

# Predicting Encoding Energy from Low-Pass Anchors for Green Video Streaming

Zoha Azimi

Institute of Information Technology, University of  
Klagenfurt  
Klagenfurt, Austria

Vignesh V Menon

Video Communication and Applications Dept, Fraunhofer  
HHI  
Berlin, Germany

Reza Farahani

Institute of Information Technology, University of  
Klagenfurt  
Klagenfurt, Austria

Christian Timmerer

Institute of Information Technology, University of  
Klagenfurt  
Klagenfurt, Austria

## Abstract

Video streaming now represents the dominant share of Internet traffic, as ever-higher-resolution content is distributed across a growing range of heterogeneous devices to sustain user Quality of Experience (QoE). However, this trend raises significant concerns about energy efficiency and carbon emissions, requiring methods to provide a trade-off between energy and QoE. This paper proposes a lightweight energy prediction method that estimates the energy consumption of high-resolution video encodings using reference encodings generated at lower resolutions (so-called anchors), eliminating the need for exhaustive per-segment energy measurements, a process that is infeasible at scale. We automatically select encoding parameters, such as resolution and quantization parameter (QP), to achieve substantial energy savings while maintaining perceptual quality, as measured by the Video Multimethod Fusion Assessment (VMAF), within acceptable limits. We implement and evaluate our approach with the open-source VVenC encoder on 100 video sequences from the Inter4K dataset across multiple encoding settings. Results show that, for an average VMAF score reduction of only 1.68, which stays below the Just Noticeable Difference (JND) threshold, our method achieves 51.22% encoding energy savings and 53.54% decoding energy savings compared to a scenario with no quality degradation.

## CCS Concepts

• Information systems → Multimedia streaming; • Computing methodologies → Artificial intelligence.

## Keywords

Video Streaming, Video on Demand, Machine Learning, Energy Efficiency.

## 1 Introduction

Video streaming applications, such as live content and Video-on-Demand (VoD), now dominate global Internet traffic as recent reports indicate that video content accounts for over 70% of total traffic today, with projections exceeding 80% by 2028 [1]. HTTP Adaptive Streaming (HAS) methods such as MPEG Dynamic Adaptive Streaming over HTTP (DASH) [2, 3] and Apple HTTP Live

Streaming (HLS) [4] have become the de facto standard video delivery method. In these methods, each video is encoded into multiple resolution–bitrate pairs, forming a bitrate ladder, from which clients dynamically select the most suitable representation according to current network and device conditions [5]. However, constructing such ladders requires encoding each video sequence into multiple representations, a process that is both computationally intensive and energy-demanding [6]. This energy cost is further intensified by modern video codecs such as High Efficiency Video Coding (HEVC) [7] and Versatile Video Coding (VVC) [8], which achieve higher compression efficiency through advanced prediction, partitioning, and transform coding tools [9, 10].

This highlights a key challenge in adaptive streaming, i.e., balancing video quality, compression efficiency, and energy consumption, where the choice of the proper encoding configuration plays a pivotal role in achieving this balance [11, 12]. Parameters such as resolution, framerate, and quantization parameter (QP) directly influence compression efficiency, perceptual quality, and energy consumed during encoding and decoding [11, 13, 14]. Since higher video quality levels typically require higher energy costs, efficient configurations become essential for sustainable video streaming. While heuristic [15] or Artificial Intelligence (AI)-driven methods [16–18] have been proposed to optimize configuration selection, they primarily focus on compression efficiency and perceptual quality, giving insufficient attention to energy considerations. Moreover, accurate energy evaluation requires per-segment energy measurements across multiple configurations, which is infeasible at scale.

This paper proposes a practical and scalable scheme that uses reference encodings generated at lower resolutions (hereafter referred to as anchors) as a proxy to predict the energy consumption of high-resolution representations. The core hypothesis is that encoding time and energy are strongly correlated, allowing patterns observed in low-resolution encodings to be leveraged for predicting the energy required at higher resolutions. For example, encoding a sequence at 360p or 540p resolution with a fixed QP typically completes much faster and consumes less energy than its 1080p or 2160p counterparts, yet still captures the content’s inherent characteristics such as motion complexity, texture richness, and scene dynamics. Using these anchors, we train machine learning (ML) models to predict higher-resolution energy consumption without exhaustive measurements, guiding an energy-aware configuration strategy that minimizes energy use while preserving perceptual



This work is licensed under a Creative Commons Attribution International 4.0 License.

quality, evaluated through Signal-to-Noise Ratio (PSNR) [19] and Video Multimethod Fusion Assessment (VMAF) [20]. The main contributions of this paper are as follows:

- **Dataset generation:** We construct a dataset of encoding and decoding time, energy consumption, and PSNR, VMAF scores for 100 video sequences from the Inter4K [21] dataset.
- **Anchor-based modeling:** We introduce the concept of low-resolution anchor encodings and demonstrate that their measurements provide meaningful insights into energy consumption trends at higher resolutions.
- **ML-based predictions:** We develop ML models that leverage features extracted from anchor encodings to accurately estimate both energy consumption and perceptual quality.
- **Energy-aware configuration strategy:** We design an encoding parameter selection method that uses predicted energy and quality values, minimizing energy consumption while maintaining visual quality.

