# Neural Video Compression with Context Modulation

Anonymous CVPR submission

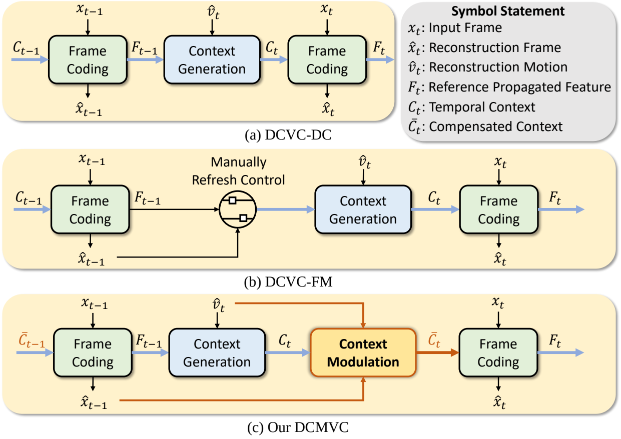
Paper ID 10317

## Abstract

*Efficient video coding is highly dependent on exploiting the temporal redundancy, which is usually achieved by extracting and leveraging the temporal context in the emerging conditional coding-based neural video codec (NVC). Although the latest NVC has achieved remarkable progress in improving the compression performance, the inherent temporal context propagation mechanism lacks the ability to sufficiently leverage the reference information, limiting further improvement. In this paper, we address the limitation by modulating the temporal context with the reference frame in two steps. Specifically, we first propose the flow orientation to mine the inter-correlation between the reference frame and prediction frame for generating the additional oriented temporal context. Moreover, we introduce the context compensation to leverage the oriented context to modulate the propagated temporal context generated from the propagated reference feature. Through the synergy mechanism and decoupling loss supervision, the irrelevant propagated information can be effectively eliminated to ensure better context modeling. Experimental results demonstrate that our codec achieves on average 22.7% bitrate reduction over the advanced traditional video codec H.266/VVC, and offers an average 10.1% bitrate saving over the previous state-of-the-art NVC DCVC-FM.*

## 1. Introduction

Temporal prediction plays an essential role in removing temporal redundancy for video compression. Since DCVC [23] shifted the paradigm from traditional residual coding to conditional coding, temporal prediction has been represented efficiently by the temporal context for the emerging conditional coding-based neural video compression framework [15, 20, 23–26, 40, 42, 43, 48]. The temporal context is regarded as the condition to encode and decode the current frame, so generating high-quality temporal context is the key to achieve a better compression performance.



(a) DCVC-DC

(b) DCVC-FM

(c) Our DCMVC

Figure 1. Comparison of context generation for our DCMVC (Deep Context Modulation for Video Compression) with previous state-of-the-art compression schemes, DCVC-DC and DCVC-FM.

Recent conditional coding-based neural video codecs (NVCs) have invested significant efforts in better leveraging the temporal context to realize effective inter-frame prediction. Based on DCVC [23], DCVC-TCM [42] further mines temporal context from the propagated reference feature and motion vector (MV), and the reference propagation structure mechanism is widely adopted in the following works [24–26, 40, 43]. As shown in Fig. 1 (a), taking our baseline DCVC-DC [25] as an example, the context of current frame $C_t$ is generated through the context generation mechanism with the input of propagated adjacent reference feature $F_{t-1}$ and reconstructed MV $\hat{v}_t$. In Fig. 1 (b), based on the DCVC-DC, considering temporal error accumulation problem in long prediction chain, DCVC-FM [26] has proposed a periodically refreshing mechanism in the process of context generation, which manually switches the input of context generation from the propagated reference feature $F_{t-1}$ to reference frame $\hat{x}_{t-1}$. Through the modulation on the temporal feature under a fixed temporal period, DCVC-FM mitigates the error propagation and maintains a good inter-frame prediction quality in long prediction chain.

Despite the progress made by DCVC-FM, we rethink the temporal context modeling in the conditional coding-based framework. Motivated by the inherent reference propagation structure mechanism and benefits of periodically refreshing mechanism in DCVC-FM, we find the propagated reference feature conveys irrelevant information in the prediction chain, though it contains more information than reference frame. Simultaneously, switching between reference frames and reference features with the fixed temporal period may not effectively exploit the reference information.

To address this issue, we propose the context modulation to generate a high-quality temporal context exploiting the reference information in both pixel and feature domain, which is shown in Fig. 1 (c). In this paper, we propose the two-step context modulation framework. First, we consider how to leverage the reference frame to learn an additional temporal context to assist the propagated temporal context generated from the propagated reference feature. Specifically, in order to adequately exploit the reference information in the pixel domain, we propose the flow orientation to extract the oriented flow between the reference frame and prediction frame. Then, we learn the oriented context from the reference frame and oriented flow for further modulating the propagated context. Second, given the propagated context and oriented context, we investigate how to manage them for temporal context modeling, and propose the context compensation to modulate the propagated context with the assistance of oriented context. To further facilitate the synergy mechanism, we also design a training constraint in the form of decoupling loss function, which leads to a better complementarity of two temporal contexts during the training. Through the context compensation, the irrelevant propagated information is removed to ensure better context modeling, which is also beneficial in alleviating error propagation.

With the aforementioned flow orientation and context compensation methods, we develop a novel NVC termed DCMVC, which stands for Deep Context Modulation for Video Compression. Experiments demonstrate that our DCMVC surpasses previous SOTA neural video compression schemes under both intra-period settings of 32 and -1. Compared to the advanced traditional codec H.266/VVC, our model achieves on average 22.7% bitrate saving on the tested videos. Our model also outperforms the best neural video codec DCVC-FM by 10.1% bitrate reduction, showing the effectiveness of our proposed scheme.

Our contribution can be summarized as follows:

- We propose the context modulation in the conditional coding-based framework to generate high-quality temporal context exploiting the reference information in both pixel and feature domain. Our model can enable better temporal context and alleviate error propagation.
- We propose the flow orientation method to mine interframe correlation between the reference frame and prediction frame. It enables our DCMVC to generate additional oriented temporal context from the reference frame.
- We propose the context compensation method to modulate the propagated temporal context with oriented context by using synergy mechanism and decoupling loss. It eliminates the irrelevant propagated information to ensure better context modeling.
- Our DCMVC can achieve 22.7% bitrate reduction over the traditional video codec H.266/VVC, and obtain on average 10.1% bitrate saving compared to the previous SOTA neural video codec DCVC-FM.

