# TLVC: Temporal Bit-rate Allocation for Learned Video Compression


1st Yifan Zhang
*Institute of Information Science, Beijing Jiaotong University*
Beijing, China
ifzhang@bjtu.edu.cn

2nd Meiqin Liu*
*Institute of Information Science, Beijing Jiaotong University*
Beijing, China
mqliu@bjtu.edu.cn

3rd Chenming Xu
*Institute of Information Science, Beijing Jiaotong University*
Beijing, China
chenming_xu@bjtu.edu.cn

4th Qi Tang
*Institute of Information Science, Beijing Jiaotong University*
Beijing, China
qitang@bjtu.edu.cn

5th Chao Yao
*School of Computer and Communication Engineering*
*University of Science and Technology Beijing*
Beijing, China
yaochao@ustb.edu.cn

6th Yao Zhao
*Institute of Information Science, Beijing Jiaotong University*
Beijing, China
yzhao@bjtu.edu.cn



***Abstract*—Most of the existing neural video compression methods adopt the hybrid coding framework, which only focus on the motion and residual coding of adjacent frames and ignore the long-term inter-frame dependency and bit-rate allocation. To address the shortcoming, we propose a temporal bit-rate allocation strategy for learned video compression (TLVC). Specifically, Motion-driven Temporal Gate (MTG) is designed to yield temporal bit-rate coefficient by considering the impact of residual coding on subsequent motion estimation. Sequentially, Texture-conditioned Spatial Gate (TSG) is proposed to take the generated coefficient to guide the residual compression with different bit-rate. Experimental results demonstrate that TLVC can achieve effective bit-rate allocation compared with the traditional codec H.266/VVC (VTM-13.2) of low delay p-frame (LDP) configuration.**



***Index Terms*—Neural video compression, Variable bit-rate allocation, Energy function**


## I. Introduction

Video compression plays a significant role in reducing the burden of storage and transmission while maintaining high reconstruction quality [1]–[3]. A series of video coding standards have emerged to get good performance, such as H.264/AVC [4], H.265/HEVC [5], and H.266/VVC [6]. They typically adopt various hand-crafted coding technologies to remove signal redundancy. And some learning-based video compression methods [7]–[15] demonstrate the excellent performance. Learning-based methods need to explore the correlation among video frames to reduce temporal redundancy and achieve bit-rate savings. Most learning-based approaches [16]–[24] adopt a hybrid coding framework consisting primarily of motion coding [25]–[27] and residual coding. For instance, M-LVC [28] introduces multiple reference frames to fully leverage inter-frame information, achieving a more accurate motion estimation. ALVC [29] can effectively predict the target frame from previously compressed frames, further eliminating inter-frame redundancy. These methods utilize the same codec to eliminate inter-frame redundancy, but they don't account for the varying extent of inter-frame information. In video sequences, the varying extent of inter-frame is not constant [30], [31], and previous methods solely rely on the adaptive capability of neural network to resist this situation. Moreover, there isn't a suitable temporal inter-frame bit-rate allocation strategy, and the temporal redundancy still exists and hinders the rate-distortion performance.

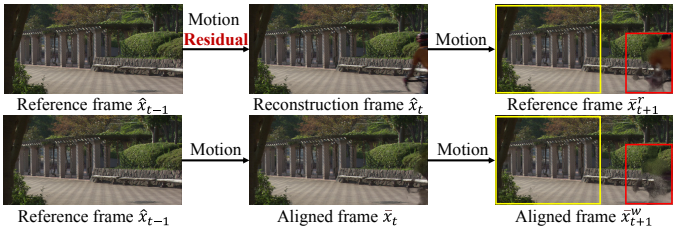



Fig. 1. The accuracy and efficiency of subsequent motion coding are directly determined by the presence of residual compensation in the reference frame.

It is challenging to directly assess the bit-rate allocated to each frame for more effective video compression. We consider the impact of motion and residual in bit-rate allocation, and assess the temporal inter-frame importance. As shown in Fig. 1, it displays two motion coding results $\bar{x}^r_{t+1}$ and $\bar{x}^w_{t+1}$. The reference frame $\hat{x}_t$ of $\bar{x}^r_{t+1}$ uses residual coding. And the reference frame $\bar{x}_t$ of $\bar{x}^w_{t+1}$ does not use residual compensation. It can be observed that, in the smoothly moving part (yellow box), excellent results are solely achieved with motion coding. However, the absence of residual compensation


* Corresponding author.


