mathematics

MDPI

Article

A Quantization-Adaptive Early Termination Method for Fast Coding Unit Partitioning in VVC

Donggeon Jo and Dongsan Jun *

Department of Computer Engineering, Dong-A University, Busan 49315, Republic of Korea; dgjo@donga-ispl.kr
* Correspondence: dsjun@dau.ac.kr

Abstract

Versatile Video Coding (VVC) achieves higher compression efficiency than the previous High Efficiency Video Coding (HEVC) standard by employing advanced coding tools, including Quad Tree (QT) and Multi-Type Tree (MTT) block partitioning, extended intra prediction modes, and affine motion compensation. Among these tools, the QT-MTT hierarchical partitioning structure significantly increases encoder complexity, since Rate-Distortion Optimization (RDO) must be performed over an exponentially growing number of partition candidates. To mitigate this complexity, a quantization-adaptive early termination method is proposed that combines neural network-based and rule-based partitioning strategies. The proposed decision mechanism significantly reduces the number of Coding Unit (CU) partition candidates, which directly lowers the number of required RDO evaluations and overall encoder complexity. Experimental results demonstrate that the proposed method achieves a 38.28% reduction in encoding time with only a 0.85% increase in Bjøntegaard Delta Bitrate (BD-BR) under the VVC common test conditions. These results indicate that the proposed method effectively balances computational complexity and rate-distortion performance.

Keywords: versatile video coding (VVC); coding unit partitioning; early termination; encoder complexity; quad tree (QT); multi-type tree (MTT); neural network

MSC: 68T07; 94A08; 68U10

1. Introduction

As multimedia technology advances and online video services become increasingly widespread, the demand for efficient video compression has grown rapidly to support large-scale transmission and storage. In response to this demand, Versatile Video Coding (VVC) [1], finalized in 2020 by the Joint Video Experts Team (JVET), was developed as the latest video coding standard through collaboration between the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG). Compared with its predecessor, High Efficiency Video Coding (HEVC) [2], VVC provides approximately 23% higher compression efficiency under the all-intra (AI) configuration. However, this improvement comes at the cost of substantially increased encoder complexity, which can reach up to 34 times that of HEVC [3]. Consequently, effective complexity reduction in VVC encoders is essential for real-time applications on resource-constrained devices with limited computing capability and memory capacity.

As illustrated in Figure 1, VVC adopts a hierarchical block partitioning structure that combines QT decomposition with Multi-Type Tree (MTT) splitting, where Binary

Academic Editors: Iliya Bouyukliev, Yansheng Wu and Shangdong Yang

Received: 27 February 2026
Revised: 16 April 2026
Accepted: 6 May 2026
Published: 7 May 2026

Tree (BT) and Ternary Tree (TT) partitions are applied in both horizontal and vertical directions. The CU is the basic unit for prediction and transform, and its optimal structure is determined through RDO over all possible partition patterns. As each candidate Coding Unit (CU) requires a full Rate-Distortion Optimization (RDO) evaluation, the optimal partition decision accounts for approximately 96% of total encoding complexity [4].

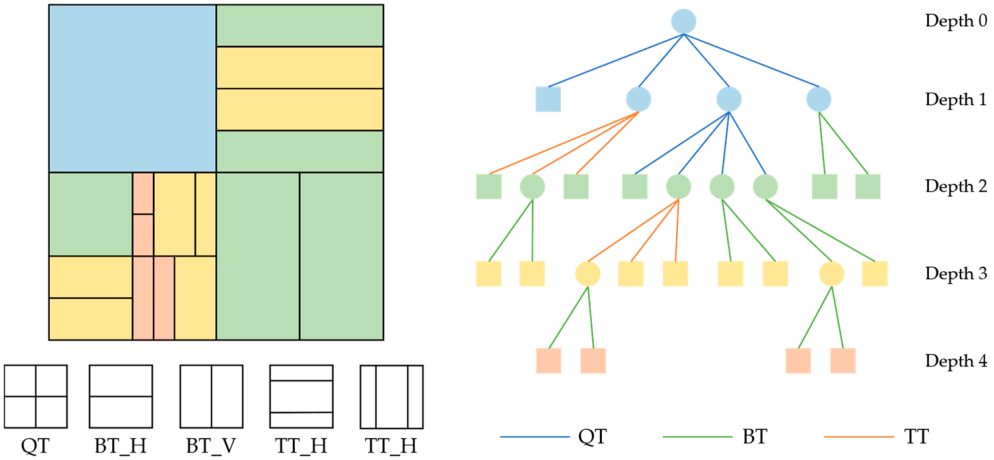

Figure 1. Illustration of Quad Tree (QT) and Multi-Type Tree (MTT) partitioning structures used in Versatile Video Coding (VVC).

To address the high computational burden caused by Quad Tree (QT) and MTT partitioning in VVC, numerous studies have focused on reducing encoder complexity by accelerating the CU partitioning process [5–20]. These approaches can be broadly categorized into rule-based and neural network-based strategies. Although rule-based methods incur low decision complexity by relying on predefined encoding features such as texture characteristics and CU size, their prediction accuracy is often less reliable for diverse video content [21]. This limitation arises because such statistical approaches lack the modeling capacity required to capture complex and nonlinear partitioning relationships. In contrast, neural network-based methods learn partition patterns from large-scale training data and exploit spatial context during inference to improve partition decisions but introduce additional computational overhead. These observations indicate that the optimal partitioning strategy should depend on the Quantization Parameter (QP) rather than remain fixed across all coding scenarios. At low QPs, the small quantization step size preserves high-frequency components and fine texture details, and incorrect partition decisions can noticeably degrade the reconstructed signal. At high QPs, the large quantization step size suppresses most high-frequency components, so differences in partition structure have a smaller impact on the reconstructed signal.

This difference highlights that the impact of partition decisions varies significantly across QP conditions. However, existing approaches rely on a fixed decision strategy regardless of the QP condition. This limitation prevents them from achieving an optimal balance between coding efficiency and encoder complexity. To address this issue, the proposed method adopts different decision strategies for low- and high-QP conditions. The main contributions of this work are summarized as follows:

- A quantization-adaptive early termination strategy is proposed that differentiates partitioning decisions between low- and high-QP conditions to balance coding efficiency and encoder complexity.
- A Hierarchical Two-stage Network (HTN) architecture is developed for low-QP scenarios, in which frame-level feature extraction is decoupled from CU-level partition decisions to significantly reduce inference overhead.

- Under the VVC CTC, the proposed method achieves an average encoding time reduction of 38.28% with only a 0.85% increase in BD-BR. These results indicate an effective balance between computational complexity and coding efficiency.

The remainder of this paper is organized as follows: Section 2 reviews existing CU partitioning methods. Section 3 describes the proposed early termination method for optimal CU partition decision. Finally, experimental results and conclusions are given in Sections 4 and 5, respectively.

