



(51) International Patent Classification:

H04N 19/132 (2014.01) H04N 19/91 (2014.01)

(21) International Application Number:

PCT/US2024/031854

(22) International Filing Date:

31 May 2024 (31.05.2024)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

202311038186 02 June 2023 (02.06.2023) IN

202311061566 13 September 2023 (13.09.2023) IN

(71) Applicant: **DOLBY LABORATORIES LICENSING CORPORATION** [US/US]; 1275 Market Street, San Francisco, California 94103 (US).

(72) Inventors: **ARORA, Arjun**; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US).

MOHANANCHETTIAR, Arunkumar; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US). **YIN, Peng**; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US). **SHINGALA, Jay Nitin**; Flat No201 Arihant Apartments, 3rd Cross Okalipuram, Bangalore 560021 (IN). **SHAO, Tong**; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US).

(74) Agent: **KONSTANTINIDES, Konstantinos** et al.; DOLBY LABORATORIES, INC., Intellectual Property Group, 1275 Market Street, San Francisco, California 94103 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA,

(54) Title: THRESHOLD MODELS FOR SKIPPING ENTROPY CODING IN END-TO-END IMAGE COMPRESSION USING NEURAL NETWORKS

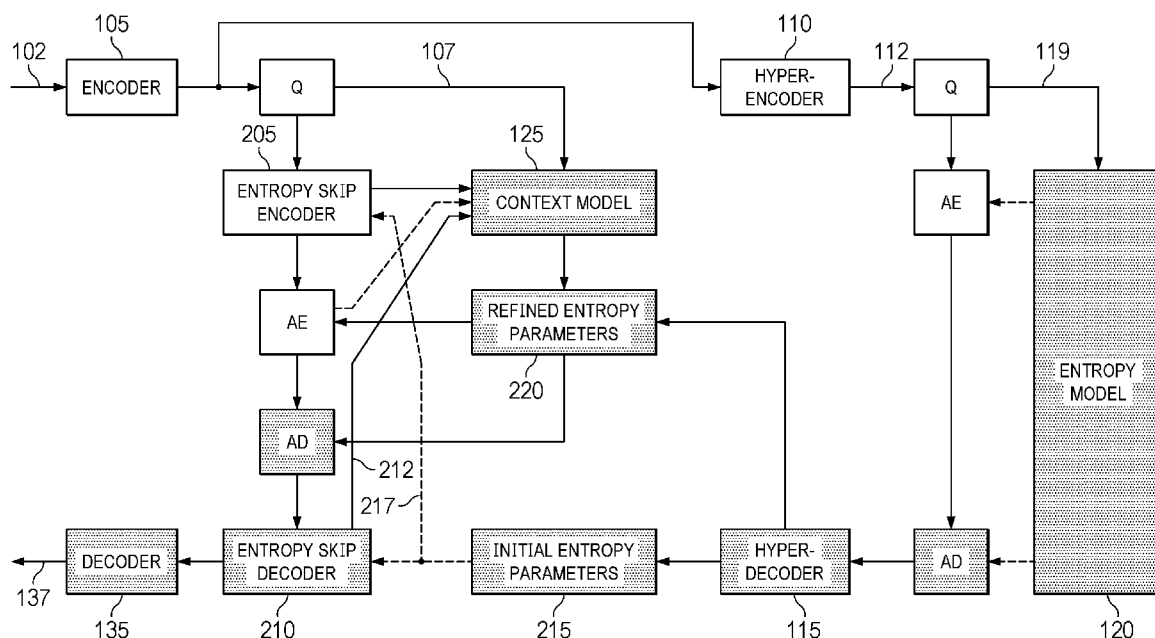


FIG. 2

(57) Abstract: Methods are described for determining thresholds to skip entropy coding of latent features in image and video coding using a neural network. A threshold based on a mean of estimated standard deviations of all latents is proposed. For autoregressive neural networks, to avoid drift between context model parameter estimates computed during training and inference, it is proposed to have two separate entropy parameter estimation networks: an initial entropy-estimation network, to be used in entropy skip encoding and decoding of quantized latents, and a refined entropy-estimation network, to be used in arithmetic encoding and decoding of the quantized latents. Methods and systems for multi-stage entropy skipping are also proposed.

NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

THRESHOLD MODELS FOR SKIPPING ENTROPY CODING IN END-TO-END IMAGE COMPRESSION USING NEURAL NETWORKS

CROSS-REFERENCE TO RELATED APPLICATIONS

- 5 [0001] This patent application claims priority to Indian Provisional Patent Application No. 202311038186, filed on June 2, 2023 and Indian Provisional Patent Application No. 202311061566, filed September 13, 2023, each of which is hereby incorporated by reference in its entirety

10 TECHNOLOGY

[0002] The present document relates generally to images. More particularly, an embodiment of the present invention relates to thresholding models for skipping entropy coding in image and video compression using neural networks.

