Content Adaptive and Error Propagation Aware Deep Video Compression

Guo Lu*[1,2], Chunlei Cai*[2], Xiaoyun Zhang[2], ✉ Li Chen[2], Wanli Ouyang[3], Dong Xu[3] and Zhiyong Gao[2]

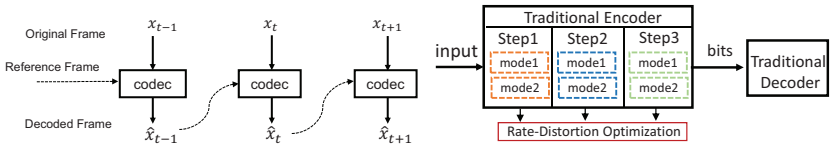
[1] Beijing Institute of Technology, China
[2] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China
[3] School of Electrical and Information Engineering, the University of Sydney, Australia

Abstract. Recently, learning based video compression methods attract increasing attention. However, the previous works suffer from error propagation due to the accumulation of reconstructed error in inter predictive coding. Meanwhile, the previous learning based video codecs are also not adaptive to different video contents. To address these two problems, we propose a content adaptive and error propagation aware video compression system. Specifically, our method employs a joint training strategy by considering the compression performance of multiple consecutive frames instead of a single frame. Based on the learned long-term temporal information, our approach effectively alleviates error propagation in reconstructed frames. More importantly, instead of using the hand-crafted coding modes in the traditional compression systems, we design an online encoder updating scheme in our system. The proposed approach updates the parameters for encoder according to the rate-distortion criterion but keeps the decoder unchanged in the inference stage. Therefore, the encoder is adaptive to different video contents and achieves better compression performance by reducing the domain gap between the training and testing datasets. Our method is simple yet effective and outperforms the state-of-the-art learning based video codecs on benchmark datasets without increasing the model size or decreasing the decoding speed.

1 Introduction

With the increasing amount of video content, it is a huge challenge to store and transmit videos. In literature, a large number of algorithms [39, 30] have been proposed to improve the video compression performance. However, all the traditional video compression algorithms [39, 30] depend on the hand-designed techniques and highly engineered modules without considering the power of end-to-end learning systems.

*First two authors contributed equally.
✉ Corresponding author: Li Chen (hilichen@sjtu.edu.cn).



(a) The error propagation issue in the video compression system.

(b) Adaptive mode selection in the traditional video compression system.

Fig. 1: Two motivations of our proposed method.

Recently, a few learning based image and video compression methods [8, 9, 24, 5, 23, 19, 40, 20] have been proposed. For example, Lu *et al.* [20] proposed an end-to-end video compression system by replacing all the key components in the traditional video compression methods with neural networks.

However, the current state-of-the-art learning based video compression algorithms [40, 20, 22] still have two drawbacks. First, the error propagation problem is not considered in the training procedure of learning based video compression systems. As shown in Fig.1(a), the previously decoded frame $\hat{x}_{t-1}$ in the coding procedure will be used as the reference frame to compress the current frame x_t. Since the video compression is a lossy procedure, the previously decoded frame $\hat{x}_{t-1}$ inevitably has reconstruction error, which will be propagated to the subsequent frames because of the inter-frame predictive coding scheme. As the encoding procedure continues, the error will be accumulated frame by frame, which will decrease the compression performance significantly. However, the current approaches [40, 20] train the codecs by only minimizing the distortion between the current frame x_t and the decoded frame $\hat{x}_t$, but ignore the influence of $\hat{x}_t$ on the subsequent encoding process for frame x_{t+1} and so on. Therefore, it is critical to build an error propagation aware training strategy for the deep video compression system.

Second, the current learning based encoders [40, 20] are not *adaptive* to different video content as the traditional codecs. As shown in Fig.1(b), the encoder in H.264 [39] or H.265 [30] selects different coding modes (*e.g.*, the size of coding unit) for videos with different contents. In contrast, once the training procedure is finished, the parameters in the learning based encoder are fixed, thus the encoder cannot adapt to different contents in videos and may not be optimal for the current video frame. Furthermore, considering the domain gap due to resolutions or motion magnitudes between the training and testing datasets, the learned encoder may achieve inferior performance for the videos with some specific contents, such as videos with complex motion scenes. To achieve content adaptive coding, it is necessary to update the encoder in the inference stage for the learning based video compression system.

In this paper, we propose a content adaptive and error propagation aware deep video compression method. Our method is a P-frame video compression method and is proposed for low-latency applications. Specifically, to alleviate error accumulation, the video compression system is optimized by minimizing the rate-distortion cost from multiple consecutive frames instead of that from a single frame only. This joint training strategy exploits the long-term information in

the coding procedure, therefore the learning based video codec not only achieves high compression performance for the current frame but also guarantees that the decoded current frame is also useful for the coding procedure of the subsequent frames. Furthermore, we propose an online encoder updating scheme to improve the video compression performance. Instead of using the hand-crafted modes in H.264/H.265, the parameters of the encoder will be updated based on the rate-distortion objective for *each* video frame. Our scheme enables adaption of the encoder according to different video contents while keeping the decoder unchanged. Experimental results demonstrate the superiority of the proposed method over the traditional codecs. Our approach is simple yet effective and outperforms the state-of-the-art method [20] without increasing the model size or computational complexity in the decoder side.

