A Video Coding Solution with Neural-network Enhancement for CLIC 2025

Keji Chen, Licheng Ma, Haixin Wang, Jing Chen, Gang Li, Xiuxin Dou, Linyan Jiang,
Xianguo Zhang and Yaqing Li
Shannon Lab.
Tencent Holdings Limited
Shenzhen, China
{kejichen, ritcheyma, haixinwang, poirotchen, derrickgli, damondou, lynjiang, codeczhang, fredli}@tencent.com

***Abstract*—In this work, a video coding solution with Neural-Network (NN) enhancement is proposed for the Challenge on Learned Image Compression (CLIC) 2025. The proposed solution is made up of ESRGAN-based video enhancement pre-processing and improved Enhanced Compression Model (ECM) codec, and the team name of the solution is TCM. The improvements of ECM include advanced Rate Control (RC) methods, frame parallel processing, and speed optimization of NN based in-loop Filter (NNLF). According to the pre-evaluation results, the proposed methods achieve more than 0.53 Mean Opinion Score (MOS) improvement compared with VTM baseline.**

I. Introduction

In recent years, various video applications have developed vigorously, and how to improve the quality of compressed videos is a topic under continuous exploration. Reducing video compression loss is a general approach. Over the past few decades, various video compression tools have been proposed, and these tools form video coding standards, such as HEVC/H.265 [1], VVC/H.266 [2], and AVS3 [3]. More advanced methods have also been explored based on VVC, eventually leading to the development of Enhanced Compression Model (ECM) [4]. While these advanced methods improve the compression efficiency, they introduce significant coding complexity. Experiments [5] show that compared with the VVC reference software VTM11.0, the encoding time of ECM17.0 increases about 9.9 times with random access configuration. And the decoding complexity increases about 11.4 times. To accelerate the coding speed, Group of Pictures (GOP) based parallel processing methods [6] are introduced. However, these methods require extra intra frames, which decrease the compression efficiency.

Beyond traditional video coding methods, Joint Video Exploration Team (JVET) has also explored Neural Network (NN) based video coding methods. Among them, NN based in-loop Filte (NNLF) were proposed to achieve better deblocking filtering. Three series of models are proposed, with significant difference in complexity, which are High Operating Point (HOP), Low Complexity Operation Point (LOP) and Very Low Complexity Operation Point (VLOP) [7]. Among these models, LOP is the medium both in complexity and coding efficiency, and achieves YUV 8.2%, 14.9%, 13.5% Bjøntegaard Delta Rate (BD-Rate) improvement, with 0.1 times increment in encoding complexity and 27 times increment in decoding complexity [8].

Besides compression efficiency, NNs have also been leveraged for image and video enhancement tasks [9], [10]. Notably, X. Wang et al. [11] proposed the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN). This network is designed to mitigate artifacts and restore fine-grained textures. And it won the first place in the PIRM2018-SR Challenge.

The Challenge on Learned Image Compression 2025 (CLIC 2025) competition [12] provides a platform for the fair comparison of various method combinations. In this competition, participants are required to compress a set of source videos to a specified target bitrate within a given time, while maximizing the Mean Opinion Score (MOS) of decoded images based on Absolute Category Rating (ACR). Most of the source videos are compressed videos with high bitrate. Furthermore, decoding time has been introduced as an important metric. Based on these requirements, we propose a high quality video coding solution, which will be described in detail in the following sections.

II. Proposed Methods

The proposed video coding solution consists of two main components. As shown in Fig. 1, at the encoding end, ESRGAN-based enhancement pre-processing is firstly applied to the source video, and then the improved ECM encoder is applied to generate the bitstream. At the decoding end, the bitstream is decoded, and the final reconstructed video is generated directly. It should be noted that the pre-processing is normally accompanied by loss of objective metrics, such as Peak Signal-to-Noise Ratio (PSNR), but the enhanced details benefit in MOS according to our subjective evaluations.

To further improve video quality, encoder is configured with only one intra frame. Then an advanced Rate Control (RC) algorithm is introduced. Moreover, the built-in NNLF tool of ECM is enabled, which introduces significant complexity.

