Neural Video Compression with Spatio-Temporal Cross-Covariance Transformers

Zhenghao Chen[†]
The University of Sydney
Sydney, Australia

Lucas Relic
ETH Zürich
Zürich, Switzerland

Roberto Azevedo[*]
DisneyResearch|Studios
Zürich, Switzerland

Yang Zhang
DisneyResearch|Studios
Zürich, Switzerland

Markus Gross
ETH Zürich
Zürich, Switzerland

Dong Xu
The University of Hong Kong
Hong Kong SAR, China

Luping Zhou
The University of Sydney
Sydney, Australia

Christopher Schroers[*]
DisneyResearch|Studios
Zürich, Switzerland

ABSTRACT

Although existing neural video compression (NVC) methods have achieved significant success, most of them focus on improving either temporal or spatial information separately. They generally use simple operations such as concatenation or subtraction to utilize this information, while such operations only partially exploit spatio-temporal redundancies. This work aims to effectively and jointly leverage robust temporal and spatial information by proposing a new 3D-based transformer module: Spatio-Temporal Cross-Covariance Transformer (ST-XCT). The ST-XCT module combines two individual extracted features into a joint spatio-temporal feature, followed by 3D convolutional operations and a novel spatio-temporal-aware cross-covariance attention mechanism. Unlike conventional transformers, the cross-covariance attention mechanism is applied across the feature channels without breaking down the spatio-temporal features into local tokens. Such design allows for modeling global cross-channel correlations of the spatio-temporal context while lowering the computational requirement. Based on ST-XCT, we introduce a novel transformer-based end-to-end optimized NVC framework. ST-XCT-based modules are integrated into various key coding components of NVC, such as feature extraction, frame reconstruction, and entropy modeling, demonstrating its generalizability. Extensive experiments show that our ST-XCT-based NVC proposal achieves state-of-the-art compression performances on various standard video benchmark datasets.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Image compression**.

KEYWORDS

Video compression, neural network, transformer

ACM Reference Format:
Zhenghao Chen[†], Lucas Relic, Roberto Azevedo[*], Yang Zhang, Markus Gross, Dong Xu, Luping Zhou, and Christopher Schroers[*]. 2023. Neural Video Compression with Spatio-Temporal Cross-Covariance Transformers. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3611960

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3611960

1 INTRODUCTION

Video compression is an important task due to the increasing demand for storing and transmitting videos. Traditional video compression standards (*e.g.*, H.266/VVC [8]) heavily rely on complex hand-crafted modules that must be individually optimized. Recently, Neural Video Compression (NVC) methods benefit from powerful end-to-end optimized neural modules (*e.g.*, neural entropy model [6, 11, 32]) and have achieved comparable performance as traditional codecs [18].

Despite attracting increasing research attention, most existing NVC methods focus on generating better temporal or spatial contexts separately. Recent approaches have adopted multi-frame alignment [21], deformable convolutional warping [16], and coarse-to-fine temporal context mining [36] to produce better temporal information. Even though a few research attempts have investigated how to take advantage of temporal and spatial information jointly, they adopt simple and non-optimal strategies. Specifically, one popular category of methods [4, 16, 26], *deep residual coding*, subtracts the temporal information (*e.g.*, aligned frame or feature) and compresses the residuals. Another category [17, 18, 24], *deep contextual coding*, concatenates the spatial and temporal contexts to build a dependency model. Indeed, effectively fusing spatial and temporal information is a non-trivial task.

Vision Transformer (ViT) has recently demonstrated an excellent ability to fuse information using its powerful attention mechanism in various video restoration tasks [19, 20, 37]. The rise of ViT has also inspired NVC research. Mentzer *et al.* [31] proposed the first ViT-based NVC method VCT, which fuses temporal and spatial information using both ViT encoder and decoder in its entropy model. However, such a proposal is still limited. First, despite bringing improvement, directly adopting a vanilla ViT also comes with a large computational burden, which makes NVC methods hard to optimize and can limit their performance. Second, VCT only exploits the spatio-temporal context in the entropy model, even though NVC also involves other essential coding components that should not be ignored.

[†]This work was done during Zhenghao Chen's internship at DisneyResearch|Studios.
E-mail: zhenghao.chen@sydney.edu.au
[*]Roberto Azevedo and Christopher Schroers are the corresponding authors.
E-mail:{roberto.azevedo, christopher.schroers}@disneyresearch.com

To effectively leverage spatio-temporal correlations and address the challenges of integrating transformers into NVC frameworks, we propose a Spatio-Temporal Cross-Covariance Transformer (ST-XCT) as a universal transformer-based feature fusion module. ST-XCT first aggregates two 2D-based individual features into a 3D-based joint spatio-temporal feature, which includes an additional temporal dimension. Then it uses a 3D convolutional operation to mix the spatio-temporal information locally while applying the attention mechanism across the entire feature channel to produce a global spatio-temporal-aware cross-covariance attention matrix. Unlike conventional ViT strategies, which decompose the feature into local patches and operate the attention mechanism among spatial dimensions, our ST-XCT module directly computes the cross-covariance attention without splitting the features into several parts. Such a design not only allows ST-XCT to model global spatio-temporal correlations but also maintains a linear complexity. To improve the information flow, we further introduce a 3D-based feed-forward gate mechanism to update the feature by using a "gate" to regulate and update the information flow. Such a "gate" is learned by 3D convolutional operations to exploit the spatio-temporal correlation between neighboring pixels.

Furthermore, we explore how to effectively deploy our universal ST-XCT in NVC. To fully benefit from ST-XCT and exploit the spatio-temporal characteristic, we integrate it into three key coding operations: feature extraction, frame reconstruction, and entropy modeling. We first apply ST-XCT in hierarchical feature extraction to progressively fuse multi-resolution features and generate latent features with spatio-temporal correlation. Then, we deploy ST-XCT to fuse two priors into a spatio-temporal-aware prior, which improves conditional entropy coding. Last, to reconstruct the frame more effectively, we adopt ST-XCT to fuse multi-scale aligned features progressively.

