IEEE Open Journal of
Circuits and Systems

Received 23 June 2021; revised 29 September 2021; accepted 26 October 2021. Date of current version 22 November 2021.

Digital Object Identifier 10.1109/OJCAS.2021.3124995

Rate-Distortion Optimized Encoding for Deep Image Compression

MICHAEL SCHÄFER[1], SOPHIE PIENTKA[1], JONATHAN PFAFF[1], HEIKO SCHWARZ[1,2,3], DETLEV MARPE[1] (Fellow, IEEE), AND THOMAS WIEGAND[1,2,4] (Fellow, IEEE)

[1]Video Communication and Applications Department, Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, 10587 Berlin, Germany

[2]Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, 10587 Berlin, Germany

[3]Department of Mathematics and Computer Science, Free University of Berlin, 14195 Berlin, Germany

[4]Department of Electrical Engineering and Computer Science, Berlin Institute of Technology, 10623 Berlin, Germany

This article was recommended by Associate Editor M. Cagnazzo.

CORRESPONDING AUTHOR: M. SCHÄFER (e-mail: michael.schaefer@hhi.fraunhofer.de)

ABSTRACT Deep-learned variational auto-encoders (VAE) have shown remarkable capabilities for lossy image compression. These neural networks typically employ non-linear convolutional layers for finding a compressible representation of the input image. Advanced techniques such as vector quantization, context-adaptive arithmetic coding and variable-rate compression have been implemented in these auto-encoders. Notably, these networks rely on an end-to-end approach, which fundamentally differs from hybrid, block-based video coding systems. Therefore, signal-dependent encoder optimizations have not been thoroughly investigated for VAEs yet. However, rate-distortion optimized encoding heavily determines the compression performance of state-of-the-art video codecs. Designing such optimizations for non-linear, multi-layered networks requires to understand the relationship between the quantization, the bit allocation of the features and the distortion. Therefore, this paper examines the rate-distortion performance of a variable-rate VAE. In particular, one demonstrates that the trained encoder network typically finds features with a near-optimal bit allocation across the channels. Furthermore, one approximates the relationship between distortion and quantization by a higher-order polynomial, whose coefficients can be robustly estimated. Based on these considerations, the authors investigate an encoding algorithm for the Lagrange optimization, which significantly improves the coding efficiency.

INDEX TERMS Deep image compression, variational auto-encoders, rate-distortion optimized encoding, non-linear transform coding.

I. INTRODUCTION

THERE is a seemingly endless variety of multimedia communication in today's world. Tailor-made solutions for high-resolution streaming, video conferencing and the storage of digital images have become easily accessible to both consumers and companies. Invisible to most users, lossy compression is at the heart of these applications because bandwidth capacities are a limiting constraint. Therefore, image and video coding technologies are capable of efficiently compressing the content at the cost of transmitting a slightly distorted version of the original.

Cutting-edge video codecs like High Efficiency Video Coding (HEVC) [1], [2], [3] and Versatile Video Coding (VVC) [4], [5], [6] employ a hybrid, block-based approach for this task. First, each frame is partitioned into blocks and for each block, a prediction signal is generated by using either intra-picture prediction or motion-compensated prediction. The prediction residual is then transformed, quantized and entropy-coded using adaptive context models.

A key to exploit the compression efficiency of hybrid video coding systems is rate-distortion optimized encoding; [7], [8]. Given a maximum budget of bits R^*, the encoder evaluates the impact of different coding decisions

such as block partitions, prediction modes and transform coefficient levels on the resulting distortion d. These coding decisions are encoded in a bitstream with bitrate R. The goal is to find a bitstream, which transmits the input image with minimal distortion and below the given rate budget R^*. Directly solving this minimization task is unfeasible, due to the sheer number of possible combinations of coding decisions. Instead, the encoder evaluates promising options by comparing their Lagrangian cost

$$\min(d + \lambda R), \tag{1}$$

where $\lambda > 0$ is the Lagrange parameter, which depends on R^*; [9]. In general, the quantization has a strong impact on the Lagrangian cost (1). For instance, the impact of different quantizers on d and R is well-known for linear, orthogonal transforms; [10], [11]. Consequently, there are several algorithms for selecting the quantization indices of scalar quantization of a transform block; [12], [13]. Furthermore, the performance capabilities of rate-distortion optimized quantization for HEVC have been investigated in [14]. In VVC, the encoder may choose between scalar quantization of transform coefficients and a low-complexity variant of vector quantization, which yields additional coding efficiency improvements; [15], [16]. Here, two different quantizers are applicable and the possible transitions between them are represented by a trellis. A suitable sequence of quantization indices can be found by applying the Viterbi algorithm [17]. In summary, the signal-dependent Lagrange optimization on the encoder side is crucial to the performance of image and video coding.

Besides the aforementioned existing video codecs, there have been promising advances in using variational auto-encoders (VAE) for end-to-end still image compression. These VAEs can be characterized as non-linear transform coding and are typically designed as artificial neural networks, which are trained on large sets of data; [18], [19]. Ballé *et al.* have made several contributions concerning the overall architecture, the choice of activation functions, and the design of a proper entropy model for the quantized features; [20], [21], [22]. Furthermore, Agustsson *et al.* investigate the effects of using soft quantizers on the performance of image compression networks; [23]. Different conditional probability models for the features have been investigated in [24], [25], [26], where the latter two ones are closely related to the works of Ballé. Remarkably, these auto-encoders are able to keep up with the compression efficiency of HEVC in an RGB-setting for still image compression. Subsequently, the authors of [27] have built a compression system using multi-scale convolutions, thereby separating the features into high and low frequency parts. In [28], the effectiveness of using trellis-coded quantization over uniform scalar quantization is demonstrated for deep-learned image compression. In [29], [30], the authors trained modulating networks for scaling the features according to the Lagrange parameter. These auto-encoders are capable of achieving multiple points on the operational rate-distortion curve.

II. MOTIVATION

In the context of deep-learned image compression systems, signal-dependent encoder optimizations have not been deeply researched yet. The authors of [31] demonstrated that the coding efficiency of a VAE is improved significantly by considering the quantized features as possible coding options and exhaustively testing different quantization levels. Still, a major obstacle is understanding the impact of the quantization due to the multi-layered, non-linear, non-orthogonal decoder network. Furthermore, given the decoder network and the original image, the non-quantized features do not necessarily minimize the distortion in the sample domain, not even locally. This is a major contrast to the orthogonal transforms, which are used in conventional video codecs. Here, the distortion in the sample domain is equal to the quantization error of the transform coefficients and the original samples can be recovered from the non-quantized coefficients, disregarding perturbations due to fixed-point arithmetics. Hence, the aim of this paper is to consider possible approaches for rate-distortion optimized encoding in deep-learned end-to-end compression systems. First, one introduces a network architecture, which is capable of achieving different rate-distortion trade-offs by suitably adapting the quantization step size. During the training stage, the network parameters are updated with respect to different Lagrange costs simultaneously. Here, one ensures that the encoder-determined features are capable of almost perfectly reconstructing the original image by optimizing with respect to a sufficiently small Lagrange parameter. This proves to be critical for designing a rate-distortion optimization algorithm: when the non-quantized features are close to local minimum of the distortion, the impact of altering the quantization levels of the features can be reasonably estimated by a higher-order polynomial. Furthermore, by avoiding the use of an auto-regressive network at the arithmetic coding stage, changing the quantization index of a single feature entry does not alter the probability estimation of the remaining ones which is beneficial for maneuvering the allocation of bits across the feature channels. Given such a variational auto-encoder, the authors demonstrate that the encoder-determined features typically inherit a near-optimal allocation of bits across the channels for high rates. From these considerations, one derives an algorithm for refining the encoder-determined features and effectively reducing the Lagrangian cost (1).

The rest of the paper is organized as follows. Section III presents the auto-encoder architecture and the training details. The following Section IV investigates the relationship between the quantization, bitrate and distortion. Using this information, the authors propose an algorithm for rate-distortion optimized encoding. Section V provides

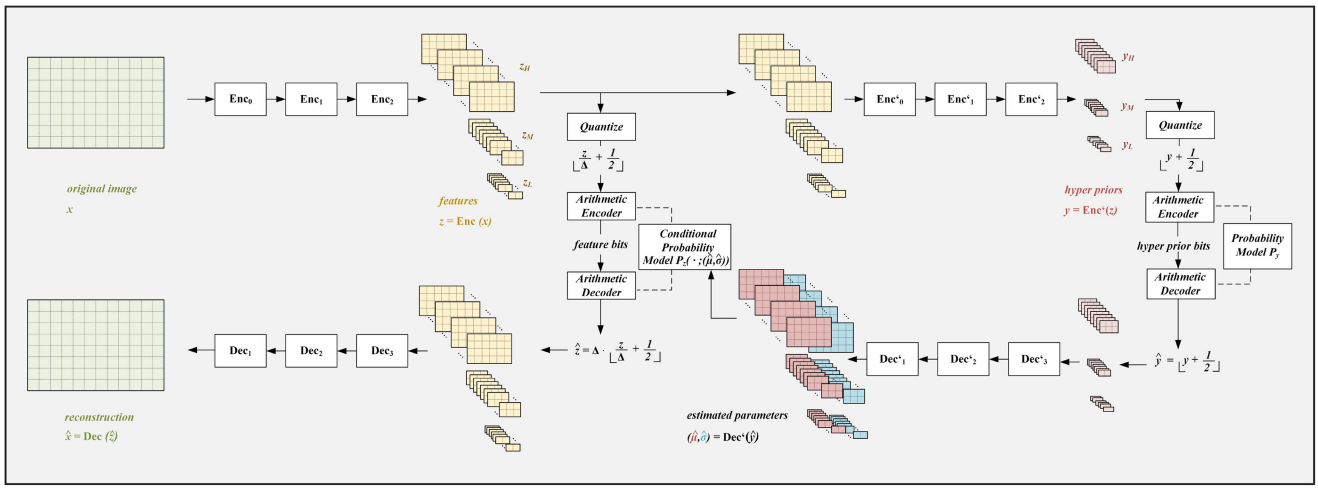


FIGURE 1. Flow chart of the proposed VAE-based image coding process. Note that each of the networks in (2) and (3) consists of three multi-scale convolutional layers. The subscripts *H*, *M*, *L* refer to high, middle and low resolution components of the latents *z* and *y*. Here, the quantization step size Δ of the features is variable while the hyper priors are rounded off to the closest integer. The parameters $(\hat{\mu}, \hat{\sigma})$ of the feature probability model are estimated from the transmitted side information.

the experimental results. Finally, the findings are discussed in Section VI the paper concludes with Section VII.