To deal with the complexity caused by above strategies, Frame Parallel Processing (FPP) method is introduced to meet the encoding time requirements. Additionally, algorithm to reduce NNLF computation is introduced in encoder. Furthermore, implementation optimizations are performed for the

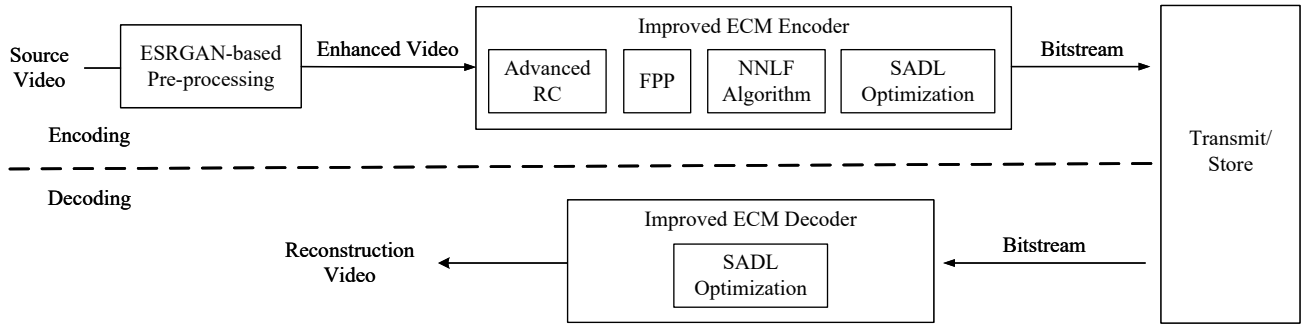


Fig. 1. The proposed video coding solution.

Small Ad-hoc Deep-Learning Library (SADL) [13], which is utilized both in the encoder and the decoder. The details of these methods are presented as follows.

A. ESRGAN-based Enhancement Pre-processing

To perform video enhancement, modifications are made to the ESRGAN. In our configuration, the original resolution is retained for video encoding. Therefore, the upsampling layer and subsequent layers in ESRGAN are removed.

The modified ESRGAN model is retrained based on a self-constructed dataset. This dataset is built by integrating public datasets such as Inter4K [14] and LDV3 [15] and generating extra distorted videos to adjust the distortion distribution. For realistic degradation simulation, mixed distortions are modeled to align the training data with real-world scenarios. Such as color-related distortions are incorporated, specifically including random saturation shifts and contrast adjustments. And randomized degradations are integrated, such as Poisson-Gaussian noise, motion blur, and H.265/H.264 compression. Notably, degradation parameters are dynamically sampled for each batch to enhance the model's robustness.

The modified ESRGAN is an RGB-based network, so color format conversion between RGB and YUV is introduced in the pre-processing stage. For the training process, to balance pixel accuracy and semantic consistency, we use a hybrid loss function combining L1 loss and Perceptual Loss. The model is trained over 600,000 iterations with a batch size of 32 and a patch size of 512x512. The initial learning rate is set to 0.0001 and halved every 10,000 iterations, using the Adam optimizer with $\beta 1 = 0.9$ and $\beta 2 = 0.99$.

Since most source videos are compressed videos in CLIC 2025, compression artifacts are thoroughly accounted for in the construction of our self-constructed dataset. This comprehensive approach enables a more realistic and robust simulation of distorted data in practical applications.

B. Advanced Rate Control Algorithm

The proposed advanced RC algorithm is based on the algorithms of x265 [16]: the Constant Rate Factor (CRF) method and block-level QP calculation method CUTree. Its frame complexity estimation relies on block costs from downsampled source frames. Specifically, in x265's lookahead stage, the downsampled current frame first undergoes simple inter and intra prediction to obtain the optimal Sum of Absolute Transformed Differences (SATD), which acts as the core block cost for complexity calculation. On this basis, frame-level complexity $complexity_i$ is derived by weighted averaging SATD values from the first frame to the current frame. Then QPs for different frames are computed with the derivation formulas below:

$$q_{\text{scale}} = \frac{complexity_i^{1-qcomp}}{factor}, \tag{1}$$

$$qp_{\text{temp}} = 12 + 6\log_2 \frac{q_{\text{scale}}}{0.85}, \tag{2}$$

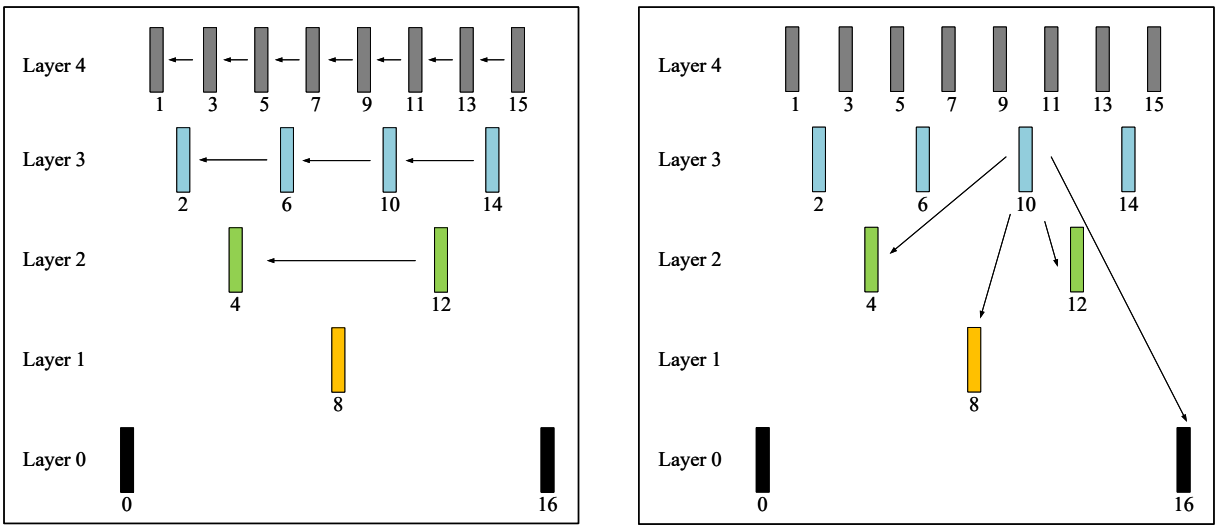
$$qp = \begin{cases} q_{\text{temp}} & \text{frame type is P or GPB} \\ q_{\text{temp}} - offset_I & \text{frame type is I} \\ q_{\text{temp}} + offset_B & \text{frame type is B} \end{cases}, \tag{3}$$

where $complexity_i$ denotes the complexity of the i-th frame (calculated via the weighted average of SATD as above). $factor$ presents the rate factor, $qcomp$ is a model parameter, $offset_I$ and $offset_B$ are parameters to control QPs of different frame types, and qp represents the final QP value.

Some improvements are proposed on the basis of CRF. In complex scenarios, the CRF scheme tends to assign large QPs for some frames, which leads to significant loss of details. To solve this problem, scene changes are considered as well as the local complexity [17]. And (1) is improved with the following formula:

$$q_{\text{scale}} = \text{clip}(\text{low}, \text{top}, \frac{inter\_cplx_i}{base\_cplx})^{1-qcomp}, \tag{4}$$

where low and top are parameters derived from the motion level, noise level, and flatness of a period of frames, $base\_cplx$ is a resolution-related parameter, and $inter\_cplx_i$ is the inter complexity of the i-th frame.



(a) Previous method

(b) Proposed method

Fig. 2. The algorithm of CABAC state initialization.

C. Frame Parallel Processing

When there is only one intra frame, each sequence contains a single GOP, and the GOP based parallel methods are not feasible. Thus, the parallel encoding within a GOP is studied, and the FPP method is introduced. During the encoding process, each frame is encoded using a distinct Frame Thread (FT). The main thread assigns frame encoding tasks to FTs and collects the compressed bitstream. One FT can start encoding only when all of its reference frames have finished in-loop filtering. FTs with no dependencies can be processed at the same time.

To reduce dependencies caused by implementation, the code flow of ECM is reorganized in detail. The non-constant global and static data structures are replaced with the local data structures. The encoding process is adjusted to cache multiple source frames during encoding. And the buffers reused by different frames are separated into multiple copies.

In addition, dependencies caused by algorithms are studied. Some algorithms utilize information or accumulated data of previously coded frames, but these previous frames are not always in the reference list. For example, as Fig. 2(a) shows, the Context-based Adaptive Binary Arithmetic Coding (CABAC) state of each frame is initialized with the final state of the previous frame with the same layer. In the proposed method, as shown in Fig. 2(b), the CABAC state is initialized with the final state of one reference frame. And which reference frame to use is based on the slice QP and distance of candidate reference frames.

D. Speed Optimization of NNLF

On the basis of the encoding configuration in previous sections, LOP model is utilized because of its balance in coding efficiency and complexity. Then the decoding time distribution is analyzed using the Visual Studio Performance Profiler [18]. As shown in Fig. 3, NNLF inference, which is implemented by the SADL library, accounts for up to 93.14% of the total decoding time. Thus, it is necessary to reduce the NNLF inference time.

