Motion Information Propagation for Neural Video Compression

Linfeng Qi[1*] Jiahao Li [2] Bin Li[2] Houqiang Li[1] Yan Lu[2]
[1] University of Science and Technology of China [2] Microsoft Research Asia
qlf324@mail.ustc.edu.cn, lihq@ustc.edu.cn, {li.jiahao, libin, yanlu}@microsoft.com

Abstract

In most existing neural video codecs, the information flow therein is uni-directional, where only motion coding provides motion vectors for frame coding. In this paper, we argue that, through information interactions, the synergy between motion coding and frame coding can be achieved. We effectively introduce bi-directional information interactions between motion coding and frame coding via our Motion Information Propagation. When generating the temporal contexts for frame coding, the high-dimension motion feature from the motion decoder serves as motion guidance to mitigate the alignment errors. Meanwhile, besides assisting frame coding at the current time step, the feature from context generation will be propagated as motion condition when coding the subsequent motion latent. Through the cycle of such interactions, feature propagation on motion coding is built, strengthening the capacity of exploiting long-range temporal correlation. In addition, we propose hybrid context generation to exploit the multi-scale context features and provide better motion condition. Experiments show that our method can achieve 12.9% bit rate saving over the previous SOTA neural video codec.

1. Introduction

How to better utilize the temporal correlation across video frames is one of the core problems throughout the line of video compression research, for both traditional video codec and neural video codec.

In traditional video codec, such as H.264 [41] and HEVC [36], temporal correlation is captured by rule-based motion-compensated prediction. It is assumed that the pixels within a block share the same motion. For the current coding block, the best motion vector corresponds to the matching block that results in the least bit cost. It is searched and then is used for motion compensation. Such motion-compensated prediction is usually a local optimization limited to two frames. The correlation across multiple frames is neglected, which limits the capacity to utilize the temporal characteristics of videos.

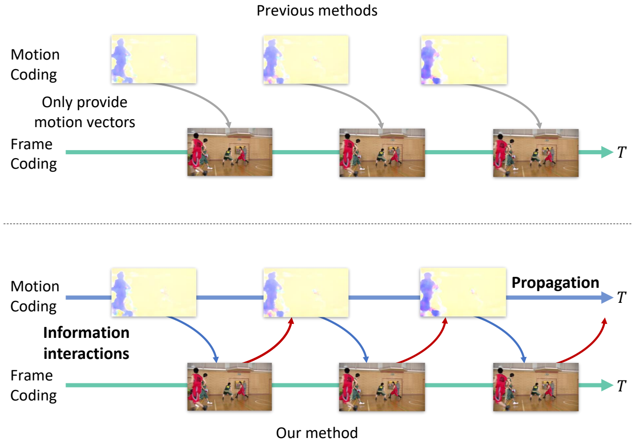
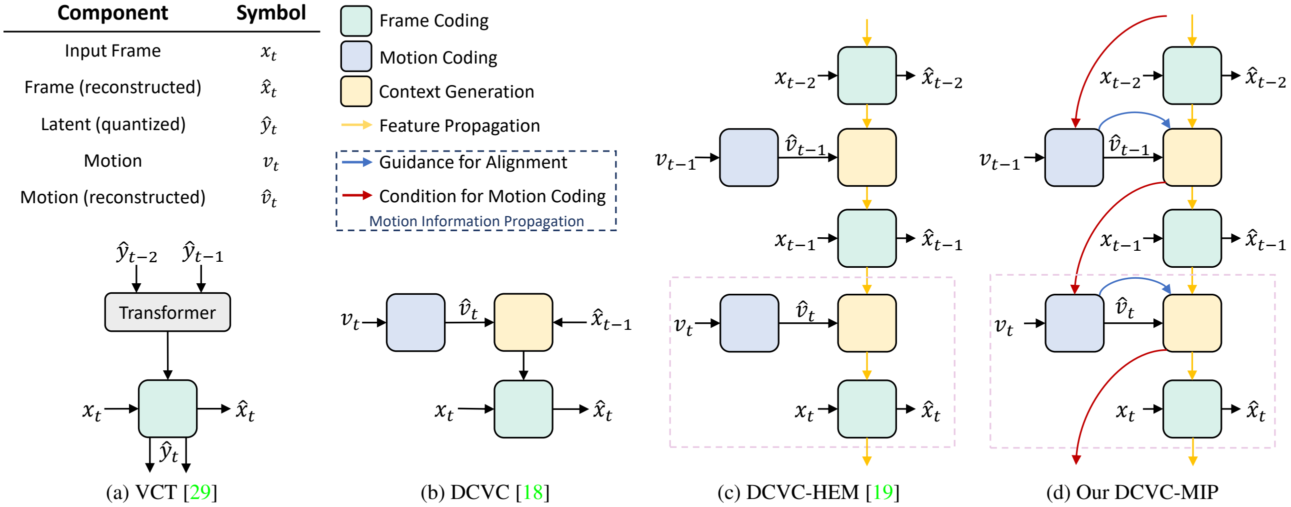


Figure 1. Compared with previous methods, besides the propagation for frame coding, our method: 1. introduces bi-directional information interactions between motion coding and frame coding. 2. enables propagation for motion coding through the interactions.

With the flourish of deep learning, it is possible for neural video codec [10, 12–14, 18, 25, 26, 45] to access pixel-wise optical flow and long-range temporal information. Typically, the entire coding process can be roughly divided into three steps: motion coding, context generation, and frame coding. The motion is firstly estimated and transmitted. Then the temporal context (prediction is also a kind of context in residual coding framework) is obtained through motion compensation, where handcrafted blockwise matching is replaced by optical flow based pixel-wise warping. Finally, with the assistance of temporal context, the frame is encoded and decoded. Frame coding draws more attention in most existing methods.