Overall, our novel end-to-end optimized NVC framework, empowered by our universal ST-XCT modules, allows us to effectively exploit spatial and temporal information across various coding operations, resulting in significantly improved video compression performance. Through extensive experiments with UVG [2], MCL [43], and HEVC [38] datasets, we demonstrate that our proposed framework not only achieves better performance than traditional video codecs (*e.g.*, H.266/VVC [8] and H.265/HEVC [38]) on most of the benchmarks (except HEVC Class C), but also outperforms state-of-the-art NVC methods (*e.g.*, DCVC* [18] and VCT [31]).

Our contributions can be summarized as follows:

- We propose a novel transformer-based feature fusion module, *ST-XCT*, which generates a spatio-temporal-aware cross-covariance attention matrix with a linear complexity and better leverages both spatial and temporal information.
- We propose an end-to-end optimized transformer-based NVC framework, which applies the ST-XCT modules to *all* key coding components. It includes transformer-based multi-scale feature extraction, spatio-temporal hybrid entropy model, and multi-scale frame reconstruction components.
- We conduct extensive experiments and ablation studies, demonstrating that our proposed NVC framework achieves state-of-the-art performance, outperforming both traditional video codecs and previous transformer-based NVC frameworks.

2 RELATED WORK

2.1 Neural Image Compression

Compared to conventional hand-engineered algorithms [7, 39, 42], neural image compression (NIC) codecs have shown superior compression performance. NIC-based methods [9, 11, 22, 29, 34, 40, 41] can be end-to-end optimized on large datasets. Hyper-prior-based methods [6, 30] utilize a hierarchical design to model dependencies across various image scales, while autoregressive-prior methods [11, 32] further capture the spatial correlation between neighboring pixels.

2.2 Neural Video Compression

Existing NVC methods can be divided into two categories: *deep residual coding* and *deep contextual coding*.

Methods in the *deep residual coding* category [4, 10, 12, 14, 15, 21, 25, 28, 45]) follow traditional video compression frameworks. They perform predictive coding (*e.g.*, motion compensation) and encode residual information. The pioneering work DVC [28] replaces all key coding operations with CNNs in the traditional residual coding pipeline, enabling end-to-end optimization. Most subsequent works build upon this pipeline and improve performance using more powerful modules and advanced techniques. For example, to produce better-aligned context (*e.g.*, frame or feature), M-LVC [21] uses a multi-frame alignment strategy, while FVC [16] adopts a deformable convolutional warping technique.

The works in the *deep contextual coding* category [13, 17, 18, 24, 36] extend the generative-based NIC methods and build spatio-temporal conditional entropy models using spatial and temporal contexts. Lombardo *et al.* [24] produced the dynamic global and local latent variables, while Habibian *et al.* [13] adopted a 3D-based VAE with a gated mechanism to generate temporal context. Different from the aforementioned methods using all accumulated information, Li *et al.* proposed DCVC [17], which directly adopts a motion compensation strategy to generate temporal context from the adjacent compressed frame. To enhance the temporal information, Sheng *et al.* [36] further improved DCVC by using multi-scale temporal context mining. The most recent version of DCVC (referred to as DCVC* in this work) with hybrid entropy models [18] outperforms the traditional video coding standard H.266/VVC [8].

Nevertheless, to the best of our knowledge, most NVC research focuses on producing better spatial and temporal information individually rather than on how to aggregate and leverage them effectively. For instance, deep residual coding methods simply subtract the temporal information, whereas deep contextual coding approaches mostly adopt common operations (such as concatenation) to combine the learned temporal and spatial contexts. Instead of such simple combinations, we propose a transformer-based module ST-XCT, which leverages the powerful cross-covariance attention mechanism to support better exploitation of the spatio-temporal correlation.

2.3 Transformers in Neural Compression

More recently, Vision Transformers (ViT) have been incorporated into NIC for building better entropy models. Qian *et al.* [34] leveraged a ViT-decoder (*i.e.*, masked mechanism) for auto-regressive

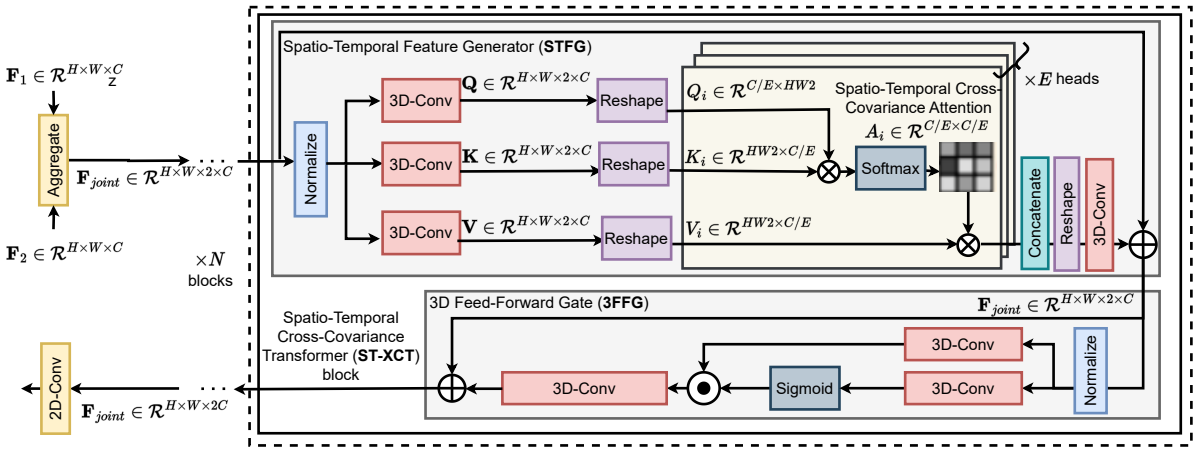


Figure 1: Overview of our proposed Spatio-Temporal Cross-Covariance Transformer (ST-XCT) module. It consists of N ST-XCT blocks to fuse the joint feature $\mathbf{F}_{joint}$. For each ST-XCT block, we first use Spatio-Temporal Cross-Covariance Generator (STCG) and apply multi-head attention mechanism with E heads to produce spatio-temporal-aware cross-covariance attention matrix A_i, which exploits the global spatio-temporal correlation. Then, we further adopt 3D Feed-Forward Gate (3FFG) mechanism to control the information flow by using 3D convolutional operation to learn a "gate" and element-wise multiplication to filter.

modeling and a ViT-encoder for hyper-prior modeling to perform entropy coding. Lu *et al.* [29] proposed a causal attention module for adaptive context modeling. Zhu *et al.* [47] adopted Swin-Transformer [23] to replace all convolutional operations in both hyper-prior [6] and auto-regressive [32] methods.