## 2. Related Work

### 2.1. Neural Video Compression

With the fast development of neural image compression [2, 3, 10, 13, 14, 30, 31, 38, 53], neural video compression has drawn increasing attention in recent years. Existing NVCs can be roughly classified into four classes: volume coding-based video codecs [12, 39, 47], implicit neural representation-based codecs [7, 8, 21, 22, 54], residual coding-based video codecs [1, 9, 16–19, 27, 29, 32–34, 45, 52], and conditional coding-based video codecs [15, 20, 23–26, 40, 42–44, 48]. The first two categories of compression schemes still have considerable room for improvement in terms of compression performance, while the latter two are currently the dominant solutions and have already achieved impressive progress. Residual coding-based video codecs leverage the optical flow estimation network [41, 46, 49] to generate the prediction frame, then its residual with the current frame is encoded to remove the temporal redundancy. Different from the residual-coding paradigm, the emerging conditional coding-based video codecs learn the temporal context in more flexible ways and have an entropy bound lower than or equal to residual coding. Our scheme adopts the conditional coding-based framework and makes improvements on temporal context learning.

### 2.2. Temporal Context Learning

In NVC, the temporal context is usually extracted from the previous reconstruction frames and the current frame, then utilized to assist the current frame coding. For residual coding-based scheme, DVC [33] utilized the optical flows and reference frame to generate the prediction frame as the temporal context. M-LVC [29] further improved the context modeling with multiple reference frames. To improve the feature alignment, FVC [18] adopted the deformable convolution [11] to obtain more accurate context in the feature domain. AlphaVC [45] proposed pixel-to-feature motion prediction to improve the accuracy of temporal context modeling at the encoder side.

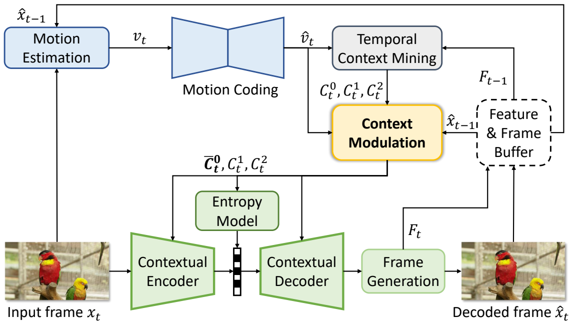


Figure 2. Our DCMVC framework. Based on the contextual-coding framework, we propose the context modulation to generate the compensated temporal context $\overline{C}_t^0$ with the input of propagated temporal context $C_t^0$, decoded flow $\hat{v}_t$, and reference frame $\hat{x}_{t-1}$.

For conditional coding-based schemes, DCVC [23] has proposed to generate the context in feature domain to remove the temporal redundancy rather than relying on the subtraction-based residual in pixel domain. The subsequent DCVC series [24–26, 40, 42] have focused on enhancing the temporal context learning for achieving better compression performance. DCVC-TCM [42] proposed the reference feature propagation mechanism to learn the multi-scale temporal context, which is widely adopted in the following works. DCVC-HEM [24] further proposed the hybrid entropy model to better capture the temporal and spatial dependency lies in the context. Group-based offset diversity [25] and hybrid context generation module [40] were proposed to improve the quality of the temporal context. The latest DCVC-FM [26] alleviated the error propagation through modulating the temporal feature. The reference frame was periodically selected to generate the context rather than from propagated reference features for stopping the quality degradation in long prediction chain.

Although these works have achieved progressive compression performance, mining high-quality temporal context remains exploration. The inherent reference propagation structure mechanism leads to quality degradation in long prediction chain, because the propagated reference feature may contain more irrelevant information than reference frame. Although DCVC-FM [26] mitigates error propagation by periodically switching the reference frame in long prediction chains, only this manual context modeling mechanism may not sufficiently exploit the reference information. In this paper, we address the limitation by proposing context modulation to sufficiently exploit the reference information in both pixel and feature domain.

# 3. Proposed Method

## 3.1. Overview

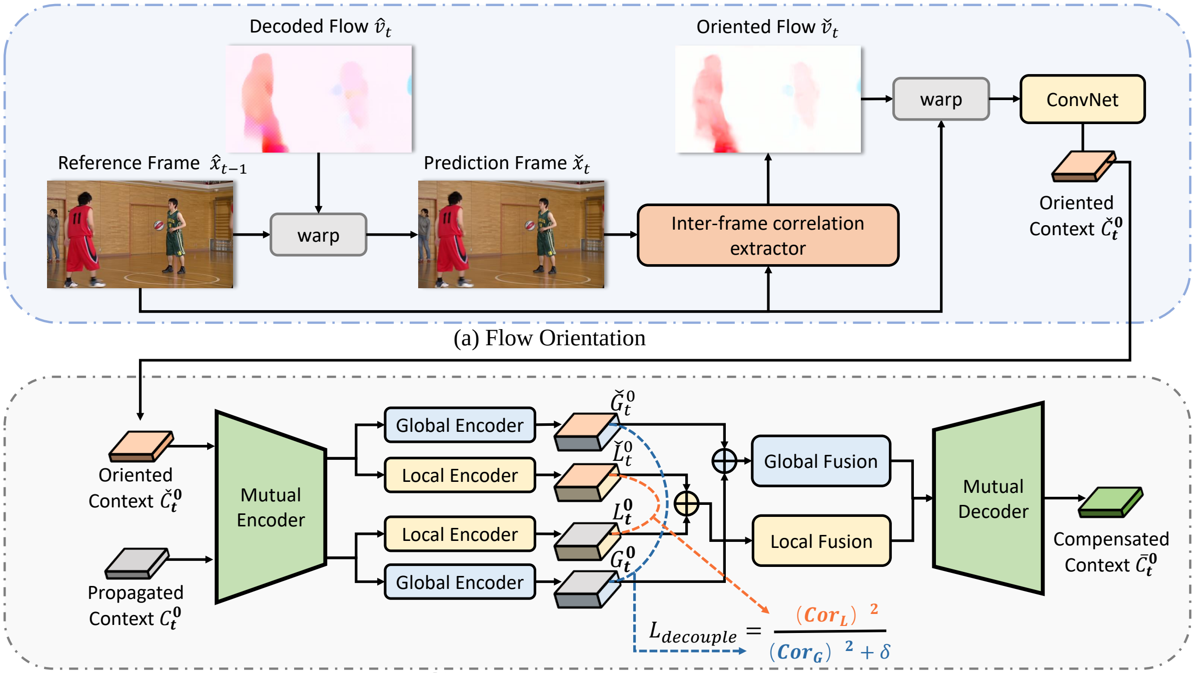
Our DCMVC is built on the emerging conditional coding-based framework, DCVC-DC [25]. In Fig. 2, our proposed framework DCMVC is illustrated. First, the current frame $x_t$ and reference frame $\hat{x}_{t-1}$ (the previous decoded frame) are input into the motion estimation for generating the MV $v_t$, which is represented by optical flows. The estimated flow $v_t$ is compressed through the motion coding module. Second, with the decoded flow $\hat{v}_t$ and the propagated reference feature $F_{t-1}$, the temporal context mining is performed to learn multi-scale propagated temporal contexts $C_t^0$, $C_t^1$, $C_t^2$. To better model temporal context, we propose the context modulation to generate the largest-scale compensated temporal context $\overline{C}_t^0$ given the reference frame $\hat{x}_{t-1}$ and the decoded flow $\hat{v}_t$. The compensated temporal context $\overline{C}_t^0$, along with $C_t^1$ and $C_t^2$, serves as conditions for the contextual encoder, entropy model, and decoder. Third, given the feature decoded by the contextual decoder, the reconstruction frame $\hat{x}_t$ and reconstruction feature $F_t$ are generated by the frame generator, and $F_t$ is propagated as the reference of next frame coding.