The contributions of our work can be summarized as follows,

1. An error propagation aware (EPA) training strategy is proposed by considering more temporal information to alleviate error accumulation for the learning based video compression system.
2. We achieve content adaptive video compression in the inference stage by allowing the online update of the video encoder.
3. The proposed method does not increase the model size or computational complexity of the decoder and outperforms the state-of-the-art learning based video codecs.

2 Related Work

2.1 Image Compression

Traditional image compression methods [35, 29, 1, 4] use hand-crafted techniques, such as discrete cosine transform (DCT)[7] and discrete wavelet transform [28] to reduce the spatial redundancy. Recently, learning based image compression approaches attracted increasing attention [32, 33, 8, 9, 31, 5, 19, 25, 23, 24, 14, 6]. In [8], a CNN based end-to-end image compression framework is proposed by considering both the rate and distortion terms. Furthermore, to obtain the accurate probability model of each symbol, Ballé *et al.* [9] estimate the hyperprior for the compressed features and improves the performance of entropy coding. Since the image compression methods rely on intra prediction, the error propagation reduction issue is not exploited in the existing image compression work.

2.2 Video Compression

The traditional video compression methods [39, 30] follow the classical block based hybrid coding framework, which uses motion-compensated prediction and transform coding. Although each module is well-designed, the traditional video compression systems cannot benefit from the power of deep neural networks.

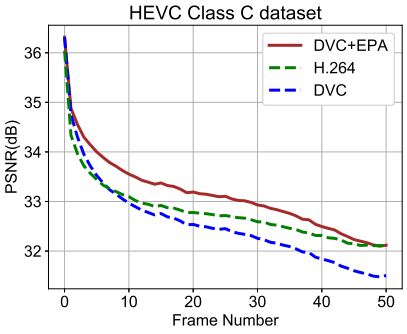


Fig. 2: The PSNR values of the reconstructed frames from different algorithms.

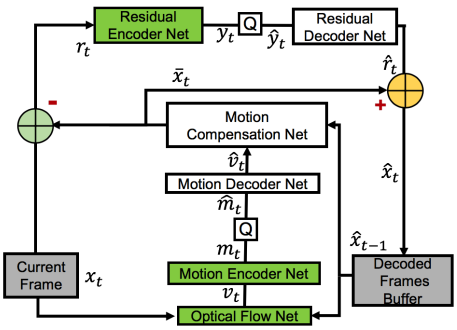


Fig. 3: The architecture of DVC in [20].

Recently, more and more end-to-end frameworks [12, 40, 20, 26, 16, 13, 34, 17] were proposed for video compression. In [40], the video compression task was formulated as frame interpolation, in which the motion information is compressed by using the traditional image compression method [4]. Lu *et al.* [20] proposed a fully end-to-end video compression system. Their approach follows the hybrid coding framework and uses neural networks to implement all components in video compression. In [26], an end-to-end video compression framework was proposed and the corresponding motion and residual information are jointly compressed. Habibian *et al.* [16] used a 3D auto-encoder to build a video compression framework and employed an auto-regressive prior as the entropy model. In [15], the residual information is computed in the latent space and the proposed framework can directly decode the motion and blending coefficients. It is worth mentioning that the methods [40, 15, 13] are based on frame interpolation and designed for B-frame compression.

We would like to highlight that the learning based video codecs in these methods [12, 40, 20, 26, 16, 13, 34, 15] are optimized by minimizing the distortion of a single frame without considering the error propagation problem for videos. More importantly, their encoders are not adaptive to different video contents. Although these methods achieve comparable or even better performance than H.264, we believe that the capability of the existing network architecture is not fully exploited, and the video compression performance can be further improved by using our proposed methods.

3 Motivations Related to Learning Based Video Compression System

3.1 Error Propagation

Error propagation is a common issue in the video compression systems, mainly due to the inter-prediction. In Fig. 2, we provide the PSNRs of the reconstructed frames from the H.264 algorithm [27] and the learning based video codec DVC

[20]. It is obvious that the PSNR drops when the time step increases. A possible explanation is that video compression is a lossy procedure and the encoding procedure of the current frame relies on the previous reconstructed frame, which is distorted and thus the error propagates to the subsequent frames. Let us take the DVC model as an example. The PSNR value of the 5^{th} reconstructed frame is 33.52dB, while the PSNR value of the 6^{th} reconstructed frame is 33.37dB(0.15dB drop). Furthermore, as the time step increases, the PSNR of the 50^{th} reconstructed frame is only 31.50dB.

Although error propagation is inevitable for such a predictive coding framework, it is possible and beneficial to alleviate the error propagation issue and further improve the compression performance (see the curve DVC+EPA in Fig. 2).

3.2 The Content Adaptive Coding Scheme

To improve the compression performance, the traditional video encoders [39, 30] use the rate-distortion costs to select the optimal mode for the current frame. For example, the encoder prefers to use a large block size for homogeneous regions while a small block size is adopted for complex regions. To this end, the encoder will calculate the rate-distortion cost for each mode in the coding procedure. In contrast, the current learning based video compression systems [20, 40] do not employ the content adaptive coding scheme. In other words, the rate-distortion technique is no longer exploited in the inference stage. Therefore, the compressed features are not optimal for the current frame.

