Offline and Online Optical Flow Enhancement for Deep Video Compression

Chuanbo Tang, Xihua Sheng, Zhuoyuan Li, Haotian Zhang, Li Li, Dong Liu*

University of Science and Technology of China
{cbtang,xhsheng,zhuoyuanli,zhanghaotian}@mail.ustc.edu.cn, {lil1,dongeliu}@ustc.edu.cn

Abstract

Video compression relies heavily on exploiting the temporal redundancy between video frames, which is usually achieved by estimating and using the motion information. The motion information is represented as optical flows in most of the existing deep video compression networks. Indeed, these networks often adopt pre-trained optical flow estimation networks for motion estimation. The optical flows, however, may be less suitable for video compression due to the following two factors. First, the optical flow estimation networks were trained to perform inter-frame prediction as accurately as possible, but the optical flows themselves may cost too many bits to encode. Second, the optical flow estimation networks were trained on synthetic data, and may not generalize well enough to real-world videos. We address the twofold limitations by enhancing the optical flows in two stages: offline and online. In the offline stage, we fine-tune a trained optical flow estimation network with the motion information provided by a traditional (non-deep) video compression scheme, e.g. H.266/VVC, as we believe the motion information of H.266/VVC achieves a better rate-distortion trade-off. In the online stage, we further optimize the latent features of the optical flows with a gradient descent-based algorithm for the video to be compressed, so as to enhance the adaptivity of the optical flows. We conduct experiments on two state-of-the-art deep video compression schemes, DCVC and DCVC-DC. Experimental results demonstrate that the proposed offline and online enhancement together achieves on average 13.4% bitrate saving for DCVC and 4.1% bitrate saving for DCVC-DC on the tested videos, without increasing the model or computational complexity of the decoder side.

Introduction

Video compression relies heavily on exploiting the temporal redundancy between video frames, which is usually achieved by estimating and using the motion information. The motion information is represented as optical flows in most of the existing deep video compression networks (Lu et al. 2019; Li, Li, and Lu 2021; Sheng et al. 2022; Lin et al. 2020; Shi et al. 2022; Li, Li, and Lu 2022, 2023; Hu and Xu 2023). Indeed, these networks often adopt pre-trained optical flow estimation networks (Ranjan and Black 2017; Ilg et al. 2017; Sun et al. 2018) to estimate the motions between video frames. Taking a widely acknowledged and highly flexible scheme, DCVC (Li, Li, and Lu 2021), as an example, the pre-trained Spynet (Ranjan and Black 2017) is used for estimating the optical flows. The optical flows can be considered as pixel-wise motion vectors (MV) and are compressed by an autoencoder-based MV encoder (Minnen, Ballé, and Toderici 2018). In the training stage, the pre-trained Spynet is first loaded, and then the whole deep video compression network is optimized in an end-to-end manner. In the inference stage, the motion information of different video contents is obtained through the fixed networks.

However, regarding the optical flows estimated by the commonly-used pre-trained optical flow estimation networks (Ranjan and Black 2017; Ilg et al. 2017; Sun et al. 2018) as motion information in deep video compression schemes may be less suitable due to the following two factors. First, the pre-trained optical flow estimation networks are trained to perform inter-frame prediction as accurately as possible, but the optical flows themselves may cost too many bits to encode. Although they can be further optimized with the whole video compression networks in an end-to-end manner, the inappropriate initial point may affect the final optimization result. Second, the optical flow estimation networks are trained on synthetic data (Dosovitskiy et al. 2015; Butler et al. 2012; Baker et al. 2011), and may not generalize well enough to real-world videos. The end-to-end optimization in video compression networks can alleviate the domain gap between the synthetic data and the real-world videos to some degree. However, once the end-to-end optimization is finished, the optical flow estimation network is "optimal" in the sense that the average performance over the entire training set is optimal, but not "optimal" in the sense that the network produced optical flows may not be the optimal for any given video sequence.

To address the twofold limitations, we consider learning the good traditions from the inter-frame prediction techniques in traditional (non-deep) video compression schemes. The latest traditional video compression standard H.266/VVC (Bross et al. 2021) has achieved great success in effectively estimating and using the motion information, which is represented by MV. Specifically, in the offline stage, various hand-crafted inter-frame prediction modes are first designed for different types of motions without op-

*Corresponding Author.

timization. Then, the optimal mode is searched online to achieve the best rate-distortion (RD) performance for each coding sequence. Such offline and online optimization is believed a promising direction for learning-based video compression as well in the reference (Huo et al. 2022).

Similar to the two-stage strategy in the traditional video compression scheme, we address the twofold limitations of the optical flows by enhancing them in two stages: offline and online. In this paper, we propose an offline and online enhancement on the optical flows to better estimate and utilize motion information under the RD constraint. Specifically, in the offline stage, the trained optical flow estimation network Spynet is fine-tuned by the MV provided by VTM (reference software of H.266/VVC), as we believe the MV of VTM achieves a better RD trade-off. With the guidance of the MV of VTM, the optical flow estimation network can provide a more appropriate initial point for end-to-end optimization in video compression networks. In the online stage, we optimize the latent features of the optical flows with a gradient descent-based algorithm for the video to be compressed, so as to enhance the adaptivity of the optical flows. Inspired by the search-based online optimization algorithm in traditional video compression schemes, our scheme enables online updating the latent features of the optical flows by minimizing the RD loss in the inference stage, which has been introduced in deep image compression (Campos et al. 2019). When online updating the latent features of the optical flows, the parameters of the whole video compression networks are fixed and the decoding time remains unchanged. With the online enhancement, the updated latent features can help the video compression networks achieve a better RD performance than the latent features obtained by a simple forward pass through the MV encoder.

We conduct experiments on the two widely acknowledged deep video compression schemes DCVC (Li, Li, and Lu 2021) and DCVC-DC (Li, Li, and Lu 2023) to verify the effectiveness of our proposed scheme. Experimental results demonstrate our scheme can outperform both baseline schemes without increasing the model size or computational complexity on the decoder side.

