Scene Matters: Model-based Deep Video Compression

Lv Tang Xinfeng Zhang* Gai Zhang Xiaoqi Ma
University of Chinese Academy of Sciences, Beijing, China
luckybird1994@gmail.com, xfzhang@ucas.ac.cn, zhanggai16@mails.ucas.ac.cn, maxiaoqi197@gmail.com

Abstract

Video compression has always been a popular research area, where many traditional and deep video compression methods have been proposed. These methods typically rely on signal prediction theory to enhance compression performance by designing high efficient intra and inter prediction strategies and compressing video frames one by one. In this paper, we propose a novel model-based video compression (MVC) framework that regards scenes as the fundamental units for video sequences. Our proposed MVC directly models the intensity variation of the entire video sequence in one scene, seeking non-redundant representations instead of reducing redundancy through spatio-temporal predictions. To achieve this, we employ implicit neural representation as our basic modeling architecture. To improve the efficiency of video modeling, we first propose context-related spatial positional embedding and frequency domain supervision in spatial context enhancement. For temporal correlation capturing, we design the scene flow constrain mechanism and temporal contrastive loss. Extensive experimental results demonstrate that our method achieves up to a 20% bitrate reduction compared to the latest video coding standard H.266 and is more efficient in decoding than existing video coding strategies.

1. Introduction

Recently, videos have become ubiquitous in people's daily lives, from short-form videos to conference and surveillance videos. Efficiently storing and transmitting video data has become a significant challenge due to the vast amounts and explosive growth of such data. To address this challenge, multiple video compression standards have been developed based on traditional hybrid video coding frameworks, such as H.264/AVC [64], H.265/HEVC [55], and H.266/VVC [4], as well as deep-learning based video compression (DLVC) methods [11, 18, 21, 29, 33, 36, 66].

*Corresponding author. This work was supported by the National Natural Science Foundation under Grant 62071449 and U20A20184, and the Fundamental Research Funds for the Central Universities.

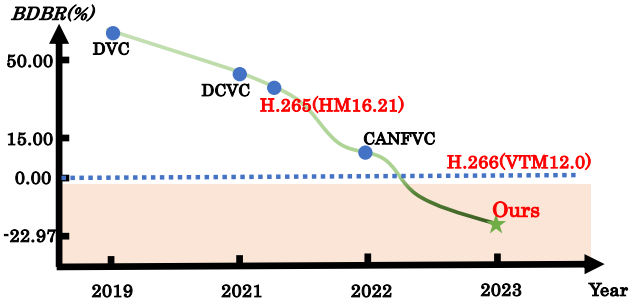

Figure 1. BDBR(%) [3] performances of different methods when compared with H.266 on the real-world surveillance video sequences in terms of PSNR. DVC [36], DCVC [29] and CANFVC [18] are three DLVC methods.

Both traditional hybrid video coding frameworks and existing deep learning-based video compression (DLVC) methods follow the same approach of compressing videos by designing various technique modules to reduce spatial and temporal redundancy. In traditional hybrid video coding, each video frame is divided into blocks, and intra- and inter-prediction techniques [63, 72] are used to reduce spatial and temporal redundancy. On the other hand, DLVC methods [18, 21, 33, 36], unlike traditional compression methods, use neural networks to design end-to-end intra- and inter-prediction modules for the entire frame. Despite the careful design of these techniques, both traditional and DLVC methods compress a video sequence progressively in a block-by-block or frame-by-frame style and only use neighboring pixels in the same frame or neighboring frames as reference to derive intra- or inter-prediction values. Since video sequences are captured at high framerates, such as 30fps or 60fps, the same scene may appear in hundreds of frames that are highly correlated in the temporal domain. However, existing compression strategies are not well-equipped to remove scene redundancy in the block- or frame-level prediction. As demonstrated in Fig. 1, the performance of existing state-of-the-art (SOTA) DLVC methods still lags behind that of the traditional H.266 standard.

To overcome the performance bottleneck in video compression, this paper proposes an innovative video coding paradigm that seeks to find a compact subspace for a video

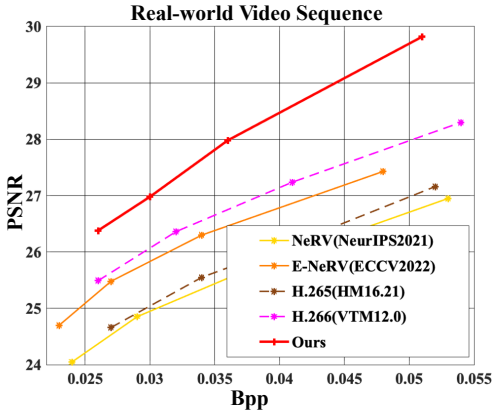

Figure 2. The performance of existing SOTA video INR methods when applied to the video compression task.

sequence of the same scene, rather than reducing spatio-temporal redundancy through block-level or frame-level prediction methods. This paradigm replaces explicit redundancy reduction through local prediction with implicit compact subspace modeling for the entire scene. Consequently, finding a suitable modeling tool to represent the scene is crucial to this paradigm. Recently, implicit neural representation (INR) has gained popularity for its strong ability to model a wide variety of signals by a deep network. INR has been already applied to various tasks to represent different objects, such as RGB images [52], 3D shapes [45, 52] and scenes [31, 43]. Considering that original signals can be implicitly encoded in network's parameters, some researchers apply the INR to the image compression task [13, 14, 54], and achieve competitive performance compared to traditional image compression method JPEG2000. Since the compressed bitstream is network's parameters, these image compression methods can be regarded as model-based image compression. Due to these characteristics, INR is a promising candidate for the backbone network of the proposed video compression paradigm.

In contrast to model-based image compression, model-based video compression (MVC) is barely explored. In MVC, sequence modeling is an extra significant factor, which is the main challenge for video compression. However, the representation ability of the primal video INR methods [7, 32] is limited. If we directly apply these methods to the video compression task, NeRV, is even inferior to traditional video coding standard H.265 [55], as shown in Fig. 2. This demonstrates that existing SOTA INR methods are unable to achieve higher-quality reconstruction results when given limited network parameters, highlighting the potential for further developments in applying video INR to video compression tasks. In this paper, we further improve the sequence modeling ability of video INR in spatial context enhancement and temporal correlation capturing

In spatial context capturing, existing video INR methods, such as those presented in [2, 7, 32], use a learnable network to reconstruct video frames from context-agnostic spatial positional embeddings. However, to handle spatial variations between different frames and achieve higher-quality reconstruction results, these methods typically require additional network parameters (*bitrates*), which can adversely impact rate-distortion (RD) performance. To address this issue, we propose a context-related spatial positional embedding (CRSPE) method in this paper. Additionally, some works [24, 25] have proposed frequency-aware operations in their networks to improve the context capturing ability and capture high-frequency image details. However, these operations often come with an added cost of network parameters that can degrade compression performance. To address this problem and maintain a balance between compression performance and reconstruction quality, we introduce a frequency domain supervision (FDS) module that can capture high-frequency details without requiring additional bitrates.