More importantly, the encoders are optimized by the rate-distortion optimization (RDO) technique in the training dataset, due to the domain gap between the training and testing datasets in terms of resolution or motion magnitudes, the learned encoders may be far from optimal for the testing dataset. For example, the average motion magnitude between neighboring frames in the training dataset is in the range of [1, 8] pixels [41]. However, the motion in some testing datasets (*e.g.,* the HEVC Class C dataset) is much larger and more complex. The experimental results in [20] also indicate that the compression performance on the HEVC Class C dataset decreases when compared with other datasets.

4 Proposed Method

4.1 Introduction of the DVC framework

In this paper, we use the framework in [20] as our baseline algorithm to demonstrate the effectiveness of our new approach. In [20], the deep video compression (DVC) framework follows the classical hybrid coding approach and designs two auto-encoder style networks to compress the motion and residual information, respectively. The architecture of DVC is shown in Fig. 3. The modules with **green color** (*i.e.,* the optical flow net, the motion encoder net and the residual encoder net) represent the **Encoder**. The other modules (*i.e.*, MV decoder, motion compensation net and residual decoder net) represent the **Decoder**. Here,

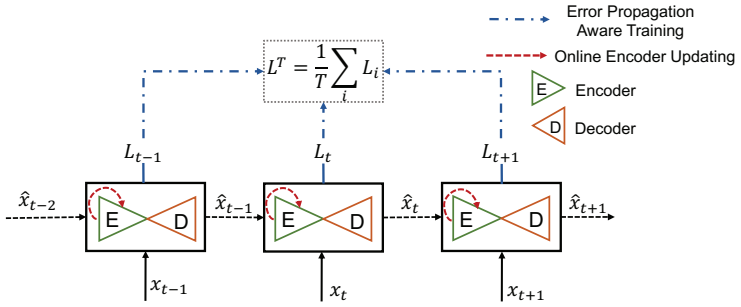


Fig. 4: The proposed content adaptive and error propagation aware deep video compression method.

we use Φ_E and Φ_D to represent the trainable parameters in the Encoder and Decoder, respectively. In the inference stage, the parameters in both Encoder and Decoder are fixed. In the coding procedure, we first estimate the motion information v_t between the current frame x_t and previous reconstructed frame $\hat{x}_t$. The the motion information will be compressed by the auto-encoder style network and the reconstructed optical flow $\hat{v}_t$ will be used for the motion compensation network. Then we obtain the predicted frame $\bar{x}_t$ and the corresponding residual information r_t. Finally, we use the residual compression network to compress the residual information and obtain the final reconstructed frame $\hat{x}_t$ based on $\bar{x}_t$ and the reconstructed residual $\hat{r}_t$.

The DVC model is optimized by minimizing the following rate-distortion (RD) trade-off,

$$L_t = \lambda D_t + R_t = \lambda d(x_t, \hat{x}_t) + [H(\hat{y}_t) + H(\hat{m}_t)] \tag{1}$$

L_t is the loss function for the current time step t. $d(\cdot,\cdot)$ is the distortion metric between x_t and $\hat{x}_t$. $\hat{y}_t$ and $\hat{m}_t$ are the compressed latent representations from residual and motion information, respectively. $H(\hat{y}_t)$ and $H(\hat{m}_t)$ are the corresponding number of bits used for compressing these latent representations. It is noticed that the whole network is optimized to minimize the rate-distortion criterion for the current time step t.

However, this scheme ignores two critical ***dependencies*** for learning based video compression. First, the compression system, including the encoder and decoder, ignores the potential influence from the reconstruction error of $\hat{x}_t$ to the next frame x_{t+1} in the training procedure and thus leads to error propagation. Second, the encoder itself is fixed and does not depend on the current frame x_t, which deteriorates the compression performance in the inference stage. In the next section, we will introduce how to address these two issues in video compression.

4.2 The Error Propagation Aware Training Strategy

To alleviate error accumulation in video compression, we propose an error propagation aware training strategy. Specifically, we 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 for better video compression performance.

The proposed training procedure is shown in Fig.4. For the current frame x_t, the corresponding reconstructed frame after the encoding and decoding procedure is $\hat{x}_t$. Given x_t and $\hat{x}_t$, we can calculate the RD cost L_t. Then $\hat{x}_t$ will be used as the reference frame in the encoding procedure of x_{t+1}, and we obtain the reconstructed frame $\hat{x}_{t+1}$ and the RD cost L_{t+1}. As the coding procedure continues, the reconstructed error will propagate to the subsequent frames. Meanwhile, we also obtain a series of RD costs, which measure the compression performance at the current time step.

Then, we propose a new objective function by considering the compression performance for both the current frame and the subsequent frames that rely on the current reconstructed frame. Therefore, the loss function is formulated as follows,

$$L^T = \frac{1}{T}\sum_t L_t = \frac{1}{T}\sum_t \{\lambda d(x_t, \hat{x}_t) + [H(\hat{y}_t) + H(\hat{m}_t)]\} \tag{2}$$

where T is the time interval(*i.e.,* the number of frames used in training procedure) and set as 5 in our experiments, L^T represents the error propagation aware loss function. Therefore, our new training objective will optimize the video codec by employing the objectives from multiple time steps.