III. ARCHITECTURE AND TRAINING FOR VARIABLE-RATE END-TO-END IMAGE COMPRESSION

Conventional video codecs are capable of quality-scaling: given a fixed decoder and bitstream syntax, the encoder is able to produce bitstreams for specific target bitrates and reconstruction qualities. Contrarily, most trained end-to-end compression systems have different sets of coefficients for each point on the operational rate-distortion curve. Another problem concerns how the encoder determines the features, especially when the network is trained with respect to a low reconstruction quality. Given the decoder of such a compression system, the encoder-determined, non-quantized features typically do not minimize the distortion with respect to the input image. The impact of altering the quantization levels is hard to estimate without knowing if the current features are far or close to such a distortion minimum. As a consequence, it becomes difficult to design a suitable algorithm for rate-distortion optimized encoding. The authors avoid this problem by using a suitable rate-distortion loss term during the training of the proposed auto-encoder. For simplicity, the authors restrict themselves to using luma-only input images for which the computation of the bitrate and distortion is unambiguous.

A. OVERVIEW OF THE IMAGE COMPRESSION PROCESS

Variational auto-encoders for image compression are composed of different convolutional neural networks where each one carries out a specific task of the conventional coding process. First, the network Enc encodes the input $x \in \mathbb{R}^{H \times W \times 1}$ into a set of features to transmit. After uniformly quantizing each entry with step size $\Delta > 0$, the resulting symbols are written into the bitstream. At the decoder side, the symbols are parsed and another network Dec reconstructs the image as

$$\left.\begin{aligned} z &= \mathrm{Enc}(x), \\ \hat{z}(\Delta) &= \Delta \cdot \left\lfloor \frac{z}{\Delta} + \frac{1}{2} \right\rfloor, \\ \hat{x} &= \mathrm{Dec}(\hat{z}). \end{aligned}\right\} \tag{2}$$

For coding the symbols, one defines a parametrized entropy model by assuming a normal distribution of the features $z \sim \mathcal{N}(\mu, \sigma^2)$. Thus, a hyper system with its own encoder Enc′ extracts side information from the features, which is also transmitted. Prior to decoding the quantized features, the hyper decoder Dec′ determines the necessary estimates $(\hat{\mu}, \hat{\sigma})$ as

$$\left.\begin{aligned} y &= \mathrm{Enc}'(z), \\ \hat{y} &= \lfloor y + \tfrac{1}{2} \rfloor, \\ \left(\hat{\mu}, \hat{\sigma}\right) &= \mathrm{Dec}'(\hat{y}). \end{aligned}\right\} \tag{3}$$

The entire image coding process is depicted in Figure 1.

An important aspect of this work is the use of multi-scale convolutional layers for representing the features and the side information at different resolutions; see [31]. The authors of [27] have demonstrated the positive impact of octave convolutions instead of regular ones on the compression efficiency of a VAE-based image codec. The benefit of this representation is that differently scaled components of the features are capable of capturing image details at different resolutions of the input. Furthermore, when a variable but uniform quantization step size is used, its impact on the individual components differs, depending on the magnitude of the feature entries in each component. Other approaches for variable-rate compression such as [29] use an additional neural network which scales each individual feature entry depending on the target bitrate. Hence, one also aims at exploiting the multi-scale representation of the features for avoiding the use of such a re-scaling network.

Concerning the conditional probability model for coding the features, we avoid using an auto-regressive network as in [25] and [31]. This simplification is beneficial for controlling the impact of the quantization on the number of bits to be transmitted. Note that the individual feature entries are assumed to be normally distributed and stochastically independent. Given that the side information is fixed, changing the quantization index of a single entry does not affect the estimated probabilities of the remaining features. As a consequence, the features can be decoded in an arbitrarily chosen but fixed scan order. Contrarily, in the aforementioned works [25] and [31], the feature probabilities in a certain spatial neighborhood and across all channels must be re-computed when the quantization index of a single entry is changed. This backwards-dependency of the probabilities turned out to be a major obstacle for adapting the quantization step size with regard to the bitrate.

Therefore, in this paper, the probability of a single feature entry $\mathbb{P}_{\hat{z}}(\hat{z}_l)$ is estimated via the cumulative distribution function with parameters $(\hat{\mu}_l, \hat{\sigma}_l)$. Thus, one writes this function as

$$P_z\big(\hat{z}_l; (\hat{\mu}_l, \hat{\sigma}_l)\big) := \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{\hat{z}_l + t - \hat{\mu}_l}{\sqrt{2}\hat{\sigma}_l}\right)\right]_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}}. \quad (4)$$

Here, $l = (l_0, (l_1, l_2, l_3))$ denotes a multi-index as follows:

- $l_0 \in \{0, 1, 2\}$ refers to the component (0 denoting the high, 1 the middle, and 2 the low resolution),
- l_1 is the channel index within this component l_0,
- l_2 is the vertical coordinate in channel l_1,
- l_3 is the horizontal coordinate in channel l_1.

Finally, for transmitting the side information, a fully connected neural network $P_y(\cdot, \phi)$ approximates the true probability $\mathbb{P}_{\hat{y}}$; see Appendix in [22]. The parameters ϕ are optimized during the training stage and remain constant afterwards.

B. ARCHITECTURE OF THE VARIATIONAL AUTO-ENCODER

As stated in the previous subsection, the variational auto-encoder investigated in this paper mainly consists of multi-scale convolutional layers. In particular, Table 1 summarizes the architecture of the encoder and decoder networks in (2) and (3). Starting with the input image x, the encoder network determines the features as

$$z = \mathrm{Enc}_{N-1} \circ \cdots \circ \mathrm{Enc}_0(x),$$
$$z \in \mathbb{R}^{w \times h \times c_0} \bigoplus \mathbb{R}^{\frac{w}{2} \times \frac{h}{2} \times c_1} \bigoplus \mathbb{R}^{\frac{w}{4} \times \frac{h}{4} \times c_2},$$

where Table 1 states the composition of the output channels in each layer. Thus, one uses channel sizes $(c_0, c_1, c_2) = (192, 48, 16)$ with spatial dimensions $(w, h) = (W/8, H/8)$ depending on the input image size. Next, for each multi-scale layer Enc_n, one defines the following functions:

- $f_{n,H\to H}, f_{n,M\to M}, f_{n,L\to L}$, which are convolutional layers with kernels and downsampling rates as in Table 1,

TABLE 1. The VAE architecture: Each row denotes a multi-scale convolutional layer. "Kernel" shows the dimensions of the convolution kernels and the down (↓) - or up (↑) - sampling rate along each spatial axis. "In" and "Out" denote the channels summed over all components. "H", "M" and "L" state the number of output channels per component. "Act" states the activations, (I)GDN stands for (inverse) generalized divisive normalization; [20]. The number of features is $(3 + \frac{13}{64})HW$.

Layer	Kernel	In	Out	H	M	L	Act
Enc_0	$5 \times 5, \downarrow 2$	1	256	192	48	16	GDN
Enc_1	$5 \times 5, \downarrow 2$	256	256	192	48	16	GDN
Enc_2	$5 \times 5, \downarrow 2$	256	256	192	48	16	None
Dec_3	$5 \times 5, \uparrow 2$	256	256	192	48	16	IGDN
Dec_2	$5 \times 5, \uparrow 2$	256	256	192	48	16	IGDN
Dec_1	$5 \times 5, \uparrow 2$	256	1	1	1	1	None
Enc'_0	$3 \times 3, \downarrow 1$	256	256	192	48	16	ReLU
Enc'_1	$5 \times 5, \downarrow 2$	256	256	192	48	16	ReLU
Enc'_2	$5 \times 5, \downarrow 2$	256	256	192	48	16	None
Dec'_3	$5 \times 5, \uparrow 2$	256	256	192	48	16	ReLU
Dec'_2	$5 \times 5, \uparrow 2$	256	384	288	72	24	ReLU
Dec'_1	$3 \times 3, \uparrow 1$	384	512	384	96	32	None

- $f_{n,H\to M}, f_{n,M\to L}$ are 5×5, which are convolutional layers with constant spatial downsampling rate 2,
- $f_{n,M\to H}, f_{n,L\to M}$ are 5×5, which are transposed convolutional layers with constant upsampling rate 2.

Note that each single convolutional layer f_n employs an activation function according to Enc_n in Table 1. The encoder network computes the features from the input image as follows. Since the original image consists of a single channel at a specific resolution, the first layer constructs a three-component multi-channel output as

$$z_1 = \mathrm{Enc}_0(x) = \begin{pmatrix} f_{0,H\to H}(x) \\ f_{0,H\to M}(z_{1,H}) \\ f_{0,M\to L}(z_{1,M}) \end{pmatrix} = \begin{pmatrix} z_{1,H} \\ z_{1,M} \\ z_{1,L} \end{pmatrix}.$$

Subsequently, each multi-scale convolutional layer updates the individual components separately from one another. Then, the updated components are re-sampled as needed and added accordingly. Hence, the outputs $z_{n+1} = \mathrm{Enc}_n(z_n)$ are computed as

$$z_{n+1} = \begin{pmatrix} f_{n,H\to H}(z_{n,H}) + f_{n,M\to H}(f_{n,M\to M}(z_{n,M})) \\ f_{n,M\to M}(z_{n,M}) + \frac{1}{2}(f_{n,H\to M}(f_{n,H\to H}(z_{n,H})) \\ \qquad + f_{n,L\to M}(f_{n,L\to L}(z_{n,L}))) \\ f_{n,L\to L}(z_{n,L}) + f_{n,M\to L}(f_{n,M\to M}(z_{n,M})) \end{pmatrix}.$$

The process is also illustrated in Figure 2.