## 2 Related work

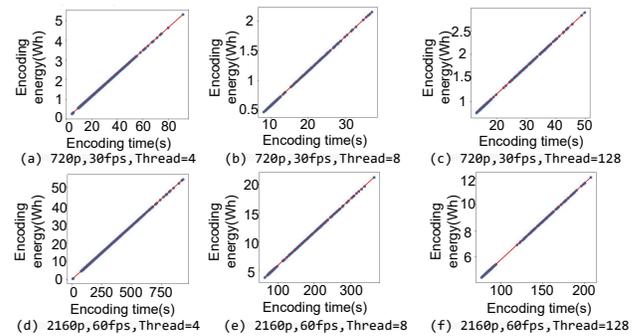
### 2.1 Energy consumption prediction

Several works have proposed methods to estimate the energy consumption during video encoding or decoding. Ghasempour *et al.* [22] proposed a lookup table method for the fast estimation of encoding and decoding energy based on video content, resolution, and framerate features. Sharrab *et al.* [23] introduced a linear regression model for encoding energy consumption using the motion estimation range of video and QP. Azimi *et al.* [13] developed Extreme Gradient Boosting (XGBoost) models to estimate encoding energy consumption using features such as video complexity, quantization parameter (QP), resolution, frame rate, and codec type. Herglotz *et al.* [24] used linear regression models to estimate decoding energy based on decoding processing time. Turkkan *et al.* [25] developed a neural network-based model to predict the decoding power consumption of a video sequence using parameters such as bitrate, resolution, framerate, and file size. Farahani *et al.* [26] introduced a relative decoding energy index (RDEI), a metric that normalizes decoding energy consumption against a baseline encoding configuration, enabling cross-platform comparability and guiding energy-efficient streaming adaptations.

*State-of-the-art limitations:* These methods rely on full encodings or decodings with specialized energy measurement tools, making them resource-intensive and limited in their ability to represent all encoding scenarios.

### 2.2 Encoding parameter configuration

Another line of research focuses on selecting encoding parameters to optimize video quality and efficiency. Lebreton *et al.* [27] designed bitrate ladders based on user quitting probabilities, improving the perceived Quality of Experience (QoE). In parallel, ML-based approaches, such as Random Forest (RF)-based models [28], have been employed to predict optimal segment resolutions for enhanced perceptual quality. Huang *et al.* [17] proposed a reinforcement learning-based method to dynamically select bitrate-resolution pairs, jointly optimizing video quality, storage cost, and adaptation to network conditions. Azimi *et al.* [29] used XGBoost models to predict the



**Figure 1: The correlation between encoding time and encoding energy for 100 video sequences, encoded with 720p/30fps and 2160p/60fps with three different number of threads.**

decoding time and used a decoding time-constrained configuration setting. Rajendran *et al.* [16] used Pareto-front analysis to predict optimized framerates, constructing decoding complexity-aware ladders. Similarly, Katsenou *et al.* [14] used video quality, decoding time, and bitrate to optimize bitrate ladder construction.

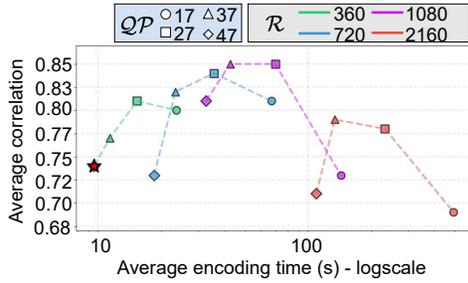
*State-of-the-art limitations:* While these approaches advance perceptual quality and compression efficiency, they largely neglect energy consumption, an increasingly critical factor for sustainable video streaming.

## 3 Motivation

Our approach is motivated by two key observations from preliminary experiments, enabling efficient energy prediction across multiple encoding settings.

First, Fig. 1 shows the correlation between encoding time and energy consumption across 100 video sequences from the Inter4K dataset [21] encoded at 720p/30fps and 2160p/60fps. The experiments were conducted using different numbers of threads (4, 8, and 128). In all configurations, a strong linear correlation between encoding time and energy consumption is observed. Similar strong correlations have been observed in other works [24]. Unlike energy measurement, measuring execution time is computationally inexpensive and does not require specialized instrumentation. Thus, we use encoding time as a reliable and practical proxy for energy consumption.

Second, we observe that encoding times across different representations of the same video are strongly correlated. Fig 2 depicts the relationship between average encoding time (in seconds, shown on a logarithmic scale) and the average correlation of encoding times across different video representations (resolutions and QPs) across our 100 test video sequences. For each resolution-QP pair (e.g., 360p, QP47), we computed the correlation between its encoding times and those of all other pairs across 100 video sequences. Overall, average pairwise correlations exceed 0.65, revealing consistent temporal behavior across representations, with the correlation peaks around 0.8 for medium encoding settings (720p–1080p, QP:27–37). To exploit this relationship, we select the representation with the lowest resolution and highest QP as the anchor value (i.e., red star) for each video sequence, as it provides the fastest encoding with



**Figure 2: Average correlation of encoding times across 100 video sequences for different resolutions and QPs. Each point shows the mean correlation of one configuration with all others, plotted against its average encoding time (log scale).**

minimal computational cost, achieving a correlation of 0.74 while requiring only 9.58 s of encoding time. Thus, by measuring the anchor’s encoding time, we can infer the energy consumption of all other representations, significantly reducing computation and energy costs. This concept is extended to predict decoding energy and quality metrics using the anchor’s decoding time and perceptual quality measurements, respectively.

## 4 System Design

### 4.1 Architecture

Fig. 3 shows the proposed architecture with three main components:

(i) *Anchor processing* encodes and decodes the lowest resolution ( $r_{min}$ ) and highest quantization parameter ( $qp_{max}$ ) representation as the anchor, measures its encoding time ( $t_{enc}^A$ ), decoding time ( $t_{dec}^A$ ), and quality metric ( $q^A$ ), such as PSNR or VMAF.