This work is supported in part by the National Key Research and Development Program of China under Grant 2022ZD0118001; and in part by the National Natural Science Foundation of China under Grant 62120106009, Grant 62372036, and Grant 62332017.

leads to the annoying alignment in the part with intense motion (red box). Therefore, residual coding performance and corresponding bit-rate allocation may influence on the subsequent motion estimation. Besides, the residual bit-rate accounts for a large proportion of video compression. Inter-frame temporal importance can be regarded as the residual temporal importance.

In this paper, we propose a temporal bit-rate allocation strategy for learned video compression (TLVC). It primarily focuses on the the rational allocation of bit-rate in both temporal and spatial domains. The challenge of bit-rate allocation in the temporal domain lies in determining the significance of individual frames. In order to deal with this issue, the temporal bit-rate coefficient is proposed to express the inter-frame dependency. Meanwhile, the Motion-driven Temporal Gate (MTG) is designed to yield the temporal bit-rate coefficient via the assessment of the residual impact on subsequent motion estimation. The generated coefficients serve as the quantitative indicator of temporal importance and are utilized to guide the residual compression. To combine the coefficients with the spatial information inherent in residual features, the Texture-conditioned Spatial Gate (TSG) is designed to estimate the spatial importance via an energy function [32] in a multi-scale manner [33]–[35]. TSG further fuses the importance predicted along spatio-temporal dimensions for variable residual compression. These modules effectively reduce the inter-frame redundancy and adaptively allocate the bit-rate. The proposed TLVC achieves savings of $11.49\%$, $3.08\%$, and $12.96\%$ compared to HM-16.20 [36], ENVC [37], and ALVC [29] on multiple datasets, respectively. In summary, our main contributions are listed as follows:

- We propose a temporal bit-rate allocation strategy for learned video compression (TLVC). It can dynamically allocate bit-rate for each frame along temporal dimension, and further reasonably allocate the bit-rate conditioned on intra-frame information.
- We design a Motion-driven Temporal Gate (MTG) to assess the inter-frame residual importance with the influence of residual compensation on subsequent motion estimation.
- We design a Texture-conditioned Spatial Gate (TSG) embedded into the hierarchical encoder-decoder. Based on the temporal importance information, it smoothly scales the residual features and enables the variable video compression.

## II. Methodology

### A. Overview

In this paper, we propose a temporal bit-rate allocation strategy for learned video compression (TLVC), as shown in Fig. 2. In order to achieve rational bit-rate allocation in spatio-temporal domains, Motion-driven Temporal Gate (MTG) and Texture-conditioned Spatial Gate (TSG) are additionally incorporated into the existing video compression network. MTG yields temporal bit-rate coefficient and TSG utilizes the generated coefficients to guide the residual compression at different bit-rate. The Motion Coding is employed to estimate and compress the motion vector between the input frame $x_t$ and the reconstructed frame $\hat{x}_{t-1}$, and aligns the $\hat{x}_{t-1}$ to the current frame as $\bar{x}_t$. $t$ is the frame index. Subsequently, MTG utilizes the aligned frame $\bar{x}_t$ and two consecutive frames $x_t$ and $x_{t+1}$ to update the temporal bit-rate coefficient $\lambda_{t-1}$ to get $\lambda_t$. Finally, the Residual Coding compresses the residual between the input frame $x_t$ and the aligned frame $\bar{x}_t$ for compensation, generating the reconstructed frame $\hat{x}_t$. TSG is embedded into the auto-encoder of Residual Coding, enabling it to compress the residual under the guidance of $\lambda_t$ with various bit-rate.

### B. Motion-driven Temporal Gate

Residual coding accounts for a significant portion in video coding, but previous methods [38]–[43] rarely utilize the inter-frame residual correlation to improve coding performance. Therefore, we utilize the influence of residuals on subsequent motion to design a Motion-driven Temporal Gate (MTG) for generating temporal bit-rate coefficients. As shown in Fig. 2 (c), MTG consists of the parameter-shared motion estimation and compensation along with a temporal gate [44], [45]. In order to evaluate the impact of the current residual compensation on the next motion, two consecutive frames $x_t$ and $x_{t+1}$ together with the motion aligned frame $\bar{x}_t$ are input into MTG for generation of aligned frames $\bar{x}^w_{t+1}$ and $\bar{x}^r_{t+1}$, as follows:

$$\bar{x}^w_{t+1} = MC(\bar{x}_t, (ME(x_{t+1}, \bar{x}_t))) \tag{1}$$

$$\bar{x}^r_{t+1} = MC(x_t, (ME(x_{t+1}, x_t))) \tag{2}$$

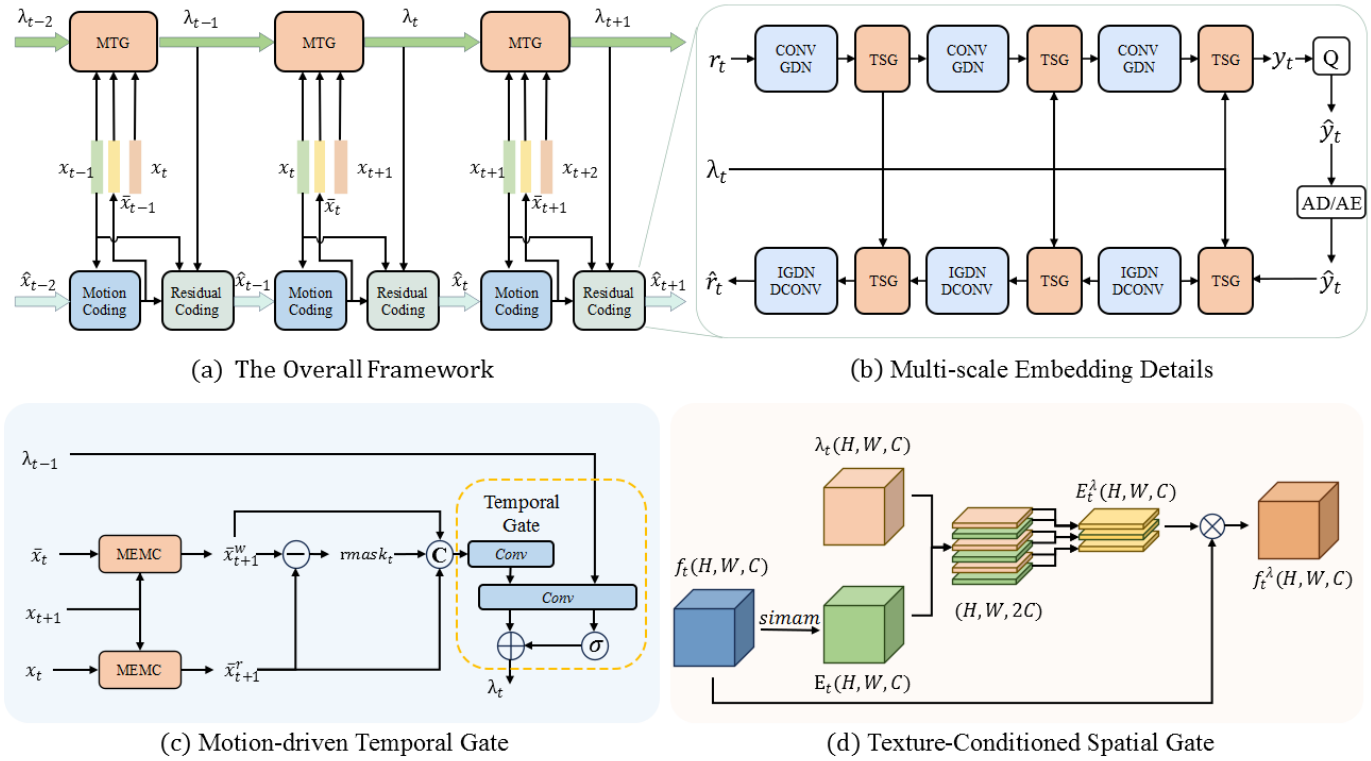
where $ME$ and $MC$ represent the operation of motion estimation and compensation. The difference between $\bar{x}_t$ and $x_t$ lies in whether there exists residual compensation. Since the reference frames of $\bar{x}^w_{t+1}$ and $\bar{x}^r_{t+1}$ are $\bar{x}_t$ and $x_t$, respectively, thus the difference between $\bar{x}^w_{t+1}$ and $\bar{x}^r_{t+1}$ can indicate the impact of residual on motion estimation from $x_t$ to $x_{t+1}$. To determine residual bit-rate using the subsequent motions, a gating mechanism is introduced to update the previous temporal bit-rate coefficient $\lambda_{t-1}$ to coefficient $\lambda_t$. The difference between $\bar{x}^w_{t+1}$ and $\bar{x}^r_{t+1}$ is extracted by convolution, and it is transformed into a feature with the same size as $\lambda_t$:

$$\theta_t = Conv(\left[\bar{x}^w_{t+1}, \bar{x}^r_{t+1}, rmask_t\right]) \tag{3}$$

where $rmask_t$ is obtained by mathematical subtraction of $\bar{x}^w_{t+1}$ and $\bar{x}^r_{t+1}$. $\theta_t$ represents the importance of the current residual to subsequent motion. '$[\cdot,\cdot]$' denotes the channel-wise concatenation operation. $\theta_t$ and $\lambda_{t-1}$ are used to determine the residual bit-rate.