Regarding NVC, VCT [31] is the first transformer-based method. It divides video frames into tokens and adopts a ViT-decoder as an autoregressive-based entropy model to perform the conditional entropy coding for each token. However, VCT has two main drawbacks. First, it is ineffective to directly use vanilla ViT with a token-based strategy, which brings higher computational complexity and makes the whole framework harder to optimize in an end-to-end manner. Second, VCT only applies ViT in the entropy model, ignoring other important coding components of NVC.

In this work, we address the above issues and investigate a more effective combination of NVC and Transformers. First, we propose the ST-XCT module to directly operate on features without partitioning them into tokens. Such a design allows our model to learn global spatio-temporal context with a linear complexity. Second, we deploy ST-XCT into multiple key coding components rather than only the entropy model. ST-XCT is inspired by previous works that transpose the spatial dimensions of 2D features and compute spatial cross-covariance attention [5, 46]. Our method is the first cross-covariance Transformer for NVC that models spatio-temporal correlation by transposing both spatial and temporal dimensions from a 3D-based feature and generating a spatio-temporal-aware cross-covariance matrix. We further adopt a 3D-based gated mechanism to enhance the produced 3D joint spatio-temporal features.

3 METHODOLOGY

3.1 ST-XCT

ST-XCT is a transformer module that produces joint spatio-temporal features by mixing two input features spatially and temporally. We designed ST-XCT as a universal module that can be easily integrated into different key coding components of NVC frameworks.

Architecture. Fig 1 details the ST-XCT architecture. It takes two individual 2D features $\mathbf{F}_1, \mathbf{F}_2 \in \mathcal{R}^{H\times W\times C}$ as inputs[1], where H, W, C respectively represent height, width, and the number of channels. Then, it aggregates these two features by creating an additional temporal channel (*i.e.*, 2 in our case), with which we produce a 3D-based joint spatio-temporal feature $\mathbf{F}_{joint} \in \mathcal{R}^{H\times W\times 2\times C}$. This joint feature is then fused by several ST-XCT blocks, which contain two components: *Spatio-Temporal Feature Generator (STFG)* and *3D Feed-Forward Gate (3FFG)*. After iteratively being fused by these two operations in each ST-XCT block, we then reshape the joint feature to $\mathbf{F}_{joint} \in \mathcal{R}^{H\times W\times 2C}$ and feed it into a 2D convolutional layer to generate a final 2D joint feature $\mathbf{F}_{joint} \in \mathcal{R}^{H\times W\times C}$.

Spatio-Temporal Feature Generator. STFG first normalizes the 3D-based joint feature, followed by applying 3D convolutional layers with $1\times1\times1$ and then $3\times3\times3$ kernels, which operate in the channel dimension to mix spatial and temporal information locally. Through this operation, we can obtain 3D-based Query (Q), Key (K), and Value (V) features, which are subsequently reshaped to $Q \in \mathcal{R}^{C\times HW2}$ and $K, V \in \mathcal{R}^{HW2\times C}$. Then, we adopt a multi-head attention mechanism to partition these features into E heads along the feature channel dimension to obtain $Q_i \in \mathcal{R}^{C/E\times HW2}$ and $K_i, V_i \in \mathcal{R}^{HW2\times C/E}$ for each head i. Using partitioned features Q_i and K_i, we compute their spatio-temporal-aware cross-covariance attention matrix $A_i \in \mathcal{R}^{C/E\times C/E}$, via the dot-product operation and Softmax function. Next, we use a dot-product operation to multiply this attention matrix with the partitioned value feature V_i, and then reshape and concatenate all partitioned product features from all heads to $\mathbf{F}_{pro} \in \mathcal{R}^{H\times W\times 2\times C}$. Finally, we add $\mathbf{F}_{pro}$ back to the input joint feature $\mathbf{F}_{joint}$ following the conventional residual transformer

[1]We define the spatial dimension $H \times W$ as one dimension (*i.e.*, spatial dimension).

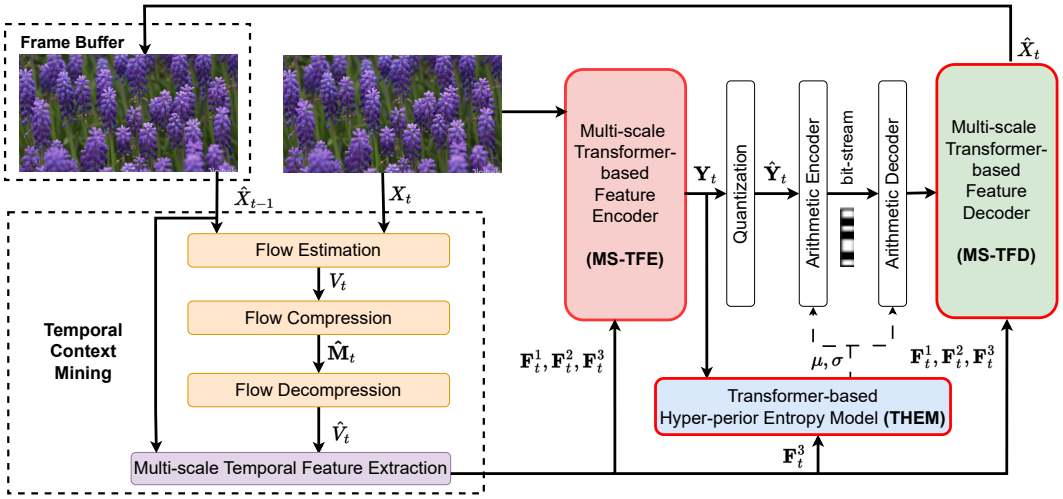


Figure 2: Overview of our proposed transformer-based neural video compression (NVC) framework. It takes the current frame X_t and reference frame $\hat{X}_{t-1}$ as inputs and produces the multi-scale temporal features $\mathbf{F}_t^1, \mathbf{F}_t^2, \mathbf{F}_t^3$ by using an optical-flow-based temporal context mining strategy. Next, it progressively fuses such temporal features with the feature extracted from the current frame X_t to produce the quantized latent feature $\hat{\mathbf{Y}}_t$ by using *Multi-scale Transformer-based Feature Encoder (MS-TFE)*. Then, it performs the entropy-coding to losslessly encode or decode $\hat{\mathbf{Y}}_t$ with the aid of *Transformer-based Hybrid Entropy Model (THEM)*. Last, we adopt *Multi-scale Transformer-based Feature Decoder (MS-TFD)* to reconstruct $\hat{\mathbf{Y}}_t$ back to the reconstructed frame $\hat{X}_t$. Note that we apply our proposed ST-XCT modules in MS-TFE, THEM and MS-TFD (highlighted in red box).