2. Related Works

2.1. Rule-Based Approaches

Rule-based approaches determine CU partition structures using features derived from encoding statistics, such as texture characteristics, block statistics, and RD costs. These methods aim to reduce unnecessary partition exploration by identifying low-probability split candidates before performing exhaustive search. Zhang et al. proposed a fast intra coding algorithm that exploits differences in CU depth and RD cost to avoid redundant QT partition evaluations [5]. Chen et al. introduced a fast QT-MTT decision method that uses variance and gradient features to identify CUs with low splitting necessity [6]. Song et al. quantified directional texture complexity using the Sum of Mean Absolute Deviations (SMAD) to prune unlikely horizontal and vertical split directions [7]. Chen et al. developed a Support Vector Machine (SVM)-based predictor to determine QT and MTT partition structures using entropy, texture contrast, and Haar coefficients [8]. Menon et al. analyzed DCT energy-based texture correlations between neighboring blocks to predict CU depth for early termination [9]. Cui et al. proposed a perceptual early termination scheme based on a Just-Noticeable-Difference (JND) model [10]. Although these methods provide low computational overhead, their performance depends heavily on manually designed features, which limits their ability to represent complex partitioning behavior and reduces generalization across diverse video content.

Although these methods provide low computational overhead and can be incorporated into the encoder with limited additional complexity, their decisions rely on manually designed statistics, thresholds, and handcrafted descriptors. As a result, their ability to represent complex and content-dependent partitioning behavior is limited, which reduces robustness across diverse video content. Consequently, these methods may not provide sufficiently accurate partition decisions when the CU structure strongly depends on content characteristics, and coding-efficiency degradation may become more noticeable when conservative partition decisions are required. This limitation may reduce the effectiveness of rule-based approaches when handling diverse content and varying QP conditions.

2.2. Neural Network-Based Approaches

Neural network-based approaches predict CU partition decisions through data-driven training. These approaches enable the modeling of complex nonlinear relationships beyond the scope of manually designed rules. Park et al. proposed a Multi-Layer Perceptron (MLP)-based predictor for fast MTT partitioning with reduced computational overhead [11]. Lee et al. developed a lightweight MLP architecture optimized for resource-constrained environments [12]. Because MLPs process features without explicit spatial modeling, their ability to capture spatial correlations among pixels is limited. To address this limitation, Convolutional Neural Network (CNN)-based approaches have been extensively investigated. Zhang et al. employed a DenseNet-based model to predict partition boundaries and reduce unnecessary RDO evaluations [13]. Wang et al. proposed a multi-stage CNN that progressively predicts partition structures in intra coding [14]. Wu et al. introduced a hierarchical grid-based Fully Convolutional Network (FCN) to estimate the entire partition

hierarchy in a single inference pass [15]. Tissier et al. combined CNN-based prediction with decision-tree refinement [16]. Peng et al. proposed a block-dependent partition decision framework based on partition-homogeneity representation [17]. Belghith et al. applied CNN models to predict hierarchical TT partition tendencies [18]. Im et al. incorporated QP and transform-domain features into a CNN-based partition prediction model [19]. Recent studies have proposed saliency-guided partitioning strategies that integrate content importance into the CU decision process [20].

Compared with rule-based approaches, neural network-based methods generally achieve higher partition prediction accuracy by learning complex spatial and contextual relationships directly from data. This advantage can reduce unnecessary RDO evaluations and help preserve coding efficiency, while at the same time requiring additional model inference. When partition decisions are repeatedly made across many CUs, the cumulative overhead can become substantial. As a result, neural approaches do not necessarily yield the most favorable balance between partition accuracy and encoder complexity under all coding conditions. In existing approaches, the QP is used as a threshold control variable in rule-based methods, while neural network models take it as an additional input feature. In both cases, the overall decision structure remains unchanged regardless of the QP condition. This limitation may affect the ability of both rule-based and neural network-based approaches to maintain a consistent balance between partition accuracy and encoder complexity across all coding conditions.

3. Proposed Method

The proposed method presents a practical CU partitioning scheme designed to reduce encoder complexity in VVC intra coding. As introduced in Section 1, two partitioning strategies are employed according to the QP. The selection between these two strategies is determined by the QP threshold defined in Equation (1):

$$T_{QP} = \lfloor maxQP \times \theta_1 \rfloor, \tag{1}$$

where $maxQP$ denotes the maximum QP value specified under the VVC Common Test Conditions (CTC) [22], and $\theta_1 \in (0, 1)$ is a scaling parameter that controls the switching point between the HTN-based and geometry-based partitioning strategies, and $\lfloor \cdot \rfloor$ denotes the floor operator that maps a real value to the largest integer not exceeding that value. If the current QP is less than T_{QP}, the HTN-based partitioning strategy is applied. Otherwise, the geometry-based partitioning strategy is applied.

3.1. HTN-Based Partitioning Strategy

When the QP is less than T_{QP}, the HTN-based partitioning strategy is employed to ensure accurate determination of CU partition structures. The overall architecture of the HTN-based strategy is illustrated in Figure 2. The HTN-based strategy consists of three hierarchical components: Frame-level Feature Extraction (FFE), CU-level Partition Mode Prediction (PMP), and rule-based decision refinement. These components operate sequentially to balance partition accuracy and computational complexity.

The HTN begins with the FFE module, which extracts a shared feature representation for the entire frame. This representation is reused for all CUs within the same frame to avoid redundant RD computations. For each CU, a spatial feature patch is extracted from the frame-level feature map and transformed into a fixed-length feature vector. The PMP module then produces probability estimates for candidate partition modes. Finally, the rule-based refinement step determines whether the most confident prediction is accepted directly or requires further RDO evaluations.