15 BACKGROUND

[0003] In 2020, the MPEG group in the International Standardization Organization (ISO), jointly with the International Telecommunications Union (ITU), released the first version of the Versatile Video Coding standard (VVC), also known as H.266. More recently, the same group has been working on the development of the next generation coding standard that
20 provides improved coding performance over existing video coding technologies. As part of this investigation, coding techniques based on artificial intelligence and deep learning are also examined. As used herein the term “deep learning” refers to neural networks having at least three layers, and preferably more than three layers.

[0004] As used herein the term “end-to-end image compression neural network” refers to a
25 neural network that jointly optimizes all components of a video compression system, from the point of input of uncompressed input to the encoder, to the point of output of a reconstructed output image of the decoder, using a single loss function.

[0005] As appreciated by the inventors here, improved techniques for the coding of images and video based on neural networks are described herein.

30 [0006] The approaches described in this section are approaches that could be pursued, but not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated, it should not be assumed that any of the approaches described in this section qualify as prior art merely by virtue of their inclusion in this section. Similarly, issues

identified with respect to one or more approaches should not assume to have been recognized in any prior art on the basis of this section, unless otherwise indicated.

BRIEF DESCRIPTION OF THE DRAWINGS

- 5 [0007] An embodiment of the present invention is illustrated by way of example, and not in way by limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:
- [0008] FIG. 1 depicts an example neural-networks model for image and video coding based on the probabilistic modeling of latent features according to prior art; and
- 10 [0009] FIG. 2 depicts a first example neural network model for image and video coding for skipping entropy coding according to an embodiment of this invention;
- [00010] FIG. 3 depicts a second example neural network model for image and video coding for skipping entropy coding according to an embodiment of this invention;
- [00011] FIG. 4A depicts an example process for a two-stage entropy encoding according to
15 an embodiment of this invention;
- [00012] FIG. 4B depicts an example process for a two-stage entropy encoding with entropy skipping according to an embodiment of this invention;
- [00013] FIG. 5A depicts an example of dividing the latent space using a quadtree partitioning according to an embodiment of this invention;
- 20 [00014] FIG. 5B depicts an example of a choice-mask to be used with entropy skipping according to an embodiment of this invention; and
- [00015] FIG. 6 depicts an example process for multi-stage entropy skipping according to an embodiment of this invention.

25 DESCRIPTION OF EXAMPLE EMBODIMENTS

- [00016] Example embodiments that relate to skipping entropy coding (also to be referred as “entropy skipping”) in image and video coding using neural networks are described herein. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the various embodiments of present
30 invention. It will be apparent, however, that the various embodiments of the present invention may be practiced without these specific details. In other instances, well-known structures and devices are not described in exhaustive detail, in order to avoid unnecessarily occluding, obscuring, or obfuscating embodiments of the present invention.

SUMMARY

[00017] Example embodiments described herein relate to image and video coding using neural networks. Methods are described for determining thresholds to skip entropy coding of latent features in image and video coding using a neural network. In a first embodiment, a threshold based on a mean of estimated standard deviations of all latents is proposed. In another embodiment, quantized latents with absolute value less than a threshold are replaced by a very small positive value and their entropy-encoding is skipped. For autoregressive neural networks, to avoid drift between estimates of context model parameter computed during training and inference, it is proposed to have two separate entropy parameter estimation networks: an initial entropy-estimation network, to be used in skipping entropy encoding and decoding of quantized latents, and a refined entropy-estimation network, to be used in arithmetic encoding and decoding of the quantized latents. Embodiments for multi-stage entropy skipping are also proposed.

15

EXAMPLE END-TO-END VIDEO CODING MODEL

[00018] FIG. 1 depicts an example process (100) for modeling image and video coding based on neural learning and the coding of latent features (Refs. [1-2]). As used herein, the terms “latent features” or “latent variables” denote features or variables that are not directly observable but are rather inferred from other observable features or variables, e.g. by processing the directly observable variables. In image and video coding, the term ‘latent space’ may refer to a representation of the compressed data in which similar data points are closer together. In video coding, examples of latent features include the representation of the transform coefficients, the residuals, the motion representation, syntax elements, model information, and the like. In the context of neural networks, latent spaces are useful for learning data features and for finding simpler representations of the image data for analysis.

25

[00019] As depicted in FIG. 1, given input images 102, there are two main sub-networks: an autoencoder, which learns a quantized latent representation of the images (Encoder 105 and Decoder 135 blocks), and a sub-network which is responsible for learning a probabilistic model over quantized latents ($\hat{y}$) (107) used for entropy coding. The subnetwork combines the Context Model (125), an autoregressive model over latents, with the hyper-network (Hyper Encoder and Hyper Decoder blocks 110 and 115), which learns to represent information useful for correcting the context-based predictions. The data from these two sources is combined by an

30

entropy modeling network (130), which generates parameters (e.g., the mean and variance) for a conditional Gaussian entropy model.