Our proposed context modulation consists of flow orientation and context compensation, and the workflows are shown in Fig. 3, which are introduced in Section 3.2 and Section 3.3.

## 3.2. Flow Orientation

Previous conditional coding-based NVCs encode and decode the current frame with the condition of the propagated temporal context. The propagated context is inherently generated through the decoded flow $\hat{v}_t$ and propagated reference feature $F_{t-1}$. Inspired by the error propagation phenomenon of the inherent reference propagation structure mechanism in DCVC-DC, observed in [26], the propagated reference feature brings irrelevant information impacting error accumulation in the prediction chain. In contrast, the reference frame $\hat{x}_{t-1}$, which is the adjacent reconstructed frame, is constrained by the distortion loss in the rate-distortion (RD) loss function, so the reference frame contains as little irrelevant information as possible compared to the propagated reference feature. To sufficiently exploit the reference frame to generate an additional temporal context, we propose the flow orientation method.

Fig. 3 illustrates the generation of the oriented temporal context $\check{C}_t^0$ in the flow orientation. With the decoded flow $\hat{v}_t$ parsed from the motion decoder, the prediction frame $\check{x}_t$ is obtained by warping the reference frame $\hat{x}_t$. Due to the RD constraint of motion bits and prediction quality in training, the representation capacity of the prediction frame for the generation of temporal context is limited. Therefore, we propose to introduce an additional extraction module to extract the inter-frame correlation for remedy. Specifically, the Spynet [41] is used as a pyramid inter-frame correlation extractor to search the temporal correlation information between the reference frame and the prediction frame, which is named as the oriented flow $\check{v}_t$. As shown in Fig. 4 (a),


Decoded Flow $\hat{v}_t$
Oriented Flow $\check{v}_t$
warp
ConvNet
Reference Frame $\hat{x}_{t-1}$
Prediction Frame $\check{x}_t$
warp
Inter-frame correlation extractor
Oriented Context $\check{C}_t^0$
Oriented Context $\check{C}_t^0$
Propagated Context $C_t^0$
Mutual Encoder
Global Encoder
Local Encoder
Local Encoder
Global Encoder
$\check{G}_t^0$
$\check{L}_t^0$
$L_t^0$
$G_t^0$
Global Fusion
Local Fusion
Mutual Decoder
Compensated Context $\overline{C}_t^0$
$L_{decouple} = \frac{(Cor_L)^2}{(Cor_G)^2 + \delta}$

(a) Flow Orientation

(b) Context Compensation

Figure 3. (a) Framework overview of the flow orientation in context modulation. $\check{C}_t^0$ is the oriented context generated from reference frame $\hat{x}_{t-1}$ and oriented flow $\check{v}_t$. $C_t^0$ is the context generated through propagated reference feature $F_{t-1}$ and decoded flow $\hat{v}_t$. $\overline{C}_t^0$ is the context generated from our proposed context modulation. (b) Framework overview of the context compensation in context modulation. $\check{G}_t^0$ and $G_t^0$ are global features extracted from $\check{C}_t^0$ and $C_t^0$, respectively, and $Cor_G$ is the cosine correlation between $\check{G}_t^0$ and $G_t^0$. Similarly, $\check{L}_t^0$ and $L_t^0$ are local features extracted from $\check{C}_t^0$ and $C_t^0$, respectively, and $Cor_L$ is the cosine correlation between $\check{L}_t^0$ and $L_t^0$. The decoupling loss function $L_{decouple}$ consists of the cosine similarity of both global features and local features.

we visualize the estimated flow $v_t$ (estimated from the reference frame and current frame), decoded flow $\hat{v}_t$ (decoded from the motion decoder), oriented flow $\check{v}_t$, and their corresponding warp frames. Due to the RD constraint of motion bits, the uncertainty of the estimated flow and decoded flow may incur prediction errors in the warp frames. In contrast, the oriented warp frame generated by oriented flow $\check{v}_t$ achieves better prediction results than those generated by $v_t$ and $\hat{v}_t$, which indicates that the oriented flow has learned additional temporal correlation information for the temporal context mining. With the guidance of oriented flow $\check{v}_t$, the reference frame $\hat{x}_t$ is aligned to extract the oriented temporal context $\check{C}_t^0$.