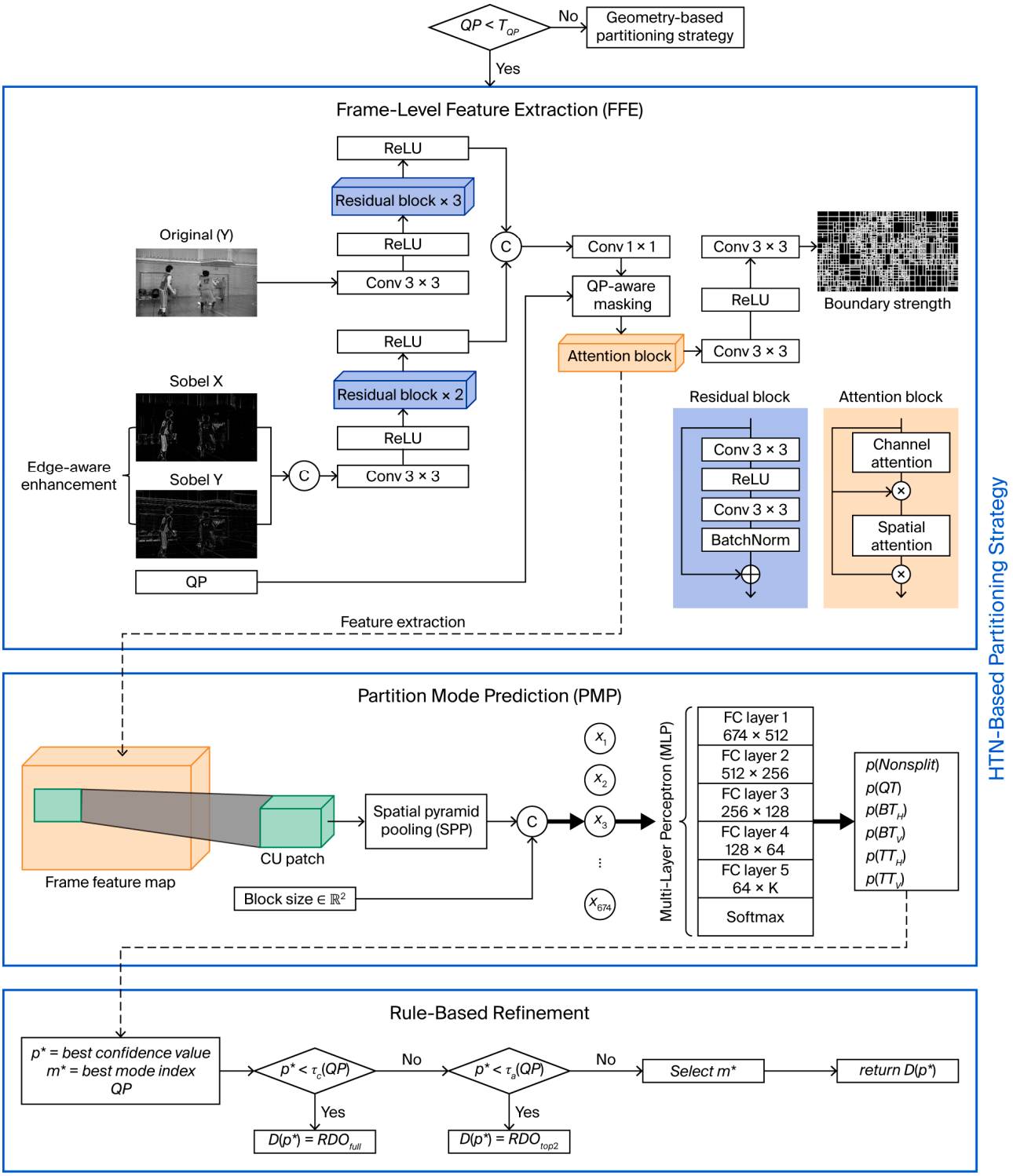

Figure 2. Overall architecture of the proposed Hierarchical Two-stage Network (HTN)-based CU partitioning strategy, including Frame-level Feature Extraction (FFE), CU-level Partition Mode Prediction (PMP), and rule-based refinement.

3.1.1. CNN-Based Frame-Level Feature Extraction

The FFE module is designed to construct a frame-level representation that captures structural, textural, and QP-dependent characteristics relevant to CU partitioning. To achieve this objective, complementary inputs are employed and normalized before network processing to ensure stable feature learning. Specifically, three types of inputs are considered: (1) the luminance (Y) channel of the original frame, which represents spatial

structure; (2) Sobel-filtered horizontal and vertical edge maps, which describe local texture variations; and (3) the QP, which reflects encoding difficulty.

As illustrated in Figure 2, the Y channel and the edge maps are processed through two parallel convolutional pathways. The first pathway applies a 3×3 convolution layer, a *ReLU* activation function, and three residual blocks [23] to learn structural representations from the luminance Y channel. The second pathway concatenates Sobel-based horizontal and vertical gradient maps along the channel dimension and applies a 3×3 convolution layer followed by a *ReLU* activation function and two residual blocks to capture edge-oriented texture information. In this design, the first pathway focuses on the luminance channel to capture coarse structural information such as object shapes and large-scale intensity variations, which strongly influence the depth of the partition tree. The second pathway is fed with horizontal and vertical gradient maps to emphasize fine edge details and directional texture complexity, which are closely related to the choice between binary and ternary splits. By processing these inputs in parallel, the network learns specialized filters for structure and texture without forcing a single pathway to represent both types of information.

The feature outputs from the two pathways are concatenated along the channel dimension to form an integrated representation F. A 1×1 convolution layer is subsequently applied to adjust the channel dimension and facilitate feature fusion. This allows subsequent layers to access both structural and edge-oriented information simultaneously, which is important for reliable partition decisions. The 1×1 convolution then performs a linear combination of the concatenated features at each spatial location, enabling adaptive weighting between structure and edge information. It also reduces channel dimensionality, thereby limiting the computational cost of subsequent modules without discarding relevant information. Overall, this design ensures that complementary features are effectively integrated while maintaining computational efficiency. To incorporate QP-dependent encoding behavior, a QP-aware masking operation [21] is introduced. Let λ_{QP} denote the normalized QP value, defined as $QP/maxQP$. The refined feature map $\widetilde{F}$ is defined as in Equation (2):

$$\widetilde{F}_c = \begin{cases} \lambda_{QP} \cdot F_c, & c \leq C/2, \\ F_c, & c > C/2, \end{cases} \tag{2}$$

where C represents the total number of channels, and F_c denotes the c-th channel of F. Here, the channel refers to the feature dimension index in the multi-channel representation F, that is, the index of individual response maps generated by convolutional filters. This masking operation scales only the first half of the channels by λ_{QP}, so that a subset of feature maps is explicitly modulated by the quantization level while the remaining channels preserve QP-independent structural information. This design encourages the network to allocate part of its capacity to model QP-dependent behavior without discarding general spatial features that are useful across all QPs.

The QP-refined feature map is further processed by an attention block [24], which performs spatial reweighting to emphasize regions near potential CU boundaries. Two additional 3×3 convolution layers are applied to enhance the representational capacity of the network. During training, the resulting representation is supervised using a ground-truth boundary strength map to guide the learning of partition-relevant structures. The boundary strength map is a binary image representing CU partitioning results in grid form. This structured supervision enables the FFE module to learn CU boundary patterns systematically and provides informative cues for subsequent partition mode prediction. During inference, the output of the attention block serves as a 16-channel frame-level feature map.

This shared representation is spatially partitioned according to individual CU positions and sizes, and each extracted patch is forwarded to the MLP-based PMP module.

3.1.2. MLP-Based Partition Mode Prediction

Following the frame-level feature extraction stage, the PMP module performs CU-level partition mode estimation. The PMP module operates on feature patches extracted from the shared frame-level representation generated by the FFE module, as illustrated in Figure 2. Because CUs vary in spatial size under the hierarchical partitioning structure of VVC, the spatial dimensions of the extracted feature patches are not fixed. To address this variability, Spatial Pyramid Pooling (SPP) [25] is employed to transform each variable-sized feature patch into a fixed-length representation. This operation preserves multi-scale spatial information while ensuring consistent feature dimensionality across different CU sizes.

Let $v \in \mathbb{R}^{672}$ denote the feature vector generated by the SPP layer. The dimensionality of 672 corresponds to the concatenation of pooled responses across all pyramid levels and pooling bins defined in the SPP configuration. In addition to appearance-based features, explicit structural information is incorporated through the normalized block size vector $s \in \mathbb{R}^2$, defined as

$$s = \frac{1}{B_{max}} \begin{bmatrix} W \\ H \end{bmatrix}, \tag{3}$$

where W and H denote the width and height of the current CU, and B_{max} represents the maximum allowable CU size defined in the coding configuration. The final input vector of the MLP is constructed by concatenating the feature vector and the normalized block size vector:

$$\S = \begin{bmatrix} v \\ s \end{bmatrix} \in \mathbb{R}^{674}, \tag{4}$$

The vector x is fed into an MLP consisting of five fully connected layers. The detailed architecture, including the number of neurons in each layer and the corresponding activation functions, is illustrated in Figure 2. The network produces a probability vector $p \in \mathbb{R}^K$, where K denotes the number of candidate partition modes. Each element of p represents the predicted likelihood of a specific partition mode. These probability estimates are subsequently forwarded to the rule-based refinement step described in Section 3.1.3 to determine the final partition mode.