As shown in Fig. 2, the video codec DVC with an error propagation aware (DVC+EPA) training strategy significantly reduces error accumulation. For example, the proposed method has 0.61dB (32.11dB vs. 31.50dB) improvement over the baseline DVC algorithm [20] for the 50^{th} frame, and the gain becomes larger when the time step increases.

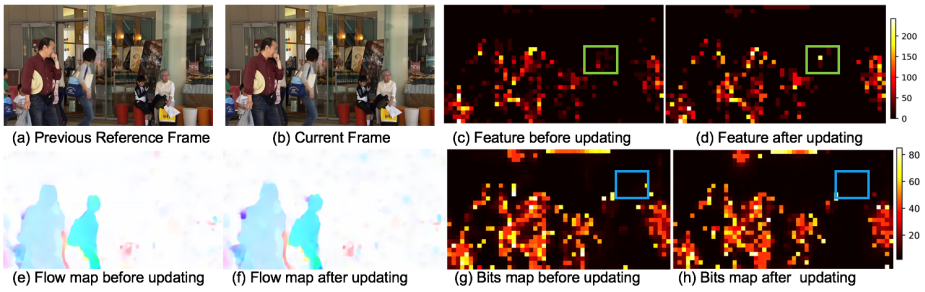


Fig. 5: Visual comparison before and after using our online encoder updating scheme.

4.3 The Online Encoder Updating Scheme

To optimize the encoder for each frame and mitigate the domain gap between training and testing data, we propose an online encoder updating scheme in the inference stage. Our method will update the encoder according to the input

image while keeping the decoder unchanged. In other words, we use the training dataset to obtain a general decoder and employ the testing dataset to update the CNN parameters of the encoder. Based on the training strategy described in the previous section, we can obtain the learned encoder(E) and decoder(D). For the given original frame x_t and the reference frame $\hat{x}_{t-1}$, the objective L_t at the current frame is obtained according to Eq. (1). Then, the parameters Φ_D in the decoder are fixed while the parameters Φ_E are updated by minimizing L_t. After several iterations, we obtain the content adaptive encoder, which is optimal for the current frame x_t. Finally, the updated encoder based on testing data and the learned decoder from training data is employed for the actual compression procedure. In our implementation, the maximum iteration number is set to 10. To reduce computational complexity, we will compare L_t between two consecutive iterations and stop the optimization procedure once the loss becomes stable.

In contrast to other low-level vision tasks, the ground-truth frame for video compression is available at the encoder side. As a result, we can update the encoder by using the original frame as long as the decoder remains unchanged.

In Fig. 5, we provide the visual results before and after the online updating procedure. It is observed that the output feature from the residual encoder (Fig.5(c) and (Fig.5(d)) has changed after the updating procedure which is optimized for the current frame. More importantly, as shown in Fig. 5(e) and Fig. 5(f), the optical flow map after the updating process contains more details, which is beneficial for accurate prediction. For example, based on the optical flow map in Fig. 5(e), the PSNR of the warped frame is 33.40dB, while the corresponding PSNR of the warped frame is 34.13dB based on the updated optical flow map in Fig. 5(f). Furthermore, for the estimated bits map shown in Fig. 5(g) and Fig. 5(h), it is observed that the bits map after the updating process allocates fewer bits for the background region. The experimental results show that the coding bits drop from 0.056bpp to 0.051bpp after the online encoder updating procedure. However, the reconstructed frame has better visual quality after the online updating procedure (36.47dB vs. 36.40dB).

5 Experiments

5.1 Experimental Setup

Datasets In the training stage, we use the Vimeo-90k dataset[41]. Vimeo-90k is a widely used dataset for low-level vision tasks [37, 21]. It is also used in the recent learning based video compression tasks [20, 15]. To evaluate the compression performance of different methods, we employ the four widely used datasets in our experiments.

Specifically, for the HEVC Common Test Sequences [30], we use Class B, Class C, Class D and Class E in our experiments. We don't include the video sequences from the HEVC Class A dataset since it requires more than 11Gb memory for evaluation, which exceeds the capacity of our 1080Ti machine. More details about the other testing datasets including Video Trace Library(VTL) [3],

Ultra Video Group(UVG) [2] and MCL-JCV [36], are provided in the supplementary material.

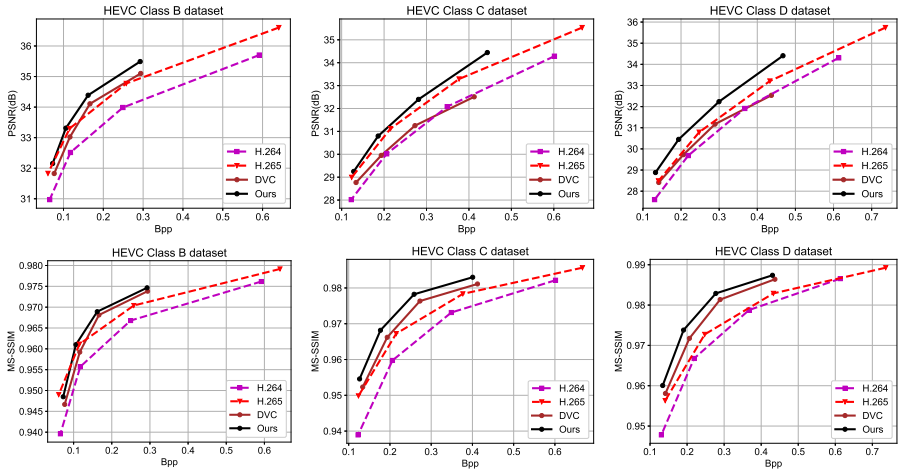


Fig. 6: Comparison between our proposed method with the learning based video codec in [20], H.264 [39] and H.265 [30] at the fixed GoP setting.