Temporal correlation is a critical factor for INR methods to improve the representation efficiency of different frames. Existing video INR methods primarily rely on different time positional encodings to distinguish between frames and expect the network to implicitly learn temporal correlation. While these encodings can capture temporal correlation to some extent, they struggle to explore complex temporal correlations, particularly for long video sequences. To address this limitation, we introduce a scene flow constraint mechanism (SFCM) for short-term temporal correlation and a temporal contrastive loss (TCL) for long-term temporal correlation in this paper. These mechanisms do not increase network parameters and are well-suited for the MVC task. As illustrated in Fig. 1, our proposed framework already outperforms H.266 [4] significantly, indicating the potential of MVC methods. Our main contributions are:

- We propose an MVC that seeks to identify more compact sub-spaces for video sequences. Unlike existing methods that rely on explicit spatio-temporal redundancy reduction through signal prediction at the block or frame level, our framework uses the correlations between all frames in a video scene simultaneously.
- To address the limitations of existing video INR methods when applied to video compression, we introduce CRSPE and FDS in spatial context enhancement, which can handle spatial variations of different frames and capture high-frequency details. We further design SFCM and TCL for temporal correlation modeling.
- Extensive experiments are conducted on different databases, and detailed analyses are provided for our designed modules. Experimental results show that our proposed method can outperform H.266 (VTM12.0), which demonstrates the superiority of our proposed method and may inspire researchers to explore video compression in a new light.

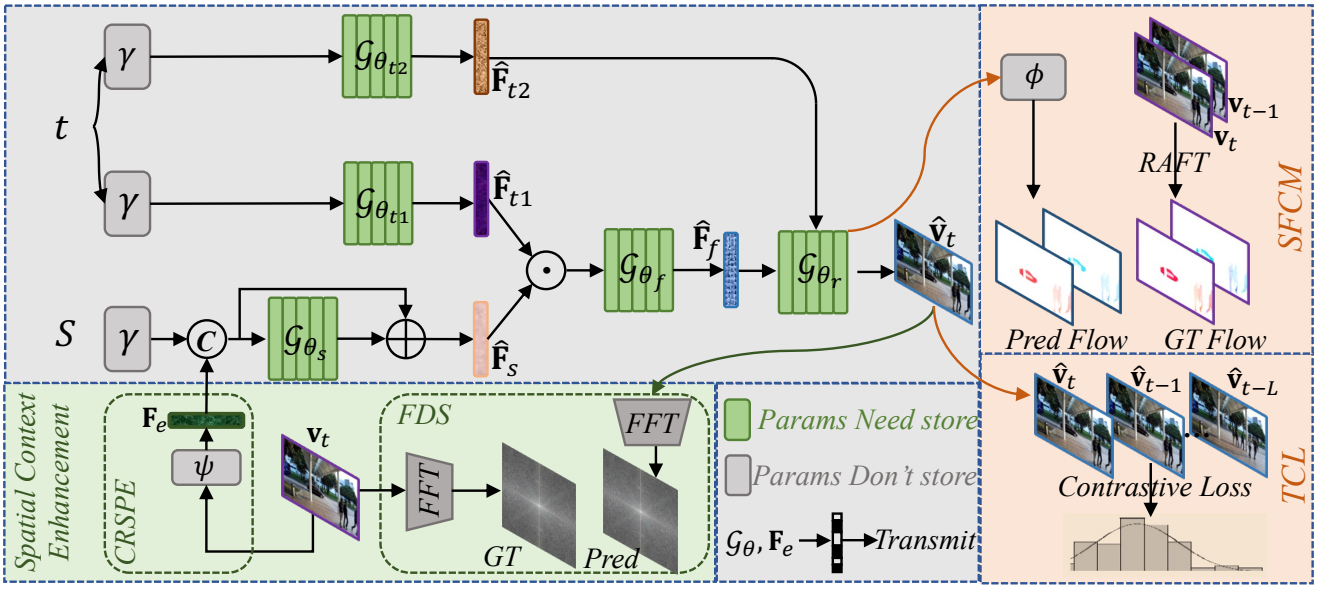

Figure 3. The framework of our proposed model-based video compression method.

2. Related works

2.1. Video Compression Methods.

Traditional video coding methods mainly follow hybrid video coding framework by improving various coding tools e.g., short-distance intra prediction [5], affine motion estimation [72] and low-rank in-loop filter [73]. However, these coding tools are designed so delicately that the traditional coding framework faces performance bottleneck [75]. Deep learning technique has also been applied to improve video coding performance, and achieved great progress in recent years. According to recent review works in [34, 40], deep learning based video compression methods can be roughly divided into two categories: deep-tool methods and deep-framework methods. Deep-tool methods [9, 35, 37, 53, 69, 76] still follow traditional hybrid video coding framework, and replace one or more coding tools by deep neural networks (DNN), such as intra/inter prediction, probability distribution prediction and in-loop filtering, in the traditional framework. However, restricted by the traditional framework, these methods cannot make full use of DNN, which limits its potential improvements. Hence, recent deep-framework methods [1, 12, 15, 16, 19–21, 26–28, 33, 36, 38, 41, 47–51, 56–60, 65, 67, 68, 70, 71, 77, 78] propose to construct end-to-end DLVC frameworks. Although DLVC methods can leverage the benefits of end-to-end learning strategy, these methods still remove temporal redundancy by only referring to one or limited neighboring frames, which limits its performance improvement. In this paper, our proposed MVC method can simultaneously leverage the correlations of all frames in the scene and find a much more compact sub-space for videos.

2.2. Implicit Neural Representation.

INR is a new paradigm to parameterize a wide range of signals, and its key idea is to represent an object as a function approximated by neural networks. DeepSDF [46] is one of the early works on INR, which is a neural network representation for 3D shapes. Recently, many works are proposed to represent different objects with INR, such as 3D shapes [42] and scene representation [31, 43], etc. Due to the representation ability of INR, some works [13, 14, 54] introduce INR to image compression task, proposing MIC methods. The work INVC [74] also applies INR to video compression network, and it directly uses INR to represent each frame then estimates motion and residual information. In fact, the INVC still follows the framework of existing common deep video compression and cannot fully explore the potential of INR. Therefore, its performance is even lower than H.264. In this paper, we propose the MVC method by designing novel modules for spatial context enhancement and temporal correlation capturing to simultaneously model video sequences in one scene, and the performance of our MVC method can outperform H.266.

3. Method

We first introduce the preliminary knowledge of this paper. Then we elaborate on our main contributions in this paper. Finally, we describe the whole compression pipeline of this work. In this paper, the original video sequence containing T frames is written as $\mathbf{V} = \{\mathbf{v}_t\}_{t=0}^{T-1}$, and the corresponding reconstruction video sequence is $\hat{\mathbf{V}} = \{\hat{\mathbf{v}}_t\}_{t=0}^{T-1}$. The framework of our proposed MVC network is in Fig. 3.