3.1.3. Rule-Based Refinement

The PMP module outputs a probability vector $p = [p_1, \ldots, p_K] \in \mathbb{R}^K$, where K denotes the total number of candidate partition modes and p_k represents the predicted probability of the k-th mode. From this probability vector, two quantities are derived. First, the maximum confidence value is defined as

$$p^* = \max_{k \in \{1,\ldots,K\}} p_k, \tag{5}$$

which represents the largest predicted probability among all candidates. Second, the corresponding partition mode index is defined as

$$m^* = \arg\max_{k \in \{1,\ldots,K\}} p_k. \tag{6}$$

Here, p^* is a scalar confidence value, while m^* specifies the index of the partition mode that attains this maximum confidence. In other words, p^* quantifies the confidence of the network prediction, whereas m^* identifies the partition mode selected if the prediction is accepted.

To evaluate the reliability of the neural prediction, a QP-adaptive confidence threshold is introduced. A reference QP is first defined as

$$Q_0 = \left\lceil \frac{maxQP}{\theta_2} \right\rceil, \tag{7}$$

The QP-adaptive confidence threshold is then formulated as

$$\tau_c(QP) = \alpha - \beta(QP - Q_0), \tag{8}$$

where α and β are experimentally determined constants. This formulation ensures that the threshold decreases as QP increases, thereby allowing more aggressive early decisions under high-QP conditions while maintaining conservative behavior in low-QP settings. In addition to the conservative confidence threshold, an aggressive acceptance threshold is defined as

$$\tau_\alpha(QP) = \begin{cases} 0.70, & QP \leq 27, \\ 0.65, & QP > 27, \end{cases} \tag{9}$$

which is empirically determined based on validation experiments. It is defined in a piecewise manner to reflect different signal characteristics at low and high QP conditions. At low QPs, where fine texture details are preserved, partition decisions have a strong impact on reconstruction quality, and a stricter threshold ($\tau = 0.70$) is applied to ensure reliable predictions. At high QPs, where the signal is heavily smoothed due to coarse quantization, the impact of partition decisions becomes less significant, and a relaxed threshold ($\tau = 0.65$) is used to allow more frequent early termination and reduce computational complexity. The final partition decision is expressed as

$$D(p^*) = \begin{cases} RDO_{full}, & p^* < \tau_c(QP), \\ RDO_{top2}, & \tau_c(QP) \leq p^* < \tau_a(QP), \\ m^*, & p^* \geq \tau_a(QP), \end{cases} \tag{10}$$

where RDO_{full} denotes full rate-distortion optimization over all candidate partition modes, and RDO_{top2} denotes RDO restricted to the two most probable modes. When the confidence value exceeds the aggressive threshold, the partition mode m^* is directly accepted without additional RDO evaluation.

This hierarchical confidence-driven refinement strategy ensures that low-confidence predictions are validated through exhaustive search, whereas high-confidence predictions bypass unnecessary RDO computations. Consequently, the proposed HTN-based partitioning strategy achieves an effective trade-off between encoder complexity and coding efficiency.

3.2. Geometry-Based Partitioning Strategy

When the quantization parameter exceeds the switching threshold described in Section 3.1, the proposed method adopts a geometry-based partitioning strategy. In high-QP environments, extensive partition exploration yields limited RD improvement while significantly increasing computational complexity. This is because coarse quantization already introduces substantial distortion, so the additional coding gain obtained by testing many fine partition candidates becomes relatively small compared with that at low-QP values. A deterministic geometry-driven rule is therefore applied to reduce the candidate search space and avoid unnecessary RDO evaluations.

Let W and H denote the width and height of the current CU, respectively. The geometric imbalance is measured using the logarithmic aspect ratio $\left|\log_8(W/H)\right|$, which increases as the block deviates from a square shape. When the imbalance reaches 1 (e.g., $W = 8H$ or $H = 8W$), a TT split is selected to better match the elongated structure. Otherwise, BT splitting is applied. The split direction is determined by the dominant dimension. It is set to horizontal when $W > H$ and to vertical otherwise. This deterministic rule produces a single geometry-driven candidate split mode, denoted by m_{geo}.

Accordingly, instead of performing an exhaustive search over all partition modes, the final decision is obtained by restricting rate-distortion optimization to a reduced candidate set:

$$\hat{m} = \arg\min_{k \in \mathcal{M}_{geo}} J(k), \quad where\ \mathcal{M}_{geo} = \{m_{geo},\ NonSplit\}. \tag{11}$$

Here, $J(k)$ denotes the RD cost associated with partition mode k. By limiting the candidate set to at most one geometry-selected split mode and the non-split mode, the proposed strategy significantly reduces encoder complexity under high-QP conditions while maintaining competitive coding efficiency.

4. Experimental Results

4.1. Experimental Framework

All experiments were conducted on a system equipped with an Intel® Core™ i7-14700KF CPU operating at 5.60 GHz, 64 GB of RAM, and the Windows 11 Pro 64-bit operating system. The GeForce RTX 4090 GPU is used exclusively for neural network training. During encoding and performance evaluation, the trained models are deployed using the CPU-only version of a C++ inference library (LibTorch 2.6.0), and no GPU acceleration is used in the encoding pipeline. Consequently, both the conventional VTM encoding process and the proposed neural network inference are executed entirely on the CPU, which ensures a fair comparison across all methods.

The proposed method is implemented on top of the VVC reference software VTM-23.4 [26]. Performance evaluation is conducted on video sequences belonging to Classes A-E defined in the VVC CTC [22]. All sequences are encoded under the AI configuration using four QP values {22, 27, 32, 37}. Encoding complexity reduction is quantified using the time saving (TS) metric, defined as

$$TS = \frac{T_{VTM} - T_{proposed}}{T_{VTM}} \times 100\%, \tag{12}$$

where T_{VTM} and $T_{proposed}$ denote the encoding times of the reference VTM encoder and the proposed method, respectively. Coding efficiency is evaluated using the BD-BR metric [27].

To train the FFE and PMP networks, the RAISE dataset [28] was used. The dataset contains raw images with four spatial resolutions: 2880 × 1920, 2304 × 1536, 1536 × 1024, and 768 × 512. A total of 1000 images was constructed by selecting 250 images from each resolution. These images are randomly divided into a training set of 800 images and a validation set of 200 images using an 8:2 split ratio and are then encoded using VTM-23.4 under the AI configuration at QP values of 22, 27, 32, and 37. The encoder extracts CU partition structures and partition modes to generate ground-truth labels for supervised learning. The overall training dataset and configuration are summarized in Table 1. The source code and trained model weights used for training have been made publicly available to facilitate reproducibility.

Table 1. Training dataset and configuration for the proposed networks.

Settings	Options
Training sequence	RAISE [28]
Number of sequences	1000
Spatial resolution	2880 × 1920, 2304 × 1536, 1536 × 1024, 768 × 512
Quantization Parameters (QP)	22, 27, 32, 37
Deep learning framework	Pytorch

The FFE network is trained for 250 epochs with a batch size of 32. The initial learning rate is set to 10^{-4}, and it is reduced by a factor of 0.5 every 50 epochs. The Adam optimizer is adopted, and network parameters are initialized using Xavier initialization. Mean Squared Error (MSE) is used as the loss function. The PMP network is trained for 200 epochs with a batch size of 32. The initial learning rate is set to 10^{-4}. A ReduceLROnPlateau [29] scheduler reduces the learning rate by a factor of 0.5 if the training loss does not decrease for three consecutive epochs. The optimizer and initialization scheme are identical to those used for the FFE network. To mitigate class imbalance among partition modes, focal loss [30] is employed.