Note that the decoder network and the hyper networks are constructed in a similar manner. For instance, the decoder computes the reconstruction from the quantized features as

$$x_N := \hat{z}, \quad x_0 = \mathrm{Dec}_1 \circ \cdots \circ \mathrm{Dec}_N(x_N),$$

where the outputs $x_{n-1} = \mathrm{Dec}_n(x_n)$ are computed as

$$x_{n-1} = \begin{pmatrix} g_{n,H\to H}(x_{n,H}) + g_{n,M\to H}(g_{n,M\to M}(x_{n,M})) \\ g_{n,M\to M}(x_{n,M}) + \frac{1}{2}(g_{n,H\to M}(g_{n,H\to H}(x_{n,H})) \\ \qquad + g_{n,L\to M}(g_{n,L\to L}(x_{n,L}))) \\ g_{n,L\to L}(x_{n,L}) + g_{n,M\to L}(g_{n,M\to M}(x_{n,M})) \end{pmatrix}.$$

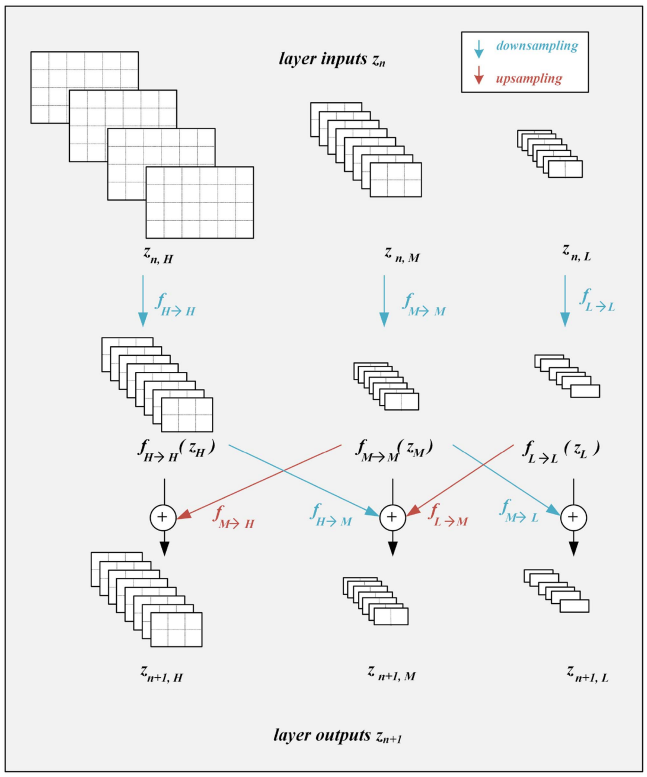


FIGURE 2. The computation of z_{n+1} via the multi-scale convolutional layer. Note that each colored arrow denotes a convolutional layer with an activation function applied to its outputs.

Here, the maps $g_{n,H\to H}$, $g_{n,M\to M}$, $g_{n,L\to L}$ are transposed convolutional layers with upsampling rates and activations as stated in Table 1. Note that the final decoder layer Dec_1 is computed as in [31, Sec. 2.2]. The reconstruction is defined as the high-resolution component from the final layer output, i.e.,

$$x_0 = \begin{pmatrix} x_{0,H} \\ x_{0,M} \\ x_{0,L} \end{pmatrix}, \quad \hat{x} \coloneqq x_{0,H}.$$

C. TRAINING DETAILS

The training algorithm requires a differentiable version of the Lagrangian cost (1). As in [21], one introduces noisy versions of the features and side information as

$$n \sim \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right), \quad \tilde{z}(\Delta) = z + \Delta \cdot n, \quad \tilde{y} = y + n, \quad (5)$$

for avoiding zero gradients due to the hard quantization. Further, let $\mathrm{MSE}(\cdot,\cdot)$ denote the mean squared error between original and reconstruction as

$$\mathrm{MSE}(x, \hat{x}) \coloneqq \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \left(x_{i,j,0} - \hat{x}_{i,j,0}\right)^2.$$

Next, one defines the following loss terms

$$d = d(z, \Delta) = \mathrm{MSE}(x, \mathrm{Dec}(\tilde{z}(\Delta))), \quad (6)$$

$$R = R(z, \Delta) = \frac{1}{HW} \sum_{l} -\log_2 P_z\left(\tilde{z}_l(\Delta); \left(\hat{\mu}_l, \hat{\sigma}_l\right)\right), \quad (7)$$

$$R' = R'(y) = \frac{1}{HW} \sum_{k} -\log_2 P_y(\tilde{y}_k; \phi), \quad (8)$$

where l and k are multi-indices as in (4). Using a pair of Lagrange parameters (λ_1, λ_2) and scalar weights (κ_1, κ_2), the training objective then becomes

$$\min \mathbb{E}\left[\sum_{i=1}^{2} \kappa_i \left(d + \lambda_i (R + R')\right)\right]. \quad (9)$$

Note that (6) and (7) are both differentiable with respect to Δ. For choosing the appropriate weights, one picks $\lambda_1 = 128$ and $\lambda_2 = 512$ and trains two separate VAEs with respect to the individual costs and constant uniform quantization. Then, one chooses weights such that the weighted training cost $\kappa_1(d+\lambda_1(R+R'))$ of the first network is approximately as large as the weighted training cost $\kappa_2(d+\lambda_2(R+R'))$ of the second network. Then, one trains a third, single VAE by setting κ_1, κ_2 to these determined values. For the optimization of (9), instead of training a modulating network for scaling the layer outputs as in [29], one fixes $\Delta_1 = 1 = \text{const}$ and optimizes the step size Δ_2 jointly with the network parameters. Consequently, the impact of each example with respect to both Lagrange costs is taken into account at each update step during the optimization.

Using middle-resolution images from ImageNet [32] as training data, one performs stochastic gradient descent over 256×256 patches of the luma component. Furthermore, one sets the batch size to 8 and processed 2500 batches per training epoch. The step size for the Adam optimizer [33] was set as $\alpha_j = 10^{-4} \cdot 1.13^{-j}$, where $j = 0, \ldots, 19$ was increased if the training loss saturated after finishing an epoch. Finally, the authors have checked that the average compression performance of the resulting auto-encoder is competitive against a set of auto-encoders, which were separately optimized over parameters $\lambda \in \{128, 256, 512, 1024, 2048\}$; see Section V for further details. The training procedure can be adapted to different pairs (λ_1, λ_2), where an optimization with respect to high bitrates and high reconstruction quality should be ensured. In particular, one observes that setting the Lagrangian parameters too large leads to a poor performance of the resulting VAE for higher bitrates. Furthermore, these VAEs are hardly capable of achieving higher reconstruction qualities, even when the features are un-quantized. On the other hand, the best results were accomplished when at least one summand in the training objective (9) rewards a high reconstruction quality (i.e., PSNR > 40).

IV. RATE-DISTORTION OPTIMIZED ENCODING

The purpose of this section is to investigate the impact of changing the quantization indices of the features on the resulting rate-distortion performance. For this, one uses a fixed VAE-based image codec whose exact architecture is described in Sections III-A and III-B. The weights of this

VAE are optimized with respect to (9) where the training procedure is carried out as explained in Section III-C. Most importantly, the described VAE is able to continuously achieve different rate-distortion trade-offs by adapting the quantization step size accordingly. Given such an image compression system, one may consider the following set of quantization indices

$$w \in \mathbb{Z}^{w \times h \times c_0} \bigoplus \mathbb{Z}^{\frac{w}{2} \times \frac{h}{2} \times c_1} \bigoplus \mathbb{Z}^{\frac{w}{4} \times \frac{h}{4} \times c_2}.$$

Provided that the side information $\hat{y}$ and the parameters $\mathrm{Dec}'(\hat{y}) = (\hat{\mu}, \hat{\sigma})$ are fixed, the resulting bitrate and distortion can be expressed as

$$d(\Delta \cdot w) = \mathrm{MSE}(x, \mathrm{Dec}(\Delta \cdot w)), \tag{10}$$

$$R(\Delta \cdot w) = \frac{1}{HW} \sum_l R_l\big(\Delta \cdot w_l; (\hat{\mu}_l, \hat{\sigma}_l)\big), \tag{11}$$

with a variable quantization step size $\Delta > 0$. The minimization task then becomes

$$\min_{w,\Delta} \big(d(\Delta \cdot w) + \lambda\big(R' + R(\Delta \cdot w)\big)\big), \tag{12}$$

where R' is the constant bitrate of the side information (8). Note that the encoder Enc typically does not find a global solution of (12). In [31], an algorithm for improving the coding efficiency of an auto-encoder was proposed, which can be characterized as an input-dependent encoder optimization. The algorithm exhaustively tries promising candidates in a neighborhood around the network-determined features and avoids multiple decoder executions by pre-estimating the distortion by a higher-order polynomial. The approach is similar to fast-search methods in modern video codecs, where the impact of different coding options on d and R is well-understood. The following subsections provide a thorough investigation of the rate-distortion performance of the proposed auto-encoder. The goal is to understand the impact of signal-dependent encoder optimizations on the rate-distortion performance.

A. ROBUST ESTIMATION OF THE DISTORTION

In [31], a distortion estimate for VAEs was derived from a Taylor approximation of the following auxiliary function with features z and the approximation error h as

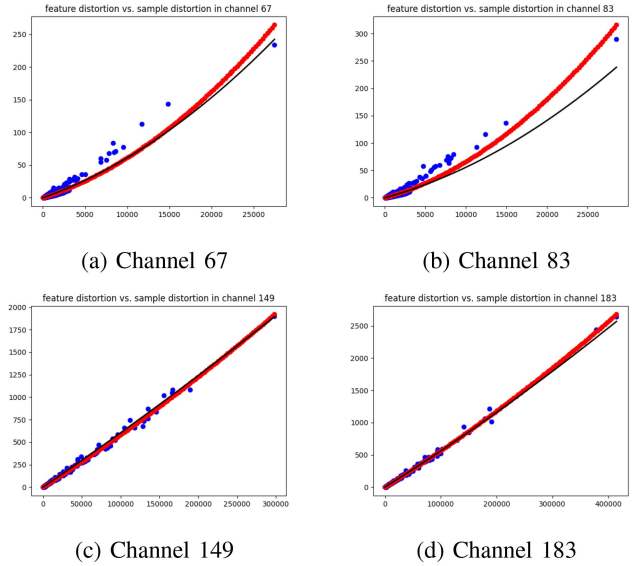
$$\epsilon(h) := \mathrm{MSE}(\mathrm{Dec}(z), \mathrm{Dec}(z + h)),$$

where z is expected to be close to a local minimum of

$$\mathrm{MSE}(x, \mathrm{Dec}(z)).$$

As $\epsilon(0) = 0$ is clearly a minimum, the gradient therefore is $\nabla\epsilon(0) = 0$ and the impact of displacements h on ϵ can be approximated by a multivariate polynomial of order two or higher. Let $z = (z^{(0,0)}, \ldots, z^{(0,c_0-1)}, \ldots, z^{(2,0)}, \ldots, z^{(2,c_2-1)})$ denote the c feature channels, i.e.,

$$z^{(i,0)}, \ldots, z^{(i,c_i-1)} \in \mathbb{R}^{\frac{h}{2^i} \times \frac{w}{2^i}}.$$



(a) Channel 67 (b) Channel 83

(c) Channel 149 (d) Channel 183

FIGURE 3. Relationship of single-channel feature displacements and the distortion in selected channels. The blue dots are evaluations ($||h||^2$, $\epsilon(h)$) on the Imagenet data, the red line is the fitted polynomial (13). The black line was fitted to the Kodak data.