3.1. Preliminaries

The typical video INR (V-INR) methods [7,32] represent the video as a mapping network $\mathcal{G}_\theta : \mathbb{R} \rightarrow \mathbb{R}^{3\times H\times W}$ parameterized by the network weight θ, where H, W are the spatial size of the video frame. As shown in Fig. 3, the whole mapping network contains five sub-networks: $\mathcal{G}_\theta = \{\mathcal{G}_{\theta_{t1}}, \mathcal{G}_{\theta_{t2}}, \mathcal{G}_{\theta_s}, \mathcal{G}_{\theta_f}, \mathcal{G}_{\theta_r}\}$, and its corresponding weights are: $\theta = \{\theta_{t1}, \theta_{t2}, \theta_s, \theta_f, \theta_r\}$.

Firstly, the V-INR performs the regular positional encoding [43] on the scalar frame index t, then maps the temporal positional encoding to a feature vector $\hat{\mathbf{F}}_{t1} \in \mathbb{R}^d$ through the MLP operation $\mathcal{G}_{\theta_{t1}}$:

$$\hat{\mathbf{F}}_{t1} = \mathcal{G}_{\theta_{t1}}(\gamma(t)), \tag{1}$$

where the $\gamma(t)$ means the positional encoding:

$$\begin{aligned}\gamma(t) = (&\sin\left(b^0\pi t\right), \cos\left(b^0\pi t\right), \ldots, \sin\left(b^{l-1}\pi t\right), \\ &\cos\left(b^{l-1}\pi t\right)).\end{aligned} \tag{2}$$

$b = 1.25$ and $l = 80$ follow the common setting in [7,32].

Secondly, the V-INR initializes the normalized grid coordinates S, which are expected to contain the spatial context. The size of S is $\mathbb{R}^{2\times h\times w}$. For S, the V-INR first encodes it into $\hat{S}$ with similar positional encoding $\gamma(\cdot)$ in Eqn. 2. Then, V-INR adopts a small transformer [62] with single-head self-attention and residual connection to encourage the feature fusion among spatial locations:

$$\hat{\mathbf{F}}_s = \mathcal{G}_{\theta_s}(\hat{S}) + \hat{S}, \tag{3}$$

where $\mathcal{G}_{\theta_s}$ is the transformer network. The "*C*" (Concatenation, Fig. 3) dose not exist in the original V-INR network, which is the contribution proposed in this paper.

Thirdly, the V-INR fuses $\hat{\mathbf{F}}_{t1}$ and $\hat{\mathbf{F}}_s$ with another MLP network $\mathcal{G}_{\theta_f}$:

$$\hat{\mathbf{F}}_f = \mathcal{G}_{\theta_f}(\hat{\mathbf{F}}_s \odot \hat{\mathbf{F}}_{t1}). \tag{4}$$

Finally, the network $\mathcal{G}_{\theta_r}$ is used to reconstruct frames:

$$\begin{aligned}\hat{\mathbf{v}}_t &= \mathcal{G}_{\theta_r}(\hat{\mathbf{F}}_f, \hat{\mathbf{F}}_{t2}), where \\ \hat{\mathbf{F}}_{t2} &= \mathcal{G}_{\theta_{t2}}(\gamma(t)).\end{aligned} \tag{5}$$

In the reconstruction stage, the V-INR further leverages the MLP $\mathcal{G}_{\theta_{t2}}$ on $\gamma(t)$ to make sufficient and thorough use of the temporal embedding. Note that, $\mathcal{G}_{\theta_r}$ contains five upsampling stages with pixel-shuffle operation. The $\mathcal{L}_1$ loss is used for optimization:

$$\mathcal{L}_{spa} = \mathcal{L}_1(\hat{\mathbf{v}}_t, \mathbf{v}_t). \tag{6}$$

Figure 4. Reconstructed results without/with CRSPE.

3.2. Spatial Context Enhancement

3.2.1 Context-related Spatial Positional Embedding

Existing video INR methods implicitly represent spatial context by fixed grid coordinates S, which are context-agnostic spatial positional embeddings. However, they suffer from spatial variations among different frame contents, and the network would spend more encoding times and larger parameters. To address this problem, we propose the CRSPE. In Fig. 4, we show two frames containing spatial variations in video sequence. With our proposed CRSPE, the network can reconstruct higher-quality results.

For the original frame $\mathbf{v}_t$, we use a 80×80 convolutional operation ψ to transform it to $\mathbf{F}_e \in \mathbb{R}^{c\times h\times w}$:

$$\mathbf{F}_e = \psi(\mathbf{v}_t). \tag{7}$$

In this paper, we set $c = 3$ and the embedding $\mathbf{F}_e$ is the same spatial size as S. Although transmitting the embedding $\mathbf{F}_e$ needs extra bitrates, the model can obtain better reconstruction quality, which would benefit the RD performance. Specifically, our proposed framework spends extra 10% (Bits per pixel, Bpp) with about 0.9dB (PSNR) increasing for 720p videos.

3.2.2 Frequency Domain Supervision

To further improve the performance of INR network, some works [24,25] utilized frequency-aware operations in their networks, which can capture high-frequency details of images. However, these operations are difficult to directly be applied to video compression task, since these operations need extra sophisticated modules and would introduce more coding bitrates. In order to keep more high-frequency detailed information, we propose the frequency-aware perceptual loss without adding network parameters. In particular, we utilize the fast Fourier transform (FFT) to transform $\hat{\mathbf{v}}_t$ and $\mathbf{v}_t$ into frequency domain, then calculate the $\mathcal{L}_1$ loss:

$$\mathcal{L}_{freq} = \mathcal{L}_1(\mathbf{FFT}(\hat{\mathbf{v}}_t), \mathbf{FFT}(\mathbf{v}_t)). \tag{8}$$

3.3. Temporal Correlation Capturing

Temporal correlation is another crucial factor for the MVC to distinguish the representation of different frames.

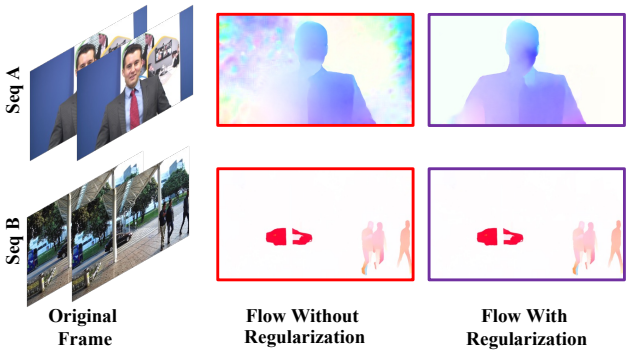

Figure 5. The scene flow without/with regularization loss.

Existing V-INR methods are not efficient in modeling complex temporal variations, especially with long-term temporal correlation, by only utilizing different positional encodings. Therefore, we first propose SFCM to capture short-term temporal correlation. We also design TCL to enhance the modeling efficiency for long-term temporal correlation.