Our contributions are summarized as follows:

- We propose an offline enhancement on the optical flows by fine-tuning the optical flow estimation network with the MV of VTM. With the guidance of the MV of VTM, the optical flow estimation network can provide a more appropriate initial point for end-to-end optimization in deep video compression networks.
- We further enhance the adaptivity of the optical flows by online optimizing the latent features of the optical flows according to the contents of different coding sequences in the inference stage without changing the network parameters.
- When equipped with our proposed offline and online optical flow enhancement methods, the baseline scheme DCVC achieves a better RD performance without increasing the model size and decoding complexity.

Related Work

Deep Video Compression

With the development of deep learning (Xiao et al. 2020; Li et al. 2023), deep video compression has explored a new direction. Deep video compression frameworks can be categorized into two main types: the motion-compensated prediction and residual coding framework and the motion-compensated prediction and conditional coding framework. DVC (Lu et al. 2019) is the pioneering work for the motion-compensated prediction and residual coding framework, which replaced each part in traditional video compression framework with neural networks. DCVC (Li, Li, and Lu 2021) introduced the motion-compensated prediction and conditional coding framework, which is able to utilize the learned temporal correlation between the current frame and predicted frame, rather than the subtraction-based residual.

Research on the frameworks. For the motion-compensated prediction and residual coding framework, the motion compression and residual compression was improved (Lu et al. 2020b; Hu et al. 2020). Multi-frame-based motion estimation and compensation (Lin et al. 2020) can reduce the temporal redundancy efficiently. The deformable convolution (Dai et al. 2017) was applied for motion estimation, compression, and compensation in feature domain (Hu, Lu, and Xu 2021), and coarse-to-fine motion compensation (Hu et al. 2022) was further proposed. Pixel-to-feature motion prediction (Shi et al. 2022) improved the inter-frame accuracy without increasing decoding complexity.

Following the motion-compensated prediction and conditional coding framework, multi-scale temporal context mining (Sheng et al. 2022) and hybrid spatial-temporal entropy model (Li, Li, and Lu 2022) were designed to improve the compression performance. DCVC-DC (Li, Li, and Lu 2023) further increased the context diversity in both temporal and spatial dimensions by introducing the group-based offset diversity and quadtree-based partition.

Research on the optimization strategy. Lu $et\ al.$ (Lu et al. 2020a) applied a new training objective with multiple time steps and adopted an online encoder updating scheme which updates the parameters of the encoder in the inference stage. A pixel-level implicit bit allocation (Xu et al. 2023) was proposed by using online optimization.

Inter-frame Prediction in Traditional Video Compression

In the past decades, several traditional compression schemes have been proposed, such as H.264/AVC (Wiegand et al. 2003), H.265 /HEVC (Sullivan et al. 2012), and H.266/VVC (Bross et al. 2021).

In the latest coding standard (H.266/VVC (Bross et al. 2021)), many advanced inter-frame prediction techniques (Huo et al. 2018; Li et al. 2023) have been proposed to attain high inter-frame coding efficiency. To estimate the accurate motion, various motion situations (translation, rotation motion model, etc.) corresponding to different inter-frame prediction modes (AMVP (Chien et al. 2021), Affine (Li et al. 2017), etc.) are executed to search for

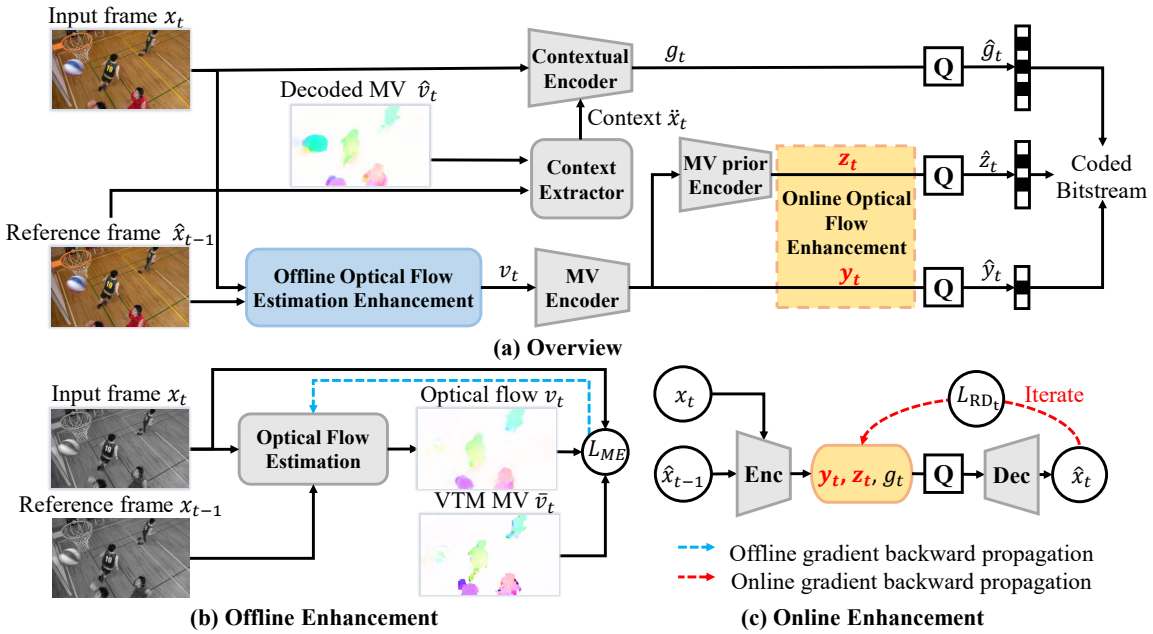

Figure 1: (a) Overview of our framework. (b) Offline enhancement for the optical flow network. The training procedure of the optical flow estimation network is supervised by the MV of VTM. (c) Online enhancement for the optical flow. The latent features of the optical flow y_t and z_t are online updated by minimizing the RD loss L_{RD_t}.

the optimal MV for each coding region via Rate-Distortion-Optimization (RDO) (Sullivan and Wiegand 1998). For each coding sequence, the optimal mode is searched online from multiple inter-frame prediction modes.