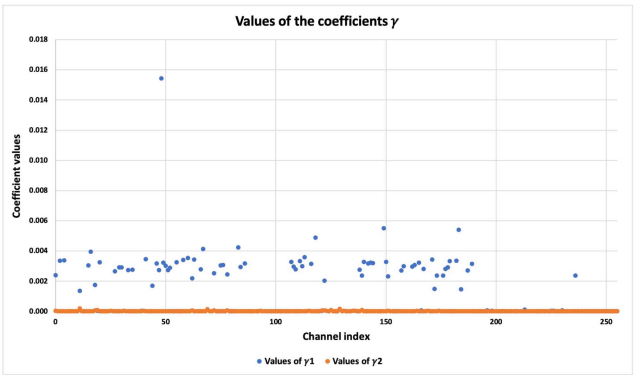


FIGURE 4. Coefficients $\gamma_1^{(i,j)}$ and $\gamma_2^{(i,j)}$ were determined on a subset of the Imagenet data. Here, the first 192 coefficients belong to the *H*-component, the next 48 coefficients to the *M*-component and the last 16 coefficients to the *L*-component.

Similarly, one writes the displacement per component and channel as

$$h = \Big(h^{(0,0)}, \ldots, h^{(0,c_0-1)}, \ldots, h^{(2,0)}, \ldots, h^{(2,c_2-1)}\Big).$$

Given a randomly chosen subset of the Imagenet data and the Kodak set, one evaluates $\epsilon(h)$ for different single-channel displacements and compared the results; see Figure 3. Given the data, one finds that $\epsilon(h)$ can be approximated robustly by a radial polynomial as in [31]. In particular, the approximation of ϵ becomes

$$\epsilon(h) \approx \sum_{i=0}^{2} \sum_{j=0}^{c_i-1} \left(\gamma_1^{(i,j)} \left\|h^{(i,j)}\right\|^2 + \gamma_2^{(i,j)} \left\|h^{(i,j)}\right\|^4\right), \tag{13}$$

where $||\cdot||$ denotes the Euclidean norm and $\gamma_1^{(i,j)} \gg \gamma_2^{(i,j)}$ holds, see Figure 4.

The authors point out that although the values $\gamma_2^{(i,j)}$ are close to zero, the fourth power of the quantization error in

the feature domain can become quite large. Hence, the corresponding term in the polynomial still affects the outcome of the approximation, in particular for larger quantization step sizes. In particular, the authors have tested to fit a quadratic polynomial for approximating the distortion, thus omitting the latter summands from (13). This has led to less accurate pre-estimates of the sample distortion from the approximation error in the feature domain, which proves non-beneficial for finding promising quantization indices in terms of the rate-distortion cost.

Next, with $\hat{z} = z + h$, the triangle inequality yields

$$d(\hat{z}) \lessapprox d(z) + \epsilon(h). \tag{14}$$

Here, the upper bound is used for estimating $d(\hat{z})$. Note that $d(z)$ is not zero, contrary to the situation in conventional codecs where orthogonal transforms are used. Since the displacement h directly depends on Δ, one can also determine a high-rate approximation of the average distortion as

$$d(\hat{z}) \approx d(z) + \hat{\gamma}_1 \Delta^2 + \hat{\gamma}_2 \Delta^4, \tag{15}$$

where $\hat{\gamma}_1, \hat{\gamma}_2$ are determined in the same way as the parameters in (13). Finally, the estimate (14) can be improved by determining a local minimum $z_{\min}$ of the un-quantized reconstruction error, which therefore satisfies

$$d(z_{\min}) \leq d(\Delta \cdot w). \tag{16}$$

As a consequence, there is a neighborhood of $z_{\min}$, in which the lower bound (16) holds.

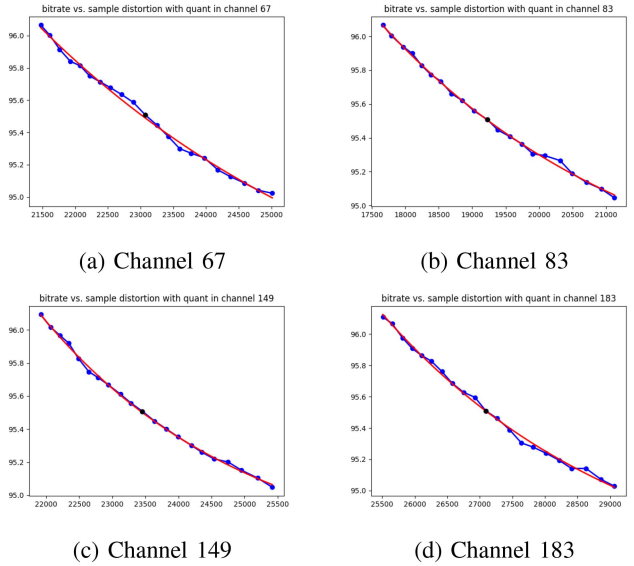
B. MODELING THE RELATIONSHIP BETWEEN DISTORTION AND BITRATE

Given a budget of R^* bits, the goal is to optimally spend this budget across the features such that the distortion is minimized. Solving this allocation problem requires to understand the functional relationship between the bitrate in each channel and the resulting distortion. Furthermore, a suitable model for the distortion-rate function helps to better assess the bit allocation of the encoder-determined features before the optimization. As both functions d and R depend on the coding options and the quantization, one approximates the distortion by the sum of the following distortion-rate functions

$$d(\Delta \cdot w) \approx \sum_{i=0}^{2} \sum_{j=0}^{c_i - 1} d_{i,j}(R_{i,j}). \tag{17}$$

Here, $R_{i,j}$ denotes the bitrate in the j-th channel of the i-th component and equals the sum of cross entropies over all spatial positions. Note that the rates $R_{i,j}$ are additive since the features are coded without context adaption. One can expect similarly-behaving distortion-rate functions in a single channel due to using convolutional layers. By using an ansatz from [7], one can parametrize the expression (17) by the following family of convex functions

$$d_{i,j}(R_{i,j}) = \alpha^2 \exp(-\beta R_{i,j}). \tag{18}$$



(a) Channel 67 (b) Channel 83

(c) Channel 149 (d) Channel 183

FIGURE 5. The blue dots denote evaluations (d, R) with variable $\Delta \in (0.8, 1.2)$ in the channels with the highest rate for the image Kodak14. The step size $\Delta = 1$ is fixed in the remaining channels. The red curves (18) were fitted to each channel.

For scalar quantizers, the rate-distortion function is known for high rates; [10], [34]. In particular, when the distortion is measured as MSE, one has $\beta = 2 \ln 2 = \text{const}$ according to the classical quantization theory. For the sake of completeness, note that vector quantization is capable of achieving rates closer to the theoretical lower bound of lossy coding than uniform scalar quantizers at the same distortion level.

Next, the coefficients $\alpha, \beta \in \mathbb{R}$ in (18) are determined channel-wise by evaluating (10) and (11) with variable step sizes $\Delta \in (0, \Delta_{\max})$ on a luma-only version of the Kodak set [35]. Here, one observes that the fitted coefficients β are quite different from channel to channel but in the same order of magnitude. The values of α also differ in each channel. The expressions (17) and (18) approximate the distortion well at high rates which was experimentally verified. Figure 5 demonstrates this for a sample image from the Kodak set.

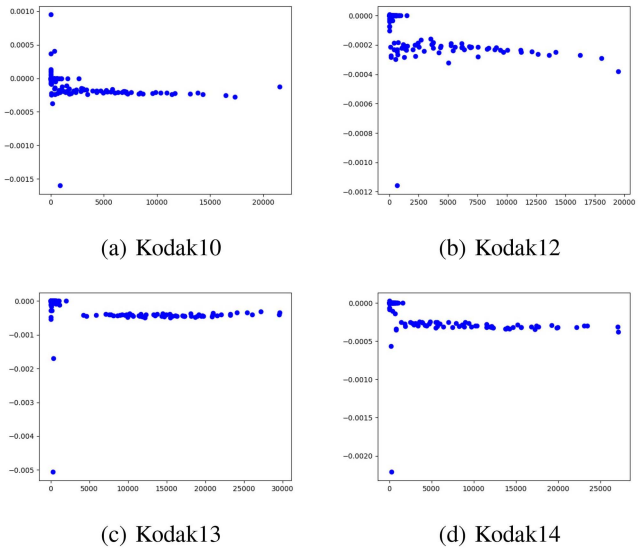
C. EQUAL-SLOPE CONDITION AND STEP SIZE SELECTION

By using the considerations from the previous subsection, one is able to state a condition for the optimal allocation of bits across the feature channels. First, minima of the Lagrangian cost (12) are expected to occur when all partial derivatives $\partial / \partial R_{i,j}$ are zero, i.e.,

$$\frac{\partial d_{i,j}}{\partial R_{i,j}}(R_{i,j}) = -\lambda. \tag{19}$$

This condition for the optimal allocation of bits is referred to as equal-slope condition; [10], [11]. By using expression (18), the partial derivatives in (19) are computed as

$$\frac{\partial d_{i,j}}{\partial R_{i,j}}(R_{i,j}) \approx -\alpha^2 \beta \exp(-\beta R_{i,j}). \tag{20}$$



(a) Kodak10 (b) Kodak12

(c) Kodak13 (d) Kodak14

FIGURE 6. The blue dots denote the slope values (20) versus the bitrate $R(\hat{z}(\Delta))$ with step size $\Delta = 1$ in each channel. The resulting slopes of the high-rate channels remain on a plateau in each image.