(ii) *Prediction module* takes the anchor processing outputs ( $t_{enc}^A$ ,  $t_{dec}^A$ ,  $q^A$ ) together with the target resolutions ( $\mathcal{R}$ ) and QPs ( $\mathcal{QP}$ ) as input and employs  $f_{enc}$ ,  $f_{dec}$ , and  $f_q$  to predict the encoding energy ( $\hat{e}_{enc}$ ), decoding energy ( $\hat{e}_{dec}$ ), and quality ( $\hat{q}$ ) for the target video representations.

$$\hat{e}_{enc} = f_{enc}(t_{enc}^A, r \in \mathcal{R}, qp \in \mathcal{QP}),$$

$$\hat{e}_{dec} = f_{dec}(t_{dec}^A, r \in \mathcal{R}, qp \in \mathcal{QP}),$$

$$\hat{q} = f_q(q^A, r \in \mathcal{R}, qp \in \mathcal{QP}).$$

(iii) *Green encoding configurations* leverages the prediction results to recommend encoding parameters ( $r$ ,  $qp$ ) based on an acceptable quality degradation factor  $\rho$ . When  $\rho = 0$ , no quality degradation is allowed, and the encoding parameters are selected to provide the highest visual quality. In contrast, when  $\rho = 1$ , the configurations prioritize minimum energy consumption, regardless of quality.

### 4.2 Execution workflow

Algo. 1 outlines the execution workflow of our proposed method. For a given video sequence, a set of target resolutions  $\mathcal{R}$  and quantization parameters  $\mathcal{QP}$  are defined, along with an acceptable quality degradation factor  $\rho$ . First, the video sequence is encoded at the lowest resolution ( $r_{min}$ ) and highest QP ( $qp_{max}$ ), which serves as an anchor. The anchor’s encoding time ( $t_{enc}^A$ ), decoding time ( $t_{dec}^A$ ),

### Algorithm 1: Proposed Green Encoding Framework

---

**Input:**  $\mathcal{R}$ ,  $\mathcal{QP}$ ,  $\rho \in [0, 1]$   
**Output:** Selected representation ( $r^*$ ,  $qp^*$ )

// Step 1: Anchor Processing

- 1  $t_{enc}^A \leftarrow \text{Encode}(r_{min}, qp_{max})$
- 2  $t_{dec}^A, q^A \leftarrow \text{Decode}(r_{min}, qp_{max})$

// Step 2: Prediction Module

- 3  $\hat{E}_{enc} \leftarrow [], \hat{E}_{dec} \leftarrow [], \hat{Q} \leftarrow []$
- 4 **for**  $r \in \mathcal{R}$  **do**
- 5     **for**  $qp \in \mathcal{QP}$  **do**
- 6          $\hat{e}_{enc} \leftarrow f_{enc}(t_{enc}^A, r, qp)$
- 7          $\hat{e}_{dec} \leftarrow f_{dec}(t_{dec}^A, r, qp)$
- 8          $\hat{q} \leftarrow f_q(q^A, r, qp)$
- 9          $\hat{E} \leftarrow \hat{e}_{enc} + \hat{e}_{dec}$
- 10          $\hat{Q} \leftarrow \hat{q}$

// Step 3: Green Configuration Selection

- 11  $\hat{q}_{max} \leftarrow \text{Max}(\hat{Q})$
- 12  $\mathcal{F} \leftarrow \{(r, qp) \mid \hat{Q} \geq (1 - \rho) \cdot \hat{q}_{max}\}$
- 13  $(r^*, qp^*) \leftarrow \arg \min_{(r, qp) \in \mathcal{F}} \hat{E}$
- 14 **return** ( $r^*$ ,  $qp^*$ )

---

and quality metric ( $q^A$ ) are then measured (lines 1–2). Next, for each resolution  $r \in \mathcal{R}$  (line 4) and quantization parameter  $qp \in \mathcal{QP}$  (line 5), the encoding energy prediction model  $f_{enc}$  (line 6), the decoding energy prediction model  $f_{dec}$  (line 7), and the quality prediction model  $f_q$  (line 8) are invoked, yielding the predicted encoding energy ( $\hat{e}_{enc}$ ), decoding energy ( $\hat{e}_{dec}$ ), and quality ( $\hat{q}$ ) for each representation. The predicted energies are then aggregated into  $\hat{E}$  (line 9), while the quality predictions are stored in  $\hat{Q}$  (line 10). After computing the maximum obtainable quality  $\hat{q}_{max}$  (line 11), all representations whose predicted quality falls within the acceptable threshold defined by  $\rho$  are identified (line 12), and the one with the lowest predicted energy consumption is selected (lines 12–13). Finally, the representation ( $r^*$ ,  $qp^*$ ) that satisfies the quality constraint and minimizes energy consumption is returned (line 14). The time complexity of Algo. 1 is  $O(|\mathcal{R}| \times |\mathcal{QP}|)$ , where  $|\mathcal{R}|$  and  $|\mathcal{QP}|$  denote the number of target resolutions and quantization parameters, respectively (i.e., the number of target representations of the bitrate ladder).

## 5 Evaluation Setup

We conducted all experiments on a server with a 128-core Intel Xeon Gold CPU and two NVIDIA Quadro GV100 GPUs. The following subsections describe dataset characteristics and analysis, ML-based prediction models, and evaluation metrics.