mechanism. We summarize the process as follows, where $[\cdot \oplus \cdot]$ denotes the concatenation operations,

$$
\begin{aligned}
\mathbf{Q}, \mathbf{K}, \mathbf{V} &= \mathbf{W}_q Norm(\mathbf{F}_{joint})), \mathbf{W}_k Norm(\mathbf{F}_{joint}), \mathbf{W}_v Norm(\mathbf{F}_{joint}) \\
Reshape(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &\rightarrow Q \in \mathcal{R}^{C \times HW2}, K \in \mathcal{R}^{HW2 \times C}, V \in \mathcal{R}^{HW2 \times C} \\
Q, K, V &= [Q_1 \oplus Q_2 ... \oplus Q_E], [K_1 \oplus K_2 ... \oplus K_E], [V_1 \oplus V_2 ... \oplus V_E] \\
A_i &= Softmax(Q_i \in \mathcal{R}^{C/E \times HW2} \cdot K_i \in \mathcal{R}^{HW2 \times C/E}) \\
F_{pro}^i &= V_i \in \mathcal{R}^{HW2 \times C/E} \cdot A_i \in \mathcal{R}^{C/E \times C/E} \\
Reshape([F_{pro}^1 \oplus F_{pro}^2 ... \oplus F_{pro}^E]) &\rightarrow \mathbf{F}_{pro} \in \mathcal{R}^{H \times W \times 2 \times C} \\
\mathbf{F}_{joint} &= \mathbf{F}_{joint} + \mathbf{W}_{pro} \mathbf{F}_{pro}
\end{aligned}
\tag{1}
$$

3D Feed-Forward Gate. We introduce 3FFG to further enhance the information flow by applying a gating mechanism commonly used in transformers. First, we take a 3D joint feature $\mathbf{F}_{joint}$ from STFG and use 3D convolutional layers with $1 \times 1 \times 1$ and then $3 \times 3 \times 3$ kernels to generate two separate features. Second, one of these features is transformed by the Sigmoid activation function to serve as a "gate", which is then element-wisely multiplied by another feature for information filtering. Lastly, the fused feature is fed into a 3D convolutional layer with $1 \times 1 \times 1$ kernel and added back to $\mathbf{F}_{joint}$. This process can be summarized as follows, where $\odot$ represents element-wise multiplication,

$$
\begin{aligned}
\mathbf{F}_{gate} &= Sigmoid(\mathbf{W}_1 Norm(\mathbf{F}_{joint})) \\
\mathbf{F}_{joint} &= \mathbf{W}_3(\mathbf{F}_{gate} \odot (\mathbf{W}_2 Norm(\mathbf{F}_{joint}))) + \mathbf{F}_{joint}
\end{aligned}
\tag{2}
$$

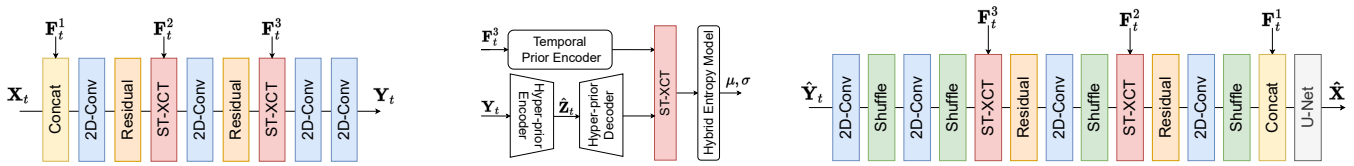
While STFG produces the joint spatio-temporal feature by exploiting global spatio-temporal correlation using a cross-covariance attention mechanism, 3FFG concentrates on better information transformation by exploring the correlation between spatio-temporal neighboring pixel positions using 3D convolutional operations.

3.2 Neural Video Compression with ST-XCT

We deploy ST-XCT in an NVC framework, as shown in Fig. 2. Our framework compresses the current frame X_t of a video sequence $\mathcal{X} = \{X_1, X_2, \ldots X_{t-1}, X_t, \ldots\}$ to obtain the reconstructed frame $\hat{X}_t$, where the subscript t represents the current time-step t. The process involves four main steps: 1) *Temporal Context Mining*; 2) *Spatio-Temporal Feature Extraction*; 3) *Entropy Coding*; and 4) *Frame Reconstruction*. This section discusses each of those steps in detail and explains how we apply ST-XCT to the different parts of our end-to-end pipeline.

3.2.1 Temporal Context Mining. We adopt an optical-flow-based compensation strategy to explore temporal information, as in most NVC methods [17, 18, 36]. We first estimate the raw optical flow V_t between previous reconstructed frame $\hat{X}_{t-1}$ (*i.e.*, reference frame) and current-coding frame X_t by using SpyNet [35], followed by using an auto-encoder-style network to compress V_t to the quantized motion feature $\hat{\mathbf{M}}_t$, and decompressing $\hat{\mathbf{M}}_t$ back to the reconstructed flow $\hat{V}_t$. Last, we use a multi-scale temporal context extraction strategy as in [36] by taking $\hat{V}_t$ and $\hat{X}_{t-1}$ as inputs. This results in three scales of temporal context information, $\mathbf{F}_t^1, \mathbf{F}_t^2, \mathbf{F}_t^3$.