As shown in Figure 1, in the previous methods, the information flow therein is uni-directional, where only motion coding provides motion vectors for frame coding. By contrast, we emphasize the synergy between motion coding and frame coding. As far as we know, we are the first to introduce effective bi-directional information interactions between motion coding and frame coding. Through the cycle of such interactions, feature propagation on motion coding

[*] This work was done when Linfeng Qi was an intern at Microsoft Research Asia.



(a) VCT [29] (b) DCVC [18] (c) DCVC-HEM [19] (d) Our DCVC-MIP

Figure 2. Comparison for several recent methods. MIP stands for motion information propagation.

is enabled to exploit long-range motion information.

Figure 2 provides a more detailed comparison for several recent methods. VCT [29] uses transformers [38] to capture the contexts among latent representations. It is free from motion coding and beyond our discussion. DCVC [18] uses the motion to extract context from the previously decoded frame. Its following work DCVC-TCM [35] and DCVC-HEM [19] further use feature propagation to improve frame coding. However, in these existing works, there are no effective information interactions between motion coding and frame coding. Furthermore, propagation mechanism which captures long-range temporal correlation is also neglected for motion coding.

Thus, based on previous SOTA DCVC-HEM [19], we build our DCVC-MIP framework, which introduces bi-directional information interactions via proposed Motion Information Propagation. When generating the temporal contexts for frame coding, the information is passed from motion coding to context generation via motion guidance, which aims at mitigating the alignment errors. Meanwhile, besides the feature propagation between context generation and frame coding, information is also passed from context generation to motion coding, and it is as motion condition to help reduce the bit cost for coding the subsequent motion latent. The bi-directional information flow can be described as follows: motion coding $\rightarrow$ context generation $\rightarrow$ frame coding, and frame coding $\rightarrow$ context generation $\rightarrow$ motion coding. Through the cycle of such interactions, feature propagation on motion coding is implicitly built, strengthening the capacity of exploiting long-range temporal correlation. We emphasize the importance of introducing Motion Information Propagation as follows: 1. Although the bit cost for motion coding is less than frame coding, it is still not neglectable. Motion Information Propagation can implicitly capture long-range motion information, helping save the bit cost for motion coding. 2. By introducing the information interactions, the synergy between motion coding and frame coding could be achieved, leading to mutual improvement.

Another advance of DCVC-MIP is the hybrid context generation. There are various types of motion in the video, from simple camera motion to sophisticated human activities. In order to handle such motion information, the capacity of learning hybrid temporal context is important. Thus, we propose our hybrid context generation module as follows: for the high-resolution context feature, we adopt offset diversity [6], which predicts multiple offsets and masks to get more accurate feature alignment and enhance the reconstruction details. The inner feature, which is used for generating offsets and masks, contains rich motion information and is suitable to be propagated as motion condition; for the low-resolution context feature, we introduce transformer based context refinement, which enlarges the receptive field and captures better correlation to refine the context. Our contributions can be summarized as follows:

- We are the first to introduce bi-directional information interactions between motion coding and frame coding, which assist to achieve the synergy between them.
- Our Motion Information Propagation not only mitigates the alignment errors, but also saves the bit cost for motion coding. With the propagated features, long-range motion information is implicitly utilized, leading to a better rate distortion trade-off.
- We propose a hybrid context generation module, which not only strengthens the multi-scale context feature mining, but also provides better motion condition for Motion Information Propagation.
- Compared with the previous SOTA neural video codec,

our method achieves 12.9% bit rate saving, which demonstrates the effectiveness of our method.

2. Related Work

2.1. Learned Video Compression

Neural video compression attracts increasing interests in recent years. The residual coding framework is explored by many works, such as [4, 8, 12–14, 21–28, 33, 34, 44]. 3D autoencoder based methods [10, 31, 37] are also proposed, which naturally expand the input dimension to compress multiple frames simultaneously. However, 3D autoencoder will significantly increase the computation cost and encoding delay. The conditional coding, which learns implicit temporal contexts, also achieves impressive performance, such as [11, 15–19, 29, 35]. Our method also belongs to this category, but focuses more on building effective information interactions and feature propagation between motion coding and frame coding.

2.2. Motion Coding

Recently, increasing attention has been paid to motion coding [4, 9, 11, 21, 32, 33, 43]. Agustsson *et al.* [4] propose scale-space flow to represent the motion and perform motion compensation, which learns to adaptively blur the reference frame according to the uncertainty of the flow map. RLVC [43] employs a recurrent model to propagate the temporal information for motion codind and frame coding. M-LVC [21] applies predictive motion coding by extrapolating a flow map from multiple previously decoded motion fields. ELF-VC [33] further estimates a residual motion field between the motion-compensated frame and the target frame. CANF-VC [11] extends the conditional coding idea to motion coding, extrapolating a reference flow map as motion condition. Gao *et al.* [9] estimate a temporally consistent motion field by aggregating the motion prediction of both the original and the decoded reference frame. Pourreza *et al.* [32] learn the second-order redundancy in motion and residual via predictors.

These existing methods usually rely on extra specially designed modules for motion coding, which increases the computation cost and may need additional pre-training stage. By contrast, we use the off-the-shelf feature from our hybrid context generation as motion condition, which just brings a little computation cost and does not change the training strategy. In particular, the information interactions between motion coding and frame coding are neglected in the previous methods. By contrast, we effectively introduce bi-directional information interactions between motion coding and frame coding to achieve mutual improvements. Through the cycle of such interactions, the feature propagation on motion coding is implicitly built. During training, long-range temporal correlation can be captured through the propagated features, implicitly reducing the higher-order redundancy.