### 5.1 Dataset analysis

We used 100 ultra-high-definition (UHD) video sequences with diverse spatiotemporal characteristics from the Inter4K dataset [21]. To verify that the selected subset is representative of the full Inter4K dataset, which contains 1000 sequences, we applied a Self-Organizing Map (SOM) [30] clustering on the content-complexity features ( $E_Y$ ,  $h$ ,  $L_Y$ ) using the Video Complexity Analyzer (VCA) tool [31] for both the full dataset and our subset. As shown in Fig. 4, the selected subset spans all clusters, confirming that it is representative of the overall dataset in terms of video complexity. Table 1 reports the statistical distribution of the content-complexity

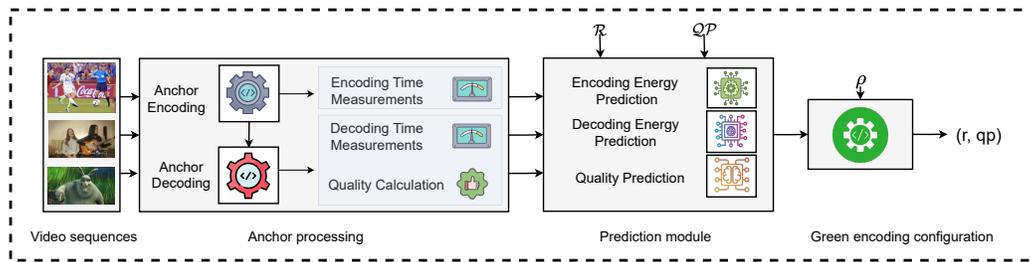


Figure 3: Proposed system architecture.

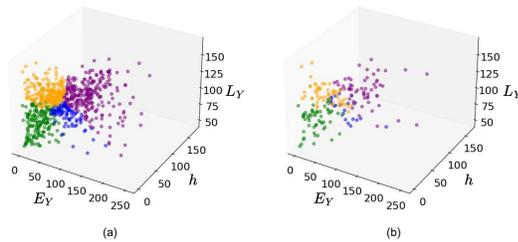


Figure 4: SOM-based clustering on the video complexity features on (a) the full dataset, and (b) our 100 subset.

Table 1: Statistical distribution of  $E_Y$ ,  $h$ , and  $L_Y$  across SOM clusters for the full dataset and our subset.

Cl.	Set	$E_Y$			$h$			$L_Y$		
		Avg.	min	max	Avg.	min	max	Avg.	min	max
0 (Purple)	Full	107.4	37.4	252.4	59.4	10.4	171.7	116.3	59.9	138.9
	Sub.	98.8	37.4	252.4	66.1	18.9	165.8	118.9	97.8	138.6
1 (Orange)	Full	41.1	2.7	90.4	25.4	1.9	72.6	123.7	99.2	166.9
	Sub.	48.6	8.8	113.6	20.5	2.6	45.3	125.2	111.0	166.9
2 (Blue)	Full	87.4	48.2	174.9	22.1	2.8	78.8	100.7	52.9	122.2
	Sub.	70.8	23.3	168.2	44.3	15.1	68.5	94.8	72.5	113.9
3 (Green)	Full	29.0	0.9	84.7	15.3	0.5	59.6	83.2	47.4	111.5
	Sub.	31.6	1.0	92.2	14.4	0.5	44.7	87.2	47.4	108.9

features ( $E_Y$ ,  $h$ ,  $L_Y$ ) across the SOM clusters for both the full dataset and our selected subset. While minor variations exist in individual cluster values (e.g., higher  $h$  and lower  $E_Y$  in Cluster 2 for the subset), the overall ranges and mean values remain consistent.

We encoded each video sequence at 60 fps using VVenC v1.11 [32] encoder with the faster preset [33]. The encoding configuration includes resolutions  $\mathcal{R} = \{360, 540, 720, 1080, 1440, 2160\}$ p and quantization parameters  $\mathcal{QP} = \{17, 22, 27, 32, 37, 42, 47\}$  [34]. The decoding process was applied using VVdeC v2.3.0 [35]. We recorded the encoding time, encoding energy consumption, decoding time, decoding energy consumption as well as quality scores measured by PSNR and VMAF. The energy consumption was measured with the CodeCarbon tool [36], which tracks the energy consumption of the underlying hardware using Running Average Power Limit (RAPL) [37] for the CPU and nvidia-ml-py [38] for the GPU.

Fig. 5 presents the impact of different  $\mathcal{R}$  on encoding and decoding energy, bitrate, PSNR, and VMAF. As expected, higher resolutions substantially increase both encoding and decoding energy, while improving video quality. Fig. 6 shows the variations in encoding and decoding time, bitrate, PSNR, and VMAF across different  $\mathcal{QP}$  levels. As  $\mathcal{QP}$  increases, encoding and decoding time and bitrate decrease, while quality metrics deteriorate accordingly.

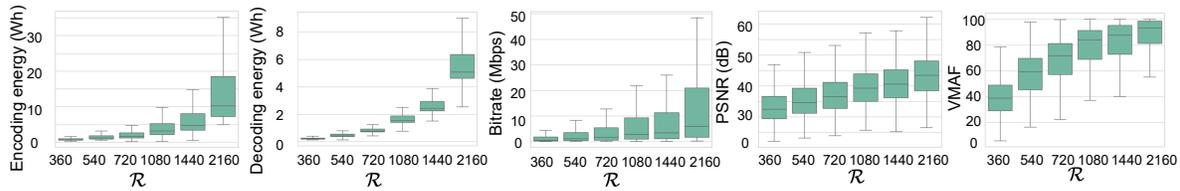
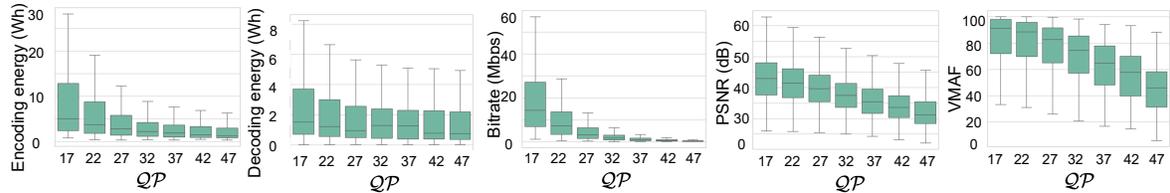
### 5.2 Prediction models

We evaluated six well-known ML models covering four different categories: (1) *Linear Regression* (LR) [39] and *Ridge Regression* (Ridge) [40] as linear models; (2) *Random Forest* (RF) [41] as a tree-based ensemble model; (3) *XGBoost* (XGB) [42] and *LightGBM* (LGBM) [43] as gradient boosting-based ensembles; (4) *Multi-Layer Perceptron* (MLP) [44], a neural network with fully connected layers.