3.2.2 Spatio-Temporal Feature Extraction. In existing *deep contextual coding* frameworks [18, 36], multi-scale temporal context features, $\mathbf{F}_t^1, \mathbf{F}_t^2, \mathbf{F}_t^3$, are extracted from previous frames and spatial



(a) Multi-scale Transformer-based Feature Encoder (MS-TFE) **(b) Transformer-based Hybrid Entropy Model (THEM)** **(c) Multi-scale Transformer-based Feature Decoder (MS-TFD)**

Figure 3: Transformer-based key coding components, where we apply our ST-XCT modules in our NVC framework.

features are extracted from the current frame. These features are directly used for encoding the current frame X_t into a latent feature $\mathbf{Y}_t$, which is then quantized to $\hat{\mathbf{Y}}_t$. However, in this work, we propose to utilize the *Multi-scale Transformer-based Feature Encoder (MS-TFE)* to extract joint spatio-temporal features.

MS-TFE (Fig. 3a) extracts the spatial information from the current frame X_t and fuses it with previously produced multi-scale temporal features $\mathbf{F}_t^1, \mathbf{F}_t^2, \mathbf{F}_t^3$. At each scale, the spatial and temporal features are combined and eventually produce the latent features $\mathbf{Y}_t$ with rich spatio-temporal information. We first concatenate the largest-scale temporal feature $\mathbf{F}_t^1$ with X_t, followed by processing this combined feature with the standard 2D convolutional operations. We subsequently fuse $\mathbf{F}_t^2$ and $\mathbf{F}_t^3$ with 4 and 6 proposed ST-XCT blocks and 2 heads for all blocks at the two subsequent scales.

3.2.3 Entropy Coding. To losslessly encode (*resp.,* decode) the produced quantized latent feature $\hat{\mathbf{Y}}_t$, we adopt arithmetic encoder (*resp.,* decoder) to convert $\hat{\mathbf{Y}}_t$ to bit-stream (*resp.,* convert bit-stream to $\hat{\mathbf{Y}}_t$). To reduce the bitrate, we adopt a *Transformer-based Hyper-prior Entropy Model (THEM)* to better estimate the distribution of $\hat{\mathbf{Y}}_t$ and improve the cross-entropy coding.

THEM (Fig. 3b) estimates the probability distribution of the quantized latent feature $\hat{\mathbf{Y}}_t$ for bitrate saving. We extend the hybrid entropy model in [18] by using our ST-XCT model instead of simple concatenation. First, we generate the temporal prior from the smallest-scale temporal feature $\mathbf{F}_t^3$ using a temporal prior encoder. Second, we produce the spatio-temporal prior from the latent feature $\mathbf{Y}_t$, for which we adopt a hyper-prior encoder to generate the quantized prior feature $\hat{\mathbf{Z}}_t$, followed by using a decoder to generate decoded spatio-temporal prior feature. Last, we apply our ST-XCT module with 16 ST-XCT blocks and 6 heads for each cross-covariance attention mechanism to fuse temporal and decoded prior features to generate a better spatio-temporal prior. Based on this prior we can then apply the hybrid entropy model in [18].

3.2.4 Frame Reconstruction. During the decoding stage, we generate the reconstructed frame $\hat{X}_t$ from the quantized latent feature $\hat{\mathbf{Y}}_t$. To better leverage spatial and temporal information, we adopt the *Multi-scale Transformer-based Feature Decoder (MS-TFD).*

MS-TFD (Fig. 3c) reconstructs the quantized latent feature $\hat{\mathbf{Y}}_t$ back to the reconstructed frame $\hat{X}_t$ using two ST-XCT modules. It uses 6 ST-XCT blocks for fusing $\mathbf{F}_t^3$, while 4 ST-XCT blocks for fusing $\mathbf{F}_t^2$, where we set the number of heads as 2 for all blocks at each scale. For fusing $\mathbf{F}_t^1$ at the highest resolution, we simply concatenate it and feed the concatenated feature into a U-Net as in [18] to generate the reconstructed frame $\hat{X}_t$.

3.2.5 Loss function. We optimize our method by solving the following rate-distortion optimization problem:

$$\mathcal{L} = \lambda D(X_t, \hat{X}_t) + R(\hat{\mathbf{Y}}_t) + R(\hat{\mathbf{Z}}_t) + R(\hat{\mathbf{M}}_t), \tag{3}$$

where $D(\cdot)$ represents the distortion between the reconstructed and original frames. $R(\cdot)$ represents the bitrate cost in the compression procedure. Here, $\hat{\mathbf{Y}}_t$, $\hat{\mathbf{Z}}_t$ and $\hat{\mathbf{M}}_t$ respectively represent the quantized latent feature, the quantized prior feature, and the quantized motion feature. We use λ as a hyper-parameter to control the trade-off between rate and distortion.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets. **Training.** We trained our models on the Vimeo-90K dataset [44], which comprises 89,800 video sequences of 7 frames each with resolution 448×256. We randomly cropped the frames to 256×256 and applied random horizontal and vertical flips for the data augmentation. **Evaluation.** To evaluate the performance of our method, we used sequences from the HEVC [38] (Class B, C, D, and E), UVG [2], and MCL-JCV [43] datasets, which are widely used as evaluation benchmarks for video compression. The resolutions of videos in HEVC dataset range from 416×240 to 1920×1080 pixels, while those in UVG and MCL-JCV are both 1920×1080 pixels. Consistent with previous benchmark [4, 14–16, 26–28, 31], we cropped the smaller dimension of all frames to a multiple of 64. To measure the compression performance, we used bits per pixel (bpp), while PSNR in RGB space between the target and reconstructed frames was utilized as the distortion metric.

4.1.2 Baselines. We assessed the effectiveness of our method compared to traditional codecs and state-of-the-art learning-based methods. We use H.265 and H.266 (and respective reference implementations HM-16.21 [1] and VTM-13.2 [3]) as traditional codecs baselines with the same configuration parameters as in [18]. We use FVC [16], C2F [15], DCVC [17], and DCVC* [18] as neural codecs baselines. FVC is a leading deep residual coding NVC, while C2F is one of NVC methods achieving comparable performance to traditional video codecs. Both DCVC and improved DCVC* were included due to similar network architecture to our method.