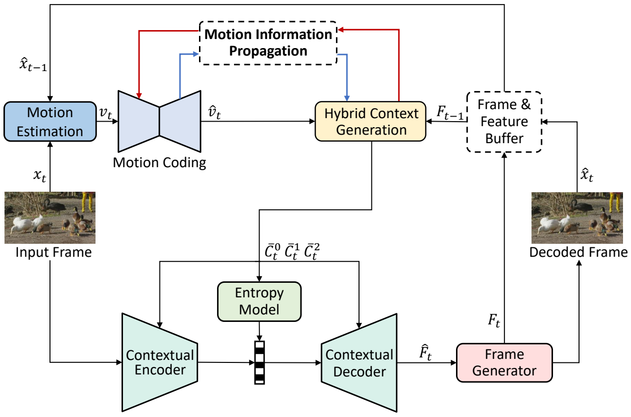


Figure 3. Illustration for our DCVC-MIP.

3. Method

3.1. Overview

Figure 3 provides a high-level overview of our proposed DCVC-MIP. It is noted that our DCVC-MIP is built on DCVC-HEM [19]. Given the input frame x_t, the objective is reconstructing the high-quality video frame $\hat{x}_t$ with bit cost as less as possible. The corresponding motion v_t between the input frame x_t and the previous decoded frame $\hat{x}_{t-1}$ is first estimated. Through motion coding, the decoded motion $\hat{v}_t$ is obtained. The proposed Motion Information Propagation associates motion coding and context generation to build effective interactions between motion coding and frame coding. The motion guidance (the blue arrow) and motion condition (the red arrow) help to achieve mutual improvements. Taking the decoded motion $\hat{v}_t$ and propagated feature F_{t-1} as input, the proposed hybrid context generation module generates a set of hybrid temporal contexts $\bar{C}_t^0, \bar{C}_t^1, \bar{C}_t^2$, which will be used for both conditional encoding and decoding. We will introduce the proposed Motion Information Propagation and hybrid context generation in Section 3.2 and Section 3.3.

3.2. Motion Information Propagation

Figure 4 illustrates how the proposed Motion Information Propagation works in the left part and the structure of offset diversity [6] in the right part. We choose offset diversity to align the propagated feature, rather than deformable convolution [7] adopted by existing methods [14, 45]. The reason is that compared with deformable convolution, offset diversity has two superiorities: 1. It provides a more flexible way to increase the diversity of offsets to get a better trade-off. We can set the number of offsets as we want, rather than a fixed number, *i.e.*, the square of the kernel size in deformable convolution. 2. Compared with deformable

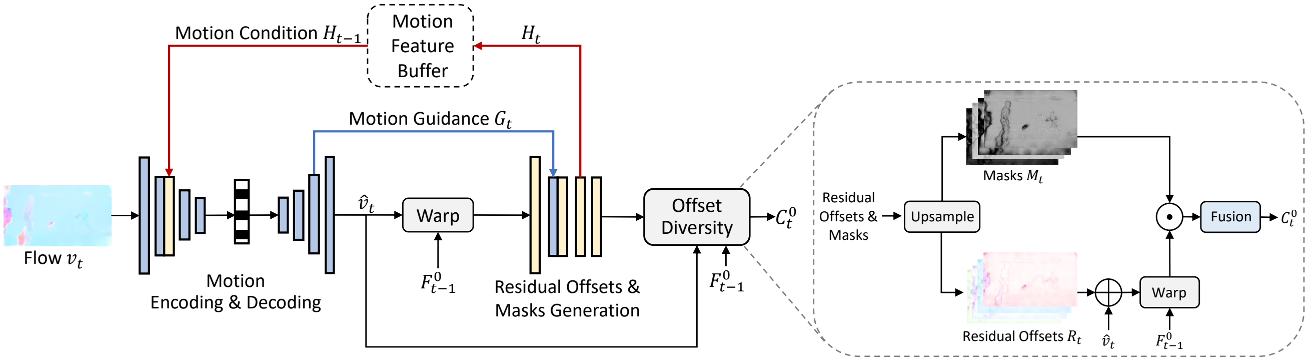


Figure 4. Illustration for the proposed Motion Information Propagation.

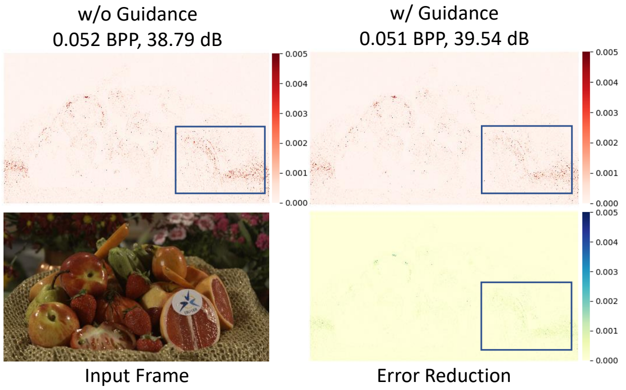


Figure 5. Visualization for the effectiveness of motion guidance. BPP means bits per pixel.

convolution, offset diversity is implemented by simpler operations and is easier to be optimized.

As shown in Figure 4, we first get the decoded motion $\hat{v}_t$ from the estimated optical flow v_t. It is used to warp the propagated feature F^0_{t-1} (extracted from F_{t-1}, see Section 3.3 for details). Then more accurate alignment is performed by the offset diversity [6] module. Specifically speaking, to align the feature F^0_{t-1} better, we will predict multiple residual offsets and masks (correspond to the confidence for the residual offsets), which are of $1/2$ resolution for saving computation cost. To get better residual offsets and masks, besides using the warped feature $Warp(F^0_{t-1}, \hat{v}_t)$ as input, we also use the motion feature G_t from the motion decoder as **motion guidance**. It is superior to 2-dimensional motion vector because G_t also contains the propagated high-dimensional information from the previous time steps. The sufficient motion information can help predict more accurate residual offsets and masks, mitigating the alignment errors.

