Efficient Video Encoder Autotuning via Offline Bayesian Optimization and Supervised Learning

Roberto Azevedo
DisneyResearch|Studios
roberto.azevedo@disneyresearch.com

Yuanyi Xue
Disney Entertainment & ESPN Tech.
yuanyi.xue@disney.com

Xuewei Meng
Disney Entertainment & ESPN Tech.
xuewei.meng@disney.com

Wenhao Zhang
Disney Entertainment & ESPN Tech.
wenhao.zhang2@disney.com

Scott Labrozzi
Disney Entertainment & ESPN Tech.
scott.labrozzi@disney.com

Christopher Schroers
DisneyResearch|Studios
christopher.schroers@disneyresearch.com

Abstract*—Modern video encoders are complex software containing dozens of parameters, which allows them to be configured to different scenarios, requirements, or specific titles or scenes. Besides the number of parameters, the inter-dependency between them adds to the complexity of finding a per-title optimized combination of encoding parameters. Even though good practices in the industry have emerged, with the definition of presets per content type (e.g., film vs. cartoon), such practices are suboptimal for specific titles or scenes. Indeed, finding the best encoding parameters for a piece of content is currently a mix of best practices and trial-and-error artwork. We propose an efficient video encoder autotuner based on offline Bayesian optimization and supervised machine learning. Our proposal uses Bayesian optimization to search for a per-title best encoding parameter set offline to generate a dataset. Then, we use the generated dataset to train machine learning models that can map features extracted from the content to the best encoding parameters. Our experiments show that our generated dataset can find a combination of parameters that improves up to approximately -14.49% BD-Rate (0.77 BD-PSNR) and -11.59% BD-Rate (2.12 BD-VMAF) when optimizing for PSNR and VMAF, respectively. In comparison, our prediction models can recover ~80% of such performance while requiring only one fast encoding (compared to hundreds of encodes of a search optimization).

***Index Terms*—Video encoder, Encoding parameters, Bayesian Optimization, Deep Learning**

I. Introduction

Video is one of the most important media for communication and entertainment in today's digital world, dominating global internet traffic. Video compression standards [1]–[3] provide the key technologies that support the successful deployment of digital video. Modern video encoders have many parameters that can be tuned to specific scenarios (e.g., on-demand vs live video) or in a per-content/scene manner (e.g., cartoon vs live action content). Examples of encoding parameters include the number of reference frames, rate-distortion optimization mode, adaptive quantization mode and strength, number of B-frames, motion estimation range, deblocking filter strength, etc. However, finding the best encoding parameters for a specific content/scene is non-trivial.

A naive approach is to brute-force all the combinations of values for different parameters, encode the content with such combinations, and then choose the one with the best quality (according to a specific quality metric). The huge number of available parameters and the exponential combination of them, however, makes such an approach impractical. An improved method is using optimization methods (e.g., genetic algorithms [4] or Bayesian optimization [5]) to guide the search for the best encoding parameters per title. Sharma et al. [6] is an example of such a work that uses genetics algorithms to find the best encoding parameter for H.265/HEVC (High-Efficiency Video Coding) [7]. However, even though such approaches can provide an approximation of the optimum encoding parameter values per content/scene, they still require hundreds of encodings for each title during inference.

Brute-force-based approaches have also been proposed for bitrate-ladder construction for HTTP-based dynamic adaptive streaming (HAS) [8]–[10]. More recently, data-driven methods have been explored for such a problem [11]–[14]. In HAS, the video content is split into short segments. Each segment is then encoded at different resolutions and quality levels, which constitutes a bitrate-ladder. During streaming, based on network conditions, display resolution, etc., the client can dynamically decide which representation to download for each segment. One key issue of bitrate ladder optimization is to predict for each target bitrate which resolution provides the best quality. Such a problem can be seen as a specific case of video encoder parameter autotuning, in which resolution is the only parameter being selected.

In this paper, we propose a data-driven method that enables a significantly faster decision process than previous video encoder autotuning methods. Our proposal is based on i) generating an offline dataset through optimization methods (e.g., Bayesian optimization or genetics algorithms) and then ii) using such data to learn a model that predicts the best encoding parameter. The main goal is for the model to learn the best encoding parameters found on such a dataset and, by imitating such behavior, extrapolate to predict the best encoding parameters on new samples. Our proposal is also *non-invasive* since it does not need any change on the encoder. Compared to previous approaches, our proposed method allows for a better exploration of the search space (due to the per-title dynamic exploration of the space of parameters) and faster inference time (due to the learned prediction models).