$$g_t, z_t = Split(Conv([\lambda_{t-1}, \theta_t])) \tag{4}$$

$$\lambda_t = Sigmoid(g_t) \odot z_t \tag{5}$$




(a) The Overall Framework (b) Multi-scale Embedding Details

(c) Motion-driven Temporal Gate (d) Texture-Conditioned Spatial Gate

Fig. 2. (a) Overall framework of TLVC. (b) Structural description of the introduced Texture-conditioned Spatial Gate in the residual encoder-decoder. (c) Illustration of Motion-driven Temporal Gate. (d) Illustration of Texture-conditioned Spatial Gate.

where $Split$ is the operation of split along the channel dimension. $z_t$ and $g_t$ respectively represent hidden features and the attention gate. '$\odot$' denotes element-wise multiplication.

### C. Texture-conditioned Spatial Gate

In order to compress the smooth region and fine structure at various bit-rate in the spatial domain, Texture-conditioned Spatial Gate (TSG) is embedded into the residual auto-encoder in a multi-scale manner. It utilizes the temporal bit-rate coefficients and the intrinsic spatial information of the features to determine the intra-frame bit-rate allocation. Since the features $f_t$ possess varying spatial scales, an expansion in the channel dimension of the temporal bit-rate coefficient $\lambda_t$ is performed to align with the current feature size. As shown in Fig. 2 (d), TSG first calculates the energy $e_t^{i,j}$ of each element in the input feature using the energy function [32]. Specifically, it estimates the attention weight of each element by calculating the mean $\hat{\mu}_t$ and variance $\hat{\sigma}_t$ within each channel:

$$e_t^{i,j} = \frac{4(\hat{\sigma}_t^2 + \beta)}{(p_t^{i,j} - \hat{\mu}_t)^2 + 2\hat{\sigma}_t^2 + 2\hat{\mu}_t} \tag{6}$$

where $p_t^{i,j}$ represents the value of the element at position $(i, j)$, $i \in \{0, 1, ..., H-1\}$, $j \in \{0, 1, ..., W-1\}$. H and W are the height and width of $f_t$ respectively. $\beta$ is a hyper-parameter. A lower energy value $e_t^{i,j}$ indicates greater dissimilarity between the current neuron and its surrounding neurons, signifying higher importance. Therefore, the importance of the current neuron is calculated as follows:

$$E_t = Sigmoid(\{\frac{1}{e_t^{i,j}} | 0 \le i \le H-1, 0 \le j \le W-1\}) \tag{7}$$

where $E_t$ groups all $e^{i,j}$ across channel and spatial dimensions, $Sigmoid$ is introduced to restrict large value of $E_t$.

Then, the extended $\lambda_t$ and the spatial importance $E_t$ are cross-arranged along the channels and fed into the group convolution [46] for scaling of spatial importance. Cross-arrangement aims to preserve the original spatial importance distribution within the channel, which is unchanged to the greatest extent. The above process realizes distilling the spatial importance $E_t$ of the current feature with $\lambda_t$ to get $E_t^\lambda$:

$$E_t^\lambda = C_g(\{(E_t^k, \lambda_t^k) | 0 \le k \le C\}) \tag{8}$$

where $C_g$ represents group convolution and $C$ is the number of channel $f_t$. Feature $f_t^\lambda$ is modified by multiplying $E_t^\lambda$ and the input features $f_t$ as:

$$f_t^\lambda = E_t^\lambda \odot f_t \tag{9}$$

Through the above process, MTG can smoothly scale the features of residuals according to the temporal bit-rate coefficient generated by TSG.

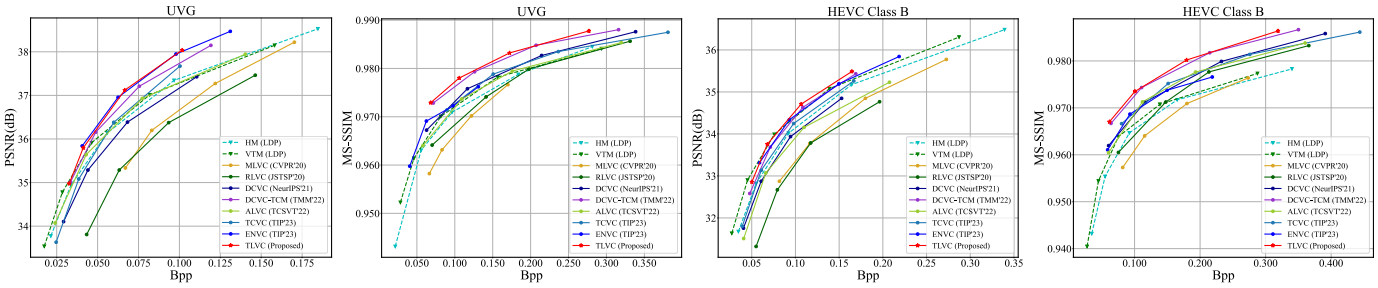



Fig. 3. PSNR (dB) and MS-SSIM of TLVC on the UVG and the HEVC Class B datasets. The lines with circles represent the learned codecs and triangles represent the traditional codecs.