The predicted residual offsets and masks are upsampled, denoted as R_t and M_t, respectively. Then the refined alignment is performed as follows: 1. The residual offsets R_t are added by the decoded motion $\hat{v}_t$ to get the final offsets, which are used for warping F^0_{t-1}. 2. The warped results will be modulated by the masks M_t and then fused to generate the context feature C^0_t for further use.

$$C^0_t = Fusion(Warp(F^0_{t-1}, R_t + \hat{v}_t) \odot M_t) \tag{1}$$

The fusion is implemented by a 1×1 convolution for aggregating the warped results of different offsets and masks.

Figure 5 provides qualitative results to show the effectiveness of motion guidance. We can see that, compared to the model without motion guidance, the model with motion guidance is able to provide obvious error reduction. It demonstrates that introducing motion guidance can effectively boost the performance. We infer that the motion feature G_t from motion decoder could assist to learn offsets and masks better, leading to more accurate alignment.

The feature for generating residual offsets and masks tends to contain rich motion information. We choose the feature H_t as **motion condition**. It is propagated to the motion encoder and also serves as an extra prior in the entropy model for the next frame (see supplementary material for details). Besides saving bit cost for coding the motion latent, motion condition can also enrich the information of its corresponding motion guidance. Such a motion information propagation mechanism implicitly captures higher-order motion information, benefiting the motion coding for multiple frames jointly. It is noted that though participating in the propagation, offset diversity [6] is a part of hybrid context generation. We just use the off-the-shelf inner features. Our Motion Information Propagation actually does not introduce extra modules, thus is very efficient.

In Figure 6, we investigate the influence of motion condition and motion guidance on motion coding. The top row shows the pixel-level bit allocation for coding the motion latent. Comparing the regions indicated by the red box in Figure 6 (a) and Figure 6 (b), we can see that introducing motion condition not only brings bit rate saving for motion coding, but also leads to a better bit allocation. More bits are allocated for objects in the foreground with larger motion and fewer bits are allocated for the background with smaller motion. In Figure 6 (c), when motion condition and motion guidance are both introduced, such effect is more

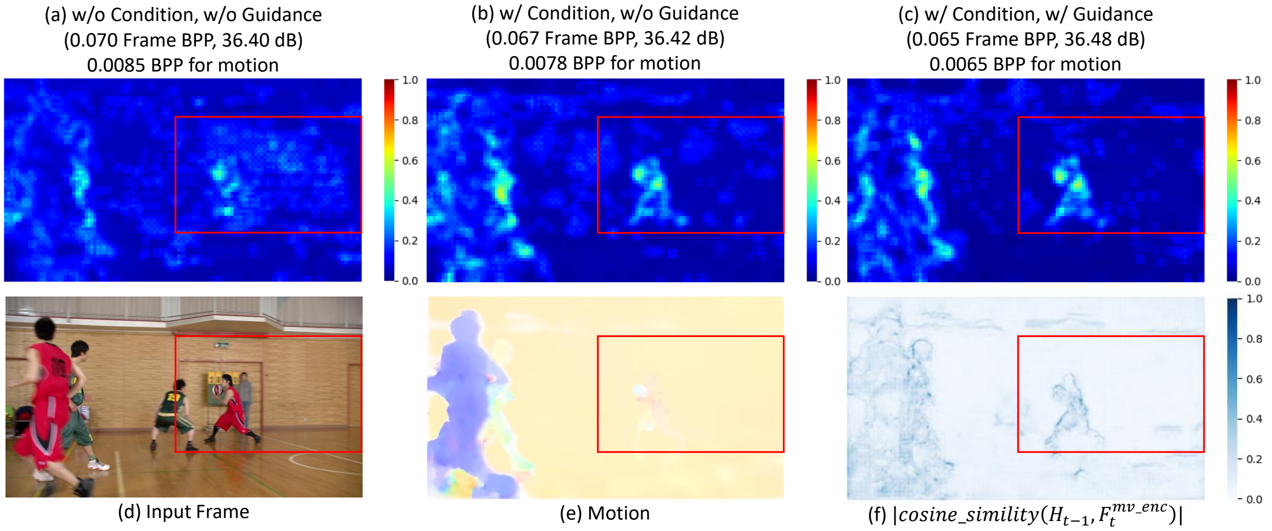

(a) w/o Condition, w/o Guidance
(0.070 Frame BPP, 36.40 dB)
0.0085 BPP for motion
(b) w/ Condition, w/o Guidance
(0.067 Frame BPP, 36.42 dB)
0.0078 BPP for motion
(c) w/ Condition, w/ Guidance
(0.065 Frame BPP, 36.48 dB)
0.0065 BPP for motion
1.0
0.8
0.6
0.4
0.2
0.0
(d) Input Frame
(e) Motion
(f) $|cosine\_simility(H_{t-1}, F_t^{mv\_enc})|$

Figure 6. Visualization for the bit rate saving on motion coding. (a), (b), and (c) illustrate the pixel-level bit allocation for motion coding.