Considering the latest traditional video codec VTM (Bross et al. 2021) searches each MV for the best RD performance for each coding sequence, the MV can achieve a better RD trade-off than the optical flows. Thus, in this paper, we enhance the optical flows with the MV of VTM in offline stage. In the online stage, inspired by the VTM searching-based strategy in motion estimation, we further optimize the latent features of the optical flows with a gradient descent-based algorithm.

Approach

In this paper, our proposed offline and online enhancement is integrated into two baseline scheme DCVC and DCVC-DC to demonstrate the effectiveness. The encoding procedure of our scheme, as illustrated in Fig. 1(a), can be divided into three parts: motion estimation, motion compression, and contextual compression.

Motion Estimation. The input frame x_t and the reference frame $\hat{x}_{t-1}$ are fed into our proposed offline enhanced optical flow estimation network to estimate the optical flows, which are considered as pixel-wise MV v_t. Following DCVC and DCVC-DC, the network is based on Spynet, but we fine-tune it with the MV of VTM.

Motion Compression. The estimated MV v_t is compressed by an autoencoder-based MV encoder (Minnen, Ballé, and Toderici 2018). The latent features of the optical flows, MV features y_t and MV hyperprior z_t, are online enhanced by updating with a gradient descent-based algorithm in the inference stage.

Contextual Compression. Following DCVC and DCVC-DC, the input frame x_t is compressed conditioned on the context $\ddot{x}_t$, which is extracted by the context extractor using the reference frame $\hat{x}_{t-1}$ and the decoded MV $\hat{v}_t$ as input.

Offline Enhancement

To alleviate the domain gap between the synthetic data and the real-world videos, and provide a more appropriate initial point for the end-to-end optimization in deep video compression networks, we propose the offline enhancement on the optical flows. Different from DCVC and DCVC-DC, we fine-tune the pre-trained Spynet with the MV searched by VTM for the best RD performance on real-world videos, which has a better RD trade-off than the optical flows.

Preliminaries. To provide the optical flow estimation network with accurate and learnable labels, we extract the block-level MV from each frame by VTM under certain configuration. To match the low-delay mode of DCVC and DCVC-DC, the reference list of VTM is set to only include the previous frame of the current frame. Besides, for acquiring finer MV on the encoder side, we set the quantization parameter (QP) to 22 and turn off the decoder-side MV refine technique (PROF (Luo, He, and Chen 2019)). As the coding block predicted by intra mode is not appropriated for the training of optical flow estimation network, we turn off the intra-prediction mode and intra-related inter technique (CIIP (Chien et al. 2021)) in VTM to obtain the MV. The extracted block-level MV is at the quarter resolution, so we use the nearest interpolation to obtain the full-resolution block-level MV $\bar{v}_t$. Besides, as the precision of VTM MV is 1/16

sample, the extracted MV is multiplied by 16 to store the integer and fractional part of MV. When as the label to fine-tune the Spynet, the MV needs to be divided by 16.

Specifically, as shown in Fig. 1(b), we fine-tune the pre-trained Spynet under the guidance of the extracted MV $\overline{v}_t$, which is searched by VTM for the best RD performance on real-world videos. To better match the warp operation in the video compression, our training objective for Spynet is to minimize both the End Point Error (EPE) loss and the Mean Squared Error (MSE) loss between the input frame and the corresponding warp frame. Let $\check{x}_t$ denote the warp frame,

$$\check{x}_t = w(x_{t-1}, v_t), \tag{1}$$

where $w(\cdot)$ denotes the warp operation. Therefore, our training objective is to minimize a weighted sum of EPE and MSE loss,

$$L_{ME} = \frac{1}{mn}\sum_{i,j}\sqrt{(v_i-\overline{v}_i)^2+(v_j-\overline{v}_j)^2} + \lambda_{ME}\cdot d(x_t,\check{x}_t). \tag{2}$$

The $m \times n$ in Eq. (2) is the image dimension and the i and j subscript indicate the horizontal and vertical components of the flow vector and motion vector. $d(x_t, \check{x}_t)$ represents the MSE metric for measuring the difference between the input frame x_t and the warp frame $\check{x}_t$. λ_{ME} controls the trade-off between the EPE and MSE loss.

Compared with Spynet, the warp frames of enhanced Spynet have an average improvement of 1.15dB (33.23dB vs. 32.08dB) in JVET CTC test sequences (Bossen et al. 2019). The improvement in inter-frame prediction accuracy indicates that the offline enhancement can alleviate the domain gap between the synthetic data and the real-world videos to some degree.

End-to-End Training

After fine-tuning the pre-trained Spynet, we deploy it into DCVC and DCVC-DC, then train the whole video compression network in an end-to-end manner which is the same as DCVC and DCVC-DC. Thus, the training loss is as follows:

$$L = \lambda\cdot D + R = \lambda d(x_t,\hat{x}_t) + H(\hat{y}_t) + H(\hat{z}_t) + H(\hat{g}_t), \tag{3}$$

where $\hat{y}_t = Q(y_t)$, $\hat{z}_t = Q(z_t)$, and $\hat{g}_t = Q(g_t)$. $Q(\cdot)$ represents the quantization operator. The term R in Eq. (3) denotes the number of bits used to encode the frame, and R is computed by adding up the number of bits $H(\hat{y}_t)$ and $H(\hat{z}_t)$ for encoding the latent features of motion information and $H(\hat{g}_t)$ for encoding the latent features of context. $d(x_t, \hat{x}_t)$ denotes the distortion between the input frame x_t and the reconstruction frame $\hat{x}_t$. λ is a hyperparameter that determines the trade-off between the number of bits R and the distortion D. The MV of VTM is searched for the best trade-off between the bits cost and the MSE loss, so we only optimize our scheme with D representing the MSE.

Online Enhancement

To further enhance the adaptivity of the optical flows and achieve a better compression performance, we propose the online enhancement on the optical flows. In the inference stage, we online optimize the latent features of the optical flows with a gradient descent-based algorithm minimizing the RD loss for the videos to be compressed.