TABLE I
BD-RATE (%) COMPARISON RESULTS OF PSNR AND MS-SSIM. THE ANCHOR IS HM-16.20. THE BEST RESULTS ARE IN **BOLD**.

| | **HM** (LDP) | **VTM** (LDP) | **DCVC** (NeurIPS'21) | **DCVC-TCM** (TMM'21) | **ALVC** (ALVC'22) | **ENVC** (TIP'23) | **TCVC** (TIP'23) | **TLVC** (Proposed) |
|---|---|---|---|---|---|---|---|---|
| UVG | 0/0 | -5.37/7.11 | 12.56/-16.59 | -13.89/-34.48 | -2.22/-6.54 | **-22.83**/-25.64 | 2.21/-14.69 | -18.84/**-36.81** |
| HEVC Class B | 0/0 | -15.65/-12.50 | 8.40/-27.27 | -10.19/-46.58 | -2.10/-27.27 | -11.74/-24.06 | -2.10/-14.38 | **-16.29/-48.62** |
| HEVC Class C | 0/0 | **-12.63**/-11.14 | 25.15/-29.43 | 1.12/-45.94 | 12.40/-25.17 | 3.42/-25.18 | 7.63/-31.56 | -4.37/**-48.63** |
| HEVC Class D | 0/0 | -10.50/-3.38 | 12.19/-39.91 | -8.72/**-53.97** | 1.92/-43.33 | -4.84/-31.04 | -4.69/-40.55 | **-14.46**/-59.00 |
| HEVC Class E | 0/0 | **-18.30**/-0.14 | 32.40/-15.50 | 3.18/-39.27 | -9.54/**-49.60** | -6.07/-2.24 | 4.28/-3.48 | -12.06/-45.21 |
| Average | 0/0 | **-12.55**/-6.85 | 18.14/-25.74 | -5.70/-44.05 | 0.09/-30.38 | -8.41/-21.63 | 1.47/-22.64 | -11.49/**-47.06** |

### *D. Objective Function*

The objective function consists of three losses, the commonly used rate-distortion (RD) loss ($L_{rd}$) and proposed two losses, average loss ($L_a$) and bias loss ($L_b$). RD loss $L_{rd}$ is formulated as:

$$L_{rd} = \frac{1}{T}\sum_{t}(\lambda_t d(x_t, \hat{x}_t) + R(\hat{mv}_t) + R(\hat{y}_t)) \tag{10}$$

where $d(x_t, \hat{x}_t)$ is the distortion between the input video frame $x_t$ and the reconstructed frame $\hat{x}_t$. $d(\cdot)$ refers to mean square error (MSE) when targeted at PSNR or 1-MS-SSIM [2] when targeted at MS-SSIM. $R(\cdot)$ is utilized to calculate bit-rate for representations compression.

Average loss $L_a$ is proposed to adapt the overall bit-rate to the user-entered $\lambda_{GOP}$, as:

$$L_a = (\frac{\sum_T \hat{\lambda}_t}{T} - \lambda_{GOP})^2 \tag{11}$$

And, bias loss $L_b$ is proposed to constrain the generation of temporal bit-rate coefficient based on the inter-frame dependency and motion complexity instead of fixed bit-rate allocation, as:

$$L_b = \sum_{T}(\hat{\lambda}_t - u_t\lambda_{GOP})^2 \tag{12}$$

where $u_t$ increases with the rise of $t$.

Finally, the loss function of TLVC is formulated:

$$L = L_{rd} + w_1 L_a + w_2 L_b \tag{13}$$

where $w_1$ and $w_2$ represent the weights of $L_a$ and $L_b$, respectively.

## III. EXPERIMENTS

### *A. Experimental Settings*

Due to the proposed method being able to adapt to different bit-rate variations, we employ the commonly used Vimeo-90K septuplet dataset [2] to train only 2 models for MSE and MS-SSIM, respectively. The AdamW [47] optimizer is used with a batch size of 4. HEVC sequences [5] and UVG [48] are used to evaluate the compression performance. For a fair comparison, all experiments are implemented on 2 NVIDIA GeForce RTX 3090 GPUs with Intel(R) Xeon(R) Gold 6226R CPUs.