We partitioned the dataset into training (70%) and testing (30%) sets at the video level, ensuring that all segments from a given video belong exclusively to one set, avoiding data leakage. We randomly shuffled video identifiers with a fixed seed, guaranteeing reproducibility and consistent splitting across the three prediction tasks. We applied GridSearchCV [45] from Scikit-learn [46] with five-fold cross-validation on the training set. We performed an exhaustive grid search over predefined hyperparameter spaces as summarized in Table 2. While the LR model has no tunable parameters, the Ridge model requires optimization of the regularization parameter  $\alpha$ . Tree-based and gradient boosting models required optimization of the number of estimators ( $n_{trees}$ ), maximum tree depth ( $d_{max}$ ), and learning rate ( $\eta$ ), along with model-specific parameters such as subsampling rates or the number of leaves. For the MLP model, we fine-tuned the number ( $h_{num}$ ) and size of hidden units ( $h_{size}$ ) and learning rate ( $\eta$ ).

### 5.3 Evaluation metrics

We use the following metrics to evaluate the accuracy and generalization performance of the prediction models: *Coefficient of determination* ( $R^2$ ) measures the proportion of variance in the ground truth explained by the model; a higher  $R^2$  indicates better predictive accuracy. *Mean absolute error* (MAE) measures the average relative prediction error; lower MAE indicates higher accuracy. *Root mean squared error* (RMSE) represents the square root of the average squared prediction error, penalizing large deviations more heavily; lower RMSE indicates better performance. *Standard deviation of absolute errors* (SDAE) measures the variability of absolute prediction errors; lower SDAE indicates more consistent predictions.


**Figure 5: Impact of resolution variations on encoding and decoding energy, bitrate, PSNR, and VMAF across 100 video sequences.**

**Figure 6: Impact of QP variations on encoding and decoding energy, bitrate, PSNR, and VMAF across 100 video sequences.**
**Table 2: Hyperparameter search space explored for ML-based prediction models.**

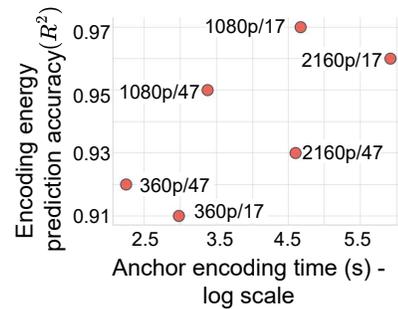
Predictive model	Explored hyperparameters
LR	None
Ridge	$\alpha \in \{0.1, 1.0, 10.0, 100.0\}$
RF	$n_{trees} \in \{50, 100, 200\}$ , $d_{max} \in \{\text{None}, 10, 20\}$ $min\_samples\_split \in \{2, 5\}$ $min\_samples\_leaf \in \{1, 2\}$
XGBoost	$n_{trees} \in \{50, 100, 200\}$ , $d_{max} \in \{3, 6, 9\}$ $\eta \in \{0.01, 0.1, 0.2\}$ , $subsample \in \{0.8, 1.0\}$
LightGBM	$n_{trees} \in \{50, 100, 200\}$ , $d_{max} \in \{3, 6, 9\}$ $\eta \in \{0.01, 0.1, 0.2\}$ , $num\_leaves \in \{31, 50, 100\}$
MLP	$h_{size} \in \{64, 128, 256\}$ , $h_{num} \in \{1, 2\}$ $\eta \in \{0.001, 0.01\}$

We also assess the green encoding configuration module via: *Energy savings* quantifies the reduction in average encoding and decoding energy consumption (Wh) compared to the highest-quality scenario ( $\rho = 0$ ). *Average quality* reports the average PSNR (dB) and VMAF scores across different encoding scenarios (varying  $\rho$ ). *Average quality drop* measures the reduction in PSNR (dB) and VMAF relative to the highest-quality scenario ( $\rho = 0$ ).

## 6 Evaluation Results

### 6.1 Anchor selection analysis

Fig. 7 shows the accuracy of energy prediction models, when a different resolution-QP pair was selected as an anchor. We evaluated six resolution-QP pairs to cover low (360p), medium (1080p), and high (2160p) resolutions, with the minimum (QP = 17) and maximum (QP = 47). The x-axis shows the average encoding time required for each anchor, averaged across 100 video sequences. The results show that higher  $R^2$  values are obtained at the expense of substantially longer encoding times (up to 372.48 s for 2160p/17), whereas the fastest configuration (9.48 s for 360p/47) still achieves reasonable accuracy ( $R^2 = 0.92$ ). These findings validate the rationale discussed in Section 3, confirming that the optimal anchor corresponds to the configuration with the lowest encoding time.


**Figure 7: Encoding energy prediction accuracy using different anchors.**

### 6.2 Prediction models analysis

We evaluated the performance of the candidate prediction models on four target metrics: encoding energy, decoding energy, PSNR, and VMAF. The reported results represent the average values across the entire test set.