We use H.264/AVC (Advanced Video Coding) [1], [15] as our target codec and evaluate our method on the PSNR (Peak-Signal-to-Noise Ratio) and VMAF (Video Multimethod Assessment Fusion) [16] quality metrics. However, our overall proposal is encoder-agnostic, and it can be easily applied to different encoders, encoder parameter sets, and target quality metrics. Experimental results show that our dataset generation method supports (without any changes on the encoder itself) an improvement of -14.49% BD-Rate (0.77 BD-PSNR) and -11.59% BD-Rate (2.12 BD-VMAF) when optimizing for PSNR and VMAF, respectively. Our prediction models can recover $\sim$80% of that performance with just one faster encoding process (compared to hundreds of encoding of optimization-based approaches).

II. Problem formulation

We consider the encoder as a function $\mathcal{E}$ that takes the frames of a video $\mathcal{V} = \{F_1, F_2, ..., F_n\}$, the specific encoding parameters $p = \{p_1, p_2, ..., p_p\}$, and the target bitrate b as input. The outputs of the encoder are a set of encoded frames $\mathcal{V}' = \{F'_1, F'_2, ...F'_n\}$, i.e.,

$$\mathcal{V}' = \mathcal{E}(\mathcal{V}, p, b). \tag{1}$$

Given the set of encoding parameters, the encoder tries its best to encode $\mathcal{V}$ into $\mathcal{V}'$ while keeping the final bitrate as close as possible to b. The final achieved quality and bitrate depend on the encoder heuristics themselves and how the user controls such heuristics based on p.

Given an objective quality metric $\mathcal{M}(\mathcal{V}, \mathcal{V}')$, finding the best set of encoding parameters can then be defined as,

$$\max_{p_1, p_2, ... \in P_1, P_2, ...} \mathcal{M}(\mathcal{V}, \mathcal{V}'), \tag{2}$$

where $\mathcal{V}'$ is given by Eq. (1). Common examples of $\mathcal{M}$ are PSNR, SSIM [17], and VMAF [16]. The goal of the above problem is to find the best parameters set $p \in \mathcal{P}$ that maximizes output quality produced by the encoder.

As aforementioned, a straightforward solution for the problem above is using optimization algorithms, e.g., genetic algorithms [6], simulated annealing [18], and Bayesian optimization [5]. Such approaches require hundreds of function evaluations to converge to the maximum solution. However, since the evaluation of the encoder function (i.e., encoding the content and compute the objective quality) is an expensive process, running such an optimization search approach per title/scene during inference time is prohibitive.

III. Proposed Method

We first use a Bayesian optimization-based approach to generate an offline dataset (Subsection III-A). The dataset is then used to train machine learning models (Subsection III-B) in a supervised way to approximate the best encoding parameters solution found in the offline dataset. Fig. 1 overviews the proposed method.

A. Dataset generation

For each video in the source dataset, we perform a Bayesian optimization-based approach to guide the search for the “best” encoding parameters for that video. Also, for each video, we extract features that can characterize its content. These features can later be used to predict the ground-truth encoding parameters found by the optimization search.

1) Bayesian Optimization: Bayesian optimization is an approach to optimizing objective functions that take a long time to evaluate. It uses the accumulated knowledge in the known area of the search space to guide sampling in the remaining area in an iterative process. For that, it builds a surrogate for the objective and quantifies the uncertainty in that surrogate using a Gaussian process regression, and then uses an acquisition function to decide where to sample. Bayesian optimization has been used in many optimization tasks when the function to be optimized needs to be treated as a black box, e.g., as hyperparameter search for deep learning [19] [20] or in parameter tuning of compilers [21].

Bayesian optimization is a good fit for our problem because it does not assume first or second-order derivatives, and thus can work with the encoder as a black-box function. However, our overall proposal is not dependent on Bayesian optimization and could perfectly work with other optimization search algorithms, e.g., genetics algorithms. One drawback of genetics algorithms, however, is that they require many more encode runs to converge when compared to Bayesian optimization.