After training, the trained neural network models are converted to TorchScript [31] format and embedded into the VTM-23.4 encoder to enable seamless integration into the encoding framework. All hyperparameters used in the proposed method are summarized in Table 2. The parameters are configured as follows. The scaling parameter $\theta_1 = 0.55$ in Equation (1) determines the QP switching threshold T_{QP}, which controls the boundary between the HTN-based and geometry-based strategies. The reference scaling parameter $\theta_2 = 0.30$ in Equation (7) defines the reference QP value Q_0 for the confidence threshold computation. The base confidence threshold $\alpha = 0.60$ and the decay rate $\beta = 0.015$ in Equation (8) define the QP-adaptive confidence threshold $\tau_c(QP)$. Specifically, α sets the initial threshold value, while β controls its decay as QP increases. This enables more aggressive early termination at higher QPs. Under the VVC CTC with QP values of 22, 27, 32, and 37, the switching threshold_T_{QP} in Equation (1) is set to 34. Accordingly, the HTN-based partitioning strategy is activated at QPs 22, 27, and 32, whereas the geometry-based partitioning strategy is used at QP 37.

Table 2. Hyperparameter settings used in the proposed method.

Hyperparameters	Options
Optimizer	Adam
Activation function	*ReLU*, Softmax
Loss function (FFE network)	Mean Squared Error (MSE)
Loss function (PMP network)	Focal loss [30]
Learning rate	10^{-4}
Number of epochs (FFE network)	250
Number of epochs (PMP network)	200
Initial weight	Xavier

4.2. Performance Comparisons

Table 3 reports the BD-BR and TS performance of the proposed method under the VVC CTC in the all-intra configuration. In Table 3, "Ours" refers to the proposed method, rather than the reference software alone. The proposed method achieves an average BD-BR increase of 0.85% while reducing encoding time by 38.28%. Compared with the existing methods, the proposed method yields the lowest BD-BR degradation (Cui [10]: 1.32%, Chih-Ying [20]: 1.24%) while maintaining competitive time savings. Here, TS/BD-BR [32]

is defined as TS divided by the BD-BR increase, where a larger value indicates greater time savings per 1% BD-BR increase. The resulting TS/BD-BR ratio of 45.04 confirms a favorable balance between complexity reduction and coding efficiency. These results demonstrate that the proposed hierarchical early termination strategy effectively limits rate-distortion degradation while substantially reducing encoder complexity.

Table 3. Bjøntegaard Delta Bitrate (BD-BR) and Time saving (TS) comparison between the proposed method and existing methods under the VVC Common Test Conditions (CTC) in the all-intra configuration. Here, ↑ indicates that higher values are better and ↓ indicates that lower values are better.

Class	Sequence	Cui (VTM-17.0) [10]		Chih-Ying (VTM-22.0) [20]		Ours (VTM-23.4)	
		BD-BR(%)↓	TS(%)↑	BD-BR(%)↓	TS(%)↑	BD-BR(%)↓	TS(%)↑
A1	Tango2	2.11	18.59	2.59	60.33	0.43	4.37
	FoodMarket4	1.23	31.85	2.06	43.30	0.40	16.75
	Campfire	1.91	19.54	1.21	49.11	0.50	19.00
A2	CatRobot1	2.20	17.25	2.14	46.41	0.79	20.14
	DaylightRoad2	1.44	26.78	1.56	51.24	0.85	22.12
	ParkRunning3	1.21	32.80	0.57	44.05	0.34	32.11
B	MarketPlace	1.64	21.10	2.02	60.49	0.44	53.83
	RitualDance	2.14	17.35	2.47	36.85	0.78	50.40
	Cactus	1.37	26.50	0.88	36.64	0.89	44.58
	BasketballDrive	2.08	17.76	0.86	32.73	0.88	39.30
	BQTerrace	0.71	31.66	0.64	30.82	0.88	50.35
C	RaceHorsesC	0.73	40.08	0.67	33.38	1.21	40.41
	BQMall	1.19	25.14	0.70	24.68	1.11	51.60
	PartyScene	0.43	45.09	0.53	26.20	0.84	49.05
	BasketballDrill	1.46	17.77	1.61	30.94	0.79	39.39
D	BasketballPass	0.56	34.38	0.49	22.89	1.08	44.23
	BQSquare	0.21	62.86	0.53	18.34	0.99	28.14
	BlowingBubbles	0.50	39.56	0.66	21.66	0.87	53.70
	RaceHorses	1.06	22.49	0.93	19.63	0.86	41.50
E	FourPeople	1.44	25.16	1.56	27.67	1.26	47.69
	Johnny	1.82	20.04	1.07	28.06	1.21	42.83
	KristenAndSara	1.53	21.02	1.50	30.67	1.23	50.60
Average		1.32	31.13	1.24	35.28	0.85	38.28
TS/BD-BR		23.58		28.45		45.04	

A resolution-wise analysis reveals consistent behavior across different content classes. For high-resolution sequences (Classes A1 and A2), the proposed method prioritizes coding efficiency and achieves an average BD-BR increase of 0.55% with moderate time savings (19.08%). For medium-resolution content (Classes B and C), it provides a balanced trade-off, with TS values of 47.69% and 45.11% and corresponding BD-BR increases of 0.77% and 0.99%, respectively. For low-resolution sequences (Classes D and E), the method maintains high time savings of 41.89% and 47.04%, while BD-BR increases remain within 0.95% and 1.23%. This behavior indicates that the proposed QP-adaptive strategy adjusts partition exploration according to content characteristics and resolution. As a result, stable rate-distortion performance is maintained across diverse scenarios.

To further examine the partitioning accuracy of the proposed early termination strategy, Figure 3 compares the CU partition mode distributions obtained using full RDO (reference VTM) and the proposed method. Representative sequences under low-QP and high-QP

conditions are included to reflect both HTN-based and geometry-based decision regimes. The close alignment between the two distributions confirms that the proposed method preserves the overall partitioning behavior of the full RDO process. This alignment explains the small average BD-BR increase reported in Table 3. Consequently, the proposed method reduces complexity without substantially altering partition mode statistics.