- (1) Encoding energy prediction. Table 3 (Encoding Energy) shows that MLP achieved the highest  $R^2$  (0.91) and the lowest RMSE among all models, with  $h_{num} = 1$ ,  $h_{size} = 64$ , and  $\eta = 0.01$ .
- (2) Decoding energy prediction Table 3 (Decoding Energy) shows that RF, XGB, LGBM, and MLP have similar performance with  $R^2$  (0.95). We selected LGBM due to its slightly lower MAE (0.01). LGBM achieved its best performance with  $n_{leaves} = 50$ ,  $n_{estimators} = 50$ ,  $d_{max} = 3$  and  $\eta = 0.1$ .
- (3) PSNR prediction Table 3 (PSNR) shows that MLP and LGBM achieved the best overall performance, with the highest  $R^2$  (0.92) among all models. However, with slightly lower RMSE (1.93) and MAE (1.33), MLP with  $h_{num} = 2$ ,  $h_{size} = 256$ , 128 and  $\eta = 0.001$  was selected for PSNR prediction.

**Table 3: Prediction results for encoding energy, decoding energy, PSNR, and VMAF across all models.**

Model	Encoding Energy				Decoding Energy				PSNR				VMAF			
	$R^2 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	SDAE $\downarrow$	$R^2 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	SDAE $\downarrow$	$R^2 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	SDAE $\downarrow$	$R^2 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	SDAE $\downarrow$
LR	0.85	2.39	1.06	2.14	0.91	0.06	0.03	0.04	0.86	2.56	1.94	1.67	0.64	13.15	10.82	7.47
Ridge	0.84	2.39	1.06	2.14	0.91	0.06	0.03	0.04	0.86	2.54	1.93	1.64	0.66	12.74	10.67	6.95
RF	0.89	1.99	0.79	1.83	0.95	0.04	0.02	0.04	0.91	1.98	1.40	1.40	0.89	7.09	4.97	5.06
XGB	0.90	1.89	0.70	1.75	0.95	0.04	0.02	0.03	0.91	2.03	1.40	1.40	0.90	6.82	5.08	4.55
LGBM	0.90	1.88	0.74	1.73	0.95	0.04	0.01	0.03	0.92	1.94	1.36	1.37	0.90	6.77	5.04	4.52
MLP	<b>0.91</b>	<b>1.82</b>	<b>0.71</b>	<b>1.67</b>	0.95	0.04	0.02	0.04	<b>0.92</b>	<b>1.93</b>	<b>1.33</b>	<b>1.40</b>	<b>0.92</b>	<b>6.40</b>	<b>4.78</b>	<b>4.26</b>

**Table 4: Average VMAF and PSNR drops, and encoding, decoding energy savings for different  $\rho$  values.**

$\rho$	Avg. VMAF $\uparrow$	Avg. PSNR $\uparrow$	VMAF drop $\downarrow$	PSNR drop $\downarrow$	Enc. energy savings% $\uparrow$	Dec. energy savings% $\uparrow$
0	99.74	50.05	0	0	0	0
0.05	98.06	46.38	1.68	3.67	51.22	53.54
0.1	94.74	44.16	5	5.89	70.75	70.28
0.3	72.97	37.57	26.77	12.48	92.64	89.90
0.5	53.06	33.97	46.68	16.07	95.82	94.25
0.7	34.83	31.30	64.91	18.74	97.32	96.88
1	31.52	30.85	68.22	19.19	97.58	97.42

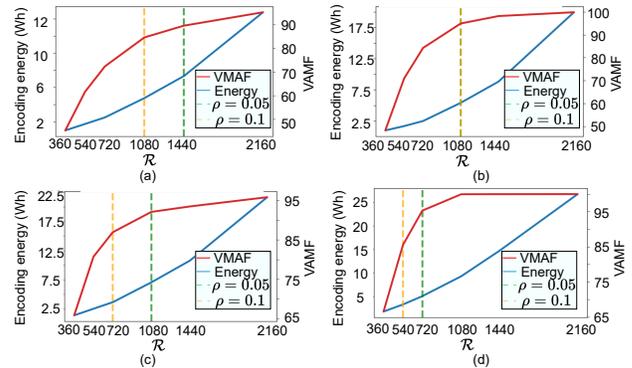
- (4) VMAF prediction Table 3 (VMAF) shows that MLP achieved the best overall performance, achieving the lowest RMSE (6.4) and MAE (4.78) and highest  $R^2$  (0.92) among all models with  $h_{num} = 2$ ,  $h_{size} = 256$ , 128 and  $\eta = 0.001$  and therefore was selected for VMAF prediction.

### 6.3 Green encoding configuration

Table 4 reports the average energy savings (encoding and decoding) and quality degradation (PSNR, VMAF) across the test video sequences for different  $\rho$  values, using the configuration with  $\rho = 0$  (no quality degradation) as the reference. We also report the average quality scores (PSNR and VMAF) corresponding to the selected encoding configurations.

For  $\rho = 0.05$ , i.e., allowing a 5% degradation in VMAF, the average quality loss remains minimal (1.68 VMAF points and 3.67dB PSNR) while achieving substantial energy saving of 51.22% in encoding and 53.54% in decoding. The minimum noticeable quality difference, referred to as the just-noticeable-difference (JND) for VMAF, has been reported to range between 2 and 6 in prior works [47–49], indicating that the observed quality loss is likely imperceptible. As  $\rho$  increases, the potential energy savings grow further but at the expense of more noticeable quality degradation. For instance, at  $\rho = 0.3$ , energy savings exceed 90%, whereas perceived quality declines by more than 25 VMAF points. These results indicate that a modest quality relaxation (e.g.,  $\rho = 0.05$ ) yields substantial energy savings while having a negligible impact on perceived visual quality.