Since $R_{i,j} = R_{i,j}(\hat{z}(\Delta))$ holds, one can evaluate the slopes (20) for different quantizations of a fixed feature representation. Figure 6 demonstrates that the bit allocation of the encoder-determined features $\hat{z}$ is close to the optimal condition (19), especially for the high-rate channels. Since the values of β vary channel-wise (see Section IV-B), the equal-slope condition does not imply that the scalar quantizers in the different channels operate at the same distortion like in conventional codecs, i.e., $d_{i,j}(R_{i,j}) \neq \text{const}$ holds here.

Analogously to (18), one models the rate-distortion function of the compression networks for high rates as

$$R(\Delta \cdot w) \approx R(d) = \frac{1}{\beta} \ln\left(\frac{\alpha^2}{d}\right). \tag{21}$$

By again combining with the Lagrangian cost (12), a minimum is expected where the derivative $\partial/\partial d$ is zero, i.e.,

$$\frac{\partial R}{\partial d}(d) = -\frac{1}{\beta d} = -\frac{1}{\lambda} \iff \lambda = \beta d. \tag{22}$$

Hence, the distortion grows linearly with the Lagrange parameter. Therefore, using (15) as a high-rate approximation of the distortion yields a simple method for adapting the quantization step size Δ accordingly.

D. AN ALGORITHM FOR RATE-DISTORTION OPTIMIZED ENCODING

Using the considerations from Sections IV-A and IV-B, one derives the following Algorithm 1 for optimizing the rate-distortion trade-off (11). Given the Lagrange parameter λ and the initialization $z = \text{Enc}(x)$, the bit allocation across the channels is typically close to the optimal condition (19) for a suitably chosen step size $\Delta(\lambda)$, at least for higher rates. Then, the minimum is initialized as $w = z$ and the current

Algorithm 1: Fast Rate-Distortion Optimization for Variable-Rate Auto-Encoders

Result: w^*
Given: $x, z, \hat{y}, R', \{(\gamma_1^{(i,j)}, \gamma_2^{(i,j)})\}, \lambda$;
$(\hat{\mu}, \hat{\sigma}) = \text{Dec}'(\hat{y})$ via (3);
Pick $\Delta := \Delta(\lambda)$;
Set $w := z, w^* := \hat{w}(\Delta), h^* := w - w^*$;
$R^* = R(w^*, (\hat{\mu}, \hat{\sigma})), d^* = d(w^*)$ via (10),(11);
$J^* := d^* + \lambda(R^* + R')$;
for *each feature position* l **do**
  Set cand $= \{\hat{\mu}_l, w_l - \Delta, w_l + \Delta\}$;
  $R_l^* = R_l(w_l^*, (\hat{\mu}_l, \hat{\sigma}_l))$ via (11);
  $\epsilon^* = \epsilon(h^*)$ via (13);
  for $k = 0, 1, 2$ **do**
    Set $w_l := \text{cand}[k], w^k := \hat{w}(\Delta), h^k := w - w^k$;
    $R_l^k = R_l(w_l^k, (\hat{\mu}_l, \hat{\sigma}_l))$ via (11);
    $R^k := R^* - R_l^* + R_l^k$;
    $\epsilon^k = \epsilon(h^k)$; $d^k := d^* - \epsilon^* + \epsilon^k$ via (13);
    if $d^k + \lambda(R^k + R') < J^*$ **then**
      $d^k = d(w^k)$ via (10);
      if $d^i + \lambda(R^i + R') = J^i < J^*$ **then**
        Set $w^* := w^k, d^* := d^k, R^* := R^k, J^* := J^k, h^* := h^k \epsilon^* := \epsilon^k, R_l^* := R_l^k$;
      end
    end
  end
end

rate and distortion of $w^* = \hat{w}(\Delta)$ are computed. Furthermore, one computes the approximation error as $h^* = w - w^*$. Next, proceed for each multi-index position l in the feature representation of w^* as follows:

1) Compute the bitrate of the feature entry w_l^* and the auxiliary value $\epsilon^* = \epsilon(h^*)$.
2) Set the quantization index candidates cand $= \{\hat{\mu}_l, w_l - \Delta, w_l + \Delta\}$, i.e., the upper and lower neighboring quantization levels of w_l^*, and $\hat{\mu}_l$, and do for $k = 0, 1, 2$:
   a) Set $w_l := \text{cand}[k], w^k := \hat{w}(\Delta)$ and compute the updated bitrate.
   b) Set $h^k = w - w^k$, compute the updated auxiliary value $\epsilon^k := \epsilon(h^k)$
   c) Pre-estimate the distortion by using (13) with the values ϵ^k and ϵ^*.
   d) When the pre-estimated rate-distortion cost is lower than the current minimum, compute the actual distortion by executing the decoder network with input w^k and re-set everything, when the rate-distortion cost is less than the current minimum.

Most importantly, the distortion is pre-estimated by (13) for avoiding the execution of the decoder network for less-promising feature candidates. Remember that the algorithm

disregards the dependency of the entropy parameters from the features, see (3). Instead, the features are optimized with respect to a fixed probability distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$, which is transmitted lossless by the hyper system. Finally note that the initial value z can be replaced by determining a local minimum $z_{\min}$ via gradient descent prior to the Lagrange optimization.

E. PRE-PROCESSING THE FEATURES BEFORE THE OPTIMIZATION

In Sections IV-A and IV-D, it is suggested to replace the initial value z by a local minimum of the un-quantized reconstruction error. The motivation behind this is that the distortion can be estimated more precisely in a neighborhood of such a minimum. Ideally, increasing the quantization error in the feature domain should lead to an increased error in the sample domain. This would be case if the networks were entirely linear, for instance. From the considerations in Section IV-A, the initial value $z_{\min}$ itself does not provide a more efficient compression of the image (in fact, the Lagrange cost (12) slighlty increases), but the optimization might benefit from this because the distortion is estimated more accurately.

Another way of pre-processing the features can be derived by considering how the encoder network is optimized. Note that the training loss (9) takes into account the rate-distortion cost of the noisy features instead of the quantized ones. Furthermore, the loss is computed with respect to different Lagrange parameters using batches of example patches as input. As the updates of the encoder and decoder are computed by minimizing the expected value of the batch cost, one and the same network is supposed to generate compressible features for different resolutions and types of image content. Hence, when a particular image and a fixed decoder is given, the encoder-determined features are not necessarily a minimum of the following optimization problem

$$\min_{z}\big(d(\tilde{z}(\Delta)) + \lambda\big(R' + R(\tilde{z}(\Delta))\big)\big). \tag{23}$$

Note that the actual rate-distortion cost of a minimum z^* of (23) is not necessarily below the cost of the original features z. On the other hand, during the training stage, one considers the training loss of the noisy features as a suitable replacement of the actual rate-distortion cost.

Hence, the outcome of Algorithm 1 might improve by determining a local minimum z^* of the noisy cost (23) and using it as initial value instead of the encoder-determined features z. Furthermore, one finds that such a local minimum can be determined with modest computational complexity via gradient descent. Here, one initializes with the original features z and computes the step size factor of the gradient at each iteration by a back-tracking line search. The number of iterations is limited to 20, where the optimization is aborted prematurely when the cost improvement is below 10^{-5}.

V. EXPERIMENTS

As baseline, the auto-encoder designed and optimized as in Section III was evaluated on the entire Kodak set [35] with luma-only versions of the images. For comparison, the authors have trained another auto-encoder whose architecture is similar to the one proposed in [25]. Furthermore, this section presents results from [31] with and without encoder optimizations.

Next, the authors have combined the baseline auto-encoder with Algorithm 1 ("fast RDO"), where one computes the final bitrate by arithmetically coding the output w^* and counting the bits; [36]. The authors have further tested Algorithm 1 with the alternative initialization $z_{\min}$ ("dist.-opt. init value"), which was determined by applying a gradient descent with line search to the un-quantized reconstruction error $\mathrm{MSE}(x, \mathrm{Dec}(z))$ with 100 iterations. In the same fashion, a local minimum of the noisy cost z^* ("RD.-opt. init value"), which was determined as stated in Section IV-E, was tested as alternative initial value. Finally, one assesses the performance of the fast RDO algorithm by actually computing the Lagrangian cost of each quantization value candidate in Algorithm 1 instead of pre-estimating the distortion ("full RDO"). The reconstruction quality in all experiments is stated as Peaked-Signal-to-Noise Ratio (PSNR), although it is not well-suited for assessing perceptual image quality. However, it is a standard measure for comparing the efficiency of different image compression methods. In the Appendix, the authors further report individual rate-distortion curves of selected images from the Kodak set.

A. RESULTS WITHOUT ENCODER OPTIMIZATIONS

The baseline auto-encoder and the two benchmarks have similar designs. Assuming a normal distribution of the features, these networks apply a hyper system, which estimates the Gaussian parameters for each feature entry. The architecture in [31] practically employs the same architecture as from Section III except additionally applying a context-adaptive network for estimating the Gaussian parameters. The same network is used in [25], where it is introduced as auto-regressive context model. Here, the authors have adapted the architecture by setting number of output channels to 205 and using three layers on the encoder and decoder side. Thereby, the number of features almost remains constant across the different network architectures. The blue and yellow rate-distortion curves in Figure 7 demonstrate that [31] consistently outperforms the architecture from [25] for luma-only content. Hence, this demonstrates the benefit of using multi-scale convolutional layers. Furthermore, the red rate-distortion curve of the proposed VAE shows that using such layers overcompensates the loss in coding efficiency from omitting the auto-regressive network from the coding stage. Note that this curve is obtained by only adapting the step size Δ to the Lagrange parameter, instead of training separate coefficients as in the quoted works.

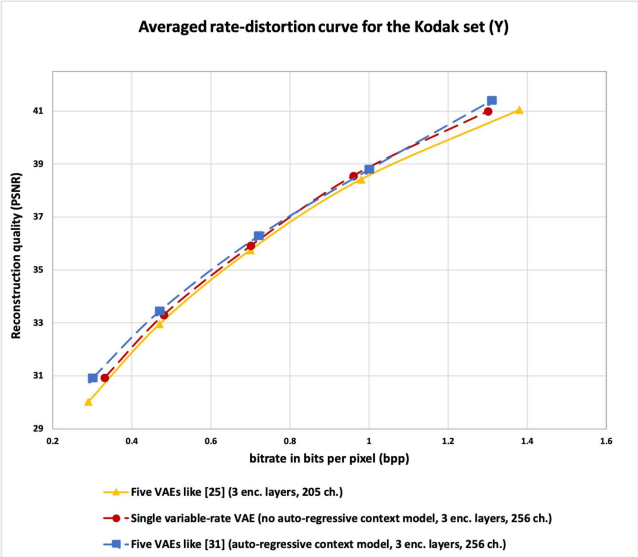


FIGURE 7. Averaged rate-distortion curves of the Kodak set (luma-only). The average PSNR is computed from the average MSE across the images. The red dashed curve denotes the variable-rate VAE from Section III. The yellow curve refers to a VAE similar to [25] with 205 output channels and three encoder layers. The blue dashed curve is generated by using the VAE from [31].