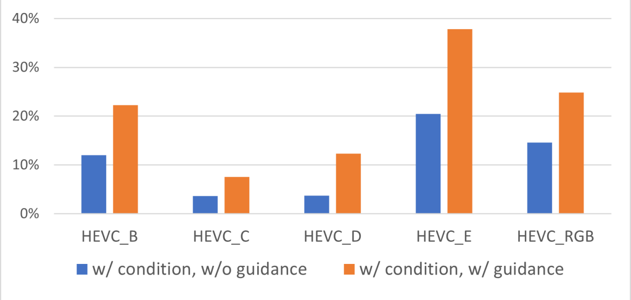


Figure 7. Relative bit cost reduction for motion coding. The results are averaged on the sequences for each dataset.

obvious. That is because the complete interaction cycle can enable the more effective feature propagation on motion coding, which can better exploit long-range temporal correlation. We also use cosine similarity to measure the correlation between the propagated feature H_{t-1} and the feature from the motion encoder $F_t^{mv\text{-}enc}$, as shown in Figure 6 (f). It can provide some information to help understand why our method is able to save the bit cost and make the bit allocation more reasonable. Figure 7 further illustrates the relative bit cost reduction for motion coding on different datasets. The result is consistent with the visualization. It is shown that introducing motion condition can reduce the bit cost for motion coding by a large margin. When the motion condition and motion guidance are both introduced to make up the full Motion Information Propagation, the bit cost will be further reduced.

3.3. Hybrid Context Generation

We follow [19] to learn multi-scale temporal contexts from the propagated feature F_{t-1}. However, in [19], temporal contexts from different scales are processed in the same way, *i.e.*, using flow warping. It neglects the characteristics of contexts from different scales, leading to sub-optimal performance. Our conceptual idea is to learn the multi-scale temporal contexts in a hybrid way, applying different strategies at different scales.

The proposed hybrid context generation is depicted in Figure 9. Firstly, we generate multi-scale features $F_{t-1}^0, F_{t-1}^1, F_{t-1}^2$ from the propagated feature F_{t-1} through convolution layers. These features will be firstly aligned by the motion of the corresponding scale. For the high-resolution context feature, more accurate alignment is expected to enhance the reconstruction details. Thus, we adopt offset diversity [6], which predicts extra offsets and modulating masks for better alignment. The inner feature for generating residual offsets and masks tends to contain rich motion information and is suitable to serve as motion condition for Motion Information Propagation. Using such off-the-shelf feature only introduces a little computation cost, but effectively boosts the performance.

With the resolution decreases, the information is aggregated progressively. For the low-resolution context feature, transformer based context refinement is applied to enlarge the receptive field and model long-range dependencies. Figure 8 provides the intuition about the effectiveness of this module. The large motion between the previously decoded frame $\hat{x}_{t-1}$ and the current frame x_t leads to an inaccurate alignment, *e.g.*, the missing pixels of the basketball annotated by the green box in the warped frame $Warp(\hat{x}_{t-1}, \hat{v}_t)$. The failures in the warped frame can reveal the collapsed regions in the context feature. We visualize the attention in the transformer based context refinement by summing the attention values corresponding to the region inside the green box. It is observed that the most relevant pixels are mainly distributed in the boundary region, which is more hopeful to be refined than the totally missing ones. Intuitively, the transformer based context refinement can focus more on these ambiguous regions and mitigate the problem caused by the

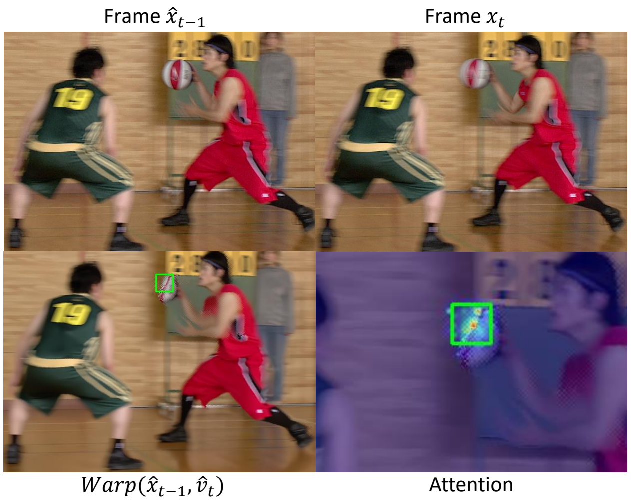


Figure 8. Visualization for transformer based context refinement.

alignment failures. This module consists of two convolution layers and a residual Swin Transformer block (RSTB) [20]. It does not bring too much computation cost because the computation is done at low-resolution. The details can be found in our supplementary material.

Finally, we follows DCVC-HEM [19] and DCVC-TCM [35] to fuse the multi-scale context features hierarchically. We upsample the context feature of the lower resolution and concatenate it with the corresponding feature of the higher resolution. Then convolution layers and residual connections are applied to generate the output hybrid temporal contexts $\bar{C}_t^0, \bar{C}_t^1, \bar{C}_t^2$, which will be propagated for coding the next video frame.

4. Experiments

4.1. Experimental Setup

Datasets. For training, we use the Vimeo-90k [42] dataset. We randomly crop the videos into 256 × 256 patches. For testing, we follow DCVC-HEM [19] to use the same datasets, *i.e.*, UVG [30], MCL-JCV [39] and HEVC [36] Class B, C, D, E and RGB.

Test settings. For the main experiments, we follow the settings of DCVC-HEM [19] to test 96 frames for each video. The intra period is set to 32, which is closer to the practical usage in the real applications. We use BD-Rate [5] for evaluating the compression ratio. VTM-13.2, which represents the best encoder of H.266, is used as the anchor for performance comparison. We follow [19] to configure VTM-13.2. Besides VTM-13.2, we also provide the results of x265 (veryslow preset) and HM-16.20. For neural video codec, we compare our proposed DCVC-MIP with the previous SOTA neural video codecs including DVCPro [27], M-LVC [21], CANF-VC [11], DCVC [18], DCVC-TCM [35] and DCVC-HEM [19].