Fig. 8 illustrates the impact of  $\rho = 0.05, 0.1$  values on four video sequences with different complexity levels. Each plot shows the changes in encoding energy consumption and VMAF across resolutions. The QP is fixed to allow visualization in a two-dimensional plane. The selected resolution differs for each sequence due to variations in content complexity, such as color dynamics, scene change frequency, motion intensity, and brightness. For example, under  $\rho = 0.05$ , sequence 72 (Fig. 8 (a)) at 1440p resolution achieves a 50.4% energy saving while maintaining acceptable quality compared to 2160p. Sequence 65 (Fig. 8 (b)) stays within 5% quality


**Figure 8: Impact of changing  $\rho$  values on video sequences: (a) sequence 72 ( $E_Y = 73.07$ ,  $h = 61.2$ ,  $L_Y = 122.84$ ;  $QP = 27$ ), (b) sequence 65 (41.61, 9.29, 90.10; 17), (c) sequence 23 (25.97, 28.37, 91.22; 22), (d) sequence 98 (37.39, 96.16, 109.35; 17).**

degradation at 1080p. Even when the degradation threshold is increased to 10% ( $\rho = 0.1$ ), the selected resolution remains 1080p, as lower resolutions would exceed the allowed quality loss. For sequence 23 (Fig. 8 (c)), the selected resolutions are 1080p and 720p for  $\rho = 0.05$  and 0.1, respectively. In contrast, for sequence 98 (Fig. 8 (d)), a lower resolution of 720p is sufficient to meet the quality constraint of  $\rho = 0.05$ , resulting in an 81.1% energy saving.

## 7 Conclusions

This paper presents a novel method for predicting energy consumption in the context of selecting configurations for green encoding by employing low-pass anchors. The proposed method involves assessing the encoding and decoding durations of the anchor encodings that are created at reduced resolutions and utilizes lightweight machine learning models to anticipate the energy usage and video quality across all other representations within a given sequence. By analyzing these predictions, the encoding configuration is meticulously chosen to strike a balance between energy efficiency and acceptable quality reduction. Our evaluation, conducted on a dataset comprising 100 video sequences from the Inter4K dataset, illustrates that restricting the decrease in the VMAF score to 5% results in substantial energy savings, specifically 51.22% in encoding energy and 53.54% in decoding energy, when compared to an encoding configuration selection that prioritizes maximum quality.

## References

- [1] AppLogic Networks, “The Global Internet Phenomena Report.” <https://www.applonetworks.com/phenomena>, 2024. Retrieved: 2025-10-10.