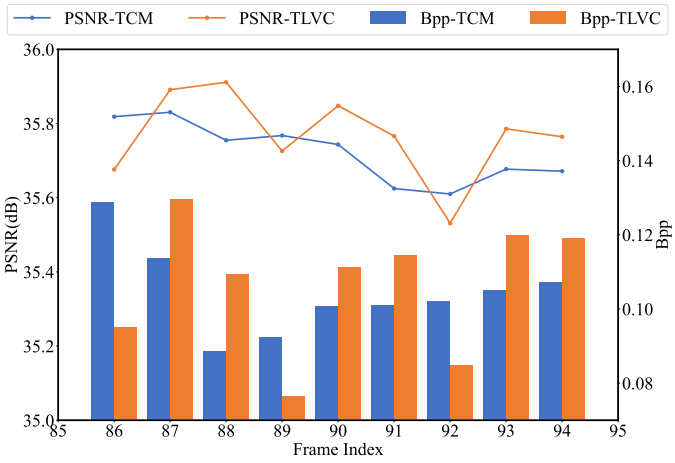



Fig. 4. Coding performance of TLVC and DCVC-TCM for the HEVC Class B BasketballDrive.

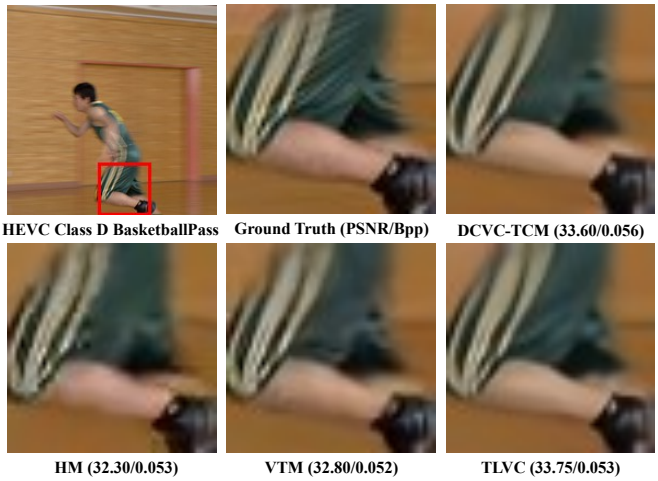



Fig. 5. Qualitative comparison on HEVC Class D BasketballPass dataset.

### B. Quantitative and Qualitative Evaluation

We conduct an RD curve comparison with traditional codecs, such as HM-16.20 [36] and VTM-13.2 [49], in LDP configuration and other state-of-the-art learned codecs [17], [28], [29], [37], [50]–[52], shown as Fig. 3. TLVC exhibits significant superiority in terms of PSNR and MS-SSIM RD curves on the UVG and HEVC Class B datasets compared with other algorithms. Additionally, we compare the BD-rate (%) of VCEG-M33 [53] for PSNR and MS-SSIM models in Tab. I, the anchor is HM-16.20. TLVC achieves an average 5.7% bit-rate reduction on the evaluation datasets compared to DCVC-TCM [50]. Specifically, it is noted that TLVC gets a 16.29% bit-rate reduction compared with HM-16.20 [36] on the HEVC Class B dataset, verifying the generalization on high-resolution videos.

To validate the temporal bit-rate allocation effectiveness of TLVC, we visualize the consecutive inter-frame coding performance of TLVC and DCVC-TCM on the HEVC Class B BasketballDrive sequence (86th to 94th frames), as shown in Fig. 4. It indicates that DCVC-TCM uses the same bit-rate for all frames, while TLVC can reasonably allocate temporal bit-rate based on the inter-frame features, achieving better compression performance.

The qualitative comparison results of HEVC Class D dataset are given in Fig. 5. HM-16.20 [36] and VTM-13.2 [49], these traditional methods lose some texture details on the pants. TLVC achieves visually pleasing texture reconstruction quality compared to DCVC-TCM [50] with a similar bit-rate.

### C. Ablation Study

To verify the effectiveness of TSG and MTG in TLVC, we separately ablate these two modules in Table II. Model B utilizes TSG with uniform temporal information without MTG. Compared to Model A, Model B successfully saves 3.03% bit-rate and indicates the effective guidance of TSG on bit-rate allocation using spatial importance. Furthermore, Model C saves 5.91% of the bit-rate and achieves further improvement. It suggests that the temporal information inferred by MTG allows for a more rational inter-frame bit-rate allocation.