Algorithm 1: Optical Flow Latent Updating in the Inference Stage

Input: input frame x_t and the reference frame $\hat{x}_{t-1}$
Parameter: MV encoder and hyperprior encoder $\mathbf{Enc_{MV}}$
The video decoder with gradient $\mathbf{Dec_T}$
The video decoder without gradient $\mathbf{Dec_I}$
N represents the total updating times
η represents the step size (learning rate)
$\lfloor\cdot\rceil$ represents rouding operation and $u \sim \mathcal{U}(-0.5, 0.5)$
Output: The reconstruction frame $\hat{x}_t$ and latent features of context $\hat{g}_t$

$y_t{}^0, z_t{}^0 \Leftarrow \mathbf{Enc_{MV}}(x_t, \hat{x}_{t-1})$
$\hat{y}_t^0, \hat{z}_t^0 \Leftarrow \lfloor y_t{}^0 \rceil, \lfloor z_t{}^0 \rceil$
$\hat{x}_t^0, \hat{g}_t^0 \Leftarrow \mathbf{Dec_I}(\hat{y}_t^0, \hat{z}_t^0)$
$\hat{y}_t^{op}, \hat{z}_t^{op} \Leftarrow \hat{y}_t^0, \hat{z}_t^0$
$\hat{L}_{RD_t}^{op} = \lambda \cdot d(x_t, \hat{x}_t^0) + H(\hat{y}_t^0) + H(\hat{z}_t^0) + H(\hat{g}_t^0)$
for $i = 0; i < N; i++$ **do**
  $\tilde{y}_t^i, \tilde{z}_t^i \Leftarrow y_t{}^i + u, z_t{}^i + u$
  $\tilde{x}_t^i, \tilde{g}_t^i \Leftarrow \mathbf{Dec_T}(\tilde{y}_t^i, \tilde{z}_t^i)$
  $\tilde{L}_{RD_t}^i = \lambda \cdot d(x_t, \tilde{x}_t^i) + H(\tilde{y}_t^i) + H(\tilde{z}_t^i) + H(\tilde{g}_t^i)$
  $y_t{}^{i+1} \Leftarrow y_t{}^i - \eta \frac{\partial \tilde{L}_{RD_t}^i}{\partial y_t{}^i}$
  $z_t{}^{i+1} \Leftarrow z_t{}^i - \eta \frac{\partial \tilde{L}_{RD_t}^i}{\partial z_t{}^i}$
  $\hat{y}_t^{i+1}, \hat{z}_t^{i+1} \Leftarrow \lfloor y_t{}^{i+1} \rceil, \lfloor z_t{}^{i+1} \rceil$
  $\hat{x}_t^{i+1}, \hat{g}_t^{i+1} \Leftarrow \mathbf{Dec_I}(\hat{y}_t^{i+1}, \hat{z}_t^{i+1})$
  $\hat{L}_{RD_t}^{i+1} = \lambda \cdot d(x_t, \hat{x}_t^{i+1}) + H(\hat{y}_t^{i+1}) + H(\hat{z}_t^{i+1}) + H(\hat{g}_t^{i+1})$
  if $\hat{L}_{RD_t}^{i+1} < \hat{L}_{RD_t}^{op}$ **then**
    $\hat{y}_t^{op}, \hat{z}_t^{op} \Leftarrow \hat{y}_t^{i+1}, \hat{z}_t^{i+1}$
    $\hat{L}_{RD_t}^{op} \Leftarrow \hat{L}_{RD_t}^{i+1}$
  end if
end for
$\hat{x}_t, \hat{g}_t \Leftarrow \mathbf{Dec_I}(\hat{y}_t^{op}, \hat{z}_t^{op})$

Single-frame online optimization. As shown in Fig. 1(c), for the input frame x_t and reference frame $\hat{x}_{t-1}$ in a group of pictures (GOP), we online update the latent features of the optical flows (MV feature y_t and the MV hyperprior z_t) by a gradient descent-based algorithm in the inference stage. After N iterations, we obtain the latent features of the optical flows $\hat{y}_t^{op}$ and $\hat{z}_t^{op}$ which are optimal for the consecutive two frames x_t and $\hat{x}_{t-1}$. Then, the latent features of context $\hat{g}_t$ and reconstruction frame $\hat{x}_t$ are generated by the $\hat{y}_t^{op}$ and $\hat{z}_t^{op}$, and we start to online update the latent features of the optical flows generated by the next input frame x_{t+1} and the reference frame $\hat{x}_t$ in a GOP.

The pipeline of the optical flow latent updating algorithm is shown in Algorithm 1. Firstly, the initial latent features of the optical flows $y_t{}^0$ and $z_t{}^0$ are generated by the input frame x_t and the reference frame $\hat{x}_{t-1}$. Secondly, the initial RD cost $\hat{L}_{RD_t}^{op}$ can be computed by feeding the initial latent features of the optical flows to the video decoder without gradient $\mathbf{Dec_I}$. Then, the latent features of the optical flows are online updated iteratively to minimize the RD loss of

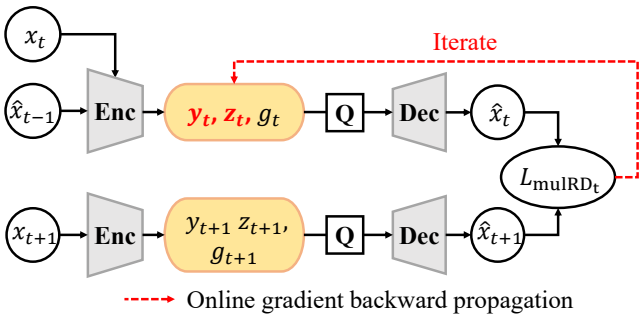

Figure 2: Overview of the three-frame online optimization.