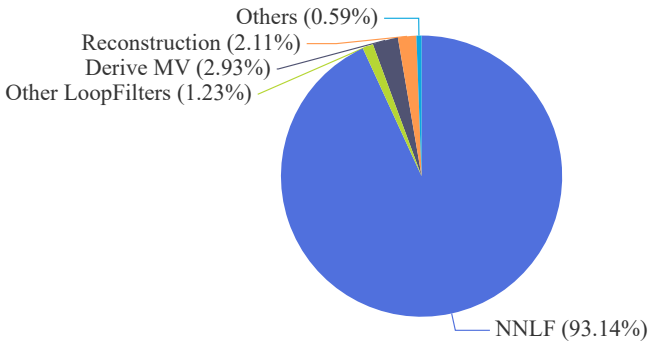


Fig. 3. The decoding time analysis of ECM decoder.

The NNLF inference is first optimized from the implementation perspective. The time consuming functions in SADL library are analyzed. The address calculation of the tensors is simplified via pointer movement. And additional AVX2 acceleration is implemented for some interpolation operations.

Encoding algorithms are also proposed to adaptively skip NNLF processing for some blocks. Whether to skip NNLF processing for a specific block is jointly determined by the texture content and the encoding QP.

III. Experimental Results

In this section, detailed experimental results are presented to show the performance of the proposed methods. All experiments are conducted using ECM17.0 as the baseline. Except for the ESRGAN-based enhancement which uses the GPU,

all the encoding and decoding methods are CPU-based. The encoding is applied on a 14-core Docker container with an Intel Platinum 8255C CPU, and the decoding is applied on a Docker container with an AMD EPYC 7K62 CPU using a single thread.

A. Performance of Intra Frame Configuration

Firstly, the intra frame configuration is evaluated. The configuration is based on Common Test Condition (CTC) [19] and extra intra frames are removed by setting IntraPeriod to a large value. As shown in Table I, 6.35% PSNR-YUV811 gain is achieved by this configuration.

TABLE I
PERFORMANCE OF INTRA FRAME CONFIGURATION

Class	Random Access			
	Y (%)	U (%)	V (%)	YUV811 (%)
Class C	-7.05	2.01	2.18	-5.22
Class D	-9.31	0.55	-0.95	-7.49
Average	-8.18	1.28	0.62	-6.35

B. Advanced RC Methods

The proposed RC method is compared with the default RC algorithm in ECM with improved configuration. The result is shown in Table II. The proposed RC method achieves 10.74% gain in PSNR-YUV811.

TABLE II
PERFORMANCE OF ADVANCED RC METHODS

Class	Random Access				
	Y (%)	U (%)	V (%)	YUV811 (%)	EncT (%)
Class C	-8.64	-28.16	-28.08	-12.54	91.0
Class D	-3.65	-30.17	-30.11	-8.95	101.6
Average	-6.15	-29.17	-29.10	-10.74	96.3

C. FPP Performance

The FPP method is compared with the default ECM configuration in CTC. For the FPP method, the count of FTs is set to 33, and the IntraPeriod setting and RC algorithm is the same to ECM. As shown in Table III, the proposed FPP method can reduce the encoding time to 24.5%, with about 0.97% PSNR-YUV811 loss.

TABLE III
FPP PERFORMANCE

Class	Random Access				
	Y (%)	U (%)	V (%)	YUV811 (%)	EncT (%)
Class C	0.76	0.87	0.75	0.77	24.9
Class D	1.17	1.34	0.96	1.17	24.1
Average	0.97	1.11	0.86	0.97	24.5

D. NNLF Decoding Optimization

The performance of NNLF decoding optimization is analyzed. Two groups of bitstreams are used, which are generated without and with the proposed NNLF skip algorithms. For convenience, they are called original and optimized bitstreams. In Table IV, FPS1 denotes the Frames Per Second (FPS) of the original ECM using original bitstream, FPS2 is the FPS of SADL optimization applied to ECM using original bitstream, and FPS3 is the FPS of SADL optimization using optimized bitstream. The results show that the decoding speed increases to 1.45 times with SADL optimization, and increases again to 1.45 times with NNLF skip algorithms. The overall speed is 2.11 times compared with the original methods.

TABLE IV
PERFORMANCE OF NNLF DECODING OPTIMIZATION

Class	QP	FPS1	FPS2	FPS3	$\frac{\text{FPS2}}{\text{FPS1}}$	$\frac{\text{FPS3}}{\text{FPS2}}$	$\frac{\text{FPS3}}{\text{FPS1}}$
Class C	22	1.21	1.68	1.99	1.39x	1.19x	1.64x
	27	1.23	1.77	2.21	1.43x	1.25x	1.80x
	32	1.34	2.02	2.86	1.51x	1.42x	2.13x
	37	1.50	2.33	4.06	1.56x	1.74x	2.71x
Class D	22	4.04	5.56	7.53	1.38x	1.35x	1.87x
	27	4.28	6.05	8.37	1.41x	1.38x	1.96x
	32	5.11	7.47	11.27	1.46x	1.51x	2.21x
	37	6.45	9.49	16.29	1.47x	1.72x	2.53x
Average					1.45x	1.45x	2.11x