TABLE II
ABLATION STUDY RESULTS.

| Models | TSG | MTG | BD-Rate | Params |
|---|---|---|---|---|
| A | | | 0.0% | 10.71M |
| B | ✓ | | -3.03% | 13.23M |
| C | ✓ | ✓ | -5.91% | 14.80M |

## IV. Conclusion

In this paper, we propose a temporal bit-rate allocation strategy for learned video compression, called TLVC. The Motion-driven Temporal Gate (MTG) and Texture-conditioned Spatial Gate (TSG) are incorporated into learning-based video compression framework. MTG estimates the temporal bit-rate coefficient based on the influence of residual compensation on subsequent motion estimation and can dynamically allocate bit-rate for residual. Concurrently, TSG combines the inherent priors of features in the spatial dimension and the coefficients generated by MTG to better guide the residual coding. Experimental results demonstrate that TLVC maximizes the utilization of both temporal and spatial information within the video sequence and greatly saves the storage space for effective compression.

## REFERENCES


[1] Jian Jin, Xingxing Zhang, Xin Fu, Huan Zhang, Weisi Lin, Jian Lou, and Yao Zhao, "Just noticeable difference for deep machine vision," *TCSVT*, 2021.

[2] Zhou Wang, Eero P Simoncelli, and Alan C Bovik, "Multiscale structural similarity for image quality assessment," in *ACSSC*, 2003.

[3] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.

[4] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra, "Overview of the H.264/AVC video coding standard," *TCSVT*, vol. 13, no. 7, pp. 560–576, 2003.

[5] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *TCSVT*, vol. 22, no. 12, pp. 1649–1668, 2012.

[6] 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," *TCSVT*, vol. 31, no. 10, pp. 3736–3764, 2021.

[7] Kai Lin, Chuanmin Jia, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao, "DMVC: Decomposed motion modeling for learned video compression," *TCSVT*, vol. 33, no. 7, pp. 3502–3515, 2022.

[8] Jiahao Li, Bin Li, and Yan Lu, "Neural video compression with diverse contexts," in *CVPR*, 2023.

[9] Linfeng Qi, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu, "Motion information propagation for neural video compression," in *CVPR*, 2023.

[10] Mu-Jung Chen, Yi-Hsin Chen, and Wen-Hsiao Peng, "B-CANF: Adaptive b-frame coding with conditional augmented normalizing flows," *TCSVT*, pp. 1–14, 2023.

[11] Jiahao Li, Bin Li, and Yan Lu, "Neural video compression with feature modulation," in *CVPR*, 2024.

[12] Seunghwa Jeong, Bumki Kim, Seunghoon Cha, Kwanggyoon Seo, Hayoung Chang, Jungjin Lee, Younghui Kim, and Junyong Noh, "Real-time CNN training and compression for neural-enhanced adaptive live streaming," *TPAMI*, pp. 1–16, 2024.

[13] Yuan Tian, Guo Lu, Yichao Yan, Guangtao Zhai, Li Chen, and Zhiyong Gao, "A coding framework and benchmark towards low-bitrate video understanding," *TPAMI*, pp. 1–19, 2024.

[14] Bowen Liu, Yu Chen, Rakesh Chowdary Machineni, Shiyu Liu, and Hun-Seok Kim, "MMVC: Learned multi-mode video compression with block-based prediction mode selection and density-adaptive entropy coding," in *CVPR*, 2023.

[15] Chenming Xu, Meiqin Liu, Chao Yao, Weisi Lin, and Yao Zhao, "IBVC: Interpolation-driven b-frame video compression," *PR*, p. 110465, 2024.

[16] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao, "DVC: An end-to-end deep video compression framework," in *CVPR*, 2019.

[17] Jiahao Li, Bin Li, and Yan Lu, "Deep contextual video compression," in *NeurIPS*, 2021.

[18] Zhihao Hu, Guo Lu, and Dong Xu, "FVC: A new framework towards deep video compression in feature space," in *CVPR*, 2021.

[19] Oren Rippel, Alexander G Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir Bourdev, "ELF-VC: Efficient learned flexible-rate video coding," in *ICCV*, 2021.

[20] Wei Jiang, Junru Li, Kai Zhang, and Li Zhang, "LVC-LGMC: Joint local and global motion compensation for learned video compression," in *ICASSP*, 2024.

[21] 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 *CVPR*, 2022.

[22] Guo Lu, Chunlei Cai, Xiaoyun Zhang, Li Chen, Wanli Ouyang, Dong Xu, and Zhiyong Gao, "Content adaptive and error propagation aware deep video compression," in *ECCV*, 2020.

[23] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici, "Scale-space flow for end-to-end optimized video compression," in *CVPR*, 2020.

[24] Shuhong Liao, Chuanmin Jia, Hongfei Fan, Jingwen Yan, and Siwei Ma, "Rate-quality based rate control model for neural video compression," in *ICASSP*, 2024.