[00020] In FIG. 1, the arithmetic encoding (AE) blocks yield the compressed representation of the latent symbols ($\hat{y}$) (107) coming from the quantizer (Q), which is stored in a file.

5 Therefore, at decoding time, any information that depends on the quantized latents may be used by the decoder once it has been decoded. In order for the context model 125 to work, it can only access previously decoded latents.

[00021] This model jointly optimizes an autoregressive component that predicts latents (y) from their causal context (Context Model 125) along with a hyperprior and the underlying autoencoder. Real-valued latent representations are quantized (Q) to create quantized integer-valued latents ($\hat{y}$) (107) and quantized hyperlatents ($\hat{z}$) (119), which are compressed into a bitstream using an arithmetic encoder (AE) and are decompressed by an arithmetic decoder (AD). Blocks with a cross-hatch background correspond to the components that are executed by the receiver to reconstruct an image (137) from a compressed bitstream.

15 [00022] As discussed in Ref. [1], a hierarchical prior (or hyperprior) z (112) is used to improve the entropy model of the latents by capturing their spatial dependencies. Such a model allows for end-to-end training, which includes joint optimization of a quantized representation of the hyperprior, the conditional entropy model, and the base autoencoder. Under this model, the compressed hyperpriors may be added to the generated bitstream as side information, which
20 allows the decoder to use the conditional entropy model. In this way, a separate entropy model (120) of the hyperpriors allows for a richer and more accurate model.

[00023] The training goal is to minimize the expected length of the bitstream as well as the expected distortion of the reconstructed image with respect to the original, giving rise to a rate-distortion (R/D) optimization problem:

$$25 \quad R + \lambda D, \quad (1)$$

where λ is the Lagrange multiplier that determines the desired rate-distortion (RD) trade-off, and R and D may be expressed as:

$$R = E_{x \sim p_x}[-\log_2 p_{\hat{y}}(q(f(x)))], \quad (2)$$

$$D = E_{x \sim p_x}[d(x, g(q(f(x))))],$$

30 where p_x denotes an unknown distribution of natural images, $q(\cdot)$ represents rounding to the nearest integer, $y = f(x)$ denotes an encoder output, $\hat{y} = q(y)$ represents the quantized latents, $p_{\hat{y}}$ is a discrete entropy model, and $\hat{x} = g(\hat{y})$ is the decoder output where $\hat{x}$ represents the reconstructed image. The rate term corresponds to the cross entropy between the marginal

distribution of the latents and the learned entropy model, which is minimized when the two distributions are identical. The distortion term may correspond to a closed-form likelihood, such as when $d(x, \hat{x})$ represents a measure of distortions, such as the Mean Squared Error (MSE), a Structural Similarity Image Measure (SSIM), Multiscale Structural Similarity (MS-SSIM), IW-SSIM (Information Content Weighted Structural Similarity Measure), FSIM (Feature Similarity Index Measure), PSNR-HVSM (Peak Signal to Noise Ratio Human Visual System Measure, taking into account a Contrast Sensitivity Function (CSF) and between-coefficient contrast masking of DCT basis functions), VMAF (Video Multi-method Assessment Fusion), VIF (Visual Information Fidelity measure), VDP2 (Visual Difference Predictor), NLPD (Normalized Laplacian Pyramid Distortion), and a learning-based distortion measure such as LPIPS (Learned Perceptual Image Patch Similarity), DISTIS (Deep Image Structure and Texture Similarity), and the like.

[00024] As in Ref. [1], because both the compressed latents and the compressed hyper-latents are part of the generated bitstream, the rate-distortion loss from equation (1) may be expanded to include the cost of transmitting $\hat{z}$. Coupled with the distortion metric D , the full loss function becomes:

$$R + \lambda D + R_z, \quad (3)$$

where

$$R_z = E_{x \sim p_x}[-\log_2 p_z(\hat{z})] \quad (4)$$

denotes the rate due to the hyper-latents.

[00025] This kind of end to end deep learning compression-based framework using neural networks in general consists of two parts: the core autoencoder and an entropy sub-network. The core autoencoder is used to learn a quantized latent vector of the input image or video signal. For this aspect, the key is to how to define an efficient neural-nets (NN) architecture. The entropy sub-network is responsible for learning a probabilistic model over the quantized latent representations, utilized for entropy coding. For this aspect, finding the right entropy model is very critical to reduce the bitrate overhead. Embodiments of the invention herein propose a new entropy modeling of latent features.