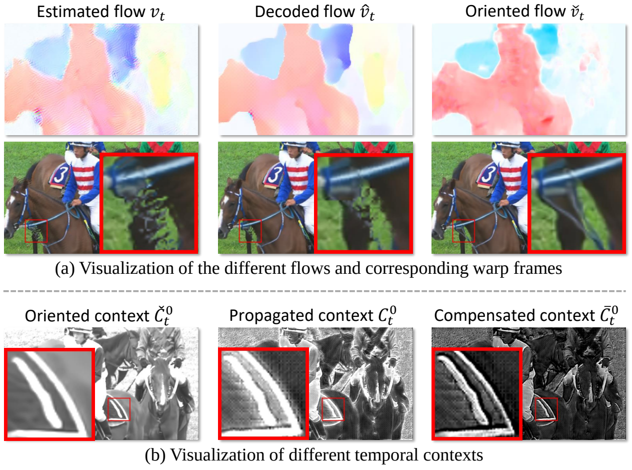
### 3.3. Context Compensation

After obtaining the oriented temporal context and the propagated temporal context from different reference mechanisms, we consider how to effectively manage these temporal contexts for better temporal context modeling. To achieve the synergy of two kinds of temporal contexts, we propose the context compensation method to modulate the propagated context with oriented context.

Motivated by the good traditions of multi-source perception in [35,36,55,56], we design a global-local context compensation network to enable better complementarity of the oriented temporal context and propagated temporal context. In addition, decoupling loss is designed to supervise the network for facilitating better synergy of different contexts.

Fig. 3 shows the workflow of our proposed context compensation. First, the oriented temporal context $\check{C}_t^0$ is generated from the flow orientation, and the largest-scale temporal context $C_t^0$ is generated from the temporal context mining, which is performed by the multi-scale alignment [25] of the propagated feature $F_t$ with decoded flow $\hat{v}_t$. Then, with the global-local synergy mechanism and the supervision of decoupling loss, the oriented context $\check{C}_t^0$ is used to modulate the propagated context $C_t^0$ for learning the compensated temporal context $\overline{C}_t^0$. In detail, the structure of context compensation can be divided into two parts: feature extraction and feature fusion.

**Feature Extraction.** Specifically, the shallow features from $\check{C}_t^0$ and $C_t^0$ are learned by the shared mutual encoder. Then, the private global encoders learn the global features from $\check{C}_t^0$ and $C_t^0$ with large receptive field, while the private local encoders learn the local features from $\check{C}_t^0$ and $C_t^0$. The two contexts $\check{C}_t^0$ and $C_t^0$ are derived from similar ref-


Estimated flow $v_t$
Decoded flow $\hat{v}_t$
Oriented flow $\check{v}_t$
Oriented context $\check{C}_t^0$
Propagated context $C_t^0$
Compensated context $\overline{C}_t^0$

(a) Visualization of the different flows and corresponding warp frames

(b) Visualization of different temporal contexts

Figure 4. (a) Visualization of the estimated flow $v_t$ (estimated from reference frame and current frame), decoded flow $\hat{v}_t$ (decoded from motion decoder), oriented flow $\check{v}_t$ (generated from reference frame and prediction frame), and their warp frames. (b) Visualization of oriented context $\check{C}_t^0$, propagated context $C_t^0$, and compensated context $\overline{C}_t^0$ obtained from the context compensation.

erence resources (reference frame and propagated reference features), so the global features of two contexts are more similar than local features. Therefore, we design the local encoder to extract and preserve as much detail information of two contexts as possible, and we adopt invertible neural networks with affine decoupling layers in both local encoder and local fusion.

**Feature Fusion.** After obtaining the global and local features from the oriented context $\check{C}_t^0$ ($\check{G}_t^0$ and $\check{L}_t^0$) and propagated context $C_t^0$ ($G_t^0$ and $L_t^0$), the global features and local features of the two contexts are added and then input to the global and local fusion network, respectively. To maintain the performance by leveraging the consistency assumption, the networks of encoder and fusion adopt the same architecture. Then, the global and local fusion net are used to generate the fused global feature and local feature, which are further concatenated to input to the shared mutual decoder for learning the final compensated context $\overline{C}_t^0$.

As illustrated in Fig. 4 (b), we visualize the oriented context $\check{C}_t^0$, propagated context $C_t^0$, and compensated context $\overline{C}_t^0$. Upon closer inspection of the contexts, we notice that the oriented context $\check{C}_t^0$ has smoother object edges, while the propagated context $C_t^0$ exhibits obvious prediction errors. With the synergy mechanism and guidance of decoupling loss, we can observe that the prediction errors are restored in compensated context $\overline{C}_t^0$.

**Decoupling losses.** To facilitate the synergy mechanism during the training, we design a decoupling loss $L_{decouple}$ shown as follows:

$$L_{decouple} = \frac{(Cor(\check{L}_t^0, L_t^0))^2}{(Cor(\check{G}_t^0, G_t^0))^2 + \delta}, \tag{1}$$

where the $Cor(\cdot,\cdot)$ represents the cosine similarity operator. $\check{L}_t^0$ and $L_t^0$ denotes the local features of the oriented context and propagated context, and $\check{G}_t^0$ and $G_t^0$ denotes the global features of the oriented context and propagated context. $\delta$ indicates the smoothing term to ensure the values are positive, and is set to 1e-6.

Under the supervision of decoupling loss, the global information, such as structure and background information, in both temporal contexts can be more correlated, while the local information, such as texture and edges, is more specific for different contexts, which can be less correlated. The ablation study in Section 4.3 shows the effectiveness of the decoupling loss and visualization of global and local features.

### 3.4. Training Loss

The whole video compression framework is trained in an end-to-end manner optimizing the loss function shown as follows:

$$L = \lambda \cdot D + \alpha \cdot L_{decouple} + R, \tag{2}$$

where $D$ represents the distortion between the input frame and the reconstructed frame. We use the Mean Squared Error (MSE) metric to measure the distortion. $\lambda$ and $\alpha$ are hyperparameters to control the weights of distortion and decoupling loss, respectively. $R$ denotes the number of bits used to encode the frame.