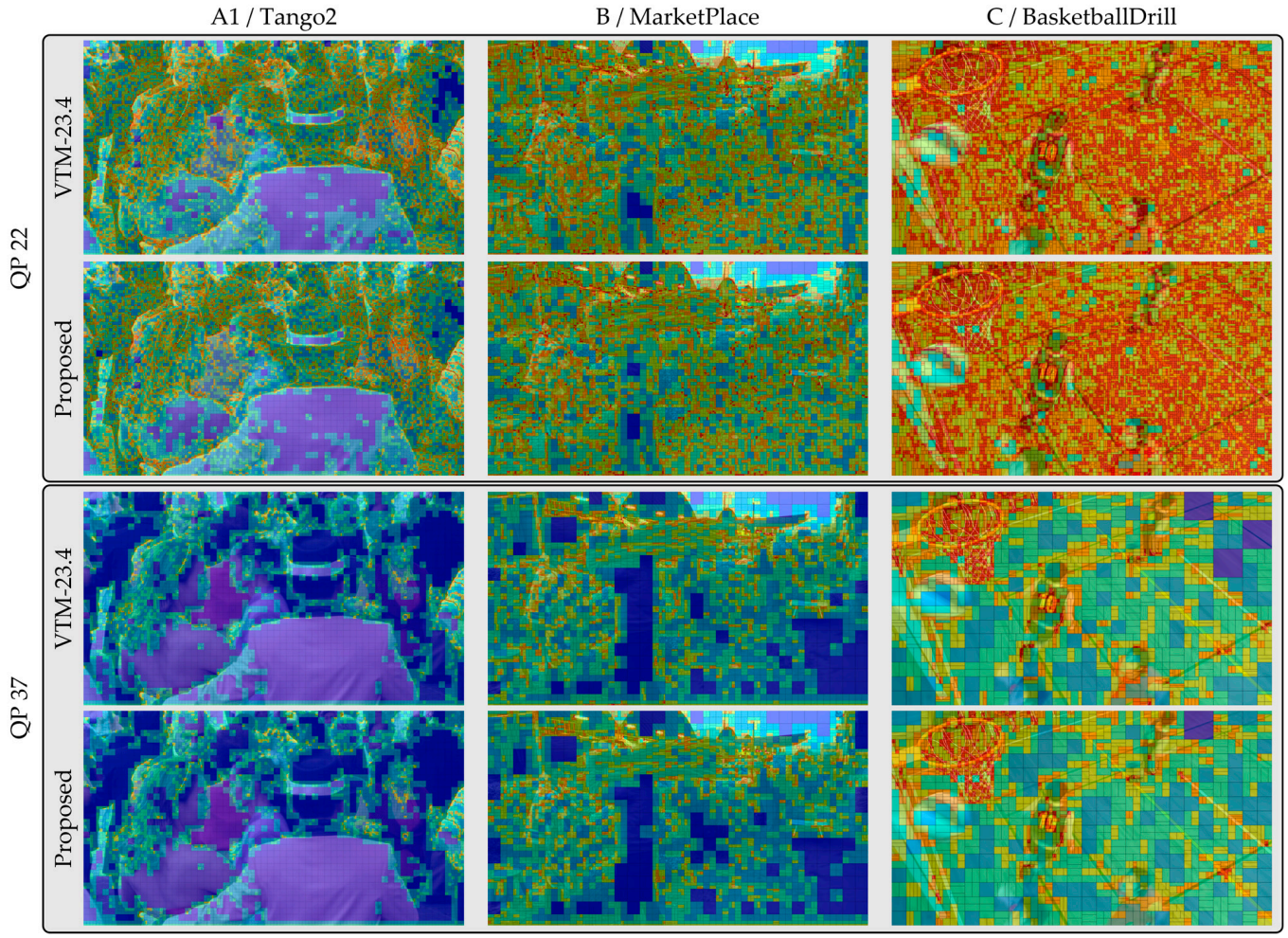

Figure 3. Comparison of Coding Unit (CU) partition mode distributions between full RDO (VTM-23.4) and the proposed early termination method under low-QP and high-QP conditions. Warmer colors indicate smaller block partitions, while cooler colors indicate larger block partitions.

To further characterize the block-level behavior of the proposed decision mechanism, Table 4 reports the proportion of each decision type, namely full RDO (RDO_{full}), top-2 RDO (RDO_{top2}), and direct mode selection (m^*), measured separately for each block size and QP. The analysis is conducted on Classes C and D at QP values of 22, 27, and 32, where the HTN-based strategy is active. QP 37 is excluded as the geometry-based strategy applies a deterministic rule uniformly across all block sizes and therefore does not involve the three-way decision process. Note that the only applicable partition types are QT and non-split for 64 × 64 blocks. This results in at most two candidates, where RDO_{top2} and RDO_{full} are equivalent by definition and the RDO_{full} entry is therefore reported as 0%.

This indicates that the HTN prediction confidence is low for square blocks, where multiple partition candidates remain competitive and the dominant split direction is ambiguous. In contrast, for rectangular blocks such as 8 × 32, 32 × 8, 16 × 32, and 32 × 16, the m^* ratio ranges from approximately 61% to 87%. This reflects the tendency of the HTN to assign high confidence to a single dominant partition mode when the block geometry

constrains the feasible split directions. Regarding the effect of QP, the overall m^* ratio decreases from 39.61% at QP 22 to 34.66% at QP 32, while the RDO_{top2} ratio increases from 20.29% to 24.20%, and the RDO_{full} ratio increases slightly from 40.09% to 41.14%. This trend suggests that as the QP increases within the HTN-active range, the HTN tends to produce predictions with intermediate confidence more frequently. As a result, the decision shifts from direct m^* acceptance toward RDO_{top2} evaluation. These results demonstrate that the proposed method does not apply a uniform decision policy across all blocks. Instead, it adapts the search intensity according to block geometry and QP. This reduces unnecessary RDO evaluations where partition decisions are predictable, while preserving full search capability in cases where uncertainty is high.

Table 4. Ratio (%) of each decision type by block size and QP for Classes C and D under the HTN-based partitioning strategy.

Block Size		QP								
Width	Height	22			27			32		
		RDO_{full}	RDO_{top2}	m^*	RDO_{full}	RDO_{top2}	m^*	RDO_{full}	RDO_{top2}	m^*
4	16	50.07	17.24	32.69	48.61	26.06	25.33	54.63	29.01	16.36
4	32	45.39	23.01	31.60	51.71	24.84	23.45	55.84	28.52	15.64
8	8	49.86	14.79	35.35	48.89	26.08	25.03	55.07	30.05	14.88
8	16	42.74	22.21	35.05	43.52	27.32	29.16	45.89	27.40	26.71
8	32	13.25	7.32	79.43	18.54	12.24	69.22	17.53	16.20	66.27
16	4	55.58	23.11	21.31	48.36	31.41	20.23	58.62	25.26	16.12
16	8	42.31	23.24	34.45	47.81	19.20	32.99	47.87	23.26	28.87
16	16	79.90	8.06	12.04	80.21	11.70	8.09	76.08	12.03	11.89
16	32	20.83	18.52	60.65	12.68	9.84	77.48	9.73	3.27	87.00
32	4	45.73	22.25	32.02	58.25	20.25	21.50	53.07	32.98	13.95
32	8	18.23	9.06	72.71	15.59	10.90	73.51	22.40	9.83	67.77
32	16	21.03	6.84	72.13	14.33	9.48	76.19	9.04	5.62	85.34
32	32	76.40	10.26	13.34	75.95	11.02	13.03	70.21	15.29	14.50
64	64	0.00	78.21	21.79	0.00	77.86	22.14	0.00	80.10	19.90
Average		40.09	20.29	39.61	40.32	22.73	36.95	41.14	24.20	34.66

To investigate the behavior of the method under different content characteristics, we compare the ParkRunning3 sequence from Class A2 (BD-BR: 0.34%, TS: 32.11%) and the FourPeople sequence from Class E (BD-BR: 1.26%, TS: 47.69%) using three content complexity metrics. Edge strength is measured as the mean gradient magnitude computed from the horizontal and vertical Sobel responses [33]. High-frequency ratio is computed from the grayscale image in the Fourier domain using the magnitude spectrum obtained by a 2D FFT [34]. It is defined as the ratio between the summed spectral magnitudes outside the central low-frequency region and the total spectral magnitude, where frequency components with a normalized radial distance greater than 0.35 from the spectrum center are regarded as high-frequency components. Laplacian variance is calculated as the variance of the Laplacian response and serves as a measure of local sharpness and intensity variation [35].