3.3.1 Scene Flow Constrain Mechanism.

In Eqn. 5, $\mathcal{G}_{\theta_r}(\cdot)$ has five up-sampling stages. The last stage contains the feature $\hat{\mathbf{Z}}_t$, which is for frame $\hat{\mathbf{v}}_t$ generation. In Fig. 3, we design the extra scene flow prediction head ϕ on $\hat{\mathbf{Z}}_t$ to predict the forward and backward flows at timestamp t. $\mathbf{O}_t^f$ means the forward flow map from $t-1$ to t, and $\mathbf{O}_t^b$ is the backward flow map from t to $t-1$. We do not evaluate the optical flow for $t=0$. To supervise the predicted flow maps, we estimate their corresponding ground truth (GT) $\{\mathbf{O}_t^{fgt}, \mathbf{O}_t^{bgt}\}$ from original images $\{\mathbf{v_{t-1}}, \mathbf{v}_t\}$, through optical flow estimation algorithm RAFT [61]. Finally, the SFCM is optimized by $\mathcal{L}_1$ loss:

$$\mathcal{L}_{vanilla-flow} = \mathcal{L}_1(\mathbf{O}_t^f, \mathbf{O}_t^{fgt}) + \mathcal{L}_1(\mathbf{O}_t^b, \mathbf{O}_t^{bgt}). \quad (9)$$

Since we do not directly supervise the SFCM from the real annotation, the generated GT from RAFT would contain some noise due to different reasons, such as algorithm accuracy. As shown in Fig. 5, the noisy GT-flow in the SeqA would disrupt the whole training process. To address this problem, we further design the regularization mechanism. Specifically, we use a 1×1 convolutional operation on $\hat{\mathbf{Z}}_t$ to evaluate the 2-channels regularization map $\mathbf{W}$, following by *Softmax* operation. Finally, we select the values in the second channel, $\mathbf{W}^{(1)}$, to re-write the $\mathcal{L}_{flow}$:

$$\begin{aligned}\mathcal{L}_{flow} =& \mathcal{L}_1(\mathbf{W}^{(1)} \cdot \mathbf{O}_t^f, \mathbf{W}^{(1)} \cdot \mathbf{O}_t^{fgt}) + \\ & \mathcal{L}_1(\mathbf{W}^{(1)} \cdot \mathbf{O}_t^b, \mathbf{W}^{(1)} \cdot \mathbf{O}_t^{bgt}).\end{aligned} \quad (10)$$

Moreover, we further design the loss $\mathcal{L}_{ent}$ to let the $\mathbf{W}$ tend to be binary. The $\mathcal{L}_{ent}$ is defined as:

$$\mathcal{L}_{ent} = -(\mathbf{W}^{(0)} \cdot log(\mathbf{W}^{(0)}) + \mathbf{W}^{(1)} \cdot log(\mathbf{W}^{(1)})). \quad (11)$$

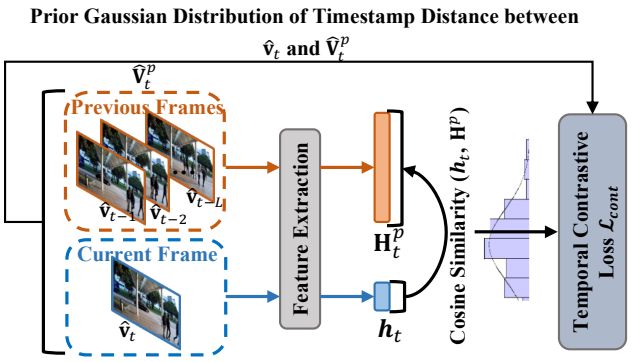

Figure 6. Illustration of the proposed TCL. The prior Gaussian distribution is computed by the timestamp distance between current frame $\hat{\mathbf{v}}_t$ and the previous frames $\hat{\mathbf{V}}^p$.

$\mathcal{L}_{ent}$ would be zero when the channels of $\mathbf{W}$ are the one-hot vector and would be maximum when they have an equal probability. As shown in Fig. 5, the proposed regularization mechanism can erase some noisy flow information and not affect other flow information.

3.3.2 Temporal Contrastive Loss

Although SFCM can capture the short-term temporal correlation of two adjacent frames, the long-term temporal correlation is also important and has not been well utilized in previous video compression methods. To further improve the representation capability of our proposed network, we aim to model the long-term temporal correlation between the current and previous reconstruction frames. However, the main challenge is the lack of labeled data. Recent works such as [6, 8, 10] propose contrastive learning mechanisms, which have proven to be powerful tools for learning representations without labeled data. Therefore, we leverage contrastive learning in this paper to address this challenge.

SimCLR [8] introduces a contrastive loss called NT-Xent, which maximizes agreement between augmented views of the same instance. In typical instance discrimination, all instances other than the positive reference are considered negatives. However, in the video task, neighboring frames around the current frame are highly correlated, and regarding them as negatives directly may hinder learning. To address this issue, we propose a novel TCL that minimizes the KL-divergence between the embedding similarity of $\{\hat{\mathbf{v}}_t, \hat{\mathbf{V}}_t^p\}$ and a prior Gaussian distribution, as shown in Fig. 6. Here, $\hat{\mathbf{V}}_t^p = \{\hat{\mathbf{v}}_{t-1}, \hat{\mathbf{v}}_{t-2}, ...\hat{\mathbf{v}}_{t-L}\}$ is the set of previous reconstruction frames from timestamp $t-L$ to $t-1$.

Concretely, we use a pre-trained feature extraction network ResNet [17] and the global pooling operation to project reconstruction frames $\{\hat{\mathbf{v}}_t, \hat{\mathbf{V}}_t^p\}$ to latent embeddings $\{\mathbf{h}_t, \mathbf{H}_t^p\}$. Due to the fact that temporally adjacent frames are more highly correlated than those faraway ones, we assume the embedding similarity between $\mathbf{h}_t$ and $\mathbf{H}_t^p$ should follow a prior Gaussian distribution of timestamp

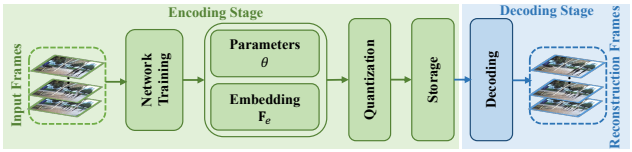

Figure 7. Compression pipeline of our proposed MVC method.

distance between $\hat{\mathbf{v}}_t$ and $\hat{\mathbf{V}}_t^p$. This assumption motivates us to use KL-divergence to optimize the embedding space. Specifically, let $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v}/\|\boldsymbol{u}\|\|\boldsymbol{v}\|$ denote cosine similarity, and $Gau(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$ denote the Gaussian function, where σ^2 is the variance. We formulate the loss of the t-th frame as:

$$\mathcal{L}_{cont} = -\sum_{j=t-L}^{t-1} w_{tj} \log \frac{\exp\left(\mathrm{sim}\left(\mathbf{h}_t, \mathbf{h}_j\right)/\tau\right)}{\sum_{k=t-L}^{t-1} \exp\left(\mathrm{sim}\left(\mathbf{h}_t, \mathbf{h}_k\right)/\tau\right)},$$
$$w_{tj} = \frac{Gau\left(t-j\right)}{\sum_{k=t-L}^{t-1} Gau\left(t-k\right)}, \tag{12}$$

where w_{tj} is the normalized Gaussian weight between timestamps t and j, and $\sigma^2 = 10$ is as default. $\tau = 0.1$ is the temperature parameter. We set $L = 80$ in this paper.