For all baselines, we employed an intra-frame period of 32 and compressed a total of 96 frames from each sequence in test datasets as in [18]. We recomputed the performance metrics for DCVC*, H.266, and H.265 using publicly available code and model weights,

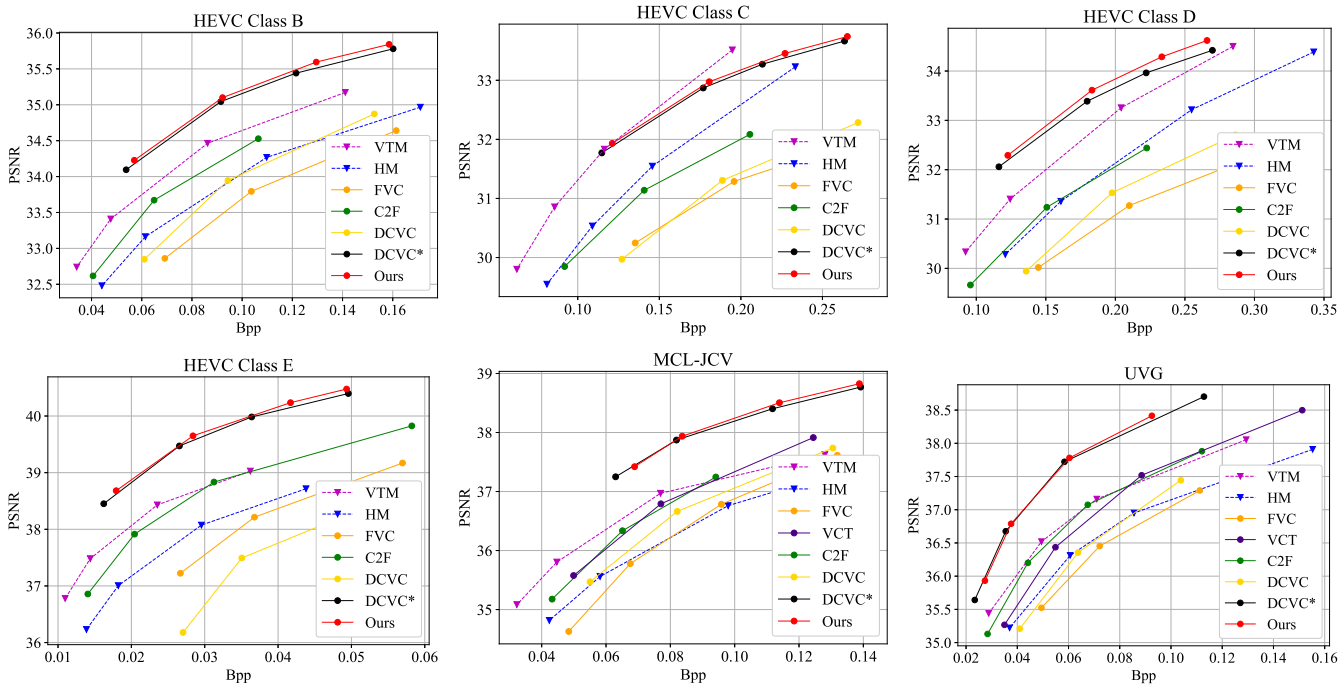


Figure 4: Rate-distortion (*i.e.*, bitrate vs PSNR) performance comparison of our and other state-of-the-art methods in the HEVC, MLC-JCV, and UVG datasets.

while FVC, C2F, DCVC, and VCT, reported numbers from the original authors were utilized. VCT did not report performance on the HEVC dataset, therefore we only compared our performance with VCT on the MCL-JCV and UVG datasets.

4.1.3 Implementation Details. We adopted a two-stage training strategy. In the first stage, we initialized the relevant modules with pre-trained weights from DCVC*, while randomly initializing all other layers, including ST-XCT modules. The weights of pre-trained layers were frozen, and the remaining layers were trained with the learning rate of 5×10^{-5} for 20K iterations. The learning rate was then reduced to 1×10^{-6} for the subsequent 30K iterations. In the second stage, we optimized all parameters end-to-end with a learning rate of 5×10^{-6}. After 40K steps, the learning rate was further reduced to 1×10^{-6}, and the model was trained for another 40K iterations. We employed a multi-frame training strategy with a batch size of 2 (corresponding to 2 sequences). Each sequence contains 7 frames, where the 1^{st} frame was treated as an intra-frame and the rest as inter-frames. As our focus was on inter-frame coding, we adopted the intra-frame coding method from [18], which was not optimized during training. Our model was implemented in PyTorch [33] and optimized using Adam. It was trained on a single NVIDIA A100 GPU, taking approximately 5 days to converge.

4.2 Results

Quantitative Comparison. Table 1 shows the BD Rate (%) performance of ours and existing methods on the evaluation datasets using H.266/VTM-13.2 [3] as the anchor. Our method significantly outperforms the existing transformer-based video codecs, VCT, and achieved better (on HEVC datasets) or comparable (on MCL-JVC and UVG datasets) performance than other NVC codecs. The experimental results indicate an average of 25.6% savings in bitrate compared to the leading traditional codec H.266 and 2.5% bitrate savings over the current state-of-the-art method DCVC*. Rate-distortion comparisons are reported in Fig. 4, where our method uses fewer bits than the baseline methods for similar reconstruction quality. We observed the largest improvements on HEVC Class D and Class E with 5.4% and 3.5% bitrate-saving from DCVC*. One potential reason is that these datasets are with lower resolution, which more closely matches our training setup. Smaller improvements were made on higher-resolution datasets such as HEVC Class B, MCL-JVC, and UVG which further supports this observation.

Qualitative Comparison. Fig. 7 shows qualitative comparisons between our method, DVC*, and VTM at similar bitrates. Generally, our method achieves a better perceptual reconstruction performance. In the *BlowingBubbles* sequence (1^{st} row), our method recovers more structural information in the tissue box, which the baseline methods are unable to achieve. Similar phenomena are visible in the *BasketballPass* and *BasketballDrive* sequences (2^{nd} and 3^{rd} row, respectively), in which some structural details are lost in the baseline methods. Furthermore, H.266 introduced ringing artifacts in the basketball in the latter sequence.