IV. SUMMARY

In this paper, we propose a video coding solution with NN enhancement for CLIC 2025. Several methods are proposed on the basis of the challenge requirements, including ESRGAN-based enhancement pre-processing, advanced RC algorithms, FPP method and speed optimization of NNLF. The proposed methods are firstly evaluated in the pre-evaluation stage, which shows about 0.53 MOS improvement compared with VTM baseline. Extra improvement is made after this stage, such as parameter fine-tuning, but the corresponding MOS result is not available until this paper is submitted. By the way, the team name of this work in challenge is Tencent Compression Model (TCM). And the coding methods of this work also play an important role in the MSU comparison [20], aiding Tencent TVC codec be the first place in 14 tracks out of all 15 tracks.

REFERENCES

[1] G. J. Sullivan, J. -R. Ohm, W. -J. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[2] B. Bross et al., "Overview of the Versatile Video Coding (VVC) Standard and its Applications," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 10, pp. 3736-3764, Oct. 2021.

[3] J. Zhang, C. Jia, M. Lei, S. Wang, S. Ma and W. Gao, "Recent Development of AVS Video Coding Standard: AVS3," 2019 Picture Coding Symposium (PCS), Ningbo, China, 2019.

[4] M. Coban, R.-L. Liao, K. Naser, J. Ström, and L. Zhang, Algorithm description of Enhanced Compression Model 17 (ECM 17), document JVET-AL2025, Joint Video Experts Team, Apr. 2025.

[5] V. Seregin, J. Chen, R. Chernyak et al., JVET AHG report: ECM software development (AHG6), document JVET-AM0006, Joint Video Experts Team, Jul. 2025.

[6] X. Ma, H. Chen, H. Yang, Simplification of the common test condition for fast simulation, document JVET-B0036, Joint Video Experts Team, Feb. 2016.

[7] F. Galpin, Yue Li, Yun Li et al., Description of algorithms version 12 and software version 14 in neural network-based video coding (NNVC), document JVET-AM02019, Joint Video Experts Team, Jul. 2025.

[8] E. Alshina, F. Galpin, S. Liu et al., JVET AHG report: Neural network-based video coding (AHG11), document JVET-AM0011, Joint Video Experts Team, Jul. 2025.
[9] Z. Wang, J. Chen and S. C. H. Hoi, "Deep Learning for Image Super-Resolution: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 10, pp. 3365-3387, 1 Oct. 2021.
[10] Liu, H., Ruan, Z., Zhao, P. et al. Video super-resolution based on deep learning: a comprehensive survey. Artif Intell Rev 55, 5981–6035 (2022).
[11] X. Wang, "ESRGAN: Enhanced super-resolution generative adversarial networks," in Proc. Eur. Conf. Comput. Vis. Workshop, 2018, pp. 63–79.
[12] CLIC 2025, "The 7exth challenge on learned image compression," https://compression.cc.
[13] F. Galpin, T. Dumas, P. Bordes et al., AHG11: Small Ad-hoc Deep-Learning Library, document JVET-W0181, Joint Video Experts Team, Jul. 2021.
[14] A. Stergiou and R. Poppe, "AdaPool: Exponential Adaptive Pooling for Information-Retaining Downsampling," in IEEE Transactions on Image Processing, vol. 32, pp. 251-266, 2023.
[15] R. Yang et al., "NTIRE 2022 Challenge on Super-Resolution and Quality Enhancement of Compressed Video: Dataset, Methods and Results," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 2022, pp. 1220-1237.
[16] x265 Documentation, [Online]. Available: https://x265.readthedocs.io/en/master/introduction.html.
[17] Y. Guo, Y. Chen, and X. Zhang, "Video Encoding Method, Apparatus, Device, Storage Medium, and Computer Program Product," CN patent ZL 202410276385.8, Jun. 14, 2024.
[18] Performance Profiler Visual Studio 2022, Version 17.0. Microsoft.
[19] M. Karczewicz, Y. Ye, Common test conditions and evaluation procedures for enhanced compression tool testing, document JVET-AI2017, Joint Video Experts Team, Jul. 2024.
[20] D. Vatolin, "MSU Video Codecs Comparison 2023-2024 Part 1, 2: FullHD Objective/Subjective", MSU Video Codecs Comparison 2023-2024, Accessed: Sep. 10, 2025 [Online]. Available: https://www.compression.ru/video/codec_comparison/2023/main_report.html.