder side, and Skip for reducing entropy and speeding up runtime. Experimental results show that AlphaVC outperforms H.266/VVC in terms of PSNR by 28% under the same configuration, meanwhile AlphaVC has the comparable decoding time compared with VTM. To the best of our knowledge, AlphaVC is the first learned video compression scheme achieving such a milestone result that outperforms VTM-IPP over the entire bitrate range and on all common test datasets.

We believe that our proposed AlphaVC provides some novel and useful techniques that can help researcheres to further develop the next generation video codecs with more powerful compression.

References

1. Agustsson, E., Minnen, D., Johnston, N., Balle, J., Hwang, S.J., Toderici, G.: Scale-space flow for end-to-end optimized video compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8503–8512 (2020)
2. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. IEEE transactions on Computers **100**(1), 90–93 (1974)
3. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimization of nonlinear transform codes for perceptual quality. In: 2016 Picture Coding Symposium (PCS). pp. 1–5. IEEE (2016)
4. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. arXiv preprint arXiv:1802.01436 (2018)
5. Bellard, F.: Bpg image format (2014). URL http://bellard. org/bpg/.[Online, Accessed 2016-08-05] **1**(2) (2016)
6. Bossen, F.: Common test conditions and software reference configurations, document jctvc-l1100. JCT-VC, San Jose, CA (2012)
7. Bross, B., Chen, J., Ohm, J.R., Sullivan, G.J., Wang, Y.K.: Developments in international video coding standardization after avc, with an overview of versatile video coding (vvc). Proceedings of the IEEE **109**(9), 1463–1493 (2021)
8. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7939–7948 (2020)
9. Christopoulos, C., Skodras, A., Ebrahimi, T.: The jpeg2000 still image coding system: an overview. IEEE transactions on consumer electronics **46**(4), 1103–1127 (2000)
10. Cisco: Cisco annual internet report (2018-2023) white paper. URL https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (2020)
11. Cui, Z., Wang, J., Gao, S., Guo, T., Feng, Y., Bai, B.: Asymmetric gained deep image compression with continuous rate adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10532–10541 (2021)
12. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
13. Duda, J.: Asymmetric numeral systems. arXiv preprint arXiv:0902.0271 (2009)
14. Feng, R., Guo, Z., Zhang, Z., Chen, Z.: Versatile learned video compression. arXiv preprint arXiv:2111.03386 (2021)
15. Guo, T., Wang, J., Cui, Z., Feng, Y., Ge, Y., Bai, B.: Variable rate image compression with content adaptive optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 122–123 (2020)
16. Guo, Z., Feng, R., Zhang, Z., Jin, X., Chen, Z.: Learning cross-scale prediction for efficient neural video compression. arXiv preprint arXiv:2112.13309 (2021)
17. Habibian, A., Rozendaal, T.v., Tomczak, J.M., Cohen, T.S.: Video compression with rate-distortion autoencoders. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7033–7042 (2019)

18. Howard, P.G., Vitter, J.S.: Arithmetic coding for data compression. Proceedings of the IEEE **82**(6), 857–865 (1994)
19. Hu, Z., Lu, G., Xu, D.: Fvc: A new framework towards deep video compression in feature space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1502–1511 (2021)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
21. Li, J., Li, B., Lu, Y.: Deep contextual video compression. Advances in Neural Information Processing Systems **34** (2021)
22. Liu, J., Wang, S., Ma, W.C., Shah, M., Hu, R., Dhawan, P., Urtasun, R.: Conditional entropy coding for efficient video compression. In: European Conference on Computer Vision. pp. 453–468. Springer (2020)
23. Lu, G., Cai, C., Zhang, X., Chen, L., Ouyang, W., Xu, D., Gao, Z.: Content adaptive and error propagation aware deep video compression. In: European Conference on Computer Vision. pp. 456–472. Springer (2020)
24. Lu, G., Zhang, X., Ouyang, W., Chen, L., Gao, Z., Xu, D.: An end-to-end learning framework for video compression. IEEE transactions on pattern analysis and machine intelligence **43**(10), 3292–3308 (2020)
25. Mercat, A., Viitanen, M., Vanne, J.: Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In: Proceedings of the 11th ACM Multimedia Systems Conference. pp. 297–302 (2020)
26. Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. Advances in neural information processing systems **31** (2018)
27. Pourreza, R., Cohen, T.: Extending neural p-frame codecs for b-frame coding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6680–6689 (2021)
28. Sheng, X., Li, J., Li, B., Li, L., Liu, D., Lu, Y.: Temporal context mining for learned video compression. arXiv preprint arXiv:2111.13850 (2021)
29. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (hevc) standard. IEEE Transactions on circuits and systems for video technology **22**(12), 1649–1668 (2012)
30. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
31. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
32. Wallace, G.K.: The jpeg still picture compression standard. IEEE transactions on consumer electronics **38**(1), xviii–xxxiv (1992)
33. Wang, H., Gan, W., Hu, S., Lin, J.Y., Jin, L., Song, L., Wang, P., Katsavounidis, I., Aaron, A., Kuo, C.C.J.: Mcl-jcv: a jnd-based h. 264/avc video quality assessment dataset. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 1509–1513. IEEE (2016)
34. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. vol. 2, pp. 1398–1402. Ieee (2003)
35. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the h. 264/avc video coding standard. IEEE Transactions on circuits and systems for video technology **13**(7), 560–576 (2003)