Total Loss. The total loss $\mathcal{L}_{total}$ in this work is written as:

$$\mathcal{L}_{total} = \mathcal{L}_{spa} + \mathcal{L}_{freq} + \mathcal{L}_{flow} + \mathcal{L}_{ent} + \mathcal{L}_{cont}. \tag{13}$$

3.4. Compression Pipeline for Proposed Method

The compression pipeline of this paper is presented in Fig. 7. To compress an input video sequence $\mathbf{V}$, at the training (encoding) stage, we first train the proposed network on $\mathbf{V}$. The network parameters θ and the embedding $\mathbf{F}_e$ are then quantized into a bitstream for storage and transmission. The network's parameters are single precision floating point numbers that require 32 bits per weight, but to reduce the memory requirement, we use the AI Model Efficiency Toolkit (AIMET)[1] for quantization. We apply quantization specific to each weight tensor such that the uniformly-spaced quantization grid is adjusted to the value range of the tensor. The bitwidth determines the number of discrete levels, i.e., quantization bins, and we find empirically that bitwidths in the range of 7-8 lead to optimal rate-distortion performance for our models as shown in the supplement material. Apart from AIMET, we can also use the quantization mechanism proposed in COOL-CHIC [23] to quantize network parameters. Finally, the quantized network parameters and the embedding are used for decoding. Although more advanced model compression techniques, such as model pruning, can further improve performance, the discussion for model compression is beyond the scope of this paper and will be considered in future work.

[1] https://quic.github.io/

3.5. Discussion of Our Method

Our proposed MVC network encodes an entire video sequence simultaneously into network parameters. The bits per pixel (Bpp) for the video sequence are calculated using the formula: $(NP + FE)/(FN \times W \times H)$. Here, *NP* denotes the bits required for network parameters, *FE* denotes the bits required for feature embeddings, and *FN* is the total number of frames, while W and H are the width and height of a frame, respectively.

The current version of the MVC network may only be suitable for the non-live (or non-delay-constrained) scene. In this scene, the video can be encoded and stored first and decoded later for analysis or video on demand (VoD), when required. Unlike traditional video coding standards and DLVC methods that decode video frames sequentially, our MVC network allows for random access to video frames at any time during decoding. This feature enables researchers to easily analyze or request saved videos.

4. Experiments

4.1. Implementation Details

Experiment Settings. We use the PyTorch framework to implement our method, accelerated by the NVIDIA RTX 3090. Following the works [7, 32], we train the model using Adam optimizer [22]. The initialization learning rate is $5e^{-4}$, and decreases 10% every 10 epochs. Each model is trained with the batchsize of 1.

Evaluation Metrics. In this paper, we use PSNR to measure the quality of the reconstructed frames, which is the commonly used quality metric in video compression. The compression rate is measured by the Bpp. Additionally, we evaluate various video compression methods using the BDBR [3]. The BDBR is a measure of how much bit rate is saved when compared to the baseline algorithm at the same quality, measured by PSNR.

Evaluation Databases. As discussed in Section.3.5, the current version of our proposed MVC network may only be applicable to certain non-live scenes. Therefore, we first evaluate the performance of our proposed method on video sequences that can be first encoded and stored, and then used for analysis or VoD when required, such as conference and surveillance videos. For conference videos, we choose three typical 720p resolution video sequences from HEVC ClassE [55]. For surveillance videos, we choose three 1080p resolution video sequences from IEEE1857 [39] and three 1080p resolution video sequences from the work [44].

4.2. Comparison Methods

We compare our method against traditional codecs, INR-based methods and DLVC approaches. Traditional video compression codecs contain H.265 [55] (HM16.21) and H.266 [4] (VTM12.0), where H.266 is still a video codec

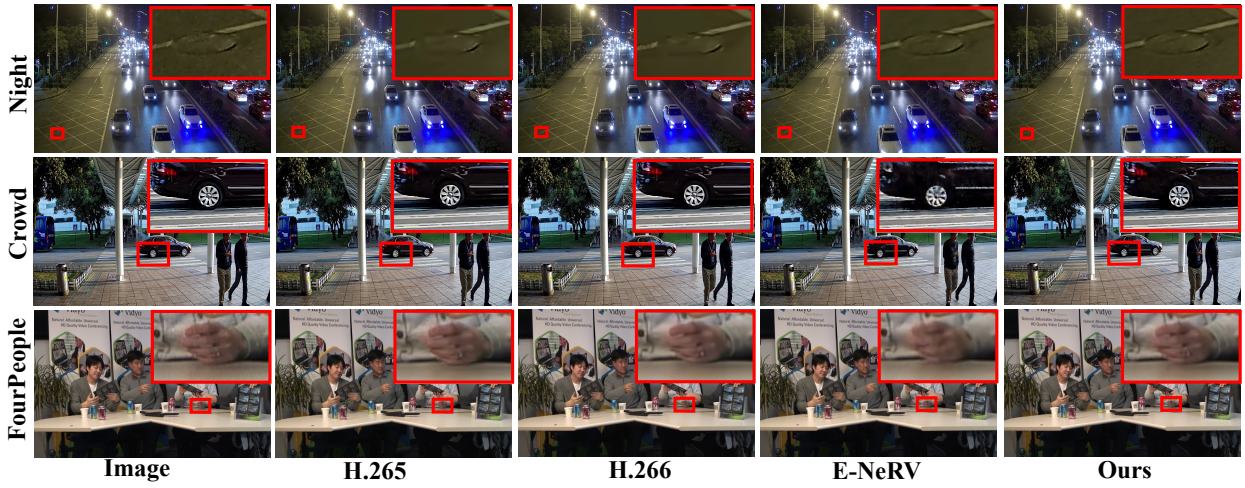

Figure 8. The qualitative results of our and other SOTA methods.

Table 1. BDBR(%) performances of different methods when compared with H.266 on the 9 different video sequences in terms of PSNR.

Models	Surveillance-1 [39]				Surveillance-2 [44]				Conference [55]			
	Crowd	Bulong	Night	*Average*	Seq001	Seq002	Seq003	*Average*	FourPeople	KristenAndSara	Johnny	*Average*
HSTE(MM2022)	7.12	13.43	12.25	10.93	7.65	5.43	8.87	7.32	16.75	15.83	5.46	12.68
CANFVC(ECCV2022)	8.79	12.76	14.36	11.97	8.79	9.32	7.93	8.68	15.58	16.95	7.34	13.29
E-NeRV(ECCV2022)	10.01	17.93	13.75	13.90	12.21	11.43	8.73	10.79	35.38	18.87	28.75	27.67
Ours	-31.13	-14.94	-22.84	-22.97	-20.97	-16.02	-22.07	-19.68	-8.45	-4.64	-8.83	-7.30

with the best performance for most cases. The INR-based method is E-NeRV [32] (ECCV2022), which is also similar to our proposed method using implicit neural representation technique. DLVC approachs are HSTE [30] (MM2022) and CANFVC [18] (ECCV2022). HSTE claims that its performance has already surpassed H.266 and it is the industry leader in deep video compression.