## 4. Experimental Results

### 4.1. Experimental Settings

**Datasets.** For training, we use the commonly used 7-frame Vimeo-90k [51] dataset to align with the training setting of the most existing NVCs, and all the videos are randomly cropped into 256 × 256 patches. Besides, we utilize the raw Vimeo videos[1] to select a subset of 9000 sequences for 32-frame cascaded training, and all the videos are randomly cropped into 256 × 384 patches. For testing, we use UVG [37], MCL-JCV [50], USTC-TD [28], and HEVC Class B, C, D, and E dataset [5].

**Test Conditions.** Following most previous neural video compression schemes, we conduct our tests under low-delay settings. We test 96 frames for each video, and the intra-period is set to 32 and -1 for each dataset as two different settings. For evaluating the compression performance, we use BD-rate metric [4] to measure the compression ratio

[1] http://toflow.csail.mit.edu

Table 1. BD-Rate (%) comparison in RGB colorspace measured with PSNR. 96 frames with intra-period=32.

| | UVG | MCL-JCV | HEVC B | HEVC C | HEVC D | HEVC E | USTC-TD | Average |
|---|---|---|---|---|---|---|---|---|
| VTM | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **0.0** | 0.0 |
| DCVC [23] | 133.9 | 106.6 | 119.6 | 152.5 | 110.9 | 274.8 | 130.4 | 147.0 |
| DCVC-TCM [42] | 23.1 | 38.2 | 32.8 | 62.1 | 29.0 | 75.8 | 73.5 | 47.8 |
| DCVC-HEM [24] | –17.2 | –1.6 | –0.7 | 16.1 | –7.1 | 20.7 | 21.5 | 4.5 |
| SDD [43] | –19.7 | –7.1 | –13.7 | –2.3 | –24.9 | –8.4 | 6.6 | –9.9 |
| DCVC-DC [25] | –25.9 | –14.4 | –13.9 | –8.8 | –27.7 | –19.1 | 7.6 | –14.6 |
| DCVC-FM [26] | –20.4 | –8.1 | –10.3 | –8.4 | –25.8 | –21.9 | 19.6 | –10.8 |
| Our DCMVC | **–30.6** | **–17.3** | **–14.5** | **–14.4** | **–31.6** | **–28.1** | 0.9 | **–19.4** |

Note: The quality indexes of DCVC-FM are set as 36, 45, 54, 63 to match the bit-rate range of DCVC-DC.

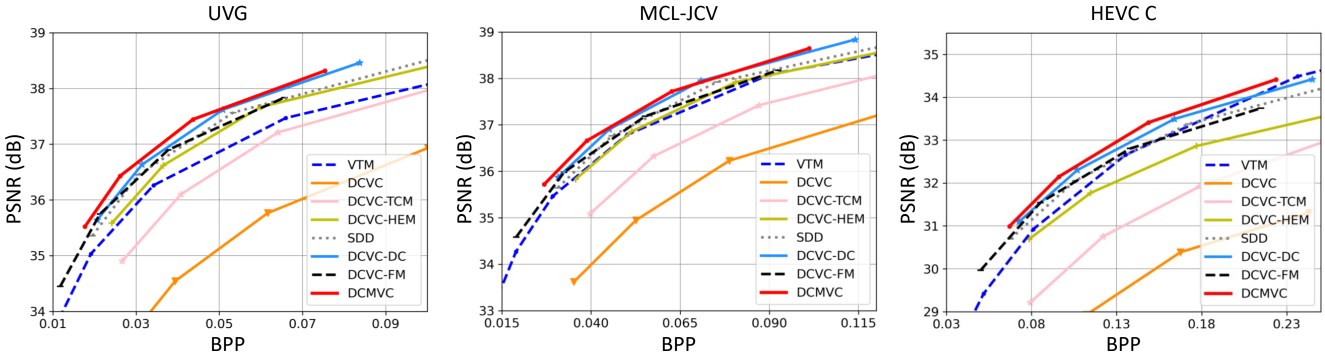


Figure 5. Rate and distortion curve for UVG, MCL-JCV, and HEVC Class C datasets. The comparison is in RGB colorspace measured with PSNR, and the intra-period is set as 32.

change. The negative numbers indicate bitrate saving, while positive numbers indicate bitrate increasing.

We evaluate our scheme against two categories of codecs. For the traditional codec, we choose VTM-13.2[2] as our benchmarks, which is the official reference software of H.266/VVC [6]. For neural video codec, we choose contextual coding-based NVCs as our benchmarks: DCVC [23], DCVC-TCM [42], DCVC-HEM [24], DCVC-DC [25], SDD [43], and DCVC-FM [26].

**Model Training.** We set four base $\lambda$ values (85, 170, 380, 840) to control the weight of distortion in Eq.(2), and set $\alpha$ to 0.2 as the weight of the proposed decoupling loss $L_{decouple}$. Besides, in order to achieve the hierarchical quality structure, we follow [25] to set the hierarchical weight $w_t$ as (0.5, 1.2, 0.5, 0.9) to vary the base $\lambda$ in cycles during the training. Considering the CUDA memory pressure brought by the 32-frame cascaded training, Forward Recomputation Backpropagation (FRB)[3] is deployed during training on 4 NVIDIA A800 PCIe GPUs.

### 4.2. Comparisons with Previous SOTA Methods

Table 1 reports the BD-Rate(%) comparison results in terms of RGB-PSNR with intra-period set as 32 for 96 frames. When testing RGB videos, we convert the testing videos in YUV420 format to RGB format using FFmpeg. As shown in Table 1, compared to H.266/VVC, our DCMVC has achieved an average of 19.4% bitrate saving. It also outperforms previous state-of-the-art (SOTA) neural video compression schemes DCVC-DC [25] (14.6% bitrate saving), and DCVC-FM [26] (10.8% bitrate saving) in each testing dataset. Fig. 5 shows the RD curves on testing datasets, where we can observe our DCMVC significantly surpasses previous SOTA schemes in RGB colorspace.