For our implementation of Bayesian optimization, we assume that there is a default preset p_{def} that we want to improve upon and only consider samples that have a better quality than p_{def}. For each target bitrate b, each video sample goes through the above Bayesian optimization approach, being encoded a maximum of N times. Algorithm 1 details such a process.

Algorithm 1 Pseudo-code for the proposed Bayesian optimization-based dataset generation.

```
for all $\mathcal{V}$ in the source content dataset $\mathcal{D}$ do
    Place a Gaussian process prior on $f$
    $n \leftarrow 0$
    $p_{best}(\mathcal{V}) \leftarrow p_{def}$
    Observe $f$ at $p_{def}$ point
    while $n \leq N$ do
        Update the posterior probability distribution on $f$ using the gathered data so far
        Let $x_n$ be a maximizer of the acquisition function over $x$, where the acquisition function is computed using the current posterior distribution.
        Observe $y_n = f(x_n)$
        $n \leftarrow n + 1$
    end while
    Save the point evaluated with the maximum $f(x)$ as best encoding parameter for $\mathcal{V}$, $p_{best}(\mathcal{V})$
end for
```

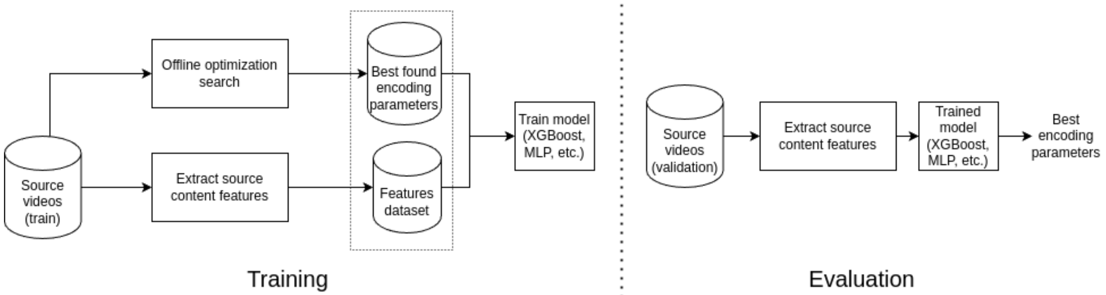
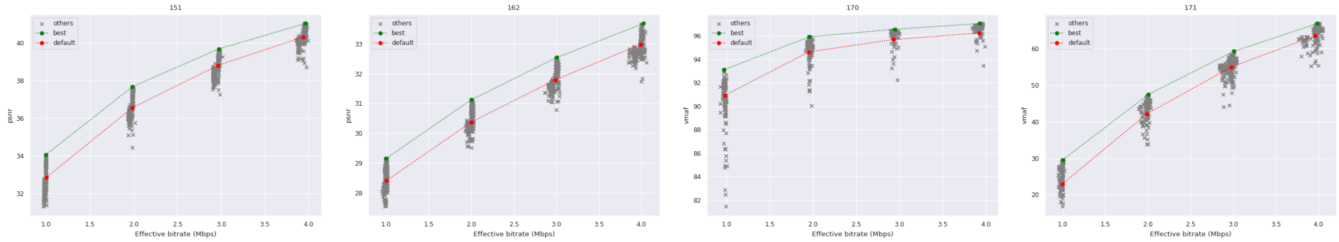

Offline optimization search
Best found encoding parameters
Source videos (train)
Extract source content features
Features dataset
Train model (XGBoost, MLP, etc.)
Training
Source videos (validation)
Extract source content features
Trained model (XGBoost, MLP, etc.)
Best encoding parameters
Evaluation

Fig. 1. Overview of the proposed approach.



(a) (b) (c) (d)

Fig. 2. Examples of the Bayesian optimization search performed for different samples from Inter4K optimized for PSNR (a)–(b) and VMAF (c)–(d).