Discussion. Overall, the above results demonstrate the effectiveness of the ST-XCT module in spatio-temporal feature encoding, entropy modeling, and frame reconstruction, confirming its performance improvements over existing methods. Although DCVC* remains competitive with our method in UVG and MCL-JCV datasets,

	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	Avg
VTM-13.2	0.0	0.0	0.0	**0.0**	0.0	0.0	0.0
HM-16.21	36.61	42.27	44.51	36.48	28.36	53.85	40.35
FVC	52.57	41.82	78.42	99.98	76.89	78.32	71.34
C2F	3.07	16.49	19.87	50.50	28.39	13.62	21.99
DCVC	38.68	24.49	50.41	93.00	55.91	140.15	67.11
DCVC*	*-35.79*	*-38.08*	*-26.01*	8.87	*-14.95*	*-36.63*	*-23.77*
VCT	10.46	9.93	-	-	-	-	10.20
Ours	**-36.19**	**-39.08**	**-27.35**	*7.20*	**-18.63**	**-39.74**	**-25.63**

Table 1: BD Rate (%) Comparison for PSNR. VTM-13.2 is used as the anchor. Negative values indicate bitrate savings and positive values indicate extra bitrate cost. The best-performing model is indicated in bold and the second best in italic.

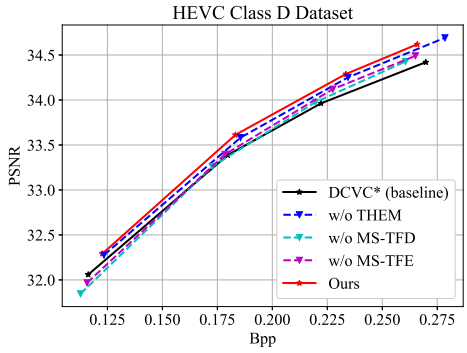


Figure 5: Ablation Study I: Removing the different transformer-based components from our NVC framework.

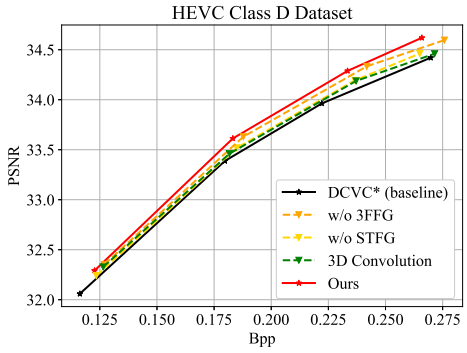


Figure 6: Ablation Study II: Removing the different individual components from our ST-XCT module.

our method shows considerable improvements on all other datasets. Moreover, our method shows significant improvements over all the other NVC methods, including the only transformer-based NVC architectures, VCT.

4.3 Ablation Studies

4.3.1 Ablation I: Individually Removing the transformer-based components from our NVC framework. To verify the effectiveness of our ST-XCT module, we conducted ablation studies on HEVC Class D dataset, where we removed the ST-XCT modules at various stages from our pipeline and replaced them with concatenation to the next block (as in our baseline DCVC*). Fig. 6 displays the rate-distortion performance of our proposed method without THEM, MS-TFD, and MS-TFE. The results indicated a noticeable drop in performance when MS-TFD and MS-TFE were removed, bringing an extra 5.6% and 4.8% bitrates. This indicates the superior ability of ST-XCT for extracting and fusing multi-scale spatio-temporal features. Also, the results showed reduced performance without THEM, which brings an extra 2.0% bitrates. It provides further evidence that ST-XCT can efficiently exploit spatio-temporal correlation to benefit entropy coding. Our findings suggest that including ST-XCT in all the stages yields the best performance.

4.3.2 Ablation II: Individually Removing Components from ST-XCT. We have also conducted ablation studies to verify the effectiveness of the STFG and 3FFG modules inside ST-XCT. We removed STFG and 3FFG individually from our ST-XCT in all coding components of our framework and evaluated the performance on the HEVC Class D dataset. In addition, we replaced our ST-XCT module (*i.e.*, STFG + 3FFG module) by directly utilizing the 3D joint feature, which is produced by aggregation and fusion by 3D convolutional operation. We observed that: i) although only using STFG (*i.e.*, w/o 3FFG) or 3FFG (*i.e.*, w/o STFG) can still improve from the baseline (2.9% and 1.5% reduced bitrates from DCVC*), they both performed worse than our proposed method with full ST-XCT blocks (resulting in an additional 2.8% and 4.2% bitrate costs from our full method); ii) the alternative method without using our ST-XCT module is also notably worse than our proposed framework (4.8% increased bitrate), but slightly better than the baseline DCVC* (1.0% reduced bitrate). The latter shows that the joint 3D feature can intuitively bring a marginal improvement but cannot sufficiently capture rich spatio-temporal information by itself.

4.3.3 Complexity Study. We also conducted runtime analysis and model size experiments, which are summarized in Table 2. Specifically, the runtime metrics were computed by compressing all sequences from the HEVC Class D dataset with a resolution of 384×192. All complexity experiments were performed on a machine with a single NVIDIA RTX 3090 GPU and Intel Core i7-6700k CPU. Compared to the other transformer-based NVC codec (VCT),

Figure 7: Qualitative comparison between our NVC method, DCVC*, and VTM-13.2. The demonstrated images are labeled as PSNR@bpp. Best viewed on screen.

	#Params (M)	GPU (MiB)	Enc (ms/Frame)
VTM	N/A	N/A	6104
VCT	121.1	461.9	268
DCVC*	17.5	66.8	19
Ours	26.8	102.2	56

Table 2: The complexity of our method and other video codecs. The GPU peak memory (*i.e.*, GPU) and encoding time (*i.e.*, Enc) are calculated by using HEVC ClassD dataset.

our compression framework has 77% fewer parameters and encodes a frame in 79% less time. Additionally, due to the linear complexity of ST-XCT, our advantage is likely to increase at higher resolutions. Our proposed framework also encodes 10× faster than the traditional codec VTM. Hence, although the ST-XCT blocks are more computationally intense than the simple concatenation operation used by DCVC*, our method is significantly less complex than VCT and VTM, indicating that our framework is practical.

Finally, we also highlight that the most complexity-consuming coding component is THEM, which accounts for around 20% of the parameters (due to to its multitude of heads and channels). Meanwhile, our ST-XCT's complexity is significantly influenced by the 3D convolutional operations, constituting approximately 30% of inference time. These are potential avenues for future exploration, and we invite the readers to delve into model compression techniques (*e.g.*, channel pruning) for such modules.