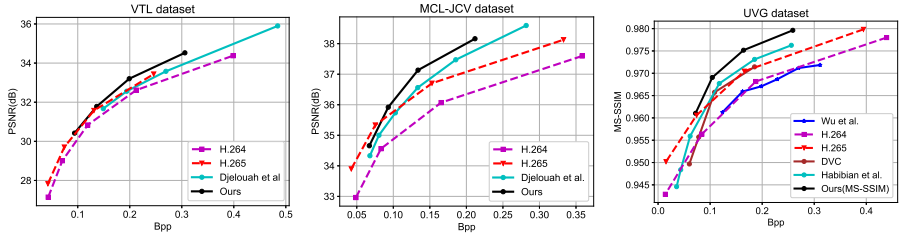


Fig. 7: Comparison between our proposed method and the learning based video codecs [15, 16, 40] at the fixed GoP setting.

Implementation details. We train four models with different λ values (256, 512, 1024, 2048) in Eq. (1). To generate the I-frame/key-frame for video compression, we use the learning based image compression method in [9], in which the corresponding λ in the image codec are empirically set to 1024, 2048, 4096 and 12000, respectively.

In our implementation, we use DVC [20] as the baseline method. In the training stage, the whole network is first optimized by using the loss in Eq.(1), then is fine-tuned based on the error propagation aware loss in Eq.(2). The corresponding batch sizes are set to 4 and 1, respectively. The resolution of the training images is 256×256. We use Adam optimizer [18] and the initial learning rate is set as $1e-4$ for the first 2M steps, and the learning rate is then set to 1e-5 for the remaining 0.5M steps. In the inference stage, the encoder is also optimized by using Adam optimizer [18] to achieve content adaptive encoding.

The proposed method is implemented based on Tensorflow. It takes about 5 days to train the whole network by using two GTX 1080Ti GPUs.

In our experiments, we use the PSNR and MS-SSIM [38] to measure the distortion between the original frame and the reconstructed frame. The bits per pixel(bpp) represents the coding bits in the compression procedure. The bpp values are estimated from the theoretical values based on the probability of the latent space values.

5.2 Comparison with the state-of-the-art methods

Evaluation Setting. To make fair comparison with the state-of-the-art learning based video compression methods and the traditional video codecs H.264/H.265, we follow the existing evaluation protocols in [40, 20, 15] to perform extensive experiments. Specifically, all the existing learning based methods [40, 20, 15] use the ***fixed GoP setting***. For example, the GoP size for the UVG dataset is set to 12 while the corresponding GoP size is set to 10 for the HEVC Common Test Sequences [40, 20]. And the corresponding GoP size for H.265/H.264 in these works is also fixed to 12 or 10. We follow the same settings and provide the experimental results in **Fig.6 and Fig.7.**

For the common testing cases of the traditional video codecs, the GoP size is usually not fixed. To further evaluate the performance of the learning based video codec and the traditional video codec (*e.g.*, H.265), we do not impose any restriction on the GoP size in the codec. Specifically, we adopt *veryfast* mode in FFmpeg with the ***default Setting***.[1] We evaluate the compression performance for all the video frames on the HEVC Class B, Class C and Class D datasets. The experimental results are provided in **Fig.8**.

Baseline Algorithms The learning based codecs in Wu *et al.* [40] and Djelouah *et al.* [15] are based on frame interpolation and designed for B-frame video compression, while the methods in [20, 16] are for P-frame based video compression. Since the B-frame based compression methods employ two reference frames, the coding performance is generally better than P-frame based compression method [30]. We use the P-frame based compression method DVC [20] as our baseline algorithm and we also demonstrate that the proposed method outperforms all the learning based methods, including the B-frame based compression methods [40, 15].

Quantitative Evaluation at the *fixed GoP setting*. As shown in Fig.6, we provide the compression performance of different methods on the HEVC Common Test Sequences. When compared with the baseline DVC [20] algorithm, our proposed method significantly improves the compression performance. For example, our proposed method has about 1dB improvement on the HEVC Class C dataset at 0.3bpp. It is also observed that the proposed method outperforms the H.264 algorithm and is comparable with H.265 in terms of PSNR. The BDBR

[1] ffmpeg -pix_fmt yuv420p -s WxH -r 50 -i video.yuv -c:v libx265 -preset veryfast -tune zerolatency -x265-params "qp=Q" output.mkv; Q is the quantization parameter. W and H are the height and width of the yuv video.

Table 1: The BDBR and BD-PSNR results of different algorithms when compared with H.264. Negative values in BDBR represent the bitrate saving.