2) Feature extraction: Aiming at predicting the best encoding parameters, we extract features from the video samples to characterize them and be used as input by machine learning algorithms. Specifically, we extract the following features:

a) Spatial Information (SI) and Temporal Information (TI): [1] SI is computed as the Root Mean Square (RMS) difference between the Sobel maps of each of the frames [22],

$$SI(v,u) = \sqrt{\frac{1}{w \times h} \sum_{i,j} |s_{ij}|^2}, \qquad (3)$$

where w and h are the width and height of the u and v frames and

$$s = S(v) - S(u), \qquad (4)$$

$$S(z) = \sqrt{(G_1 * z)^2 + (G_1^T * z)^2}, \qquad (5)$$

where $*$ denotes the 2-dimensional convolution operation, and G_1 is the vertical Sobel filter. TI is based on the motion between adjacent frames, $M_t(i,j)$, defined as the difference of the pixel luminance at the same location, at time t, i.e.,

$$M_t(i,j) = F_t(i,j) - F_{t-1}(i,j), \qquad (6)$$

where $F_t(i,j)$ is the pixel at the $(i,\, j)$ of the t-th frame. TI is computed as the maximum over time of the standard deviation over space of $M_n(i,j)$ over all i and j.

b) Energy-based Video Complexity Features: we compute the per-frame *average spatial energy* (E) and *average temporal energy* (h), following the definition of [23][2].

c) First pass features from x264: [3] Together with the above features, we also run a fast encode of x264 which allows us to extract the following features[4]: *Q*: Average of macroblocks QPs before adaptive quantization; *AQ*: average of macroblocks QPs after adaptive quantization decided by the rate control; *MV*: bits used by the motion vectors; *Tex*: Number of bits used by the texture component; and *Misc*: bits spend in other signalization, e.g., slice header and skip flags.

SI, TI, Energy-based Video Complexity, and 1st-pass features are computed per frame, and then statistics on those features (mean, standard deviation, minimum, and maximum) are computed for each video sample.

B. Prediction Models

Given a dataset that maps features to the best-found encoding parameters, machine learning methods can then be trained in a supervised way to predict such values. The goal is to learn a model that can learn $M_\theta(V) \approx p_{best}(\mathcal{V})$ for any $\mathcal{V}$, where θ are the model's parameters. Two main approaches are possible: *classification* or *regression*. From a small number of combinations, it is straightforward to train a classification model. However, this limits the approach to a pre-defined number of presets. Since our found best encoding parameters are not pre-defined, i.e., the values are dynamically chosen based on the optimization search approach, we opt for using a regression approach. In our experiments, we focused mainly on XGBoost and Multi-Layer Perceptron (MLP) models, but our general proposal is not restricted to them. We also experimented with SVM (Support Vector Machines) and Random

[1] https://github.com/Telecommunication-Telemedia-Assessment/SITI
[2] https://github.com/cd-athena/VCA
[3] https://www.videolan.org/developers/x264.html
[4] A more detailed description of those features can be found at the x264 documentation.

Forest models, but omitted these results here since XGBoost and MLP consistently performed better in our experiments.

IV. Experiments

A. Datasets generation and analysis

We experiment with the freely available dataset **Inter4k**[5], which is composed of one thousand 4k videos of 5 seconds duration each. We downsampled all the sample videos to 1920x1080 resolution and used that as our source dataset for the following experiments. This source dataset is named **Inter4K-HD** in the rest of this document. We generated three versions of this initial dataset, which we named **Inter4K-HD/PSNR**, **Inter4K-HD/VMAF**, and **Inter4K-HD/VMAF-MultiRes**. **Inter4K-HD/PSNR** and **Inter4K-HD/VMAF** use, respectively, PSNR and VMAF as the target metric for the Bayesian optimization discussed in Section III, whereas **Inter4K/VMAF-MultiRes** is similar to **Inter4K/VMAF** but also allows to configure the resolution of the output video as an additional encoding parameter.

For all the three dataset variants above, we focus on H.264/AVC as our target codec, using the *very-slow* preset from x264 as our default preset p_{def}. Table I details the range of encoding parameters and the default values used in the Bayesian optimization. The "resolution" parameter is only used for **Inter4k-HD/VMAF-MultiRes** dataset.

TABLE I
H.264/AVC Encoding parameters and ranges used during Bayesian optimization for our dataset generation.