each iteration $\tilde{L}^i_{RD_t}$:

$$\tilde{L}^i_{RD_t} = \lambda \cdot d(x_t, \tilde{x}^i_t) + H(\tilde{y}^i_t) + H(\tilde{z}^i_t) + H(\tilde{g}^i_t), \quad (4)$$

where ${y_t}^i$ denotes the MV feature of the current frame after i steps of update, and ${z_t}^i$ denotes the MV hyperprior of the current frame after i steps of update. Following the work (Ballé, Laparra, and Simoncelli 2016), we use adding uniform noise to approximate the rounding during training, $\tilde{y}^i_t = {y_t}^i + u$, $\tilde{z}^i_t = {z_t}^i + u$, and $u \sim \mathcal{U}(-0.5, 0.5)$. The latent features of context $\tilde{g}^i_t$ and reconstruction frame $\tilde{x}^i_t$ are generated by feeding the latent features of the optical flows $\tilde{y}^i_t$ and $\tilde{z}^i_t$ to the video decoder with gradient $\mathbf{Dec_T}$ during the online optimization.

The only difference between $\mathbf{Dec_T}$ and $\mathbf{Dec_I}$ lies in the quantization. To allow online optimization via gradient descent, the quantization in $\mathbf{Dec_T}$ is replaced by adding uniform noise, while the quantization in $\mathbf{Dec_I}$ is using rounding operation directly. During the updating iterations, the latent features of the optical flows are updated by minimizing the RD loss of each iteration $\tilde{L}^i_{RD_t}$, which is computed by sending latent features of the optical flows to $\mathbf{Dec_T}$. Then, the updated latent features of the optical flows are sent to $\mathbf{Dec_I}$ to compute the RD cost of each iteration $\hat{L}^i_{RD_t}$, and we only save the optimal latent features of the optical flows $\hat{y}^{op}_t$ and $\hat{z}^{op}_t$ which lead to the minimal RD cost $\hat{L}^{op}_{RD_t}$.

Multi-frame online optimization. Considering the error propagation in deep video compression frameworks, we further extend the single-frame online optimization algorithm to a multi-frame online optimization algorithm. We design a sliding-window-based online optimization algorithm to update the latent features of the optical flows by minimizing the multi-frame RD loss of each iteration $\tilde{L}^i_{mulRD_t}$ for all frames inside a window:

$$\tilde{L}^i_{mulRD_t} = \sum_{j=t}^{W} \alpha_j [\lambda d(x_j, \tilde{x}^i_j) + H(\tilde{y}^i_j) + H(\tilde{z}^i_j) + H(\tilde{g}^i_j)], \quad (5)$$

where window size W denotes the number of frames inside a window and α_j is a hyperparameter that determines the weight of RD loss for different frames.

Specifically, the overview of three-frame online optimization is shown in Fig. 2. The consecutive three frames in a GOP $\hat{x}_{t-1}$, x_t, and x_{t+1} are sent into the sliding window to update the latent features of the optical flows y_t and z_t iteratively minimizing multi-frame RD loss of each iteration $\tilde{L}^i_{mulRD_t}$. After N iterations, we obtain the latent features of the optical flows $\hat{y}^{op}_t$ and $\hat{z}^{op}_t$ which are optimal for the consecutive three frames $\hat{x}_{t-1}$, x_t, and x_{t+1}, leading to the minimal multi-frame RD cost $\hat{L}^{op}_{mulRD_t}$. The reconstruction frame $\hat{x}_t$ is generated by the updated latent features $\hat{y}^{op}_t$ and $\hat{z}^{op}_t$, then the next consecutive three frames $\hat{x}_t$, x_{t+1}, and x_{t+2} will be sent to the sliding window. When the sliding window includes the last frame of the GOP, the window size W will decrease by 1 until it equals to 2.

Experiments

Experimental Setup

Training Data. We use BVI-DVC (Ma, Zhang, and Bull 2021) dataset for fine-tuning Spynet. The BVI-DVC dataset contains 800 sequences at various spatial resolutions from 270p to 2160p. The motion vectors are extracted by VTM-10.0[1]. The commonly-used Vimeo-90k (Xue et al. 2019) dataset is used for training DCVC and DCVC-DC in an end-to-end manner. During the training, all the videos of training sets are randomly cropped into 256 × 256 patches.

Testing Data and Conditions. We use the JVET CTC test sequences (Bossen et al. 2019) for evaluating the fine-tuning of Spynet. UVG (Mercat, Viitanen, and Vanne 2020), MCL-JCV (Wang et al. 2016) and HEVC (Bossen et al. 2013) datasets are used for testing our scheme. The UVG the MCL-JCV dataset has 37 1080p sequences. The HEVC dataset contains 16 sequences including Class B, C, D, and E. In addition, HEVC RGB dataset (Flynn, Sharman, and Rosewarne 2013) are also evaluated. We test 96 frames for each video, and the intra period is set to 12 for each dataset. Besides, we use $Cheng2020Anchor$ (Cheng et al. 2020) implemented by CompressAI (Bégaint et al. 2020) for intra-frame coding in DCVC.

Implementation Details. Our scheme includes three training stages, which consist of the fine-tuning of Spynet, offline training of the video codec (DCVC and DCVC-DC), and online optimization of the video codec with the enhanced Spynet. In the first stage, we set λ_{ME} to 100, and fine-tune the Spynet using the extracted MV for 1,000,000 iterations. In the second stage, we deploy the enhanced Spynet into the video codec and train the whole video compression network for 5,000,000 iterations until converge. Finally, we set the updating times N in Algorithm 1 to 1500 according to the ablation study. The initial learning rate for the first two steps is 1e-4, then decrease to 5e-5 at the 800,000th iteration and 4,000,000th iteration respectively. The initial learning rate for online optimization is 5e-3, which is decreased by 50% at the 1200th iteration. The Adam optimizer (Kingma and Ba 2014) is used, and the batch size is set to 16 for the first training stage and 4 for the second training stage.