We also focus on the evaluation under the intra-period setting of –1 to evaluate the model performance in longer prediction chain. Table 2 illustrates the BD-Rate (%) comparison results in terms of RGB-PSNR with intra-period set as –1. From the table, we can see our DCMVC also maintains the best RD performance compared with other benchmarks under both intra-period settings. Specifically, our scheme achieves an average of 22.7% bitrate saving compared with H.266/VVC, and significantly performs better than the previous SOTA neural video compression scheme DCVC-FM [26] (12.6% bitrate saving), which is also re-

[2] https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-13.2

[3] https://qywu.github.io/2019/05/22/explore-gradient-checkpointing.html

Table 2. BD-Rate (%) comparison in RGB colorspace measured with PSNR. 96 frames with intra-period=-1.

| | UVG | MCL-JCV | HEVC B | HEVC C | HEVC D | HEVC E | USTC-TD | Average |
|---|---|---|---|---|---|---|---|---|
| VTM | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **0.0** | 0.0 |
| DCVC [23] | 259.5 | 160.0 | 212.2 | 254.8 | 180.6 | 858.4 | 168.2 | 276.2 |
| DCVC-TCM [42] | 61.6 | 55.5 | 61.7 | 99.6 | 50.4 | 213.9 | 90.2 | 90.4 |
| DCVC-HEM [24] | 1.2 | 4.9 | 10.0 | 30.0 | -1.1 | 68.6 | 27.2 | 20.1 |
| SDD [43] | –5.5 | –0.3 | –2.2 | 16.9 | –18.2 | 46.4 | 12.3 | 7.1 |
| DCVC-DC [25] | –21.2 | –13.0 | –10.8 | –0.1 | –24.2 | –7.7 | 11.9 | –9.3 |
| DCVC-FM [26] | –24.3 | –12.5 | –11.7 | –8.2 | –28.5 | –26.6 | 23.9 | –12.6 |
| Our DCMVC | **–35.6** | **–22.8** | **–16.8** | **–14.8** | **–34.6** | **–36.5** | 1.9 | **-22.7** |

Note: The quality indexes of DCVC-FM are set as 36, 45, 54, 63 to match the bit-rate range of DCVC-DC.

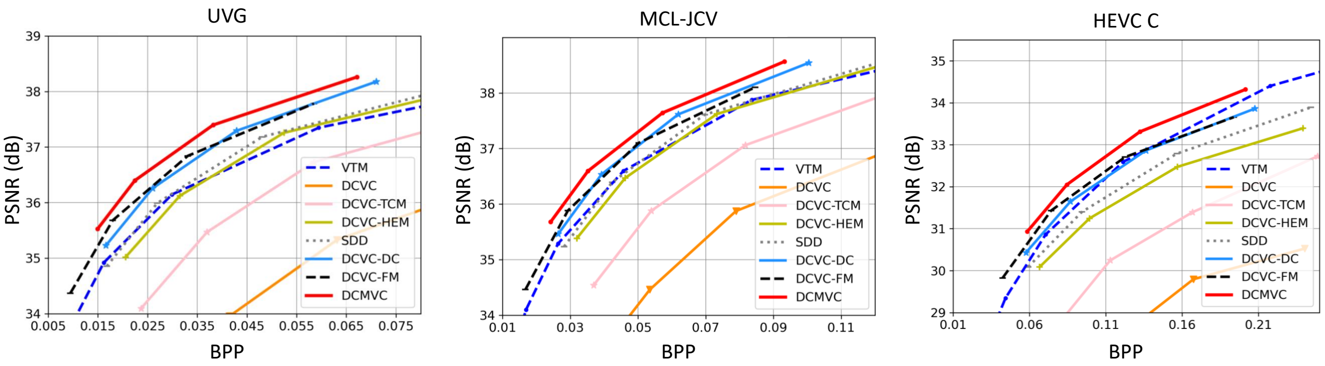
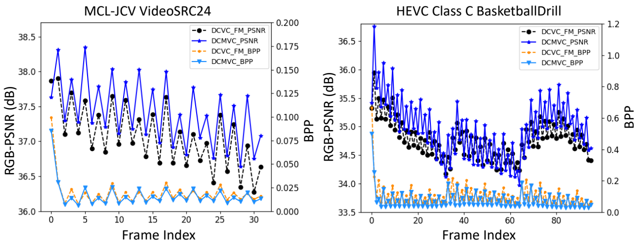


Figure 6. Rate and distortion curve for UVG, MCL-JCV, and HEVC Class C datasets. The comparison is in RGB colorspace measured with PSNR, and the intra-period is set as -1.



(a) Intra-Period 32

(b) Intra-Period -1

Figure 7. Quality and bitrate cost comparison across frames of DCVC-FM and our DCMVC.

ported to be trained with 32-frame datasets. We provide the RD curves in Fig. 6, where we can find our DCMVC brings obvious improvements compared to other benchmarks.

As illustrated in Fig. 7, we analyze the effectiveness of mitigating the error propagation in DCVC-FM and our DCMVC. In Fig. 7 (a), we provide the quality and bitrate cost across frames under the intra-period setting of 32. During the 32-frame temporal period, our DCMVC can better maintain the quality across frames with lower bitrate cost compared to DCVC-FM. Similarly, under the intra-period setting of -1, our DCMVC can better alleviate the quality degradation than DCVC-FM while costing fewer bits as shown in Fig. 7 (b). Observed from the aforementioned experimental results, it demonstrates that our context modulation performs effectively in both short intra-period and long intra-period measured with RD performances while alleviating the error propagation.

Table 3. Ablation study on main techniques (%)

| | $M_a$ | $M_b$ | $M_c$ | $M_d$ | $M_e$ | $M_f$ |
|---|---|---|---|---|---|---|
| flow orientation | | ✓ | | ✓ | ✓ | ✓ |
| Context compensation | | | ✓ | ✓ | ✓ | ✓ |
| Decoupling loss | | | | | ✓ | ✓ |
| long-sequence training | | | | | | ✓ |
| BD-Rate (%) | 0.0 | -1.9 | -3.5 | -4.4 | -5.4 | -10.3 |

## 4.3. Ablation Study

To verify the effectiveness of each proposed method, we conduct ablation studies by evaluating the average RD performance on HEVC datasets with the intra-period set of 32. Table 3 illustrates the bitrate saving of each proposed method, where we can observe that each one achieves performance improvement by progressively adding each of them. We categorize our proposed methods into two main types based on the aspects of improvement: model architecture and optimization strategy.