Parameter	Range	Default
aq_mode	$(0, 2)$	1
aq_strength	$(0, 1.0)$	1.0
bframes	$(0, 16)$	3
deblock_alpha	$(-6.0, 6.0)$	-1.0
deblock_beta	$(-6.0, 6.0)$	-1.0
ipratio	$(0, 1.6)$	1.4
mbtree	$(0, 1)$	1
merange	$(4, 32)$	16
qcomp	$(0, 1.0)$	0.6
ref	$(1, 16)$	4
subme	$(1, 8)$	7
target-bitrate	(fixed)	1Mbps–5Mbps
max-rate	(fixed)	(1.5×target-bitrate)
bufsize	(fixed)	(2.0×target-bitrate)
psy-rd	(fixed)	1
psy-trellis	(fixed)	0.15
VMAF-MultiRes Only		
resolution	{1080p, 720p, 540p, 360p}	1080p

For each target bitrate in our dataset (1Mbps, 2Mbps, 3Mbps, 4Mbps, and 5Mbps), each video sample goes through the above Bayesian optimization approach, being encoded a maximum of $N = 200$ times. In total, for each version of the Inter4K-HD dataset, we generate 5 target bitrate × 200 encodings × 1000 videos, i.e., 1 million unique encodes. Finally, for each video and target bitrate, we selected the best metric found from these encodings as the ground truth of our offline datasets, following Algorithm 1. For illustrative purposes, Fig. 2 shows examples of performing our Bayesian optimization approach for sample titles in the dataset, comparing the performance of the default preset to the best encoding parameter found during the optimization.

Table II and Fig 3(a) show the statistics of the **Inter4K-HD/PSNR**, in which, we can find a parameter set that provides up to +0.91 PSNR in the low bitrate regime compared to the default preset. Table III and Figs 3(b) show the statistics of the **Inter4K-HD/VMAF**, in which, we find a parameter set supporting up to +4.70 VMAF scores on average in the lower bitrate regime when compared to the default preset. Finally, Table IV and Fig 3(c) show the statistics of the **Inter4K-HD/VMAF-MultiRes** dataset.

TABLE II
Inter4K-HD/PSNR dataset statistics. Values are reported in the format: "Avg. PSNR (standard deviation)".

Bitrate	Inter4K-HD/PSNR Default	 Best	 Avg. $\triangle$PSNR
1Mbps	33.56 (5.76)	34.47 (5.88)	+0.91 (2.56)
2Mbps	37.05 (5.82)	37.81 (5.91)	+0.76 (0.48)
3Mbps	39.02 (5.81)	39.70 (5.89)	+0.68 (0.44)
4Mbps	40.07 (5.25)	40.70 (5.36)	+0.64 (0.44)
5Mbps	41.45 (5.75)	42.06 (5.86)	+0.61 (0.45)

TABLE III
Inter4k-HD/VMAF dataset statistics. Values are reported in the format: "Avg. VMAF (standard deviation)".

