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*

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


KEYWORDS

Video compression, neural network, transformer

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:

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, 26] 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
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 2D features \( F_1, F_2 \in \mathbb{R}^{H \times W \times C} \) as inputs, 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 \( F_{joint} \in \mathbb{R}^{H \times W \times 3 \times C} \). 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 \( F_{joint} \in \mathbb{R}^{H \times W \times 2 \times C} \) and feed it into a 2D convolutional layer to generate a final 2D joint feature \( F_{joint} \in \mathbb{R}^{H \times W \times C} \).

Spatio-Temporal Feature Generator. STFG first normalizes the 3D-based joint feature, followed by applying 3D convolutional layers with \( 1 \times 1 \times 1 \) and then \( 3 \times 3 \times 3 \) kernels, which operate in the channel dimension to mix spatio-temporal information locally. Through this operation, we can obtain 3D-based Query (Q), Key (K), and Value (V) features, which are subsequently reshaped to \( Q \in \mathbb{R}^{C \times H \times W \times 2} \) and \( K, V \in \mathbb{R}^{C \times W \times 2 \times C} \). Then, we adopt a multi-head attention mechanism to partition these features into \( E \) heads along the feature channel dimension to obtain \( Q_i \in \mathbb{R}^{C \times E \times H \times W \times 2} \) and \( K_i, V_i \in \mathbb{R}^{C \times W \times 2 \times C \times E} \) for each head \( i \). Using partitioned features \( Q_i \) and \( K_i \), we compute their spatio-temporal-aware cross-covariance attention matrix \( A_i \in \mathbb{R}^{E \times E \times C \times C} \), 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 \( F_{pro} \in \mathbb{R}^{H \times W \times 2 \times 2 \times C} \). Finally, we add \( F_{pro} \) back to the input joint feature \( F_{joint} \) following the conventional residual transformer.
Temporal Context Mining

Multi-scale Temporal Feature Extraction

Frame Buffer

Flow Estimation

Flow Compression

Flow Decompression

Multi-scale Temporal Feature Extraction (MS-TFE)

Multi-scale Transformer-based Feature Decoder (MS-TFD)

Multi-scale Transformer-based Feature Encordo (MS-TFE)

Entropori Encoder

Arithmetic Encoder

Arithmetic Decoder

Hyperprior Entropy Model (THEM)

Figure 2: Overview of our proposed transformer-based neural video compression (NVC) framework. It takes the current frame $X_t$ and reference frame $X_{t-1}$ as inputs and produces the multi-scale temporal features $F^t_1, F^t_2, F^t_3$ 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 $\hat{Y}_t$ by using Multi-scale Transformer-based Feature Encoder (MS-TFE). Then, it performs the entropy-coding to losslessly encode or decode $\hat{Y}_t$ with the aid of Transformer-based Hyper-prior Entropy Model (THEM). Last, we adopt Multi-scale Transformer-based Feature Decoder (MS-TFD) to reconstruct $\hat{Y}_t$ 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.

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 $X = \{X_1, X_2, \ldots, X_{t-1}, X_t, \ldots\}$ 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 $M_t$, and decompressing $M_t$ 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, $F^t_1, F^t_2, F^t_3$.

3.2.2 Spatio-Temporal Feature Extraction. In existing deep contextual coding frameworks [18, 36], multi-scale temporal context features, $F^t_1, F^t_2, F^t_3$, are extracted from previous frames and spatial
Figure 3: Transformer-based key coding components, where we apply our ST-XCT modules in our NVC framework.

We subsequently fuse

\[ F \]

at each scale. For fusing

\[ F \]

It uses

\[ \hat{F} \]

Multi-scale Transformer-based Feature Decoder (MS-TFD)

\[ \hat{F} \]

Multi-scale Transformer-based Feature Encoder (MS-TFE)

\[ \hat{F} \]

Hyper-prior

\[ \hat{F} \]

Shuffle

\[ \text{Concat} \]

U-Net

\[ \text{2D-Conv} \]

Residual

\[ \text{ST-XCT} \]

Concat

\[ \text{2D-Conv} \]

Shuffle

Temporal

Prior

Encoder

Decoder

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

\[
\mathcal{L} = \lambda D(X_t, \hat{X}_t) + R(\hat{Y}_t) + R(\hat{Z}_t) + R(M_t),
\]

where \( D(\cdot) \) represents the distortion between the reconstructed and original frames. \( R(\cdot) \) represents the bitrate cost in the compression procedure. Here, \( \hat{Y}_t, \hat{Z}_t \) and \( M_t \) respectively represent the quantized latent feature, the quantized prior feature, and the quantized motion feature. We use \( \lambda \) 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,
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 $5 \times 10^{-5}$ for 20K iterations. The learning rate was then reduced to $1 \times 10^{-6}$ for the subsequent 30K iterations. In the second stage, we optimized all parameters end-to-end with a learning rate of $5 \times 10^{-6}$. After 40K steps, the learning rate was further reduced to $1 \times 10^{-6}$, 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 1st 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 (1st 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 (2nd and 3rd 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,
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.

<table>
<thead>
<tr>
<th></th>
<th>UVG</th>
<th>MCL-JCV</th>
<th>HEVC B</th>
<th>HEVC C</th>
<th>HEVC D</th>
<th>HEVC E</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTM-13.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td><strong>0.0</strong></td>
<td>0.0</td>
<td>0.0</td>
<td><strong>0.0</strong></td>
</tr>
<tr>
<td>HM-16.21</td>
<td>36.61</td>
<td>42.27</td>
<td>44.51</td>
<td>36.48</td>
<td>28.36</td>
<td>53.85</td>
<td>40.35</td>
</tr>
<tr>
<td>FVC</td>
<td>52.57</td>
<td>41.82</td>
<td>78.42</td>
<td>76.89</td>
<td>76.32</td>
<td>71.34</td>
<td></td>
</tr>
<tr>
<td>DCVC</td>
<td>38.68</td>
<td>24.49</td>
<td>50.41</td>
<td>93.00</td>
<td>55.91</td>
<td>140.15</td>
<td>67.11</td>
</tr>
<tr>
<td>DCVC*</td>
<td>-35.79</td>
<td>-38.08</td>
<td>-26.01</td>
<td>8.87</td>
<td>-14.95</td>
<td>-36.63</td>
<td>-23.77</td>
</tr>
<tr>
<td>VCT</td>
<td>10.46</td>
<td>9.93</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7.20</td>
<td>10.20</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>-36.19</strong></td>
<td><strong>-39.08</strong></td>
<td><strong>-27.35</strong></td>
<td><strong>7.20</strong></td>
<td><strong>-18.63</strong></td>
<td><strong>-39.74</strong></td>
<td><strong>-25.63</strong></td>
</tr>
</tbody>
</table>

4.3 Ablation Studies

4.3.1 Ablation I: Individually Removing the transformer-based components from our NVC framework.

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

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 architecture, VCT.

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.
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.

<table>
<thead>
<tr>
<th></th>
<th>#Params (M)</th>
<th>GPU (MiB)</th>
<th>Enc (ms/Frame)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTM</td>
<td>N/A</td>
<td>N/A</td>
<td>6104</td>
</tr>
<tr>
<td>VCT</td>
<td>121.1</td>
<td>461.9</td>
<td>268</td>
</tr>
<tr>
<td>DCVC*</td>
<td>17.5</td>
<td>66.8</td>
<td>19</td>
</tr>
<tr>
<td>Ours</td>
<td>26.8</td>
<td>102.2</td>
<td>56</td>
</tr>
</tbody>
</table>

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 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.