**Model Structure.** In Table 3, $M_a$ represents our reproduced baseline model DCVC-DC since its training code is

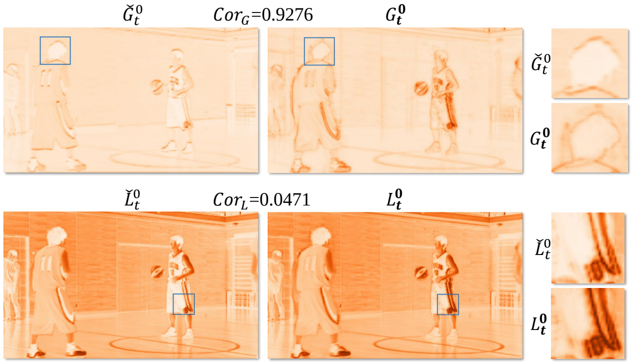


Figure 8. Visualization of the global and local features extracted from the oriented context and propagated context.

not open-sourced. We first evaluate the two main techniques of model structure: flow orientation and context compensation. When the model is equipped with the flow orientation alone ($M_b$), the oriented temporal context $\check{C}_t^0$ generated from the flow orientation and the propagated temporal context $C_t^0$ are concatenated directly, then the concatenated context is refined to generate the compensated context, which can reduce BD-rate by 1.9% over $M_a$. To evaluate the effectiveness of the context compensation, we also design the $M_c$ model that the oriented temporal context $\check{C}_t^0$ is extracted from the prediction frame directly. Besides, we scale up the model scale of $M_c$ model to the final model ($M_d$) to exclude the performance improvement brought by the increase in model complexity. From Table 3, we can find that $M_c$ model achieves 3.5% bitrate saving compared to the baseline model $M_a$. In addition, when enabling flow orientation and context compensation together ($M_d$), there is a total 4.4% BD-rate reduction over $M_a$. Observed the performance of $M_d$ model is better than $M_c$ model, we find that the extracted additional inter-frame correlation in flow orientation can provide a better oriented temporal context, which enables better temporal context modeling in the context compensation.

**Optimization Strategy.** Table 3 also shows the performance improvement achieved by implementing two enhancements in the training strategy without increasing model complexity. Adding the decoupling loss ($M_e$) into the objective function can not only enhance the interpretability of the model but also lead to a 1.0% BD-rate reduction based on $M_c$ model. Fig. 8 illustrates the global and local features extracted from the oriented temporal context and propagated temporal context. First, we can obviously observe that the global encoders pay more attention to the background information, while the local encoders focus more on the texture information. Besides, with the supervision of decoupling loss, the global features are more correlated than the local features. Moreover, the visualization also shows that the global and local features extracted from the propagated temporal context contain more prediction errors than the features extracted from the oriented temporal context. Meanwhile, training with longer sequences (32 frames) can bring more performance improvement, and $M_f$ model can achieve a total 10.3% bitrate saving over the baseline model $M_a$.

Table 4. Complexity comparison.

| | MACs | Params | Encoding Time | Decoding Time |
|---|---|---|---|---|
| DCVC-DC [25] | 2764G | 19.78M | 663ms | 557ms |
| DCVC-FM [26] | 2334G | 18.57M | 587ms | 495ms |
| SDD [43] | 3830G | 21.77M | 968ms | 775ms |
| Our DCMVC | 4131G | 20.98M | 932ms | 810ms |

Note: Tested on NVIDIA 3090 with using 1080p sequences as input.

## 4.4. Complexity Analysis

Table 4 presents the complexity comparison between our DCMVC and three latest SOTA NVCs: DCVC-DC [25], DCVC-FM [26], and SDD [43]. We measure the model complexity with MACs, network parameters, encoding time, and decoding time. The inference is conducted with 1080P frames on a NVIDIA 3090 GPU. Compared with DCVC-DC, SDD, and DCVC-FM, our DCMVC exhibits significantly higher RD performance at the cost of higher MACs and decoding time, while our encoding time is less than SDD. The moderate increase in MACs is mainly due to the complex computation in the largest-resolution temporal context. For the network parameters, our DCMVC has a limited increase in network parameters compared to DCVC-DC and DCVC-FM, while having fewer network parameters over SDD.

# 5. Conclusion

In this paper, context modulation is proposed to address the limitation of inherent reference propagation structure mechanism in conditional coding-based NVC. The proposed flow orientation leverages the reference frame to generate an additional oriented temporal context, then it is used for modulating the propagated temporal context in the proposed context compensation. Powered by our context modulation, DCMVC has further improved the compression ratio than previous SOTA NVCs, while mitigating the error propagation in the prediction chain.

Based on our exploration, investigating efficient temporal context modeling mechanisms still exhibits considerable potential. In temporal context generation, we still adopt rigid warp operation for alignment. For future work, learnable warp can be designed and utilized for more flexible temporal alignment. Besides, we only constrain MV indirectly by RD loss function during the training, which may be suboptimal for temporal dependency modeling. In the future, multiple predefined motion patterns can be learned as priors for efficient temporal context modeling.

## References

[1] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8503–8512, 2020. 2

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

[3] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[4] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *VCEG-M33*, 2001. 5

[5] Frank Bossen. Common hm test conditions and software reference configurations (JCTVC-l1100). *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG*, 2013. 5

[6] Benjamin Bross, Jianle Chen, Jens-Rainer Ohm, Gary J Sullivan, and Ye-Kui Wang. Developments in international video coding standardization after avc, with an overview of versatile video coding (VVC). *Proceedings of the IEEE*, 2021. 6

[7] Hao Chen, Matthew Gwilliam, Ser-Nam Lim, and Abhinav Shrivastava. Hnerv: A hybrid neural representation for videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10270–10279, 2023. 2

[8] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021. 2

[9] Zhenghao Chen, Luping Zhou, Zhihao Hu, and Dong Xu. Group-aware parameter-efficient updating for content-adaptive neural video compression. *arXiv preprint arXiv:2405.04274*, 2024. 2

[10] 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 (CVPR)*, 2020. 2

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 2

[12] Amirhossein Habibian, Ties van Rozendaal, Jakub M. Tomczak, and Taco S. Cohen. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[13] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5718–5727, 2022. 2