Implementation details. Considering the excellent performance and extensibility of DCVC-HEM [19], our codec is built on it. During training, our target is to get better RD trade-off, with the loss function:

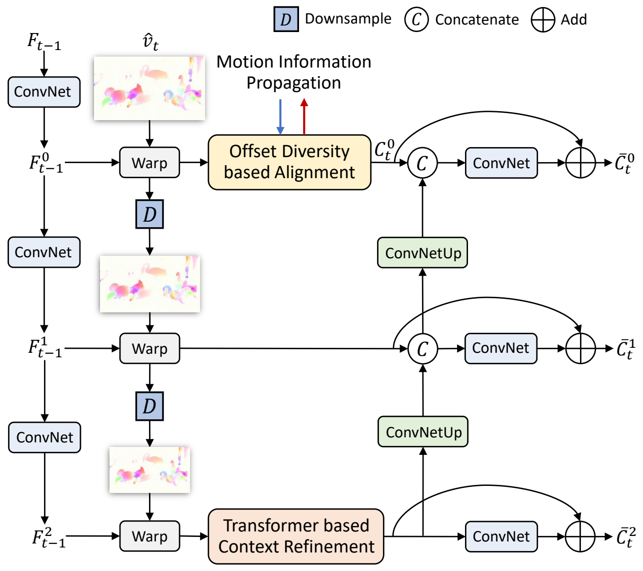


Figure 9. Illustration for hybrid context generation.

$$L = \lambda D + R = \lambda d(x_t, \hat{x}_t) + R_{\hat{v}} + R_{\hat{f}} \tag{2}$$

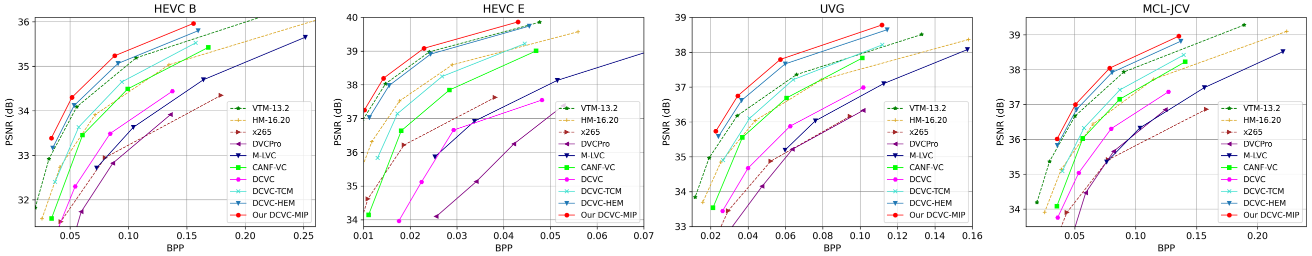
where $d(x_t, \hat{x}_t)$ is the distortion between the input video frame x_t and the reconstructed frame $\hat{x}_t$. $d(\cdot)$ refers to mean square error (MSE) when targeted at PSNR or 1 - MS-SSIM [40] when targeted at SSIM. $R_{\hat{v}}$ and $R_{\hat{f}}$ refer to the bit rate of the motion coding and the frame coding. 4 λ values (85, 170, 380, 840) are used for training variable bitrate model.

4.2. Comparisons with SOTA Methods

Table 1 and Table 2 report the BD-Rate(%) results in terms of PSNR and MS-SSIM. The anchor is VTM-13.2. The negative number indicates bit rate saving while the positive number indicates bit rate increase. As shown in Table 1, our DCVC-MIP gets an average of 16.6% bit rate saving over VTM-13.2 in terms of PSNR. It also achieves consistent performance improvements over DCVC-HEM [19], on all the datasets. The performance gain is especially obvious on HEVC E, where our method outperforms VTM-13.2 by 10.0% but DCVC-HEM has 7.1% bit rate increase. When using DCVC-HEM as anchor, our average bit rate saving is 12.9%. Figure 10 illustrates the RD curves to verify the effectiveness of our method. When targeting at MS-SSIM, our method also brings improvements. As shown in Table 2, our method gets an average of 50.8% bit rate saving compared with VTM-13.2. Because the performance of DCVC-HEM [19] is very high, there is not too much room for improvement. The relative gain is not as large as that in terms of PSNR.

Table 1. BD-Rate (%) results for PSNR in comparison with VTM-13.2.

Method	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	HEVC RGB	Average
VTM-13.2 [2]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HM-16.20 [1]	40.5	45.4	40.4	40.9	36.0	46.2	42.1	41.6
x265 [3]	191.5	160.3	143.4	105.2	96.1	128.4	151.2	139.4
DVCPro [27]	227.0	180.8	209.8	220.6	166.4	446.2	178.5	232.8
M-LVC [21]	113.6	124.0	118.0	213.7	166.5	237.6	151.2	160.7
CANF-VC [11]	49.4	52.0	52.5	68.4	52.9	110.6	74.3	65.7
DCVC [18]	126.1	98.2	115.0	150.8	109.6	266.2	109.6	139.4
DCVC-TCM [35]	17.1	30.6	28.5	60.5	27.8	67.3	17.9	35.7
DCVC-HEM [19]	-18.2	-6.4	-5.1	15.0	-8.9	7.1	-16.4	-4.7
Our DCVC-MIP	**-27.7**	**-14.1**	**-17.3**	**0.8**	**-22.1**	**-10.0**	**-25.5**	**-16.6**



(a) HEVC Class B, PSNR (b) HEVC Class E, PSNR (c) UVG, PSNR (d) MCL-JCV, PSNR

Figure 10. RD-Curves with intra perid 32.