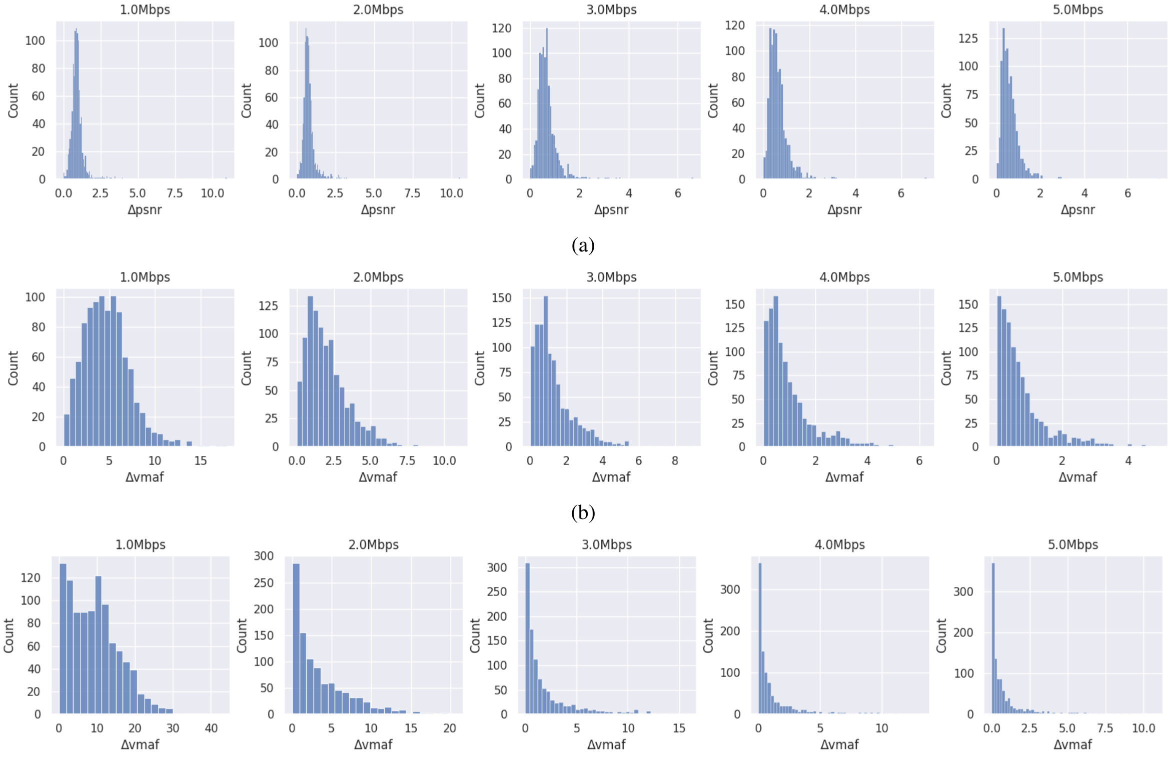
Bitrate	Inter4K-HD/VMAF Default	 Best	 Avg. $\triangle$VMAF
1Mbps	60.65 (18.23)	65.35 (16.66)	+4.70 (2.56)
2Mbps	79.19 (13.13)	81.25 (12.36)	+2.06 (1.50)
3Mbps	86.72 (10.21)	88.02 (9.67)	+1.30 (1.14)
4Mbps	90.69 (8.25)	91.64 (7.83)	+0.95 (0.93)
5Mbps	93.06 (6.82)	93.82 (6.49)	+0.75 (0.78)

TABLE IV
Inter4k-HD/VMAF-MultiRes dataset statistics .Values are reported in the format: "Avg. VMAF (standard deviation)".

Bitrate	Inter4K-HD/VMAF Default	 Best	 Avg. $\triangle$VMAF
1Mbps	60.65 (18.23)	70.01 (13.86)	+9.37 (2.56)
2Mbps	79.19 (13.13)	82.75 (10.69)	+3.57 (1.50)
3Mbps	86.72 (10.21)	88.59 (8.59)	+2.57 (1.14)
4Mbps	90.69 (8.25)	91.90 (7.05)	+1.21 (0.93)
5Mbps	93.06 (6.82)	93.97 (5.88)	+0.91 (0.78)

Our results show that the improvements supported by the Bayesian optimization are inversely proportional to the target bitrate, which is expected since at lower bitrates choosing the encoding parameters carefully is more important. Also, when comparing the results from Table III and Table IV it is clear the significant VMAF improvements on lower target bitrates in **Inter4K-HD/VMAF-MultiRes**. Such improvements comes from the ability to choose a lower resolution to compensate for higher compression artifacts. (see Fig. 4)

[5] https://alexandrosstergiou.github.io/datasets/Inter4K/



(a)

(b)

(c)

Fig. 3. Inter4K-HD/H.264 dataset distribution when optimizing for (a) PSNR, (b) VMAF, and (c) VMAF/MultiRes.

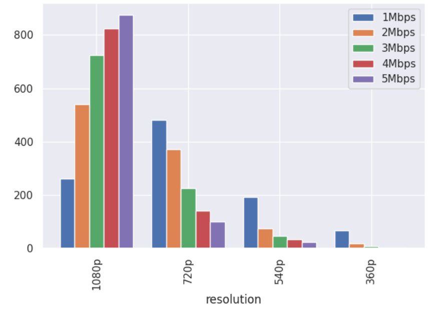


Fig. 4. Histogram of best chosen resolution for Inter4K-HD/VMAF-MultiRes.