	BDBR(%)			BD-PSNR(dB)		
Dataset	H.265	DVC	Ours	H.265	DVC	Ours
Class B	-32.0	-27.9	-41.7	0.78	0.71	1.12
Class C	-20.8	-3.5	-25.9	0.91	0.13	1.18
Class D	-12.3	-6.2	-25.1	0.57	0.26	1.25

and BD-PSNR results when compared with H.264 are provided in Table 1. The experimental results on Class E are provided in the supplementary material.

We also provide the experimental results when the distortion is evaluated by MS-SSIM. As shown in Fig.6, our approach outperforms H.265 in terms of MS-SSIM. One possible explanation is that the traditional codecs [39, 30] use the block based coding scheme, which inevitably generates the block artifacts.

In Fig.7, we evaluate the compression performance on the MCL-JCV, VTL and UVG datasets. We compare our proposed method with the recent learning based method [15], which utilizes B-frame based compression scheme. As shown in Fig.7, although we only use one reference frame, the proposed method still achieves better compression performance on the VTL dataset.

In Fig.7, we also compare the proposed method with another state-of-the-art learning based video compression method [16] on the UVG dataset. For fair comparison with [16], we also use MS-SSIM as the loss function to optimize the network. The experimental results demonstrate that the proposed approach outperforms [16] by a large margin.

Quantitative Evaluation at the *default setting*. In this section, we also compare the results when the traditional codecs use variable GoP sizes. As shown in Fig.8, our method outperforms the previous DVC algorithm [20] by a large margin, especially for the HEVC Class C dataset. A possible explanation is that error propagation is more severe as the GoP size becomes larger, which means our proposed scheme will bring more improvements. Although the proposed method cannot outperform H.265 at the default setting, the compression performance of these two methods is generally comparable. Considering that the traditional video codecs exploit other coding techniques, such as multiple reference frames or adaptive quantization parameters, which are not used by the current learning based video compression systems, it is possible to further improve the performance of learning based video codec in the future.

5.3 Ablation Study

The Error Propagation Aware Training Scheme. To demonstrate the effectiveness of our proposed error propagation aware training strategy, we compare the compression performance of different methods in Fig. 9. Specifically, the brown line represents the DVC algorithm [20], while the green line represents the DVC algorithm with the error propagation aware (EPA) training strategy. It is noticed that the proposed training scheme improves the performance by

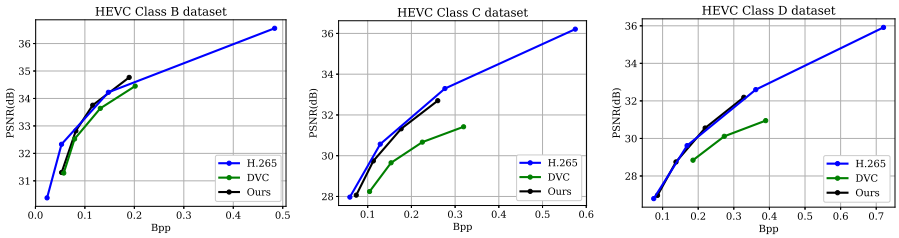


Fig. 8: Evaluation results for all video frames on the HEVC Class B, Class C and Class D at the default setting.

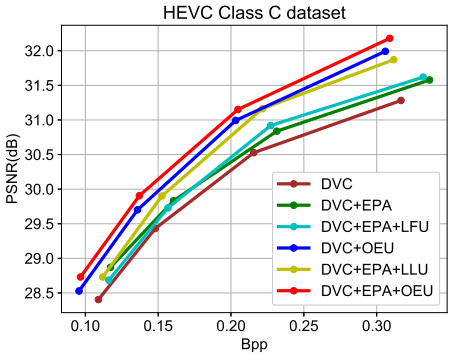


Fig. 9: Ablation study.

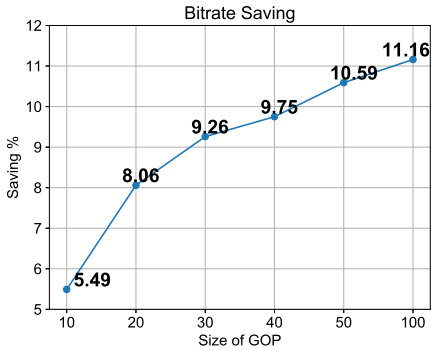


Fig. 10: The bitrate saving when comparing DVC+EPA with DVC [20] at different GoP sizes.

Table 2: BDBR(%) performance at different time intervals (*i.e.*, T in Eq.(2)).

T	2	3	4	5	6
BDBR	-0.42	-2.12	-3.68	-5.59	-5.61

0.2dB on the HEVC Class C dataset(GoP=20), which demonstrates that the proposed scheme can alleviate error accumulation by exploiting temporal neighboring frames in the training stage.

In practical applications, the GoP size for video compression is usually set as 50 or larger to reduce the bandwidth. And error accumulation is more severe as the GoP size increases. In Fig.10, we investigate the effectiveness of our newly proposed error propagation aware training scheme when the GoP sizes are set as different numbers. We use BDBR [10] to measure the bitrate saving when compared with the baseline DVC method [20]. Specifically, the proposed scheme saves 5.49% bitrate when the GoP size is set to 10 and saves up to 10.59% bitrate when the GoP size is set to 50. The experimental results demonstrate that the proposed method has achieved better compression performance for video sequences with the large GoP size.