5 CONCLUSION

In this work, we investigate how to effectively leverage both spatial and temporal information to improve video compression and propose a module, Spatial-Temporal Cross-Covariance Transformer. We conduct extensive experiments to demonstrate its effectiveness by integrating it into various components of an end-to-end neural video compression framework. A thorough set of experiments and ablation studies was performed to showcase the generalization capabilities of the ST-XCT in different coding components, ultimately resulting in superior performance compared to previous state-of-the-art video compression algorithms. Overall, our work conducts a solid baseline for the transformer-based video compression method, which will facilitate the subsequent research on effective combinations of transformers and neural video codecs.

REFERENCES

[1] [n. d.]. Hevc test model (hm). https://hevc.hhi.fraunhofer.de/HM-doc/. Accessed: 2023-03-06.

[2] [n. d.]. Ultra video group test sequences. http://ultravideo.cs.tut.fi. Accessed: 2023-03-06.

[3] [n. d.]. VVC Reference Model (VTM). https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/. Accessed: 2023-03-06.

[4] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. 2020. Scale-Space Flow for End-to-End Optimized Video Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8503–8512.

[5] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. 2021. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems* 34 (2021), 20014–20027.

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

[7] Fabrice Bellard. 2015. BPG Image format. *URL https://bellard. org/bpg* (2015).

[8] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. 2021. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 10 (2021), 3736–3764.

[9] Zhenghao Chen, Shuhang Gu, Guo Lu, and Dong Xu. 2022. Exploiting intra-slice and inter-slice redundancy for learning-based lossless volumetric image compression. *IEEE Transactions on Image Processing* 31 (2022), 1697–1707.

[10] Zhenghao Chen, Guo Lu, Zhihao Hu, Shan Liu, Wei Jiang, and Dong Xu. 2022. LSVC: A learning-based stereo video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6073–6082.

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. 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*. 7939–7948.

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

[13] Amirhossein Habibian, Ties van Rozendaal, Jakub M Tomczak, and Taco S Cohen. 2019. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE International Conference on Computer Vision*. 7033–7042.

[14] Zhihao Hu, Zhenghao Chen, Dong Xu, Guo Lu, Wanli Ouyang, and Shuhang Gu. 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*. Springer, 193–209.

[15] Zhihao Hu, Guo Lu, Jinyang Guo, Shan Liu, Wei Jiang, and Dong Xu. 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*.

[16] Zhihao Hu, Guo Lu, and Dong Xu. 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*. 1502–1511.

[17] Jiahao Li, Bin Li, and Yan Lu. 2021. Deep contextual video compression. *Advances in Neural Information Processing Systems* 34 (2021), 18114–18125.

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

[19] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1833–1844.

[20] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc V Gool. 2022. Recurrent video restoration transformer with guided deformable attention. *Advances in Neural Information Processing Systems* 35 (2022), 378–393.

[21] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. 2020. M-LVC: Multiple Frames Prediction for Learned Video Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3546–3554.

[22] Lei Liu, Zhihao Hu, Zhenghao Chen, and Dong Xu. 2023. ICMH-Net: Neural Image Compression Towards both Machine Vision and Human Vision. In *Proceedings of the 31th ACM International Conference on Multimedia*. ACM. https://doi.org/10.1145/3581783.3612041

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10012–10022.

[24] Salvator Lombardo, Jun Han, Christopher Schroers, and Stephan Mandt. 2019. Deep generative video compression. In *Advances in Neural Information Processing Systems*. 9287–9298.

[25] Guo Lu, Chunlei Cai, Xiaoyun Zhang, Li Chen, Wanli Ouyang, Dong Xu, and Zhiyong Gao. 2020. Content adaptive and error propagation aware deep video compression. In *European Conference on Computer Vision*. Springer, 456–472.

[26] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. 2019. DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11006–11015.

[27] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. [n. d.]. An End-to-End Learning Framework for Video Compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* in Press ([n. d.]), 1–1. https://doi.org/10.1109/TPAMI.2020.2988453

[28] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. 2020. An end-to-end learning framework for video compression. *IEEE transactions on pattern analysis and machine intelligence* 43, 10 (2020), 3292–3308.

[29] Ming Lu, Peiyao Guo, Huiqing Shi, Chuntong Cao, and Zhan Ma. 2022. Transformer-based Image Compression. (2022), 469–469.

[30] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. 2019. Practical Full Resolution Learned Lossless Image Compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[31] Fabian Mentzer, George D Toderici, David Minnen, Sergi Caelles, Sung Jin Hwang, Mario Lucic, and Eirikur Agustsson. 2022. VCT: A Video Compression Transformer. *Advances in Neural Information Processing Systems* 35 (2022), 13091–13103.

[32] David Minnen, Johannes Ballé, and George D Toderici. 2018. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*. 10771–10780.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library.

[34] Yichen Qian, Ming Lin, Xiuyu Sun, Zhiyu Tan, and Rong Jin. 2022. Entroformer: A Transformer-based Entropy Model for Learned Image Compression. (May 2022).

[35] Anurag Ranjan and Michael J Black. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4161–4170.

[36] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. 2022. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia* (2022).

[37] Mingyang Song, Yang Zhang, and Tunc O Aydin. 2022. TempFormer: Temporally Consistent Transformer for Video Denoising. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX*. Springer, 481–496.

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

[39] David S Taubman and Michael W Marcellin. 2002. JPEG2000: Standard for interactive imaging. *Proc. IEEE* 90, 8 (2002), 1336–1357.

[40] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. 2017. Lossy image compression with compressive autoencoders. *International Conference for Learning Representations* (2017).

[41] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. 2017. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5306–5314.

[42] Gregory K Wallace. 1992. The JPEG still picture compression standard. *IEEE transactions on consumer electronics* 38, 1 (1992), xviii–xxxiv.

[43] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C-C Jay Kuo. 2016. MCL-JCV: a JND-based H. 264/AVC video quality assessment dataset. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1509–1513.

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

[45] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. 2020. Learning for video compression with hierarchical quality and recurrent enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6628–6637.

[46] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5728–5739.

[47] Yinhao Zhu, Yang Yang, and Taco Cohen. 2022. Transformer-based transform coding. In *International Conference on Learning Representations*.

Received 11 May 2023; revised 7 July 2023; accepted 25 July 2023