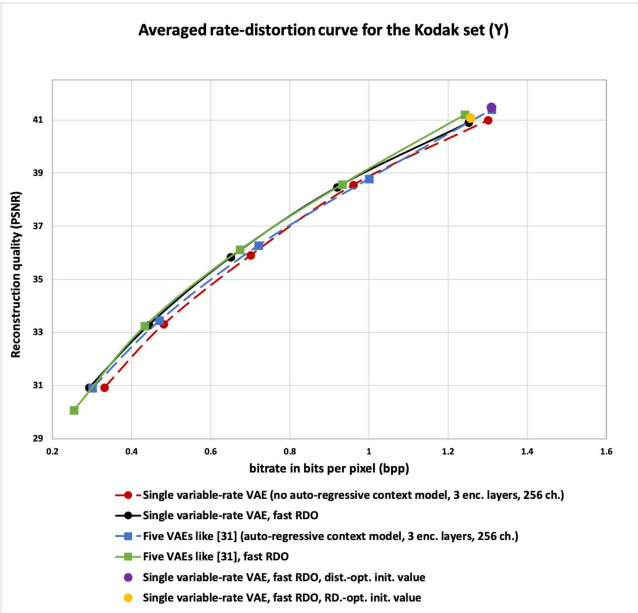


FIGURE 8. Averaged rate-distortion curves of the Kodak set (luma-only). The average PSNR is computed from the average MSE across the images. The red dashed curve denotes the variable-rate VAE from Section III. The black solid curve is generated by applying Algorithm 1 ("fast RDO") to the features. The violet dot is generated by using $z_{\min}$ at the highest rate. The yellow dot is generated by using z^* at the highest rate. The blue dashed and green solid curve are results by using the VAE from [31] with and without the fast RDO algorithm.

B. RESULTS WITH ENCODER OPTIMIZATIONS

Using the baseline auto-encoder, the authors have evaluated the impact of Algorithm 1 on the rate-distortion performance; see Figure 8. According to Table 2, the improvement of the proposed variable-rate VAE in terms of the Bjøntegaard-Delta bit rate (BD-rate) [37] due to using Algorithm 1 ranges from -2.4% to -7.0%. In comparison to [31], the compression efficiency of the un-optimized variable-rate VAE is lower than the un-optimized VAE from the quoted reference, mainly due to not using an auto-regressive context adaption of the feature probabilities and a single network for all Lagrange parameters (compare the red and blue curves). However, the rate-distortion performance of the variable-rate VAE keeps up with the results from [31] except for the highest reconstruction quality (compare the black and green curves), when both VAE architectures are combined with the fast RDO algorithm for optimizing the quantization of the encoder-determined features. As described in Section IV-E, Algorithm 1 is tested with different initial values. These initializations are derived from the encoder-determined features by applying a gradient descent with respect to either the non-quantized distortion or a smooth version of the Lagrangian cost. However, one observes that pre-processing the initial value does not significantly change the outcome of Algorithm 1 except for the highest reconstruction quality. Hence, the authors have plotted the impact of pre-processing the initial value for the operational point with the highest rate. Here, the use of the initialization $z_{\min}$ instead of z consistently leads to a higher reconstruction quality with moderate increase of the bitrate (see the violet dot). Also, the replacement of z by z^* in Algorithm 1 does not significantly alter the resulting bitrate, but increases the PSNR value moderately by 0.2 (see the yellow dot). Figures 10–15 further present individual rate-distortion curves for selected images from the Kodak set (see Appendix). Here, one considers the results from exhaustively testing each feature candidate ("full RDO") as an upper performance limit of the proposed encoder optimizations. The fast RDO algorithm closely approaches this upper limit of the rate-distortion performance for both the auto-encoder network from [31] (compare the green and orange curves) and the present variable-rate network (compare the black and brown curves).

Note that exhaustively testing each candidate requires about $10HW$ decoder network executions. This accounts for roughly 3.9 million executions for a Kodak image of size 756×512. In contrast to this, Table 3 states that Algorithm 1

TABLE 2. BD-rate savings due to applying Algorithm 1 to the features of the variable-rate VAE from Section III. Note that the stated savings were accomplished using one and the same decoder. "Index" states image index within the Kodak set. "Y in %" states the BD-rate saving in the luma component.

Index	Y in %	Index	Y in %	Index	Y in %
01	−3.12	09	−4.65	17	−4.91
02	−7.00	10	−4.77	18	−3.58
03	−3.83	11	−4.40	19	−4.50
04	−6.27	12	−6.21	20	−2.41
05	−2.74	13	−2.83	21	−3.53
06	−3.60	14	−3.64	22	−4.43
07	−3.66	15	−6.00	23	−5.05
08	−3.28	16	−4.85	24	−3.30

TABLE 3. The statistics of Algorithm 1 for different λ values. "Init." shows the initial value. "Num. dec." states the number of decoder executions. "cand[k]" states how often the *k*-th candidate was selected. The top number in each cell denotes the mean and the bottom one the standard deviation, taken over the Kodak set.

Init.: z	**Num. dec.**	**cand[0]**	**cand[1]**	**cand[2]**
128	29528.5	4966.8	408.7	405.0
	±7146.9	±1352.0	±211.3	±210.3
256	23854.3	4528.1	463.3	462.4
	±7769.7	±955.8	±269.1	±273.8
512	18475.5	4514.2	373.6	383.4
	±7437.2	±1431.8	±240.6	±248.0
1024	13354.5	4045.7	225.0	228.3
	±6117.1	±1681.6	±154.2	±160.9
2048	9024.3	3126.5	106.6	107.3
	±4716.3	±1574.0	±81.6	±83.8
Init.: $z_{\min}$	**Num. dec.**	**cand[0]**	**cand[1]**	**cand[2]**
128	30342.1	9688.2	880.1	863.3
	±6339.5	±2651.6	±389.4	±388.6
Init.: z^*	**Num. dec.**	**cand[0]**	**cand[1]**	**cand[2]**
128	35198.6	4158.5	359.4	365.8
	±10189.6	±907.9	±199.4	±210.1

consistently performs 100 to 400 times less decoder executions while providing significant coding gain against the un-optimized features. Here, the total number of decoder executions consistently decreases for larger Lagrange parameters. This hints towards a bias of the distortion estimation: for large Lagrange parameters, the quantization error in the feature domain also increases, and the inequality (14) apparently estimates the resulting distortion in the sample domain too large. As a consequence, more candidates are excluded in Algorithm 1. Interestingly, the percentage of successfully selected candidates from these tests consistently increases, from roughly 19% for the highest reconstruction quality to 37% for the lowest quality. Here, the increase is mainly caused by more frequently selecting the mean value candidate cand [0], which explains the steady decrease of the bitrate after the optimization. The algorithm mainly targets at decreasing the bitrate as much as possible without severely deteriorating the image quality. Finally, note that replacing the initial value z with $z_{\min}$ increases the percentage of successfully selected candidates at roughly 40%, see Table 3.

Specifically, the mean value candidate is consistently selected one out of three times, when the features are displaced towards a local minimum of the distortion prior to the actual rate-distortion optimization. When the initial value z is replaced with z^*, the percentage of successfully selected candidates drops to roughly 14% for the highest reconstruction quality, while the total number of tests significantly increases. In other words, more decoder evaluations are carried out and less successful candidates are found in this case. Note that the algorithm still checks each individual feature position, even when z^* was optimized with respect to a smooth version of the rate-distortion trade-off. It appears that checking different quantization indices at already altered feature positions is superfluous in most cases. However, one observes that only checking the non-altered positions in Algorithm 1 leads to a worsened outcome compared to checking all positions. The computation of the initial values z^* and $z_{\min}$ only requires a few iterations of gradient descent. Hence, the pre-processing step itself does not substantially contribute to the total complexity of the rate-distortion optimization.

VI. DISCUSSION

The present work extends the results from [31] by putting them in perspective with the rate-distortion theory from image and video coding. Here, the proposed image compression system was trained by jointly optimizing with respect to specific Lagrange parameters. This technique did not only ensure that the resulting network is capable of efficiently compressing images at different rates. For investigating the potential of rate-distortion optimized encoding, it has proven beneficial that the network can steadily navigate the rate-distortion curve. Although there is a lot of remarkable work on the capabilities of vector quantizers for both conventional and learned image coding, the presented results demonstrate that the coding efficiency can still be improved by rather simple signal processing methods. The methods in Sections IV-A and IV-B mainly rely on suitable approximations of the bitrate and the distortion for uniform scalar quantizers. However, these approximations enable the implementation of Algorithm 1, which significantly improves the coding efficiency of the proposed image compression system. From the perspective of the deep learning stage, it seems interesting that the optimized encoder network is capable of finding a near-optimal allocation of bits across the feature channels for a variety of different images. This is somehow unexpected because this condition is not explicitly stated in the training loss. Nonetheless, the presented results motivate further research on the potential of signal-dependent encoder optimizations.