To further investigate the proposed error propagation aware training strategy, we provide the compression results when the method is optimized by using

different time intervals T. As shown in Table 2, the proposed scheme saves more bitrates when T increases. For example, the proposed training scheme saves 2.12% bitrate when setting $T = 3$, while the corresponding bitrate saving is 5.59% when setting $T = 5$. One explanation is that we can use long-term temporal information when T increases, which effectively alleviates error accumulation. In our experiments, we set T to 5 by default.

The Online Encoder Updating Scheme To demonstrate the effectiveness of our proposed online encoder updating (OEU) scheme in the inference stage, we compare the compression performance of the baseline algorithm with or without using our updating scheme. In Fig.9, the proposed online encoder updating scheme (DVC+OEU, the blue line) significantly improves the compression performance by more than 0.5dB. Besides, the red line represents the full model of our proposed method, which achieves the best compression performance by using both the online updating scheme and the error propagation aware training strategy.

In [11], Campos *et al.* adaptively refined the latent representations of the learning based image codecs for better compression performance. Furthermore, we provide the experimental result for the latent features updating (LFU) scheme, where $\hat{m}_t$ and $\hat{y}_t$ are updated and the encoder itself is fixed. The corresponding RD curve (DVC+EPA+LFU) is depicted by the cyan line. Compared with our proposed training scheme (DVC+EPA), we observe that the performance can be further improved by optimizing the latent representation at a high bitrate. However, it is obvious that adaptively optimizing the whole encoder (red line in Fig. 9) achieves better performance. A possible explanation is that updating the encoder provides a larger search range and thus it is more likely to obtain an optimal encoder for the current frame.

Besides, we also provide the compression results when only partial neural networks are updated in the inference stage. Specifically, we use the last layers updating (LLU) scheme, where only the last layers in the residual encoder and motion encoder are updated according to the rate-distortion technique. The experimental results are denoted by the yellow line (DVC+EPA+LLU) in Fig. 9. It is observed that the partial updating strategy is also useful for video compression. However, the performance is inferior to the proposed approach, where all components in the encoder are updated.

5.4 Discussion

Computational Complexity In this paper, we use an adaptive encoder in the inference stage to improve compression performance. Since the online rate-distortion optimization scheme is required at the encoder side, it will increase the computational complexity. However, it is noticed that the numbers of iterations for different video sequences are different.

For the video sequences with simple motion scenes, such as the HEVC Class B dataset, the encoder learned from the training dataset is already near-optimal and it only requires 3 iterations to obtain the optimal parameters. For the videos with complex motion scenes, such as the HEVC Class C dataset, more iterations

are required to learn the optimal encoder. However, we also obtain a larger improvement(∼1dB). And the corresponding encoding speed of our approach is 1.4fps while the speed of baseline DVC is 7.1fps when using 10 iterations. More performance improvement for some test sequences can also be observed by using more iterations. It is noted that the runtime of our approach is evaluated on one Nvidia 1080Ti GPU and we use the plain Tensorflow operations without any specific optimization.

More importantly, a lot of applications, such as video-on-demand applications, are not sensitive to the computational complexity at the encoder side. Considering that our approach is generic and boosts the compression performance without increasing the decoding time, it is feasible to integrate the proposed techniques with other learning based video codecs, such as [40, 15], to further improve the compression performance.

Entropy Coding We use the same entropy coding methods as in the DVC baseline, in which the entropy coding methods in [8, 9] are employed for motion coding and residual coding, respectively. While the advanced entropy coding methods may partially alleviate the domain gap, the existing advanced entropy models like [24] usually adopt the autoregressive prior technique, which increases the runtime in the decoder side significantly. In contrast, our approach keeps the decoder unchanged without increasing the computational complexity. More importantly, our online encoder updating (OEU) scheme not only improves the internal entropy coding module but also optimizes the whole encoder (including motion estimation, motion compensation, etc), which is more effective for video compression.

6 Conclusion

In this paper, we have proposed a content adaptive and error propagation aware deep video compression method. Our approach alleviates error accumulation in the training stage and achieves content adaptive coding by using the online encoder updating scheme in the inference stage. The proposed method is fairly simple yet effective and improves compression performance without increasing the model size or decreasing the decoding speed. The experimental results show that the compression performance of our proposed method outperforms the state-of-the-art learning based video compression methods.

Acknowledgment This work was supported in part by National Natural Science Foundation of China (61771306) Natural Science Foundation of Shanghai(18ZR1418100),111 plan (B07022), Shanghai Key Laboratory of Digital Media Processing and Transmissions(STCSM 18DZ2270700). Dong Xu was partially supported by the Australian Research Council (ARC) Future Fellowship under Grant FT180100116. Wanli Ouyang was supported by SenseTime, the Australian Research Council Grant DP200103223, and Australian Medical Research Future Fund MRFAI000085.