As shown in Table 5, ParkRunning3 exhibits higher values across all three metrics (edge strength: 0.06, high-frequency ratio: 0.46, Laplacian variance: 0.022). This indicates globally consistent structural boundaries that allow the HTN to generate high-confidence

partition predictions, and results in a low BD-BR increase of 0.34%. In contrast, FourPeople presents lower values (edge strength: 0.03, high-frequency ratio: 0.25, Laplacian variance: 0.004), a pattern that reflects spatially heterogeneous regions that mix complex foreground objects with smooth backgrounds. This heterogeneity introduces diverse CU-level partition characteristics that are more difficult to predict consistently, and results in a BD-BR increase of 1.26% despite a higher TS of 47.69%. These results demonstrate that the proposed method achieves lower BD-BR degradation on content with globally consistent structural boundaries, whereas spatially heterogeneous content tends to incur a larger quality trade-off.

Table 5. Content complexity metrics (edge strength, high-frequency ratio, and Laplacian variance) and coding performance (BD-BR, TS) for representative sequences in Classes A2 and E.

Sequence		
	A2/ParkRunning3	**E/FourPeople**
BD-BR (%)	0.34	1.26
TS (%)	32.11	47.69
Edge Strength	0.06	0.03
High-frequency Ratio (%)	0.46	0.25
Laplacian Variance	0.022	0.004

Because prior works were implemented on different VTM versions, the potential impact of encoder version differences is analyzed in Table 6, where all three entries represent unmodified VVC reference encoders without the proposed method. The comparison focuses on recent VTM releases (VTM-17.0 and VTM-22.0) relative to VTM-23.4 under identical VVC CTC. The BD-BR differences remain within $\pm 0.03\%$ and encoding time variations are generally below 10% for most classes. TS is defined as the relative encoding time reduction within the same encoder version. Therefore, differences in absolute encoding time across VTM versions do not materially affect the relative complexity reduction ratio. The comparative conclusions drawn in Table 3 remain valid across recent VTM implementations. Therefore, the competitive performance of the proposed method primarily reflects the effectiveness of the algorithmic design rather than version-specific factors.

Table 6. Relative BD-BR (%) and encoding time (EncT) differences in VTM-23.4 compared with VTM-17.0 and VTM-22.0 under the VVC CTC.

Class	**VTM-23.4 Over VTM-17.0**		**VTM-23.4 Over VTM-22.0**	
	BD-BR (%)	**EncT (%)**	**BD-BR (%)**	**EncT (%)**
A1	0.00	89.01	0.00	92.09
A2	0.00	91.84	0.00	94.16
B	0.00	92.37	0.00	94.70
C	0.00	94.09	0.00	97.70
D	−0.02	95.07	−0.02	97.50
E	0.00	92.34	−0.01	95.17
Overall	0.00	92.45	−0.01	95.05

To quantify the computational overhead introduced by the proposed neural network, we separately measure the proportion of encoding time attributable to HTN inference. In addition, QP 37 is excluded from this analysis because the HTN is not used at this QP. On average, HTN inference accounts for 29.47% of the total encoding time of the proposed method. Although HTN inference introduces additional computational cost, this overhead is compensated by the reduction in RDO evaluations. As a result, the proposed method achieves an average encoding time saving of 46.04% compared to the VTM anchor. A detailed analysis of HTN inference time relative to the total encoding time is presented in Table 7. Note that Table 7 covers Classes C and D only, where the proposed method achieves a TS of 46.04%, while the overall TS of 38.28% is obtained across all classes under the full VVC CTC as reported in Table 3.

Table 7. Encoding time (EncT) analysis of the proposed method for Classes C and D at QPs 22, 27, and 32. All encoding times are reported in seconds.

Class	QP	VTM EncT (s)	Ours	
			Proposed EncT (s)	HTN Inference (s)
C	22	8204.91	4205.93	1126.10
	27	7109.43	3354.79	975.05
	32	4436.32	2742.58	1004.46
D	22	1827.31	1219.17	286.55
	27	1809.11	1026.38	290.32
	32	1434.60	844.95	274.96
Average		4136.95	2232.30	659.57

4.3. Ablation Study

To isolate the individual contributions of the HTN-based and geometry-based partitioning strategies, we conduct an ablation study on Classes C and D, in which the full proposed method and two ablated variants are evaluated. Note that the two ablated variants are evaluated under different effective QP conditions, and therefore, their averaged results should not be interpreted as a direct numerical comparison. However, the tool-off evaluation still provides consistent trends within each QP condition, which allows us to analyze the relative behavior of each method.

The first variant applies only the HTN-based method, while the second variant applies only the geometry-based method. The results are summarized in Table 8. When only the HTN-based method is applied, the time saving decreases from 43.52% to 35.01%, while BD-BR improves to 0.90%. This indicates that the HTN-based method preserves coding accuracy but provides a limited reduction in computational complexity.

Table 8. Ablation study results on Classes C and D. TS and BD-BR are reported for the proposed method and two ablated variants.

Class	Ours		HTN-Only		Geometry-Only	
	BD-BR (%)	TS (%)	BD-BR (%)	TS (%)	BD-BR (%)	TS (%)
C	0.98	45.11	0.90	39.09	4.81	77.38
D	0.95	41.94	0.91	30.93	5.37	77.99
Average	0.96	43.52	0.90	35.01	5.09	77.68

In contrast, when only the geometry-based method is applied, the time saving increases to 77.68%, but BD-BR exhibits a severe degradation of 5.09%. This behavior occurs because the geometry-based method reduces partition complexity but does not provide

sufficient accuracy for partition decisions. These results indicate that relying only on a single method leads to an imbalance between coding efficiency and encoding time. In particular, the HTN-based method focuses on coding efficiency, whereas the geometry-based method focuses on reducing encoding complexity. By combining these two complementary properties, the proposed method achieves a better trade-off between coding efficiency and encoding time reduction than either method alone.

5. Conclusions

This paper addresses the problem of high computational complexity caused by QT-MTT-based CU partitioning in VVC intra coding. To reduce encoder complexity while preserving coding efficiency, a QP-adaptive hierarchical early termination strategy is developed. The proposed method integrates two complementary partitioning strategies under a QP-dependent switching mechanism. Under low-QP conditions, an HTN predicts CU partition modes to maintain partitioning accuracy. Under high-QP conditions, a lightweight geometry-based rule limits unnecessary partition exploration. This selective activation mechanism controls computational cost according to coding conditions while preserving stable partition behavior. Experimental results under the VVC CTC show that the method achieves an average encoding time reduction of 38.28% with a BD-BR increase of 0.85%. The partition mode distributions generated by the early termination strategy remain closely aligned with those obtained through full RDO. This alignment explains the limited rate-distortion degradation. These results confirm that the proposed method reduces CU partitioning complexity without significantly affecting coding performance. Accordingly, the method provides a practical and stable complexity reduction solution for VVC intra encoders across diverse content characteristics.

Author Contributions: Conceptualization, D.J. (Donggeon Jo) and D.J. (Dongsan Jun); methodology, D.J. (Donggeon Jo) and D.J. (Dongsan Jun); software, D.J. (Donggeon Jo); formal analysis, D.J. (Dongsan Jun); investigation, D.J. (Donggeon Jo) and D.J. (Dongsan Jun); resources, D.J. (Dongsan Jun); data curation, D.J. (Donggeon Jo); writing—original draft preparation, D.J. (Donggeon Jo); writing—review and editing, D.J. (Dongsan Jun); visualization, D.J. (Donggeon Jo); supervision, D.J. (Dongsan Jun); project administration, D.J. (Dongsan Jun).; funding acquisition, D.J. (Dongsan Jun). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (RS-2023-00246528) and by the AI Computing Infrastructure Enhancement (GPU Rental Support) User Support Program funded by the Ministry of Science and ICT (MSIT), Republic of Korea.