Finally, the authors compare the rate-distortion performance of the investigated VAE-based image codec against several state-of-the-art-benchmarks. Fair comparisons are difficult to achieve since the majority of learning-based image compression networks is optimized for RGB content. Thus, we have encoded luma-only Kodak images using the reference software implementations of the latest video coding standards HEVC (HM-16.9, see [38]) and VVC (VTM-14.0, see [39]). Figure 9 suggests that HEVC (yellow curve) slightly outperforms our RD-optimized variable-rate VAE (black curve) for high rates and significantly does so for low rates. Hence, as expected, the compression efficiency of VVC (violet

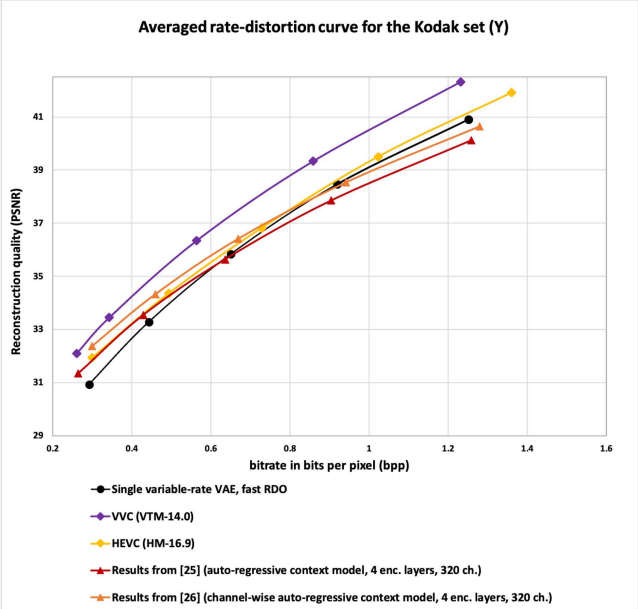


FIGURE 9. Averaged rate-distortion curves of the Kodak set (luma-only). The average PSNR is computed from the average MSE across the images. The black solid curve is generated by combining Algorithm 1 ("fast RDO") with the variable-rate VAE from Section III. The yellow line is generated by using the HEVC Test Model (HM-16.9); see [38]. The violet line is generated by using VVC Test Model (VTM-14.0); see [39]. The red curve are results published in [25]. The orange curve are results published in [26].

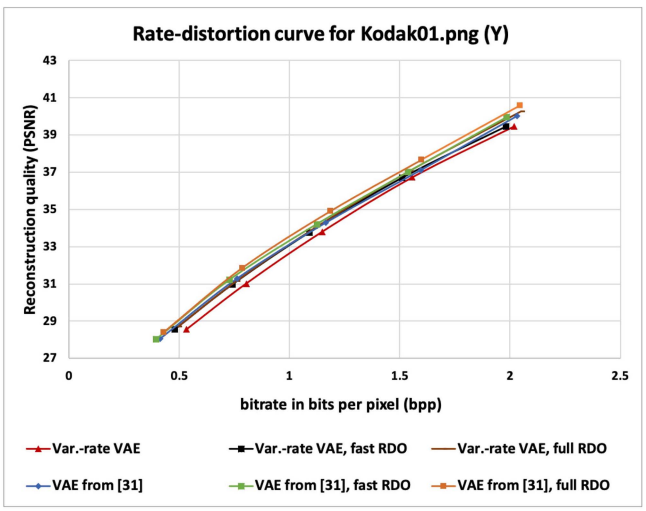


FIGURE 10. Rate-distortion curve of Kodak image 01 (Y).

curve) clearly surpasses the variable-rate VAE, regardless of the rate-distortion optimization. Furthermore, the authors of [25], [26] report luma-only results on the Kodak set using VAEs whose architectures are similar to the one in this paper (red and orange curve). Notably, both references optimize distinct networks for each point on the rate-distortion curve, and each network employs a different context-adaptive probability model for coding the features. On the other hand, theses networks use conventional convolutional layers and do not employ rate-distortion optimized quantization. In terms of rate-distortion performance, the variable-rate

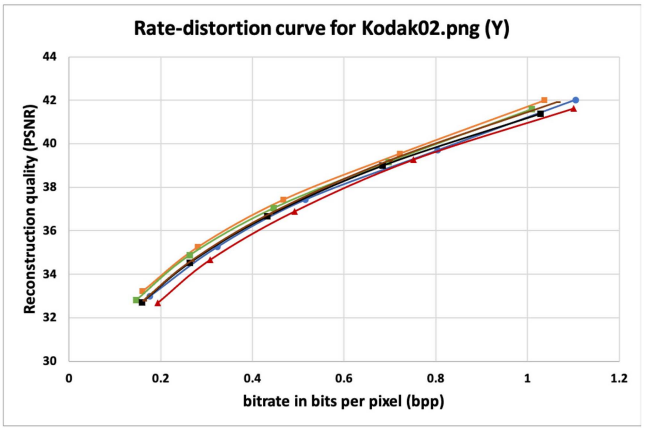


FIGURE 11. Rate-distortion curve of Kodak image 02 (Y).

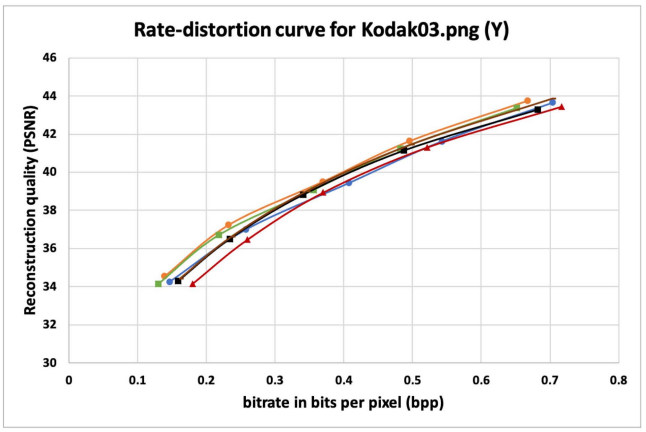


FIGURE 12. Rate-distortion curve of Kodak image 03 (Y).

VAE in this paper is competitive against both benchmarks. While [26] performs exceedingly well for lower bitrates (even better than HEVC), the variable-rate VAE has higher compression efficiency than both benchmarks for high rates larger than 1.0 bpp.

With respect to the computational complexity, there are several aspects to consider. It is clear that auto-regressive networks for coding the features are less practical because they require a fixed scan order. However, in this work, the majority of the computational burden comes from the more complex encoding process, which can be handled by massive hardware and parallelization techniques. Most importantly, the number of multiplications carried out by the networks themselves has increased strongly due to the use of multi-scale layers. Beyond optimizing the network architecture and improving the encoding process, constraining the decoding complexity remains a difficult task in learning-based image compression.

VII. CONCLUSION

This paper has investigated the rate-distortion performance of deep-learned end-to-end image compression networks. The

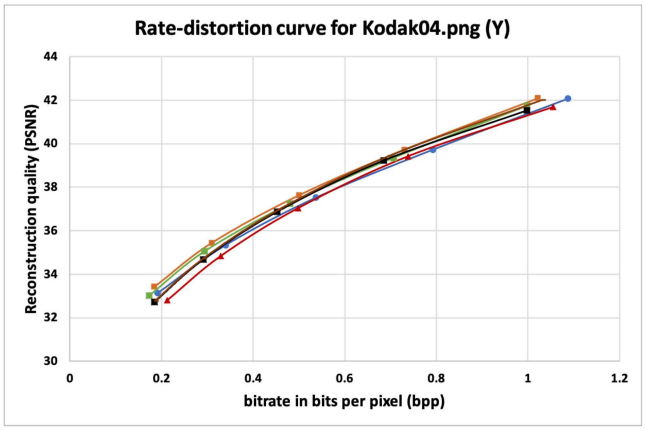


FIGURE 13. Rate-distortion curve of Kodak image 04 (Y).

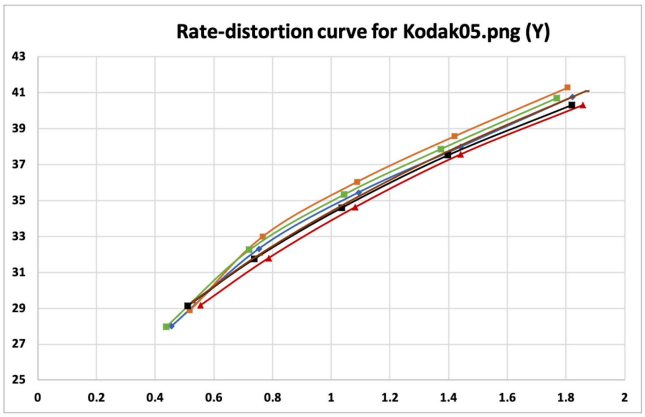


FIGURE 14. Rate-distortion curve of Kodak image 05 (Y).

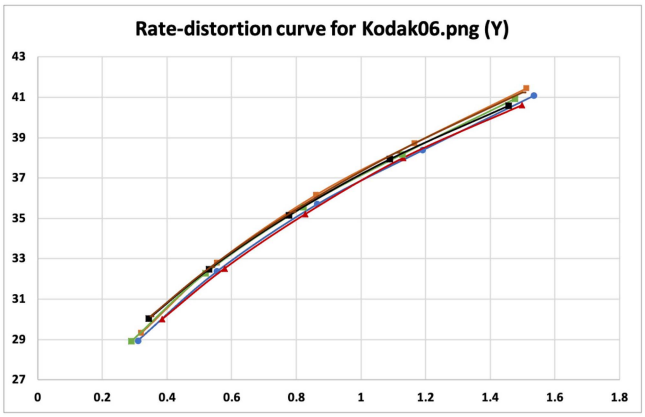


FIGURE 15. Rate-distortion curve of Kodak image 06 (Y).

training of such a network can be designed such that the impact of the quantization on the distortion and the bitrate can be estimated using conventional signal processing methods. In particular, the encoder in such a compression system is capable of finding a suitable allocation of bits across the feature channels. Furthermore, the quantization error in the individual channels can be used for estimating the distortion without necessarily executing the decoder network. Given a performant network architecture, the proposed algorithm for optimizing the quantization of the features greatly improves the coding efficiency. Nonetheless, the approach is applicable to any deep-learned image compression network.

APPENDIX

Figures 10 to 15 show the rate-distortion curves for each Kodak image and different encoding methods.

REFERENCES

[1] W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] "High efficiency video coding," Int. Telecommun. Union, Geneva, Switzerland, ITU-Recommendation H.265, 2013.

[3] M. Wien, *High Efficiency Video Coding—Coding Tools and Specification*, 1st ed. Heidelberg, Germany: Springer-Verlag, 2015, pp. 1–314. [Online]. Available: https://link.springer.com/book/10.1007/978-3-662-44276-0 (Accessed: Nov. 4, 2021).

[4] B. Bross, J. Chen, J. R. Ohm, G. J. Sullivan, and Y. K. Wang, "Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC)," *Proc. IEEE*, vol. 109, no. 9, pp. 1463–1493, Sep. 2021.

[5] "Versatile video coding," Int. Telecommun. Union, Geneva, Switzerland, ITU-Recommendation H.266, 2020.

[6] M. Wien and B. Bross, "Versatile video coding—Algorithms and specification," in *Proc. IEEE Int. Conf. Visual Commun. Image Process. (VCIP)*, 2020, pp. 1–3.