References

1. F. bellard, bpg image format. http://bellard.org/bpg/, accessed: 2018-10-30
2. Ultra video group test sequences. http://ultravideo.cs.tut.fi, accessed: 2018-10-30
3. Video trace library(vtl) dataset. http://trace.kom.aau.dk/, accessed: 2018-10-30
4. Webp. https://developers.google.com/speed/webp/, accessed: 2018-10-30
5. Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., Gool, L.V.: Soft-to-hard vector quantization for end-to-end learning compressible representations. In: NIPS. pp. 1141–1151 (2017)
6. Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., Gool, L.V.: Generative adversarial networks for extreme learned image compression. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019. pp. 221–231. IEEE (2019)
7. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. IEEE transactions on Computers **100**(1), 90–93 (1974)
8. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: 5th International Conference on Learning Representations, ICLR (2017)
9. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. In: 6th International Conference on Learning Representations, ICLR (2018)
10. Bjontegaard, G.: Calculation of average psnr differences between rd-curves. VCEG-M33 (2001)
11. Campos, J., Meierhans, S., Djelouah, A., Schroers, C.: Content adaptive optimization for neural image compression. In: IEEE CVPR Workshops 2019. p. 0 (2019)
12. Chen, Z., He, T., Jin, X., Wu, F.: Learning for video compression. IEEE Trans. Circuits Syst. Video Techn. **30**(2), 566–576 (2020). https://doi.org/10.1109/TCSVT.2019.2892608
13. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learning image and video compression through spatial-temporal energy compaction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 10071–10080 (2019)
14. Choi, Y., El-Khamy, M., Lee, J.: Variable rate deep image compression with a conditional autoencoder. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019. pp. 3146–3154. IEEE (2019)
15. Djelouah, A., Campos, J., Schaub-Meyer, S., Schroers, C.: Neural inter-frame compression for video coding. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
16. Habibian, A., van Rozendaal, T., Tomczak, J.M., Cohen, T.: Video compression with rate-distortion autoencoders. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019. pp. 7032–7041. IEEE (2019)
17. Hu, Z., Chen, Z., Xu, D., Lu, G., Ouyang, W., Gu, S.: Improving deep video compression by resolution-adaptive flow coding. In: ECCV (September 2020)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Li, M., Zuo, W., Gu, S., Zhao, D., Zhang, D.: Learning convolutional networks for content-weighted image compression. In: CVPR (June 2018)
20. Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., Gao, Z.: DVC: An end-to-end deep video compression framework. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,CVPR. pp. 11006–11015 (2019)
21. Lu, G., Ouyang, W., Xu, D., Zhang, X., Gao, Z., Sun, M.T.: Deep kalman filtering network for video compression artifact reduction. In: ECCV (September 2018)

22. 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 (2020)
23. Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Van Gool, L.: Conditional probability models for deep image compression. In: CVPR. p. 3. No. 2 (2018)
24. Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. In: Advances in Neural Information Processing Systems. pp. 10771–10780 (2018)
25. Rippel, O., Bourdev, L.: Real-time adaptive image compression. In: ICML (2017)
26. Rippel, O., Nair, S., Lew, C., Branson, S., Anderson, A.G., Bourdev, L.D.: Learned video compression. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019. pp. 3453–3462. IEEE (2019)
27. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Transactions on circuits and systems for video technology **17**(9), 1103–1120 (2007)
28. Shensa, M.J.: The discrete wavelet transform: wedding the a trous and mallat algorithms. IEEE Transactions on signal processing **40**(10), 2464–2482 (1992)
29. Skodras, A., Christopoulos, C., Ebrahimi, T.: The jpeg 2000 still image compression standard. IEEE Signal Processing Magazine **18**(5), 36–58 (2001)
30. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T., et al.: Overview of the high efficiency video coding(HEVC) standard. TCSVT **22**(12), 1649–1668 (2012)
31. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. In: 5th International Conference on Learning Representations, ICLR (2017)
32. Toderici, G., O'Malley, S.M., Hwang, S.J., Vincent, D., Minnen, D., Baluja, S., Covell, M., Sukthankar, R.: Variable rate image compression with recurrent neural networks. In: 4th International Conference on Learning Representations, ICLR (2016)
33. Toderici, G., Vincent, D., Johnston, N., Hwang, S.J., Minnen, D., Shor, J., Covell, M.: Full resolution image compression with recurrent neural networks. In: CVPR. pp. 5435–5443 (2017)
34. Tsai, Y.H., Liu, M.Y., Sun, D., Yang, M.H., Kautz, J.: Learning binary residual representations for domain-specific video streaming. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
35. Wallace, G.K.: The jpeg still picture compression standard. IEEE Transactions on Consumer Electronics **38**(1), xviii–xxxiv (1992)
36. 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)
37. Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: Edvr: Video restoration with enhanced deformable convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
38. Wang, Z., Simoncelli, E., Bovik, A., et al.: Multi-scale structural similarity for image quality assessment. In: ASILOMAR CONFERENCE ON SIGNALS SYSTEMS AND COMPUTERS. vol. 2, pp. 1398–1402. IEEE; 1998 (2003)
39. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. TCSVT **13**(7), 560–576 (2003)
40. Wu, C.Y., Singhal, N., Krahenbuhl, P.: Video compression through image interpolation. In: ECCV (September 2018)

41. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. International Journal of Computer Vision, IJCV **127**(8), 1106–1125 (2019)