Finally, Fig. 5 depicts the average rate-distortion curve of the default preset and the optimal per-content found for each of the three dataset variants, while the first row of Table V shows the improvement in terms of BD-Rate and BD-Metric.

B. Prediction model results

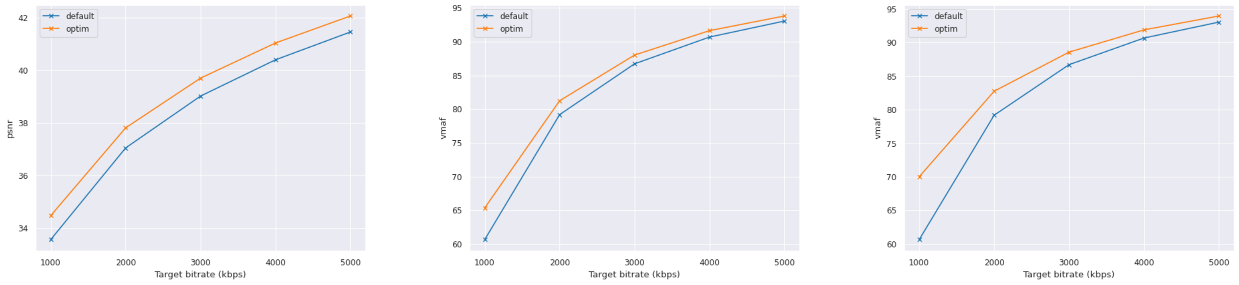
We independently trained prediction models on the different dataset variants: **Inter4K-HD/PSNR**, **Inter4K-HD/VMAF**, and **Inter4K-HD/VMAF-MultiRes**. The data was split in 80% for training and 20% for validation and the same split was used for all the dataset variants on the results presented below. When splitting the dataset into train/test, we make sure that for a given content, all the target bitrate data will appear only either in the training set or in the test set. Thus, we guarantee that our tests are only performed in video content that was not seen during training. All the features extracted from the source content and the encoding parameters are min/max normalized. For the XGBoost model, we use the default python xgboost library[6] with $max\_depth = 0$, while the MLP is composed of 5 layers, each with 512 neurons. For the MLP training, we use an adaptive learning rating starting at 1×10^{-3} being divided by 5 every time that 2 consecutive epochs fail to decrease the loss function, until the tolerance of 1×10^{-7}.

Table V shows the BD-Rate and BD-PSNR/VMAF for evaluating the different trained models (XGBoost and MLP) on the three dataset variants we generated. It also reports the BD-Rate and BD-PSNR/VMAF computed on the whole dataset and such metrics computed only on the test set. It is expected that the upper bound of the prediction model reported metrics are the ones of “Best (test set)”, while the “Best (full dataset)” is kept just for reference. From the results, it is clear that the prediction models are able to recover most of the performance of the search optimization, while requiring just one fast encoding and feature extraction step.

V. Conclusion

We introduce a video encoder autotuning framework that takes advantage of Bayesian optimization to search the space of encoder parameters and build an offline dataset which is then used to train supervised machine learning methods. Our method supports an automated and efficient search of encoding parameters while offering better performance than previous

[6] https://github.com/dmlc/xgboost



(a) (b) (c)

Fig. 5. Inter4K-HD rate-distortion curves when optimizing for PSNR (a), VMAF (b), and VMAF/MultiRes (c).

TABLE V

BD-RATE/METRIC COMPARING THE BEST IN THE DATASET AND PREDICTED PARAMETERS. DEFAULT PRESET (X264, VERY SLOW) IS USED AS ANCHOR TO COMPUTE BD-RATE/METRIC.