[25] Anurag Ranjan and Michael J Black, "Optical flow estimation using a spatial pyramid network," in *CVPR*, 2017.

[26] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, "Deformable convolutional networks," in *ICCV*, 2017.

[27] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai, "Deformable ConvNets v2: More deformable, better results," in *CVPR*, 2019.

[28] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu, "M-LVC: Multiple frames prediction for learned video compression," in *CVPR*, 2020.

[29] Ren Yang, Radu Timofte, and Luc Van Gool, "Advancing learned video compression with in-loop frame prediction," *TCSVT*, vol. 33, no. 5, pp. 2410–2423, 2023.

[30] Meiqin Liu, Chenming Xu, Chao Yao, Chunyu Lin, and Yao Zhao, "JNMR: Joint non-Linear motion regression for video frame interpolation," *TIP*, vol. 32, pp. 5283–5295, 2023.

[31] Qi Tang, Yao Zhao, Meiqin Liu, Jian Jin, and Chao Yao, "Semantic lens: Instance-centric semantic alignment for video super-resolution," in *AAAI*, 2024.

[32] Lingxiao Yang, Ru-Yuan Zhang, Lida Li, and Xiaohua Xie, "SimAM: A simple, parameter-free attention module for convolutional neural networks," in *ICML*, 2021.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[34] Jiaming Liang, Meiqin Liu, Chao Yao, Chunyu Lin, and Yao Zhao, "SIGVIC: Spatial importance guided variable-rate image compression," in *ICASSP*, 2023.

[35] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li, "Uformer: A general U-shaped transformer for image restoration," in *CVPR*, 2022.

[36] "HM-16.20," https://vcgit.hhi.fraunhofer.de/jvet/HM/, Accessed: 2022-07-05.

[37] Zongyu Guo, Runsen Feng, Zhizheng Zhang, Xin Jin, and Zhibo Chen, "Learning cross-scale weighted prediction for efficient neural video compression," *TIP*, vol. 32, pp. 3567–3579, 2023.

[38] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte, "Learning for video compression with hierarchical quality and recurrent enhancement," in *CVPR*, 2020.

[39] Haifeng Guo, Sam Kwong, Dongjie Ye, and Shiqi Wang, "Enhanced context mining and filtering for learned video compression," *TIP*, pp. 1–13, 2023.

[40] Haojie Liu, Ming Lu, Zhiqi Chen, Xun Cao, Zhan Ma, and Yao Wang, "End-to-end neural video coding using a compound spatiotemporal representation," *TCSVT*, vol. 32, no. 8, pp. 5650–5662, 2022.

[41] Haojie Liu, Ming Lu, Zhan Ma, Fan Wang, Zhihuang Xie, Xun Cao, and Yao Wang, "Neural video coding using multiscale motion compensation and spatiotemporal context model," *TCSVT*, vol. 31, no. 8, pp. 3182–3196, 2020.

[42] Jiahao Li, Bin Li, and Yan Lu, "Hybrid spatial-temporal entropy modelling for neural video compression," in *ACM MM*, 2022.

[43] Zhihao Hu and Dong Xu, "Complexity-guided slimmable decoder for efficient deep video compression," in *CVPR*, 2023.

[44] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li, "Simvp: Simpler yet better video prediction," in *CVPR*, 2022.

[45] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, S Yu Philip, and Mingsheng Long, "Predrnn: A recurrent neural network for spatiotemporal predictive learning," *TPAMI*, vol. 45, no. 2, pp. 2208–2225, 2022.

[46] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[47] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *ICLR*, 2018.

[48] Alexandre Mercat, Marko Viitanen, and Jarno Vanne, "UVG dataset: 50/120fps 4K sequences for video codec analysis and development," in *ACM MM*, 2020.

[49] "VTM-13.2," https://vcgit.hhi.fraunhofer.de/jvet/VVCS-oftware_VTM/, Accessed: 2022-03-02.

[50] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu, "Temporal context mining for learned video compression," *TMM*, vol. 25, pp. 7311–7322, 2023.

[51] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte, "Learning for video compression with recurrent auto-encoder and recurrent probability model," *JSTSP*, vol. 15, no. 2, pp. 388–401, 2021.

[52] Dengchao Jin, Jianjun Lei, Bo Peng, Zhaoqing Pan, Li Li, and Nam Ling, "Learned video compression with efficient temporal context learning," *TIP*, vol. 32, pp. 3188–3198, 2023.

[53] Gisle Bjontegaard, "Calculation of average PSNR differences between RD-curves.," VCEG-M33, 2001.