4.3. Quantitative and Qualitative Evaluation

The quantitative results are presented in Table. 1, where we report the BDBR performance. It can be observed that our proposed MVC method consistently outperforms the traditional video compression codec H.266 in all video sequences by a significant margin. Specifically, in the surveillance videos, our method achieves an approximately 1dB PSNR improvement at a similar Bpp compared to H.266, which represents a remarkable performance boost in the video compression task. In Fig. 8, we observe that although the SOTA INR-based method E-NeRV is capable of reconstructing the video scene to some extent, its performance is inferior to that of H.266. For instance, in the "Crowd" sequence, detailed information about the wheel is missing. This illustrates that the existing SOTA INR method fails to capture the spatial context of the current video sequences accurately when given limited network parameters. A straightforward solution for INR networks is to increase the network parameters to enhance the representation capability. However, for the video compression task, we need to

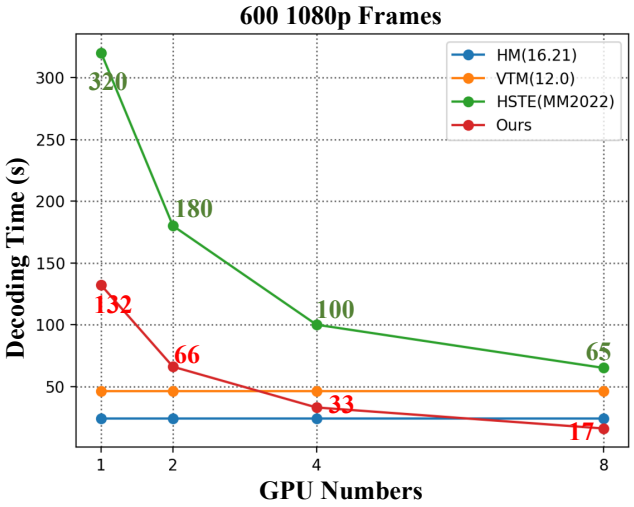

Figure 9. Decoding time (s) of our proposed method and other traditional video compression codecs.

focus more on rate-distortion performance. Therefore, we propose CRSPE to enable the INR network to learn from context-related spatial positional embeddings and introduce FDS to capture high-frequency details.

Another limitation of the existing SOTA INR-based methods is that they struggle to fully capture the temporal correlation of video sequences. For example, when reconstructing regions with complex motions, such as finger regions in the "FourPeople" sequence, there are noticeable blurs. To address this issue, we have designed the SFCM

Table 2. The BDBR(%) performances of different settings when compared with H.266. "w/o" means "without" operation.

Video Groups	Architecture					SFCM		TCL			
	Baseline	+CRSPE	+CRSPE +FDS	+CRSPE +FDS+SFCM	+CRSPE +FDS+SFCM+TCL	w/o Regularization	w/o $\mathcal{L}_{ent}$	w/o PGD	$L = 40$	$L = 80$	$L = 120$
Surveillance-1	**13.82**	**3.15**	**-3.53**	**-18.65**	**-22.97**	**-3.65**	**-16.69**	**-18.12**	**-19.85**	**-22.97**	**-21.97**
Surveillance-2	**10.74**	**3.34**	**-3.61**	**-15.67**	**-19.68**	**-3.68**	**-13.47**	**-15.52**	**-17.03**	**-19.68**	**-18.86**
Conference	**27.59**	**14.24**	**3.21**	**-4.21**	**-7.30**	**4.33**	**-3.31**	**-3.97**	**-5.04**	**-7.30**	**-6.15**

and TCL modules, which are specifically aimed at making INR networks more effective for complex scenes. As shown in Fig. 8 and Table. 1, our proposed MVC method is better able to handle conference scenes with complex motions compared to E-NeRV. Note that, to align the bitrate points of our method and HSTE and CANFVC, we retrain these two methods using the source code provided by the authors.

4.4. Decoding Time Comparison

We analyze our method's decoding time. The platform is Intel(R) Xeon(R) Gold 5218R CPU and NVIDIA RTX3090 GPU. The results are shown in Fig. 9. When decoding 600 1080p video frames, HM16.21 takes approximately 24 seconds, and VTM12.0 takes about 47 seconds. With one GPU, our proposed method takes around 132 seconds, while HSTE takes about 320 seconds to decode the same number of frames. However, as the number of GPUs increases, our proposed method becomes faster and finally achieves real-time decoding at 35 frames per second (FPS). Our proposed method utilizes a network that directly models the entire 600 frames, making it parallel-friendly and capable of utilizing all GPUs to their full potential. In contrast, HSTE decodes frames using a sequential process, making it parallel-unfriendly and unable to fully utilize all GPUs. Note that, our network allows for random access to video frames at any time during decoding. This feature enables researchers to easily analyze or request saved videos.

4.5. Ablation Studies

About the whole architecture. In this paper, we mainly propose four modules to improve the performance, including CRSPE, FDS, SFCM and TCL. We progressively integrate these modules into the ***Baseline*** network, and the performance change is shown in Table. 2 (***Architecture***). The architecture of ***Baseline*** is similar to the gray regions in Fig. 3 (Without "*C*" operation). It can be seen that all these four modules can consistently improve the network performance. The series of experiments validate the effectiveness of our proposed modules in this paper.

About SFCM. In the proposed SFCM, we design two important mechanisms, including flow regularization and $\mathcal{L}_{ent}$. Their ablation studies are shown in Table. 2 (***SFCM***). Firstly, without flow regularization operation ***(SFCM(w/o Regularization))***, the performance of our proposed network has no obvious improvement, compared to the (***+CRSPE+FDS***). Regretfully, we even find that there is a bit of performance loss in the Conference group. The main reason is that without a flow regularization mechanism, the supervised labels would contain noise. The noise information would disrupt the whole training process. Moreover, we further design the $\mathcal{L}_{ent}$ to constrain the regularization map $\mathbf{W}$ (Eqn. 11). Without $\mathcal{L}_{ent}$, the $\mathbf{W}$ can be viewed as the one-channel attention map, and the final reconstruction results of the whole MVC network would indirectly supervise the $\mathbf{W}$. Hence, there exist the risks that the $\mathbf{W}$ cannot be well learned, leading to network performance (***SFCM(w/o*** $\mathcal{L}_{ent}$***)***) drop to some extent, compared to the (***+CRSPE+FDS+SFCM***). Adding $\mathcal{L}_{ent}$, it directly supervises the regularization map $\mathbf{W}$ and makes the map tend to be binary, which lets the network be converged better.