	Inter4k-HD/PSNR		**Inter4k-HD/VMAF**		**Inter4k-HD/VMAF-MultiRes)**	
Model	**BD-Rate ↓**	**BD-PSNR ↑**	**BD-Rate ↓**	**BD-VMAF ↑**	**BD-Rate ↓**	**BD-VMAF ↑**
Best (full dataset)	-14.49	0.77	-11.60	2.32	-16.25	3.76
Best (test set)	-13.47	0.71	-11.56	2.08	-16.27	3.70
XGBoost	-10.89 (80.8%)	0.56 (78.9%)	-9.21 (79.7%)	1.64 (78.8%)	-13.25 (81.4%)	3.27 (88.4%)
MLP	-10.98 (81.5%)	0.58 (81.7%)	-8.73 (75.5%)	1.57 (75.5%)	-13.44 (82.6%)	3.17 (85.7%)

fixed parameter search methods. Moreover, after training, our method provides an efficient solution, which only requires one fast encoding of the content plus a pass of feature extraction. Specifically, we demonstrate that using x264, we are able to find a parameter set that achieves up to 14.49% and 11.59% BD-Rate reduction compared to very-slow encoding parameter preset when optimizing for PSNR and VMAF, respectively, and can recover $\sim 80\%$ of such performance with a much more efficient prediction model. Our proposed framework also opens up new avenues for future work. Although we focus only on simple hand-designed features and more traditional machine learning algorithms in our experiments, more advanced features (e.g., deep learning-based ones) and models (e.g., Transformers) can be easily integrated into our framework. The experimentation of our method with other codecs (e.g., HEVC and AV1) is another interesting future work.

REFERENCES

[1] T. Wiegand, G. J. Sullivan *et al.*, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[2] B. Bross, Y.-K. Wang *et al.*, "Overview of the Versatile Video Coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[3] J. Han, B. Li *et al.*, "A technical overview of AV1," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1435–1462, 2021.

[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[5] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[6] R. R. Sharma and K. V. Arya, "Parameter optimization for HEVC/H.265 encoder using multi-objective optimization technique," in *2016 11th International Conference on Industrial and Information Systems (ICIIS)*. Roorkee, India: IEEE, Dec. 2016, pp. 592–597.

[7] G. J. Sullivan, J.-R. Ohm *et al.*, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[8] A. Aaron, Z. Li *et al.*, "Per-title encode optimization," Tech. Rep., 2015. [Online]. Available: https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2

[9] K. S. Durbha, H. Tmar *et al.*, "Bitrate ladder construction using visual information fidelity," *arXiv preprint arXiv:2312.07780*, 2023.

[10] A. Telili, W. Hamidouche *et al.*, "Bitrate ladder prediction methods for adaptive video streaming: A review and benchmark," *arXiv preprint arXiv:2310.15163*, 2023.

[11] A. V. Katsenou, J. Sole, and D. R. Bull, "Efficient bitrate ladder construction for content-optimized adaptive video streaming," *IEEE Open Journal of Signal Processing*, vol. 2, pp. 496–511, 2021.

[12] F. Nasiri, W. Hamidouche *et al.*, "Multi-preset video encoder bitrate ladder prediction," ser. ViSNext '22. New York, NY, USA: ACM, 2022, p. 8–13.

[13] A. Telili, W. Hamidouche *et al.*, "Efficient per-shot transformer-based bitrate ladder prediction for adaptive video streaming," in *2023 IEEE ICIP*. IEEE, 2023, pp. 1835–1839.

[14] J. Yang, M. Guo *et al.*, "Optimal transcoding resolution prediction for efficient per-title bitrate ladder estimation," *arXiv preprint arXiv:2401.04405*, 2024.

[15] "Recommendation itu-t h.264. advanced video coding for generic audiovisual services," 2021.

[16] "VMAF: Video multimethod assessment fusion." [Online]. Available: https://github.com/Netflix/vmaf

[17] Z. Wang, A. C. Bovik *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[18] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical science*, vol. 8, no. 1, pp. 10–15, 1993.

[19] J. Wu, X.-Y. Chen *et al.*, "Hyperparameter optimization for machine learning models based on bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019.

[20] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using bayesian optimization," *Evolving Systems*, vol. 12, no. 1, pp. 217–223, 2021.

[21] A. H. Ashouri, G. Mariani *et al.*, "Cobayn: Compiler autotuning framework using bayesian networks," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 13, no. 2, pp. 1–25, 2016.

[22] "Recommendation p.910 : Subjective video quality assessment methods for multimedia applications," 2023.

[23] V. V. Menon, C. Feldmann *et al.*, "Vca: video complexity analyzer," in *Proceedings of the 13th ACM Multimedia Systems Conference (MMSys '22)*. New York, NY, USA: ACM, 2022, p. 259–264.