Comparisons with Baseline and SOTA Methods

Comparisons with Baseline Method. Fig. 3 shows RD curves on HEVC Class B, Class C, Class D, Class E, Class

[1]https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-10.0

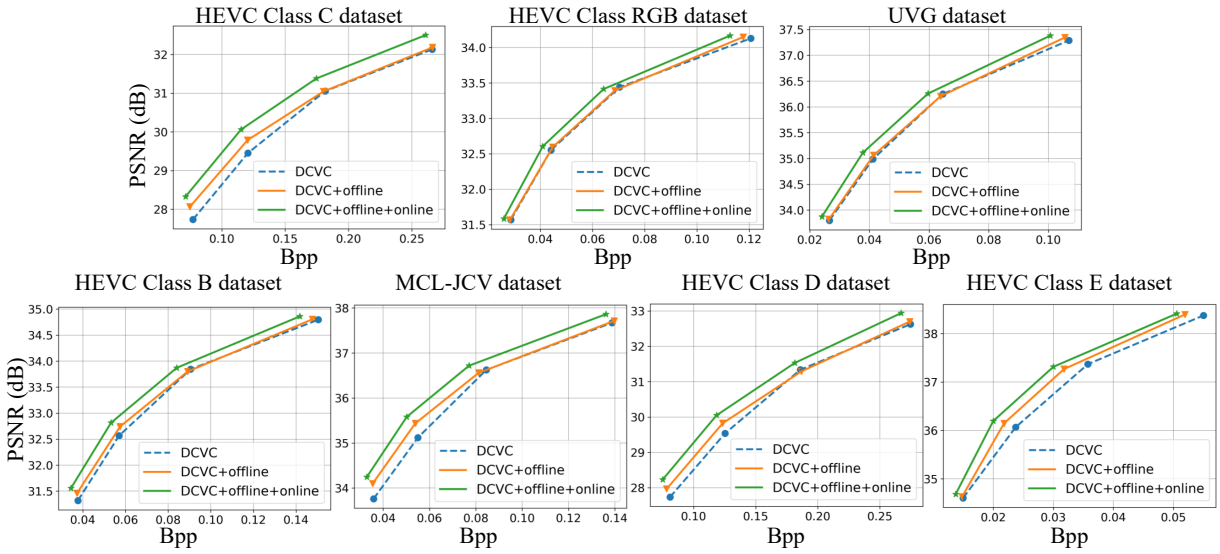

Figure 3: Rate-distortion curves of our scheme and DCVC on the testing datasets.

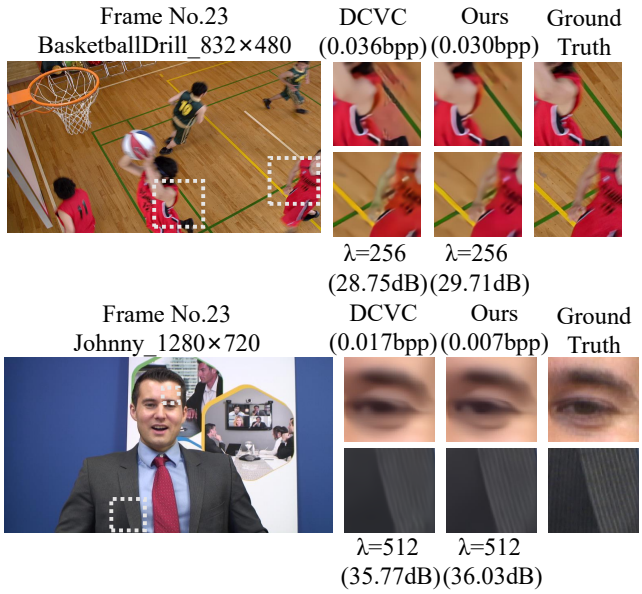

Figure 4: Reconstruction frame of DCVC and our scheme (DCVC + Offline + Online) and the ground truth in different sequences from HEVC dataset.

RGB, UVG, and MCL-JCV datasets. Our baseline scheme is DCVC, and it's obvious that DCVC with the offline and online enhancement on the optical flows can outperform DCVC in all rate points. Besides, both the offline and online enhancement on the optical flows don't change the network structure of DCVC and only optimize the encoder side of DCVC, leading to no increase in the model size or computational complexity on the decoder side. The proposed offline and online enhancement together achieves an average of 13.4% bitrate saving on all testing datasets over DCVC, and the offline enhancement can achieve an average of 4.3% bitrate saving on all testing datasets over DCVC

	B	C	D	UVG	Average
DCVC-DC	0.0	0.0	0.0	0.0	0.0
DCVC	66.6	79.7	76.7	78.7	75.4
DCVC-DC + offline	-0.7	-1.0	-2.1	-0.4	-1.1
DCVC-DC + offline + online	-2.8	-4.9	-4.6	-4.2	-4.1

Table 1: Effectiveness of the offline and online enhancement on SOTA method DCVC-DC. BD-Rate(%) comparison for PSNR. Negative values in BDBR represent the bitrate saving.

In Fig 4, we present visual results of the reconstruction frames of DCVC and our scheme across different sequences. With the offline and online enhancement, our scheme can achieve higher quality reconstruction, retaining more details in the boundaries of the motion and the regions with rich texture, while using fewer bits than DCVC.

Comparisons with SOTA Method. To evaluate that our method can be effective on other scheme, we also conduct experiments on the SOTA (state-of-the-art) deep video codec DCVC-DC (Li, Li, and Lu 2023). We report the BD-rate (Bjontegaard 2001) results of HEVC Class B, Class C, Class D, and UVG datasets in Table 1, which still verify that both our offline enhancement and online enhancement can be effective in other schemes. The temporal context mining, group-based offset diversity, and motion information propagation in DCVC-DC have helped it achieve a more accurate temporal prediction than DCVC, so the bitrate saving for DCVC-DC is not as much as that for DCVC.