4.3. Ablation Study

Main Components. The proposed hybrid context generation and Motion Information Propagation are closely related. We divide them into several main components and conduct the ablation study on these components. As shown in Table 3, we progressively add each of them to demonstrate the effectiveness of our method. It is noted that some of these models may serve as anchor in other tables. For convenience, we denote them as Model A, B, C, D and E. All the components are disabled in the anchor method Model A. Through introducing the offset diversity [6], Model B outperforms Model A by 2.6%. Equipped with motion guidance, Model C achieves an additional 3.3% gain. By enabling motion condition, Model D brings an extra 3.5% performance improvement. Model E applies transformer based context refinement to refine the low-resolution feature and boosts the performance by 3.5% further, amounting to 10.4% bit rate saving.

Separate bit comparison. Table 4 show the separate bit comparison of the rate point which is trained with $\lambda = 840$. Through the proposed bi-directional information interactions, improved motion coding can provide better motion guidance to mitigate the alignment errors, and then enhance frame coding. In turn, enhanced frame coding can provide better motion condition to help motion coding. Compared with the anchor, besides improving frame coding, our model also provides 23.81% bit reduction for motion coding.

Motion Information Propagation. For Motion Information Propagation, we explore the influence of motion guidance and motion condition. As shown in Table 5, Model B serves as the anchor, where motion guidance and motion condition are both disabled. It is shown that using the decoded motion $\hat{v}_t$ as motion guidance can bring a 2.4% gain. The motion feature G_t is a more effective choice, which shows a 3.2% performance improvement over the anchor. It is shown that propagating the feature H_{t-1} as motion condition can further boost the performance, leading to 6.9% bit rate saving. It is because H_{t-1} is used for generating the residual offsets and masks, which contains rich motion information. We also try using the predicted residual offsets R_{t-1} as motion condition, but its performance is not as good as the inner feature H_{t-1}.

Transformer Based Context Refinement. Considering that transformer is drawing increasing interests, we specially perform ablation study for the transformer based context refinement (denoted as TCR for simplicity) in Table 6. We use Model D as the anchor, which is denoted as w/o TCR. It is shown that introducing TCR to different scales all brings improvements. However, applying TCR at the original scale and $1/2$ scale will cost plenty of computation, but

Table 2. BD-Rate (%) results for MS-SSIM in comparison with VTM-13.2.

Method	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	HEVC RGB	Average
VTM-13.2 [2]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HM-16.20 [1]	36.9	43.7	36.7	38.7	34.9	40.5	37.2	38.4
x265 [3]	150.5	137.6	129.3	109.5	101.8	109.0	121.9	122.8
DVCPro [27]	68.1	37.8	61.7	59.1	23.9	212.5	57.3	74.3
CANF-VC [11]	27.7	7.4	25.3	23.8	10.6	128.7	47.0	38.6
DCVC [18]	33.6	4.7	31.0	22.8	1.2	124.7	36.5	36.4
DCVC-TCM [35]	-10.1	-24.4	-24.1	-23.3	-37.3	-8.0	-25.9	-21.9
DCVC-HEM [19]	-35.1	-46.8	-48.1	-44.6	-55.7	-47.5	-47.0	-46.4
Our DCVC-MIP	**-37.1**	**-49.2**	**-51.9**	**-52.2**	**-60.0**	**-54.0**	**-51.5**	**-50.8**

Table 3. Ablation study on the main components.

Model	Offset Diversity [6]	Motion Guidance	Motion Condition	Transformer based Context Refinement	Avg
A	-	-	-	-	0.0
B	✓	-	-	-	-2.6
C	✓	✓	-	-	-5.9
D	✓	✓	✓	-	-9.4
E	✓	✓	✓	✓	**-12.9**

Table 4. Bit comparison of rate point trained with $\lambda = 840$.

Model	bpp_{motion}	bpp_{frame}	Reduction on bpp_{motion}	Reduction on bpp_{frame}	bpp_{both}	PSNR
A	0.0084	0.1408	0.0%	0.0%	0.1492	35.19
B	0.0079	0.1401	-5.95%	-0.50%	0.1480	35.19
C	0.0078	0.1378	-7.14%	-2.13%	0.1456	35.21
D	0.0068	0.1355	-19.05%	-3.76%	0.1423	35.20
E	**0.0064**	**0.1351**	**-23.81%**	**-4.05%**	**0.1415**	**35.22**

only bring 1.8% and 1.4% bit rate saving. Applying TCR at the original scale is slightly better. It may be due to that introducing attention can help fuse the features warped by the diverse offsets. When TCR is applied at 1/4 scale, we can get a 3.2% gain without too much extra complexity. We also try using offset diversity [6] at 1/4 scale, but it only brings a 1.2% gain. It seems that attention mechanism is more suitable for the feature of 1/4 scale. Some works also apply transformers to the entropy model for capturing long-range dependencies in the probability distribution estimation. For comparison, we try applying 2 RSTB [20] to the entropy model (1 RSTB for each prior fusion module), where each RSTB contains 2 Swin Transformer layers. It brings less gain but needs more computation cost than applying TCR at $1/4$ scale, which proves that our choice is reasonable and can achieve a better trade-off.

Table 5. Ablation study on motion information propagation.

Motion Guidance	-	Flow $\hat{v_t}$	Feature G_t	Feature G_t	Feature G_t
Motion Condition	-	-	-	Feature H_{t-1}	Residual offsets R_{t-1}
Avg	0	-2.4	-3.2	**-6.9**	-4.6

Table 6. Ablation study on transformer based context refinement.