- [2] A. Bentalab, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP," in *IEEE Communications Surveys Tutorials*, vol. 21, pp. 562–585, 2019.
- [3] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP – Standards and Design Principles," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, p. 133–144, 2011.
- [4] Apple Inc., "HLS Authoring Specification for Apple Devices."
- [5] R. Farahani, A. Bentalab, C. Timmerer, M. Shojafar, R. Prodan, and H. Hellwagner, "SARENA: SFC-Enabled Architecture for Adaptive Video Streaming Applications," in *ICC 2023-IEEE International Conference on Communications*, IEEE, 2023.
- [6] R. Farahani, E. Çetinkaya, C. Timmerer, M. Shojafar, M. Ghanbari, and H. Hellwagner, "Alive: A Latency- and Cost-Aware Hybrid P2P-CDN Framework for Live Video Streaming," *IEEE Transactions on Network and Service Management*, 2023.
- [7] 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 (TCSVT)*, vol. 22, pp. 1649–1668, IEEE, 2012.
- [8] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the Versatile Video Coding (VVC) Standard and its Applications," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 31, no. 10, pp. 3736–3764, 2021.
- [9] A. Katsenou, J. Mao, and I. Mavromatis, "Energy-rate-quality Tradeoffs of State-of-the-art Video Codecs," in *Picture Coding Symposium (PCS)*, pp. 265–269, IEEE, 2022.
- [10] T. Chachou, W. Hamidouche, S. A. Fezza, and G. Belalem, "Energy Consumption and Carbon Emissions of Modern Software Video Encoders," *IEEE Consumer Electronics Magazine*, pp. 1–16, 2023.
- [11] R. Farahani, Z. Azimi, C. Timmerer, and R. Prodan, "Towards AI-assisted Sustainable Adaptive Video Streaming Systems: Tutorial and Survey," *arXiv preprint arXiv:2406.02302*, 2024.
- [12] R. Farahani, H. Amirpour, F. Tashtarian, A. Bentalab, C. Timmerer, H. Hellwagner, and R. Zimmermann, "RICHTER: Hybrid P2P-CDN Architecture for Low Latency Live Video Streaming," in *Proceedings of the 1st Mile-High Video Conference*, 2022.
- [13] Z. Azimi Ourimi, R. Farahani, V. V. Menon, C. Timmerer, and R. Prodan, "Towards ML-Driven Video Encoding Parameter Selection for Quality and Energy Optimization," *Accepted in 16th International Conference on Quality of Multimedia Experience (QoMEX)*, 2024.
- [14] A. Katsenou, V. V. Menon, A. Wiecekowski, B. Bross, and D. Marpe, "Decoding Complexity-Rate-Quality Pareto-Front for Adaptive VVC Streaming," in *IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pp. 1–5, 2024.
- [15] A. V. Katsenou *et al.*, "Content-agnostic Bitrate Ladder Prediction for Adaptive Video Streaming," in *Picture Coding Symposium (PCS)*, IEEE, 2019.
- [16] P. T. Rajendran, S. Afzal, V. V. Menon, and C. Timmerer, "Energy-Quality-Aware Variable Framerate Pareto-Front for Adaptive Video Streaming," in *IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pp. 1–5, 2024.
- [17] T. Huang *et al.*, "Deep Reinforced Bitrate Ladders for Adaptive Video Streaming," in *Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2021.
- [18] V. V. Menon, R. Farahani, P. T. Rajendran, M. Ghanbari, H. Hellwagner, and C. Timmerer, "Transcoding Quality Prediction for Adaptive Video Streaming," in *Proceedings of the 2nd Mile-High Video Conference*, 2023.
- [19] Alliance For Telecommunications Industry Solutions, "Objective Video Quality Measurement Using A Peak-Signal-to-Noise Ratio Full Reference Technique," in *T1.TR.74-2001*, 2001.
- [20] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. Cock, "VMAF: The Journey Continues," *Netflix Technology Blog*, vol. 25, no. 1, 2018.
- [21] A. Stergiou and R. Poppe, "Adapool: Exponential Adaptive Pooling for Information-retaining Downsampling," *IEEE Transactions on Image Processing*, vol. 32, pp. 251–266, 2022.
- [22] M. Ghasempour *et al.*, "Real-Time Quality-and Energy-Aware Bitrate Ladder Construction for Live Video Streaming," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2025.
- [23] Y. O. Sharrab and N. J. Sarhan, "Aggregate Power Consumption Modeling of Live Video Streaming Systems," in *Proceedings of the 4th ACM Multimedia Systems Conference (MMSys)*, pp. 60–71, 2013.
- [24] C. Herglotz, E. Walencik, and A. Kaup, "Estimating the HEVC Decoding Energy Using the Decoder Processing Time," in *IEEE International Symposium on Circuits and Systems*, pp. 513–516, IEEE, 2015.
- [25] B. O. Turkkun, T. Dai, A. Raman, T. Kosar, C. Chen, M. F. Bulut, J. Zola, and D. Sow, "GreenABR: Energy-Aware Adaptive Bitrate Streaming with Deep Reinforcement Learning," in *Proceedings of the 13th ACM Multimedia Systems Conference (MMSys)*, pp. 150–163, 2022.
- [26] R. Farahani, V. V. Menon, and C. Timmerer, "Machine Learning-Based Decoding Energy Modeling for VVC Streaming," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2671–2676, 2025.
- [27] P. Lebreton and K. Yamagishi, "Quitting Ratio-based Bitrate Ladder Selection Mechanism for Adaptive Bitrate Video Streaming," *IEEE Transactions on Multimedia (TMM)*, 2023.
- [28] M. Bhat *et al.*, "Combining Video Quality Metrics to Select Perceptually Accurate Resolution in a Wide Quality Range: A Case Study," in *IEEE International Conference on Image Processing (ICIP)*, IEEE, 2021.
- [29] Z. Azimi *et al.*, "Decoding Complexity-Aware Bitrate-Ladder Estimation for Adaptive VVC Streaming," in *32nd European Signal Processing Conference*, 2024.
- [30] T. Kohonen, "Self-Organizing Maps," *Springer Science & Business Media*, 2012.
- [31] V. V. Menon, C. Feldmann, K. Schoeffmann, M. Ghanbari, and C. Timmerer, "Green Video Complexity Analysis for Efficient Encoding in Adaptive Video Streaming," in *Proceedings of the First International Workshop on Green Multimedia Systems*, p. 16–18, 2023.
- [32] A. Wiecekowski, J. Brandenburg, T. Hinz, C. Bartnik, V. George, G. Hege, C. Helmrich, A. Henkel, C. Lehmann, C. Stoffers, *et al.*, "VVenC: An Open and Optimized VVC Encoder Implementation," in *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–2, IEEE, 2021.
- [33] J. Brandenburg, A. Wiecekowski, A. Henkel, B. Bross, and D. Marpe, "Pareto-optimized Coding Configurations for VVenC, a Fast and Efficient VVC Encoder," in *IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, IEEE, 2021.
- [34] J. Boyce, K. Suehring, X. Li, and V. Seregin, *JVET-J1010: JVET Common Test Conditions and Software Reference Configurations*. 2018.
- [35] A. Wiecekowski, G. Hege, C. Bartnik, C. Lehmann, C. Stoffers, B. Bross, and D. Marpe, "Towards a Live Software Decoder Implementation for the Upcoming Versatile Video Coding (VVC) Codec," in *IEEE International Conference on Image Processing (ICIP)*, pp. 3124–3128, IEEE, 2020.
- [36] BCG-GAMMA and MILA, "CodeCarbon." Retrieved: 2025-10-10.
- [37] "Running Average Power Limit Energy Reporting." Retrieved: 2025-10-10.
- [38] "Python bindings to the NVIDIA Management Library." Retrieved: 2025-10-10.
- [39] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. John Wiley & Sons, 2021.
- [40] G. C. McDonald, "Ridge Regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 1, pp. 93–100, 2009.
- [41] L. Breiman, "Random Forests," in *Machine Learning*, vol. 45, 2001.
- [42] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.
- [43] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *Advances in Neural Information Processing Systems*, 2017.
- [44] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer Perceptron and Neural Networks," *WSEAS Trans. on Circuits and Systems*, 2009.
- [45] "GridSearchCV." Retrieved: 2025-10-10.
- [46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine Learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [47] H. Amirpour, R. Schatz, and C. Timmerer, "Between Two and Six? Towards Correct Estimation of JND Step Sizes for VMAF-based Bitrate Laddering," in *14th International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–4, IEEE, 2022.
- [48] A. Kah, C. Friedrich, T. Rusert, C. Burgmair, W. Ruppel, and M. Narroschke, "Fundamental Relationships Between Subjective Quality, User Acceptance, and the VMAF Metric for a Quality-based Bit-rate Ladder Design for Over-the-top Video Streaming Services," in *Applications of Digital Image Processing XLIV*, vol. 11842, pp. 316–325, SPIE, 2021.
- [49] J. Ozer, "Finding the Just Noticeable Difference with Netflix VMAF," *Streaming Learning Center*, 2017.