Data Availability Statement: Our implementation and pre-trained models are available at [https://github.com/ISPL-dgjo/A-Quantization-Adaptive-Early-Termination-Method-for-Fast-Coding-Unit-Partitioning-in-VVC] (accessed on 5 May 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bross, B.; Wang, Y.K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; Ohm, J.R. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3736–3764. [CrossRef]
2. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [CrossRef]
3. Mercat, A.; Mäkinen, A.; Sainio, J.; Lemmetti, A.; Viitanen, M.; Vanne, J. Comparative rate-distortion-complexity analysis of VVC and HEVC video codecs. *IEEE Access* **2021**, *9*, 67813–67828. [CrossRef]
4. Si, L.; Zhu, W.; Zhang, Q. Fast adaptive CU partition decision algorithm for VVC intra coding. *IEEE Access* **2023**, *11*, 119766–119778. [CrossRef]

5. Zhang, T.; Sun, M.T.; Zhao, D.; Gao, W. Fast intra-mode and CU size decision for HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 1714–1726. [CrossRef]
6. Chen, J.; Sun, H.; Katto, J.; Zeng, X.; Fan, Y. Fast QTMT partition decision algorithm in VVC intra coding based on variance and gradient. In Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, 1–4 December 2019.
7. Song, Y.; Zeng, B.; Wang, M.; Deng, Z. An efficient low-complexity block partition scheme for VVC intra coding. *J. Real-Time Image Process.* **2022**, *19*, 161–172. [CrossRef]
8. Chen, F.; Ren, Y.; Peng, Z.; Jiang, G.; Cui, X. A fast CU size decision algorithm for VVC intra prediction based on support vector machine. *Multimed. Tools Appl.* **2020**, *79*, 27923–27939. [CrossRef]
9. Menon, V.V.; Amirpour, H.; Timmerer, C.; Ghanbari, M. INCEPT: Intra CU depth prediction for HEVC. In Proceedings of the 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP), Tampere, Finland, 6–8 October 2021.
10. Cui, X.; Lin, H.; Zhu, G.; Zhong, H. Perceptual based fast CU partition algorithm for VVC intra coding. In Proceedings of the TENCON 2023–2023 IEEE Region 10 Conference (TENCON), Chiang Mai, Thailand, 17–20 October 2023. [CrossRef]
11. Park, S.; Kang, J. Fast multi-type tree partitioning for versatile video coding using a lightweight neural network. *IEEE Trans. Multimed.* **2021**, *23*, 4388–4399. [CrossRef]
12. Lee, T.; Jun, D.; Park, S.H.; Kim, B.G.; Yun, J.; Yun, K.; Cheong, W.S. Medical image compression method using light-weight multi-layer perceptron for mobile healthcare applications. *Comput. Mater. Contin.* **2022**, *70*, 2013–2029. [CrossRef]
13. Zhang, Q.; Guo, R.; Jiang, B.; Su, R. Fast CU decision-making algorithm based on DenseNet network for VVC. *IEEE Access* **2021**, *9*, 119289–119297. [CrossRef]
14. Wang, Y.; Dai, P.; Zhao, J.; Zhang, Q. Fast CU partition decision algorithm for VVC intra coding using an MET-CNN. *Electronics* **2022**, *11*, 3090. [CrossRef]
15. Wu, S.; Shi, J.; Chen, Z. HG-FCN: Hierarchical grid fully convolutional network for fast VVC intra coding. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 5638–5649. [CrossRef]
16. Tissier, A.; Hamidouche, W.; Mdalsi, S.B.D.; Vanne, J.; Galpin, F.; Menard, D. Machine learning based efficient QT-MTT partitioning scheme for VVC intra encoders. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 4279–4293. [CrossRef]
17. Peng, Z.; Shen, L.; Ding, Q.; Dong, X.; Zheng, L. Block-dependent partition decision for fast intra coding of VVC. *IEEE Trans. Consum. Electron.* **2023**, *70*, 277–289. [CrossRef]
18. Belghith, F.; Abdallah, B.; Ben Jdidia, S.; Ben Ayed, M.A.; Masmoudi, N. CNN-based ternary tree partition approach for VVC intra-QTMT coding. *Signal Image Video Process.* **2024**, *18*, 3587–3594. [CrossRef]
19. Im, S.-K.; Chan, K.-H. Faster intra-prediction of versatile video coding using a concatenate-designed CNN via DCT coefficients. *Electronics* **2024**, *13*, 2214. [CrossRef]
20. Chih-Ying, L.; Jia-Yu, C.; Sheng-Chen, W.; Tzu-Der, C. Perception-driven and object-aware fast MTT partitioning for H.266/VVC: A saliency-guided complexity reduction framework. *Electronics* **2025**, *14*, 133. [CrossRef]
21. Li, T.; Xu, M.; Tang, R.; Chen, Y.; Xing, Q. DeepQTMT: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC. *IEEE Trans. Image Process.* **2021**, *30*, 5377–5390. [CrossRef]
22. Bossen, F.; Boyce, J.; Sühring, K.; Li, X.; Seregin, V. VTM common test conditions and software reference configurations for SDR video. In Proceedings of the 20th Meeting of Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Teleconference, 7–16 October 2020.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
24. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018. [CrossRef]
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
26. JVET. Versatile Video Coding Test Model (VTM) Reference Software of the JVET of ITU-T VCEG and ISO/IEC MPEG. Available online: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM (accessed on 28 January 2026).
27. Bjontegaard, G. *Calculation of Average PSNR Differences Between RD-Curves*; ITU-Telecommunications Standardization Sector: Geneva, Switzerland, 2001.
28. Dang-Nguyen, D.T.; Pasquini, C.; Conotter, V.; Boato, G. RAISE: A raw images dataset for digital image forensics. In Proceedings of the 6th ACM Multimedia Systems Conference, Portland, OR, USA, 18–20 March 2015. [CrossRef]
29. PyTorch. Torch.optim.lr_scheduler.ReduceLROnPlateau. Available online: https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html (accessed on 2 February 2026).
30. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017. [CrossRef]
31. PyTorch. TorchScript Documentation. Available online: https://pytorch.org/docs/stable/jit.html (accessed on 2 February 2026).

32. Tang, G.; Jing, M.; Zeng, X.; Fan, Y. Adaptive CU split decision with pooling variable CNN for VVC intra encoding. In Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, 1–4 December 2019.
33. Van der Walt, S.; Schönberger, J.L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.D.; Yager, N.; Gouillart, E.; Yu, T. scikit-image: Image processing in Python. *PeerJ* **2014**, *2*, e453. [CrossRef]
34. De, K.; Masilamani, V. Image sharpness measure for blurred images in frequency domain. *Procedia Eng.* **2013**, *64*, 149–158. [CrossRef]
35. Pech-Pacheco, J.L.; Cristóbal, G.; Chamorro-Martínez, J.; Fernández-Valdivia, J. Diatom autofocusing in brightfield microscopy: A comparative study. In Proceedings of the 15th International Conference on Pattern Recognition (ICPR), Barcelona, Spain, 3–8 September 2000.