Model	Avg	GMac
w/o TCR	0.0	3368
TCR at original scale	-1.8	3659
TCR at $1/2$ scale	-1.4	3441
TCR at $1/4$ scale	**-3.2**	3386
offset diversity at $1/4$ scale	-1.2	3373
RSTB at Entropy Model	-1.4	3395

5. Conclusion

As far as we know, we are the first to introduce effective bi-directional information interactions between motion coding and frame coding, enabling their synergy. In our proposed DCVC-MIP, the motion feature from the motion decoder is used as motion guidance to mitigate the alignment errors in context generation. Meanwhile, the feature from context generation is propagated as motion condition, helping save the bit cost for coding the subsequent motion latent. Through the cycle of such interactions, feature propagation for motion coding is enabled to exploit long-range temporal correlation, leading to a better RD trade-off. In addition, hybrid context generation is introduced to exploit the multi-scale context features and provide better moiton condition for Motion Information Propagation. Our method achieves the SOTA (state-of-the-art) performance on the benchmark datasets. Compared with the previous SOTA neural video codec, our method can achieve 12.9% bit rate saving.

References

[1] HM-16.20. https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tree/HM-16.20. Accessed: 2022-09-01. 7, 8

[2] VTM-13.2. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-13.2. Accessed: 2022-09-01. 7, 8

[3] x265. https://www.videolan.org/developers/x265.html. Accessed: 2022-09-01. 7, 8

[4] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2020. 3

[5] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *VCEG-M33*, 2001. 6

[6] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Understanding deformable alignment in video super-resolution. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 973–981, 2021. 2, 3, 4, 5, 7, 8

[7] 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*, pages 764–773, 2017. 3

[8] Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers. Neural inter-frame compression for video coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6421–6429, 2019. 3

[9] Han Gao, Jinzhong Cui, Mao Ye, Shuai Li, Yu Zhao, and Xiatian Zhu. Structure-preserving motion estimation for learned video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3055–3063, 2022. 3

[10] 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*, pages 7033–7042, 2019. 1, 3

[11] Yung-Han Ho, Chih-Peng Chang, Peng-Yu Chen, Alessandro Gnutti, and Wen-Hsiao Peng. Canf-vc: Conditional augmented normalizing flows for video compression. *arXiv preprint arXiv:2207.05315*, 2022. 3, 6, 7, 8

[12] 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*, pages 193–209. Springer, 2020. 1, 3

[13] 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*, pages 5921–5930, 2022. 1, 3

[14] 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*, pages 1502–1511, 2021. 1, 3

[15] Théo Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier Déforges. Optical flow and mode selection for learning-based video coding. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2020. 3

[16] Théo Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier Déforges. Conditional coding and variable bitrate for practical learned video coding. *arXiv preprint arXiv:2104.09103*, 2021. 3

[17] Théo Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier Déforges. Conditional coding for flexible learned video compression. *arXiv preprint arXiv:2104.07930*, 2021. 3

[18] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2, 3, 6, 7, 8

[19] 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*, 2022. 2, 3, 5, 6, 7, 8

[20] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 6, 8

[21] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-lvc: Multiple frames prediction for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3554, 2020. 3, 6, 7

[22] Haojie Liu, Ming Lu, Zhan Ma, Fan Wang, Zhihuang Xie, Xun Cao, and Yao Wang. Neural video coding using multiscale motion compensation and spatiotemporal context model. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(8):3182–3196, 2020. 3

[23] Haojie Liu, Han Shen, Lichao Huang, Ming Lu, Tong Chen, and Zhan Ma. Learned video compression via joint spatial-temporal correlation exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11580–11587, 2020. 3

[24] 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*, pages 456–472. Springer, 2020. 3

[25] 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*, pages 11006–11015, 2019. 1, 3

[26] 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*, 43(10):3292–3308, 2020. 1, 3

[27] 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*, 43(10):3292–3308, 2020. 3, 6, 7, 8

[28] Wufei Ma, Jiahao Li, Bin Li, and Yan Lu. Uncertainty-aware deep video compression with ensembles, 2021. 3

[29] Fabian Mentzer, George Toderici, David Minnen, Sung-Jin Hwang, Sergi Caelles, Mario Lucic, and Eirikur Agustsson. Vct: A video compression transformer. *arXiv preprint arXiv:2206.07307*, 2022. 2, 3

[30] 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. 6

[31] 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. 3

[32] Reza Pourreza, Hoang Le, Amir Said, Guillaume Sautiere, and Auke Wiggers. Boosting neural video codecs by exploiting hierarchical redundancy. *arXiv preprint arXiv:2208.04303*, 2022. 3

[33] Oren Rippel, Alexander G Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir Bourdev. Elf-vc: Efficient learned flexible-rate video coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14479–14488, 2021. 3

[34] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G Anderson, and Lubomir Bourdev. Learned video compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3454–3463, 2019. 3

[35] 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. 2, 3, 6, 7, 8

[36] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012. 1, 6

[37] 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 *European Conference on Computer Vision*, pages 239–254. Springer, 2020. 3

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[39] 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. 6

[40] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. 6

[41] Thomas Wieg, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003. 1

[42] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 6

[43] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with recurrent autoencoder and recurrent probability model. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):388–401, 2020. 3

[44] Ruihan Yang, Yibo Yang, Joseph Marino, and Stephan Mandt. Hierarchical autoregressive modeling for neural video compression. *arXiv preprint arXiv:2010.10258*, 2020. 3

[45] Minyi Zhao, Yi Xu, and Shuigeng Zhou. Recursive fusion and deformable spatiotemporal attention for video compression artifact reduction. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5646–5654, 2021. 1, 3