[14] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14771–14780, 2021. 2

[15] Yung-Han Ho, Chih-Peng Chang, Peng-Yu Chen, Alessandro Gnutti, and Wen-Hsiao Peng. Canf-vc: Conditional augmented normalizing flows for video compression. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*, pages 207–223. Springer, 2022. 1, 2

[16] Zhihao Hu, Zhenghao Chen, Dong Xu, Guo Lu, Wanli Ouyang, and Shuhang Gu. Improving deep video compression by resolution-adaptive flow coding. In *European Conference on Computer Vision (ECCV)*, pages 193–209. Springer, 2020. 2

[17] Zhihao Hu, Guo Lu, Jinyang Guo, Shan Liu, Wei Jiang, and Dong Xu. Coarse-to-fine deep video coding with hyperprior-guided mode prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5921–5930, 2022. 2

[18] Zhihao Hu, Guo Lu, and Dong Xu. FVC: A new framework towards deep video compression in feature space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1502–1511, 2021. 2

[19] Zhihao Hu and Dong Xu. Complexity-guided slimmable decoder for efficient deep video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14358–14367, 2023. 2

[20] Wei Jiang, Junru Li, Kai Zhang, and Li Zhang. Ecvc: Exploiting non-local correlations in multiple frames for contextual video compression. *arXiv preprint arXiv:2410.09706*, 2024. 1, 2

[21] Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, and Emilien Dupont. C3: High-performance and low-complexity neural compression from a single image or video. *arXiv preprint arXiv:2312.02753*, 2023. 2

[22] Ho Man Kwan, Ge Gao, Fan Zhang, Andrew Gower, and David Bull. Hinerv: Video compression with hierarchical encoding-based neural representation. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. 2

[23] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:18114–18125, 2021. 1, 2, 3, 6, 7

[24] Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1503–1511, 2022. 1, 2, 3, 6, 7

[25] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22616–22626, 2023. 1, 2, 3, 4, 6, 7, 8

[26] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with feature modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26099–26108, 2024. 1, 2, 3, 6, 7, 8

[27] Meng Li, Yibo Shi, Jing Wang, and Yunqi Huang. High visual-fidelity learned video compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8057–8066, 2023. 2

[28] Zhuoyuan Li, Junqi Liao, Chuanbo Tang, Haotian Zhang, Yuqi Li, Yifan Bian, Xihua Sheng, Xinmin Feng, Yao Li, Changsheng Gao, et al. Ustc-td: A test dataset and benchmark for image and video coding in 2020s. *arXiv preprint arXiv:2409.08481*, 2024. 5

[29] 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 (CVPR)*, pages 3546–3554, 2020. 2

[30] Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14388–14397, 2023. 2

[31] Yuxi Liu, Wenhan Yang, Huihui Bai, Yunchao Wei, and Yao Zhao. Region-adaptive transform with segmentation prior for image compression. In *European Conference on Computer Vision (ECCV)*, pages 181–197. Springer, 2025. 2

[32] 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 *European Conference on Computer Vision (ECCV)*, pages 456–472. Springer, 2020. 2

[33] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11006–11015, 2019. 2

[34] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. An end-to-end learning framework for video compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[35] Kede Ma, Zhengfang Duanmu, Hanwei Zhu, Yuming Fang, and Zhou Wang. Deep guided learning for fast multi-exposure image fusion. *IEEE Transactions on Image Processing*, 29:2808–2819, 2019. 4

[36] Kede Ma, Hui Li, Hongwei Yong, Zhou Wang, Deyu Meng, and Lei Zhang. Robust multi-exposure image fusion: a structural patch decomposition approach. *IEEE Transactions on Image Processing*, 26(5):2519–2532, 2017. 4

[37] 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. 5

[38] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. IEEE, 2020. 2

[39] Jorge Pessoa, Helena Aidos, Pedro Tomás, and Mário 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. 2

[40] Linfeng Qi, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Motion information propagation for neural video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6111–6120, 2023. 1, 2, 3

[41] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4161–4170, 2017. 2, 3

[42] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia*, 2022. 1, 2, 3, 6, 7

[43] Xihua Sheng, Li Li, Dong Liu, and Houqiang Li. Spatial decomposition and temporal fusion based inter prediction for learned video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 1, 2, 6, 7, 8

[44] Xihua Sheng, Chuanbo Tang, Li Li, Dong Liu, and Feng Wu. Nvc-1b: A large neural video coding model. *arXiv preprint arXiv:2407.19402*, 2024. 2

[45] Yibo Shi, Yunying Ge, Jing Wang, and Jue Mao. Alphavc: High-performance and efficient learned video compression. In *European Conference on Computer Vision (ECCV)*, pages 616–631. Springer, 2022. 2

[46] 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 (CVPR)*, pages 8934–8943, 2018. 2

[47] Wenyu Sun, Chen Tang, Weigui Li, Zhuqing Yuan, Huazhong Yang, and Yongpan Liu. High-quality single-model deep video compression with frame-conv3d and multi-frame differential modulation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 239–254. Springer, 2020. 2

[48] Chuanbo Tang, Xihua Sheng, Zhuoyuan Li, Haotian Zhang, Li Li, and Dong Liu. Offline and online optical flow enhancement for deep video compression. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 5118–5126, 2024. 1, 2

[49] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 2

[50] 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. 5

[51] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019. 5

[52] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6628–6637, 2020. 2

[53] Haotian Zhang, Feihong Mei, Junqi Liao, Li Li, Houqiang Li, and Dong Liu. Practical learned image compression with online encoder optimization. In *2024 Picture Coding Symposium (PCS)*, pages 1–5. IEEE, 2024. 2

[54] Qi Zhao, M Salman Asif, and Zhan Ma. Dnerv: Modeling inherent dynamics via difference neural representation for videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2031–2040, 2023. 2

[55] Zixiang Zhao, Haowen Bai, Jiangshe Zhang, Yulun Zhang, Shuang Xu, Zudi Lin, Radu Timofte, and Luc Van Gool. Cddfuse: Correlation-driven dual-branch feature decomposition for multi-modality image fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5906–5916, 2023. 4

[56] Zixiang Zhao, Shuang Xu, Jiangshe Zhang, Chengyang Liang, Chunxia Zhang, and Junmin Liu. Efficient and model-based infrared and visible image fusion via algorithm unrolling. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1186–1196, 2021. 4