Ablation Study

Effectiveness of Offline and Online Enhancement. To verify the effectiveness of the offline and online enhancement on the optical flows respectively, we compare the compression performance of the baseline scheme (DCVC) with or

Offline	Online	Class B	Class C	Class D	Class E	Class RGB	UVG	MCL-JCV	Average
✗	✗	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
✓	✗	-3.0	-5.9	-4.4	-7.9	-0.7	-1.3	-6.7	-4.3
✗	✓	-10.7	-14.3	-11.1	-9.0	-8.5	-10.1	-11.3	-10.7
✓	✓	-12.0	-17.1	-13.1	-15.3	-8.8	-10.5	-16.9	-13.4

Table 2: Effectiveness of the offline and online enhancement. BD-Rate(%) comparison for PSNR. The anchor is DCVC.

N	C	D	${E_T}^C$(s)	${D_T}^C$(s)	${E_T}^D$(s)	${D_T}^D$(s)
0	0.0	0.0	2.71	6.94	0.70	1.91
100	-6.1	-5.1	28.15	6.84	10.42	1.90
500	-9.6	-7.9	132.78	6.95	48.99	1.87
1000	-10.8	-8.6	269.20	6.73	92.58	1.89
1500	-11.2	-8.7	388.73	6.86	141.03	1.91
2000	-11.5	-9.1	530.10	6.84	190.64	1.89
2500	-11.6	-9.2	674.54	6.89	239.05	1.88

Table 3: BD-Rate(%) comparison for PSNR, Encoding time E_T, and Decoding time D_T for different online updating times N. The anchor is DCVC + Offline ($N = 0$).

W	C	D	${E_T}^C$(s)	${D_T}^C$(s)	${E_T}^D$(s)	${D_T}^D$(s)
2	0.0	0.0	518.25	6.84	187.82	1.89
3	-0.5	-0.4	1631.35	6.84	546.99	1.88
4	-0.7	-0.7	2187.58	6.82	683.04	1.88
5	-0.8	-0.8	2706.39	6.87	874.56	1.86

Table 4: BD-Rate(%) comparison for PSNR, Encoding time E_T, and Decoding time D_T for different window size W in the multi-frame online enhancement. The anchor is DCVC + Offline + Online, which adopts the single-frame online optimization ($W = 2$).

without the enhancement. We report the BD-rate results in Table 2. For online enhancement, the updating times are set to 1500. From the comparison results, we find that the offline enhancement on the optical flows brings 4.3% bitrate saving and the online enhancement brings 10.7% bitrate saving. With both offline and online enhancement, 13.4% bitrate saving is achieved. The experimental results indicate that our offline enhancement on the optical flows can provide a more appropriate initial point for the online optimization.

Influence on updating times of online enhancement. To study the influence of the total updating times in online enhancement, we change the updating times from 100 to 2500. For simplification, we only use HEVC Class C and Class D datasets to explore the reasonable updating times considering the trade-off between compression performance and encoding time. The anchor is DCVC with the offline enhancement on the optical flows (DCVC + Offline). Table 3 reports the BD-rate results, which indicate that the RD performance is improved as the updating times N increases. To balance the trade-off between the compression performance and encoding time complexity, in this paper, we set the updating times N to 1500 for online enhancement. Besides, the decoding time in Table 3 demonstrates that our proposed method doesn't increase decoding complexity.

Multi-frame Online Optimization

In this paper, we adopt the single-frame online optimization in the inference stage to improve the compression performance. Besides, we also provide the compression results adopting the multi-frame online optimization which updates the latent features of the optical flows by minimizing multi-frame RD loss. We wish to explore the potential of multi-frame online optimization on the motion information with a limited number of frames in a GOP.

For simplification, we only conduct experiments on HEVC Class C and Class D datasets. We set the DCVC with offline and online enhancement on the optical flows (DCVC + Offline + Online) as the anchor, which adopts the single-frame online optimization. The single-frame online optimization is the same as setting the window size W in Eq. (5) to 2. The hyperparameters α_0, α_1, α_2, and α_3 in Eq. (5) are set to 1, 0.5, 0.2, and 0.1 respectively.

Table 4 reports the BD-rate with updating times N set to 2000. We compare the compression performance, encoding time, and decoding time of DCVC with multi-frame online enhancement on the optical flows with window size W set from 2 to 5 in HEVC Class C and Class D datasets. Table 4 shows that increasing the window size cannot improve the compression ratio greatly, but the encoding time has increased a lot when the window size exceeds 2. In this paper, we currently adopt the single-frame online optimization.

CONCLUSION

In this paper, we have proposed an offline and online enhancement on the optical flows to better estimate and utilize the motion information in the deep video compression network. Specifically, in the offline enhancement, we fine-tune the optical flow estimation network with the MV of VTM, which is searched for the best RD performance on real-world videos. In the online enhancement, we online update the latent features of the optical flows under the RD metric for different coding sequences in the inference stage. Our scheme can effectively improve the compression performance without increasing the model size or computational complexity on the decoder side. The experimental results show that our scheme can outperform DCVC and DCVC-DC in terms of PSNR by 13.4% and 4.1% respectively under the same configuration.

Acknowledgments

This work was supported by the Natural Science Foundation of China under Grants 62036005 and 62021001, and by the Fundamental Research Funds for the Central Universities under No. WK3490000006.

References

Baker, S.; Scharstein, D.; Lewis, J.; Roth, S.; Black, M. J.; and Szeliski, R. 2011. A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 92: 1–31.

Ballé, J.; Laparra, V.; and Simoncelli, E. P. 2016. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*.

Bégaint, J.; Racapé, F.; Feltman, S.; and Pushparaja, A. 2020. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*.

Bjontegaard, G. 2001. Calculation of average PSNR differences between RD-curves. *ITU SG16 Doc. VCEG-M33*.

Bossen, F.; Boyce, J.; Li, X.; Seregin, V.; and Sühring, K. 2019. JVET common test conditions and software reference configurations for SDR video. *Joint Video Experts Team (JVET) of ITU-T SG*, 16: 19–27.

Bossen, F.; et al. 2013. Common test conditions and software reference configurations. *JCTVC-L1100*, 12(7).

Bross, B.; Chen, J.; Ohm, J.-R.; Sullivan, G. J.; and Wang, Y.-K. 2021. Developments in international video coding standardization after avc, with an overview of versatile video coding (VVC). *Proceedings of the IEEE*, 109(9): 1463–1493.