About TCL. In our proposed TCL, we design the prior Gaussian distribution (PGD) to ensure the TCL can correctly select positive samples. As shown in Table. 2 (***TCL***), without PGD (***w/o PGD***), the proposed contrastive loss has no contribution to long-term temporal correlation modeling. Moreover, we also find that the length variable L of selected previous frames would affect the performance of TCL. Compared with larger values L=80 and 120, the smaller value would slightly hurt the performance, but it still can significantly improve the performance compared to the model without using TCL (***+CRSPE+FDS+SFCM***). In this paper, we set $L = 80$, since we find that a longer L cannot further improve the performance.

5. Conclusion

In this paper, we propose a novel MVC framework for the video compression task. We leverage the INR network as our backbone network, and discuss the limitations of existing INR networks when they are applied to the video coding task. To address these limitations, we propose context-related spatial positional embedding and frequency domain supervision to enhance the spatial context ability of existing INR networks. Moreover, we design the scene flow constrain mechanism and temporal contrastive loss to improve the temporal modeling ability. In experiments, our proposed MVC method consistently outperforms H.266 for all the test video sequences, which may inspire researchers to explore the video compression task in a new light.

References

[1] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Ballé, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8500–8509, 2020. 3

[2] Yunpeng Bai, Chao Dong, and Cairong Wang. Ps-nerv: Patch-wise stylized neural representations for videos. *CoRR*, abs/2208.03742, 2022. 2

[3] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *VCEG-M33*, 2001. 1, 6

[4] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Trans. Circuits Syst. Video Technol.*, 31(10):3736–3764, 2021. 1, 2, 6

[5] Xiaoran Cao, Changcai Lai, Yunfei Wang, Lingzhi Liu, Jianhua Zheng, and Yun He. Short distance intra coding scheme for high efficiency video coding. *IEEE Transactions on Image Processing*, 22(2):790–801, 2012. 3

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Adv. Neural Inform. Process. Syst.*, 2020. 5

[7] 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, 4, 6

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int. Conf. Machin. Learn.*, volume 119, pages 1597–1607. PMLR, 2020. 5

[9] Tong Chen, Haojie Liu, Qiu Shen, Tao Yue, Xun Cao, and Zhan Ma. Deepcoder: A deep neural network based video compression. In *Visual Communications and Image Processing*, pages 1–4, 2017. 3

[10] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. 5

[11] Wenxue Cui, Tao Zhang, Shengping Zhang, Feng Jiang, Wangmeng Zuo, Zhaolin Wan, and Debin Zhao. Convolutional neural networks based intra prediction for HEVC. In *Data Compression Conference*, page 436, 2017. 1

[12] Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers. Neural inter-frame compression for video coding. In *Int. Conf. Comput. Vis.*, pages 6420–6428, 2019. 3

[13] Emilien Dupont, Adam Golinski, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: COmpression with implicit neural representations. In *Neural Compression: From Information Theory to Applications – Workshop @ Int. Conf. Learn. Represent.*, 2021. 2, 3

[14] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Golinski, Yee Whye Teh, and Arnaud Doucet. COIN++: data agnostic neural compression. *CoRR*, abs/2201.12904, 2022. 2, 3

[15] Noor Fathima Ghouse, Jens Petersen, Guillaume Sautière, Auke Wiggers, and Reza Pourreza. A neural video codec with spatial rate-distortion control. In *WACV*, pages 5354–5363. IEEE, 2023. 3

[16] AmirHossein Habibian, Ties van Rozendaal, Jakub M. Tomczak, and Taco Cohen. Video compression with rate-distortion autoencoders. In *Int. Conf. Comput. Vis.*, pages 7032–7041, 2019. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 5

[18] 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 *Eur. Conf. Comput. Vis.*, pages 207–223. Springer, 2022. 1, 7

[19] Zhihao Hu, Zhenghao Chen, Dong Xu, Guo Lu, Wanli Ouyang, and Shuhang Gu. Improving deep video compression by resolution-adaptive flow coding. In *Eur. Conf. Comput. Vis.*, pages 193–209, 2020. 3

[20] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5921–5930, June 2022. 3

[21] Zhihao Hu, Guo Lu, and Dong Xu. FVC: A new framework towards deep video compression in feature space. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1502–1511, 2021. 1, 3

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2015. 6

[23] Théo Ladune, Pierrick Philippe, Félix Henry, and Gordon Clare. Cool-chic: Coordinate-based low complexity hierarchical image codec. *arXiv preprint arXiv:2212.05458*, 2022. 6

[24] Jaewon Lee, Kwang Pyo Choi, and Kyong Hwan Jin. Learning local implicit fourier representation for image warping. *Eur. Conf. Comput. Vis.*, 2022. 2, 4

[25] Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1919–1928, 2022. 2, 4

[26] Bo Li, Zhengxing Sun, Lv Tang, and Anqi Hu. Two-b-real net: Two-branch network for real-time salient object detection. In *ICASSP*, pages 1662–1666. IEEE, 2019. 3

[27] Bo Li, Zhengxing Sun, Lv Tang, Yunhan Sun, and Jinlong Shi. Detecting robust co-saliency with recurrent co-attention neural network. In *IJCAI*, pages 818–825. ijcai.org, 2019. 3

[28] Bo Li, Lv Tang, Senyun Kuang, Mofei Song, and Shouhong Ding. Toward stable co-saliency detection and object co-segmentation. *IEEE Trans. Image Process.*, 31:6532–6547, 2022. 3

[29] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. In *Adv. Neural Inform. Process. Syst.*, pages 18114–18125, 2021. 1

[30] Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *ACM Multimedia*, pages 1503–1511. ACM, 2022. 7

[31] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 2, 3

[32] Zizhang Li, Mengmeng Wang, Huaijin Pi, Kechun Xu, Jianbiao Mei, and Yong Liu. E-nerv: Expedite neural video representation with disentangled spatial-temporal context. *Eur. Conf. Comput. Vis.*, 2022. 2, 4, 6, 7

[33] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-LVC: multiple frames prediction for learned video compression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3543–3551, 2020. 1, 3

[34] Dong Liu, Yue Li, Jianping Lin, Houqiang Li, and Feng Wu. Deep learning-based video coding: A review and a case study. *ACM Comput. Surv.*, 53(1):11:1–11:35, 2020. 3

[35] Zhenyu Liu, Xianyu Yu, Yuan Gao, Shaolin Chen, Xiangyang Ji, and Dongsheng Wang. CU partition mode decision for HEVC hardwired intra encoder using convolution neural network. *IEEE Trans. Image Process.*, 25(11):5088–5103, 2016. 3

[36] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: an end-to-end deep video compression framework. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11006–11015, 2019. 1, 3

[37] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Zhiyong Gao, and Ming-Ting Sun. Deep kalman filtering network for video compression artifact reduction. In *Eur. Conf. Comput. Vis.*, pages 591–608, 2018. 3