[7] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[8] J.-R. Ohm, G. Sullivan, H. Schwarz, T. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—Including high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[9] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[10] V. K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 9–21, Sep. 2001.

[11] T. Wiegand and H. Schwarz, "Source coding: Part I of fundamentals of source and video coding," in *Foundations and Trends in Signal Processing*, vol. 4. Hanover, MA, USA: Now Publ., 2011, pp. 176–211.

[12] K. Ramchandran and M. Vetterli, "Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility," *IEEE Trans. Image Process.*, vol. 3, pp. 700–704, Sep. 1994.

[13] M. Karczewicz, Y. Ye, and I. Chong, *Rate Distortion Optimized Quantization*, document ITU-T SG16/Q6 (VCEG), Int. Telecommun. Union, Geneva, Switzerland, Jan. 2008.

[14] J. Stankowski, C. Korzeniewski, M. Domański, and T. Grajek, "Rate-distortion optimized quantization in HEVC: Performance limitations," in *Proc. Picture Coding Symp. (PCS)*, 2015, pp. 85–89.

[15] H. Schwarz *et al.*, "Improved quantization and transform coefficient coding for the emerging versatile video coding (VVC) standard," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2019, pp. 1183–1187.

[16] H. Schwarz, T. Nguyen, D. Marpe, and T. Wiegand, "Hybrid video coding with trellis-coded quantization," in *Proc. Data Compression Conf. (DCC)*, 2019, pp. 182–191.

[17] G. D. Forney, "The viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[18] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," in *Proc. Picture Coding Symp. (PCS)*, 2016, pp. 1–5.

[19] J. Ballé *et al.*, "Nonlinear transform coding," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 2, pp. 339–353, Feb. 2021.
[20] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2016, pp. 1–14. [Online]. Available: https://arxiv.org/abs/1511.06281
[21] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–27. [Online]. Available: https://arxiv.org/abs/1611.01704
[22] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–23. [Online]. Available: https://arxiv.org/abs/1802.01436
[23] E. Agustsson *et al.*, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *Advances in Neural Information Processing Systems*, vol. 30. Red Hook, NY, USA: Curran Assoc., Inc., 2017. [Online]. Available: https://arxiv.org/abs/1704.00648
[24] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool, "Conditional probability models for deep image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4394–4402. [Online]. Available: https://arxiv.org/abs/1801.04260
[25] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Red Hook, NY, USA: Curran Assoc., 2018, pp. 10771–10780. [Online]. Available: https://arxiv.org/abs/1809.02736
[26] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2020, pp. 3339–3343. [Online]. Available: https://github.com/tensorflow/compression/blob/master/results/image_compression/kodak/PSNR_sRGB_Y
[27] M. Akbari, J. Liang, J. Han, and C. Tu, "Generalized octave convolutions for learned multi-frequency image compression," 2020, *arxiv:2002.10032*.
[28] B. Li, M. Akbari, J. Liang, and Y. Wang, "Deep learning-based image compression with trellis coded quantization," in *Proc. Data Compression Conf. (DCC)*, 2020, pp. 13–22.
[29] F. Yang, L. Herranz, J. V. D. Weijer, J. A. I. Guitián, A. M. López, and M. G. Mozerov, "Variable rate deep image compression with modulated autoencoder," *IEEE Signal Process. Lett.*, vol. 27, pp. 331–335, Jan. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8977394 (Accessed: Nov. 4, 2021).
[30] J. Lin *et al.*, "Variable-rate multi-frequency image compression using modulated generalized octave convolution," in *Proc. MMSP*, 2020, pp. 1–6.
[31] M. Schäfer, S. Pientka, J. Pfaff, H. Schwarz, D. Marpe, and T. Wiegand, "Rate-distortion-optimization for deep image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2021, pp. 3737–3741.
[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.
[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[34] H. Gish and J. Pierce, "Asymptotically efficient quantizing," *IEEE Trans. Inf. Theory*, vol. 14, no. 5, pp. 676–683, Sep. 1968.
[35] Sep. 21, 2021, "Kodak Image Dataset." [Online]. Available: http://r0k.us/graphics/kodak/ (Accessed: Nov. 4, 2021).
[36] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.
[37] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," in *Proc. 13th Meeting ITU-T Video Coding Experts Group (VCEG)*, Jan. 2001. [Online]. Available: https://www.itu.int/wftp3/av-arch/video-site/0104_Aus/VCEG-M33.doc (Accessed: Nov. 4, 2021).
[38] F. Bossen, *Common HM Test Conditions and Software Reference Configurations*, document JCTVC-L1100, Joint Collaborative Team on Video Coding (JCT-VC), Int. Telecommun. Union, Geneva, Switzerland, Sep. 2013.
[39] A. Browne, J. Chen, Y. Ye, and S. Kim, *Algorithm Description for Versatile Video Coding and Test Model 14 (VTM 14)*, document JVET-T2002, Joint Video Experts Team (JVET), Int. Telecommun. Union, Geneva, Switzerland, Jul. 2021.

MICHAEL SCHÄFER received the M.Sc. degree in mathematics from Freie Universität Berlin in 2017.

He joined the Video Communication and Applications Department, Fraunhofer Heinrich-Hertz-Institute, Berlin, Germany, in 2015. He contributed to the efforts of the ITU-T Joint Video Experts Team during the development of the Versatile Video Coding standard from 2017 to 2020. He has authored several publications concerning intra-picture prediction in video coding. His current research interests concern end-to-end image compression using artificial neural networks.

SOPHIE PIENTKA received the M.Sc. degree in mathematics from the RheinMain University of Applied Sciences in 2020.

She joined the Video Communication and Applications Department, Fraunhofer Heinrich-Hertz-Institute, Berlin, Germany, in 2020. Her current research interests concern end-to-end image compression using artificial neural networks.

JONATHAN PFAFF received the Diploma and Dr. rer. nat. degrees in mathematics from Bonn University, Bonn, Germany, in 2010 and 2012, respectively.

After a postdoctoral research stay with Stanford University, he joined the Video Coding and Analytics Department, Heinrich Hertz Institute, Berlin, Germany, in 2015, where he has been heading the research group on video coding technologies since 2020. He has been successfully contributed to the efforts of the ITU-T Video Coding Experts Group in developing the Versatile Video Coding standard since 2017. His current research interests include image and video coding and machine learning.

HEIKO SCHWARZ received the Dipl.-Ing. degree in electrical engineering and the Dr.-Ing. degree from the University of Rostock, Rostock, Germany, in 1996 and 2000, respectively.

He joined the Fraunhofer Heinrich Hertz Institute, Berlin, Germany, in 1999. Since 2010, he has been heading the research group Video Coding Technologies (formerly, Image and Video Coding) with the Fraunhofer Heinrich Hertz Institute. He became a Professor of Image Processing with the Free University of Berlin, in October 2017. He has actively participated in the standardization activities of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Pictures Experts Group. He successfully contributed to the video coding standards H.264/AVC, H.265/HEVC, and H.266/VVC.

Dr. Schwarz served as a reviewer for various international journals and international conferences. From 2016 to 2019, he was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He was appointed as a Co-Editor of H.264/AVC and as a Software Coordinator for the SVC reference software. He co-chaired various ad hoc groups of the standardization bodies and coordinated core experiments.

DETLEV MARPE (Fellow, IEEE) received the Dipl.-Math. degree (Hons.) from the Technical University of Berlin, Berlin, Germany, in 1990, and the Dr.-Ing. degree from the University of Rostock, Rostock, Germany, in 2004.

In 1999, he joined the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin, where he is currently the Head of the Video Coding and Analytics Department and the Image and Video Coding Group. He was a major Technical Contributor to the entire process of development of the H.264/MPEG-4 Advanced Video Coding (AVC) standard and the H.265/MPEG High Efficiency Video Coding (HEVC) standard, including several generations of major enhancement extensions. In addition to the CABAC contributions for both standards, he particularly contributed to the fidelity range extensions (which include the high profile that received the Emmy Award in 2008) and the scalable video coding extensions of H.264/MPEG-4 AVC. During the development of its successor H.265/MPEGHEVC, he also successfully contributed to the first model of the corresponding standardization project and further refinements. He made successful proposals to the standardization of its range extensions and 3D extensions. He has authored numerous publications in the research area of lmage and video coding, and holds several hundreds of internationally issued patents and patent applications in this area. His current research interests include still image and video coding, signal processing for communications and computer vision, machine learning, and information theory.

Dr. Marpe received several best paper awards for his publications. He was a recipient of the Karl Heinz Beckurts Award in 2011 and the Joseph von Fraunhofer Prize in 2004. He was a co-recipient of three Technical Emmy Awards as a Key Contributor and a Co-Editor of the H.264/MPEG-4 AVC standard in 2008 and 2009, respectively, and as a Key Contributor of H.265/MPEG-HEVC in 2017. In recognition of his dedicated contributions and excellent management of the review process, the 2016 Best Associate Editor Award of the IEEE Circuits and Systems Society. From 2014 to 2018, he served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is a member of the Informationstechnische Gesellschaft of the Verband der Elektrotechnik Elektronik Informationstechnik e.V.

THOMAS WIEGAND (Fellow, IEEE) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg–Harburg, Germany, in 1995, and the Dr.-Ing. degree from the University of Erlangen–Nuremberg, Germany, in 2000.

He was a Visiting Researcher with Kobe University, Kobe, Japan; University of California at Santa Barbara, Santa Barbara, CA, USA; and Stanford University, Stanford, CA, USA, where he also returned as a Visiting Professor. He served as a consultant to several start-up ventures. He is currently a Professor with the Department of Electrical Engineering and Computer Science, Technical University of Berlin, and is jointly heading the Fraunhofer Heinrich Hertz Institute, Berlin, Germany.

Dr. Wiegand received numerous awards and multiple Best Paper Awards for his publications for his research. The projects that he co-chaired for the development of the H.264/MPEGAVC standard have been recognized by the ATAS Primetime Emmy Engineering Award and a pair of NATAS Technology and Engineering Emmy Awards. He has been an active participant in standardization for video coding multimedia with many successful submissions to ITU-T and ISO/IEC. Since 2018, he has been the Chair of the ITU/WHO Focus Group on artificial intelligence for health. Since 2014, his name is listed in the World's Most Influential Scientific Minds as one of the most cited researchers in his field. He is an Associated Rapporteur of ITU-T VCEG. He has been elected to the German National Academy of Engineering (Acatech) and the National Academy of Science (Leopoldina).