Butler, D. J.; Wulff, J.; Stanley, G. B.; and Black, M. J. 2012. A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision (ECCV), Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, 611–625. Springer.

Campos, J.; Meierhans, S.; Djelouah, A.; and Schroers, C. 2019. Content adaptive optimization for neural image compression. *arXiv preprint arXiv:1906.01223*.

Cheng, Z.; Sun, H.; Takeuchi, M.; and Katto, J. 2020. 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)*, 7939–7948.

Chien, W.-J.; Zhang, L.; Winken, M.; Li, X.; Liao, R.-L.; Gao, H.; Hsu, C.-W.; Liu, H.; and Chen, C.-C. 2021. Motion vector coding and block merging in the versatile video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10): 3848–3861.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 764–773.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2758–2766.

Flynn, D.; Sharman, K.; and Rosewarne, C. 2013. Common test conditions and software reference configurations for HEVC range extensions. In *Proceedings of the 14th Meeting of Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*.

Hu, Z.; Chen, Z.; Xu, D.; Lu, G.; Ouyang, W.; and Gu, S. 2020. Improving deep video compression by resolution-adaptive flow coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 193–209. Springer.

Hu, Z.; Lu, G.; Guo, J.; Liu, S.; Jiang, W.; and Xu, D. 2022. 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)*, 5921–5930.

Hu, Z.; Lu, G.; and Xu, D. 2021. 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)*, 1502–1511.

Hu, Z.; and Xu, D. 2023. Complexity-Guided Slimmable Decoder for Efficient Deep Video Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14358–14367.

Huo, S.; Liu, D.; Li, L.; Ma, S.; Wu, F.; and Gao, W. 2022. Towards Hybrid-Optimization Video Coding. *arXiv preprint arXiv:2207.05565*.

Huo, S.; Liu, D.; Wu, F.; and Li, H. 2018. Convolutional neural network-based motion compensation refinement for video coding. In *2018 IEEE International symposium on circuits and systems (ISCAS)*, 1–4. IEEE.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2462–2470.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, J.; Li, B.; and Lu, Y. 2021. Deep contextual video compression. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 18114–18125.

Li, J.; Li, B.; and Lu, Y. 2022. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1503–1511.

Li, J.; Li, B.; and Lu, Y. 2023. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 22616–22626.

Li, L.; Li, H.; Liu, D.; Li, Z.; Yang, H.; Lin, S.; Chen, H.; and Wu, F. 2017. An efficient four-parameter affine motion model for video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8): 1934–1948.

Li, W.; Zhao, X.-L.; Ma, Z.; Wang, X.; Fan, X.; and Tian, Y. 2023. Motion-Decoupled Spiking Transformer for Audio-Visual Zero-Shot Learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, 3994–4002.

Lin, J.; Liu, D.; Li, H.; and Wu, F. 2020. M-LVC: Multiple frames prediction for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3546–3554.

Lu, G.; Cai, C.; Zhang, X.; Chen, L.; Ouyang, W.; Xu, D.; and Gao, Z. 2020a. Content adaptive and error propagation aware deep video compression. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 456–472. Springer.

Lu, G.; Ouyang, W.; Xu, D.; Zhang, X.; Cai, C.; and Gao, Z. 2019. DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11006–11015.

Lu, G.; Zhang, X.; Ouyang, W.; Chen, L.; Gao, Z.; and Xu, D. 2020b. An end-to-end learning framework for video compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10): 3292–3308.

Luo, J.; He, Y.; and Chen, W. 2019. Prediction refinement with optical flow for affine motion compensation. In *2019 IEEE Visual Communications and Image Processing (VCIP)*, 1–4. IEEE.

Ma, D.; Zhang, F.; and Bull, D. R. 2021. BVI-DVC: A training database for deep video compression. *IEEE Transactions on Multimedia*, 24: 3847–3858.

Mercat, A.; Viitanen, M.; and Vanne, J. 2020. UVG dataset: 50/120fps 4K sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, 297–302.

Minnen, D.; Ballé, J.; and Toderici, G. D. 2018. Joint autoregressive and hierarchical priors for learned image compression. *Advances in Neural Information Processing Systems (NeurIPS)*, 31.

Ranjan, A.; and Black, M. J. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4161–4170.

Sheng, X.; Li, J.; Li, B.; Li, L.; Liu, D.; and Lu, Y. 2022. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia*.

Shi, Y.; Ge, Y.; Wang, J.; and Mao, J. 2022. AlphaVC: High-Performance and Efficient Learned Video Compression. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX*, 616–631. Springer.

Sullivan, G. J.; Ohm, J.-R.; Han, W.-J.; and Wiegand, T. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12): 1649–1668.

Sullivan, G. J.; and Wiegand, T. 1998. Rate-distortion optimization for video compression. *IEEE signal processing magazine*, 15(6): 74–90.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. 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)*, 8934–8943.

Wang, H.; Gan, W.; Hu, S.; Lin, J. Y.; Jin, L.; Song, L.; Wang, P.; Katsavounidis, I.; Aaron, A.; and Kuo, C.-C. J. 2016. MCL-JCV: a JND-based H. 264/AVC video quality assessment dataset. In *2016 IEEE International Conference on Image Processing (ICIP)*, 1509–1513. IEEE.

Wiegand, T.; Sullivan, G. J.; Bjontegaard, G.; and Luthra, A. 2003. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7): 560–576.

Xiao, Z.; Xiong, Z.; Fu, X.; Liu, D.; and Zha, Z.-J. 2020. Space-time video super-resolution using temporal profiles. In *Proceedings of the 28th ACM International Conference on Multimedia*, 664–672.

Xu, T.; Gao, H.; Gao, C.; Wang, Y.; He, D.; Pi, J.; Luo, J.; Zhu, Z.; Ye, M.; Qin, H.; et al. 2023. Bit allocation using optimization. In *International Conference on Machine Learning*, 38377–38399. PMLR.

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