[38] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. An end-to-end learning framework for video compression. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3292–3308, 2021. 3

[39] Siwei Ma, Cliff Reader, Tiejun Huang, Feng Wu, and Wen Gao. Ieee audio video coding working group (1857wg). In *IEEE https://sagroups.ieee.org/1857/*. 6, 7

[40] Siwei Ma, Xinfeng Zhang, Chuanmin Jia, Zhenghui Zhao, Shiqi Wang, and Shanshe Wang. Image and video compression with neural networks: A review. *IEEE Trans. Circuits Syst. Video Technol.*, 30(6):1683–1698, 2020. 3

[41] Fabian Mentzer, George Toderici, David Minnen, Sung Jin Hwang, Sergi Caelles, Mario Lucic, and Eirikur Agustsson. VCT: A video compression transformer. *Adv. Neural Inform. Process. Syst.*, 2022. 3

[42] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4460–4470. Computer Vision Foundation / IEEE, 2019. 3

[43] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis.*, volume 12346, pages 405–421. Springer, 2020. 2, 3, 4

[44] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3153–3160. IEEE, 2011. 6, 7

[45] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2

[46] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 165–174, 2019. 3

[47] Jorge Pessoa, Helena Aidos, Pedro Tomás, and Mário A. T. Figueiredo. End-to-end learning of video compression using spatio-temporal autoencoders. In *SiPS*, pages 1–6, 2020. 3

[48] Oren Rippel, Alexander G. Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir D. Bourdev. ELF-VC: efficient learned flexible-rate video coding. In *IEEE Int. Conf. Comput. Vis.*, pages 14459–14468. IEEE, 2021. 3

[49] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G. Anderson, and Lubomir D. Bourdev. Learned video compression. In *IEEE Int. Conf. Comput. Vis.*, pages 3453–3462, 2019. 3

[50] Yibo Shi, Yunying Ge, Jing Wang, and Jue Mao. Alphavc: High-performance and efficient learned video compression. *Eur. Conf. Comput. Vis.*, 2022. 3

[51] Yibo Shi, Yunying Ge, Jing Wang, and Jue Mao. Alphavc: High-performance and efficient learned video compression. In *Eur. Conf. Comput. Vis.*, volume 13679, pages 616–631. Springer, 2022. 3

[52] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 2

[53] Rui Song, Dong Liu, Houqiang Li, and Feng Wu. Neural network-based arithmetic coding of intra prediction modes in HEVC. In *Visual Communications and Image Processing*, pages 1–4, 2017. 3

[54] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *Eur. Conf. Comput. Vis.*, 2022. 2, 3

[55] Gary J. Sullivan, Jens-Rainer Ohm, Woojin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12):1649–1668, 2012. 1, 2, 6, 7

[56] Lv Tang. Cosformer: Detecting co-salient object with transformers. *CoRR*, abs/2104.14729, 2021. 3

[57] Lv Tang and Bo Li. CLASS: cross-level attention and supervision for salient objects detection. In *ACCV*, volume 12624, pages 420–436. Springer, 2020. 3

[58] Lv Tang, Bo Li, Senyun Kuang, Mofei Song, and Shouhong Ding. Re-thinking the relations in co-saliency detection. *IEEE Trans. Circuits Syst. Video Technol.*, 32(8):5453–5466, 2022. 3

[59] Lv Tang, Bo Li, Yanliang Wu, Bo Xiao, and Shouhong Ding. Fast: Feature aggregation for detecting salient object in real-time. In *ICASSP*, pages 1525–1529. IEEE, 2021. 3

[60] Lv Tang, Bo Li, Yijie Zhong, Shouhong Ding, and Mofei Song. Disentangled high quality salient object detection. In *IEEE Int. Conf. Comput. Vis.*, pages 3560–3570. IEEE, 2021. 3

[61] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *Eur. Conf. Comput. Vis.*, volume 12347, pages 402–419. Springer, 2020. 5

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, pages 5998–6008, 2017. 4

[63] Zhao Wang, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, and Siwei Ma. Three-zone segmentation-based motion compensation for video compression. *IEEE Trans. Image Process.*, 28(10):5091–5104, 2019. 1

[64] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):560–576, 2003. 1

[65] Chao-Yuan Wu, Nayan Singhal, and Philipp Krähenbühl. Video compression through image interpolation. In *Eur. Conf. Comput. Vis.*, pages 425–440, 2018. 3

[66] Ning Yan, Dong Liu, Houqiang Li, Bin Li, Li Li, and Feng Wu. Convolutional neural network-based fractional-pixel motion compensation. *IEEE Trans. Circuits Syst. Video Technol.*, 29(3):840–853, 2019. 1

[67] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with hierarchical quality and recurrent enhancement. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6627–6636, 2020. 3

[68] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with recurrent auto-encoder and recurrent probability model. *IEEE J. Sel. Top. Signal Process.*, 15(2):388–401, 2021. 3

[69] Ren Yang, Mai Xu, Zulin Wang, and Tianyi Li. Multi-frame quality enhancement for compressed video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6664–6673, 2018. 3

[70] Ruihan Yang, Yibo Yang, Joseph Marino, and Stephan Mandt. Hierarchical autoregressive modeling for neural video compression. In *Int. Conf. Learn. Represent.*, 2021. 3

[71] M. Akin Yilmaz and A. Murat Tekalp. End-to-end rate-distortion optimized learned hierarchical bi-directional video compression. *IEEE Trans. Image Process.*, 31:974–983, 2022. 3

[72] Kai Zhang, Yi-Wen Chen, Li Zhang, Wei-Jung Chien, and Marta Karczewicz. An improved framework of affine motion compensation in video coding. *IEEE Transactions on Image Processing*, 28(3):1456–1469, 2018. 1, 3

[73] Xinfeng Zhang, Ruiqin Xiong, Weisi Lin, Jian Zhang, Shiqi Wang, Siwei Ma, and Wen Gao. Low-rank-based nonlocal adaptive loop filter for high-efficiency video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(10):2177–2188, 2016. 3

[74] Yunfan Zhang, Ties van Rozendaal, Johann Brehmer, Markus Nagel, and Taco Cohen. Implicit neural video compression. *CoRR*, abs/2112.11312, 2021. 3

[75] Yongfei Zhang, Chao Zhang, Rui Fan, Siwei Ma, Zhibo Chen, and C.-C. Jay Kuo. Recent advances on HEVC inter-frame coding: From optimization to implementation and beyond. *IEEE Trans. Circuits Syst. Video Technol.*, 30(11):4321–4339, 2020. 3

[76] Lei Zhao, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. Enhanced motion-compensated video coding with deep virtual reference frame generation. *IEEE Transactions on Image Processing*, 28(10):4832–4844, 2019. 3

[77] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding. Detecting camouflaged object in frequency domain. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4494–4503. IEEE, 2022. 3

[78] Yijie Zhong, Bo Li, Lv Tang, Hao Tang, and Shouhong Ding. Highly efficient natural image matting. In *BMVC*, page 325. BMVA Press, 2021. 3