[00026] In Ref. [1], the latent features $\hat{y}$ are modeled by a Gaussian distribution $N(\mu, \sigma^2)$ with mean μ and standard deviation (or scale) σ . The mean and the standard deviation (which may also be referred to as a scale parameter) are estimated jointly using the auto-regressive context model parameters Φ (127) from previously reconstructed latents $\hat{y}_{<i}$ and learned hyperprior features parameters Ψ (117) derived from the hyperprior latents $\hat{z}$ coded in the

bitstream as side information. The distribution of $\hat{y}_i$ latents is considered independently-conditioned on the hyperpriors and the context model.

Entropy Skipping

5

[00027] As discussed in Ref. [3], in video coding, due to the use of reference frames, the entropy of the quantized latents is relatively small, thus having distribution functions with relatively small variance. It was proposed to skip entropy-encoding and entropy-decoding certain elements and simply replace them with the peak of their probability distribution. This can save both bit-rate and runtime in entropy encoding with little error expectations.

10

[00028] In Ref. [3], it was proposed to skip encoding those latents y_i which have high enough probability, q_i , that they are within ± 0.5 of the center of their distribution. Using the erf() function

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt, \quad (5)$$

15

given the standard deviation σ_i , one computes the probability

$$q_i = \text{erf}\left(\frac{1}{2\sqrt{2}\sigma_i}\right). \quad (6)$$

Given this probability, one can apply a threshold τ to decide whether to encode or skip this element. That decision is described as

$$\hat{y}_i = \begin{cases} \mu_i, & \text{if } q_i \geq \tau \\ [y_i], & \text{if } q_i < \tau \end{cases}, \quad (7)$$

20

where μ_i is the center (mean) of the Gaussian distribution, σ_i is the standard deviation, $\tau \in [0, 1]$, for $i = 1, 2, \dots, N$, N denoting the total number of latents. One can observe that q_i is a monotonic function of σ_i so one can instead replace the error function calculation with a specially chosen threshold value for σ_i , denoted here as τ_σ , which is associated with the value chosen for τ (e.g., $\tau = 0.8$), or

25

$$\hat{y}_i = \begin{cases} \mu_i, & \text{if } \sigma_i < \tau_\sigma \\ [y_i], & \text{if } \sigma_i \geq \tau_\sigma \end{cases}, \quad (8)$$

where $[]$ denotes the quantization operation.

Threshold based on $\text{mean}(\sigma_i)$

30

[00029] In an embodiment, it is proposed to improve upon the threshold proposed in Ref. [3] by taking into account all latent distributions. Rather than relying on an absolute threshold decided by the probability of y_i , instead, the proposed threshold relies on local thresholding and encourages the network to discard ~50% of the latents and skip their encoding. This is done by calculating the mean standard deviation term within the current set of latents and determining the skip threshold based on that value. Based on experiments, this results in ~60-70% of the latents being skipped. This entropy skipping model can be expressed as:

$$\hat{y}_i = \begin{cases} \mu_i, & \text{if } \sigma_i < \text{mean}(\sigma_i) \\ [y_i], & \text{if } \sigma_i \geq \text{mean}(\sigma_i) \end{cases}, \quad (9)$$

10 where, for N latents

$$\text{mean}(\sigma_i) = \frac{1}{N} \sum_1^N \sigma_i.$$

For example, given input video patches of size $W \times H$, and a CNN encoder network with M output channels, $N = W/k \times H/k \times M$, where k denotes a down-sampling ratio of the encoder, e.g., $W=H=256$, $M = 64$ or 192 , and $k = 16$.

[00030] In an alternative embodiment, one determines which values to skip depending on the absolute magnitude of each y_i . In this method, if the absolute value of y_i is below some threshold τ , its value is not encoded into the bitstream and instead it is replaced with some non-zero epsilon (e.g., $\varepsilon = 1E^{-6}$). Since each latent is in a feature space, this method is analogous to DCT quantization, used commonly in the JPEG standard. Alternatively, since y_i values are not available in the decoder, the threshold may be defined in terms of the estimated mean, hence,

25

$$\hat{y}_i = \begin{cases} [y_i], & \text{if } |\mu_i| \geq \tau \\ \varepsilon * \text{sign}(\mu_i), & \text{if } |\mu_i| < \tau \end{cases}, \quad (10)$$

where $\text{sign}(x) = 1$ if x is positive and -1 if x is negative, and threshold τ can be optimized via experimental results.

30

Entropy skipping in auto-regressive models

[00031] In an auto-regressive network, as the one depicted in FIG. 1, the entropy (or context) model of the latents (125) is conditioned on the previously decoded latents in addition to the hyper-prior parameters (117). For example, in FIG. 1, this is depicted by a link from the AD block prior to decoder (135) to the context model (125). Since the conditioning in the entropy model uses more context, an autoregressive model is significantly more efficient in the compression of latents.

[00032] For such a neural network model, a direct application of the entropy-skipping scheme of equation (9) would require computing the entropy skipping threshold, e.g., $mean(\sigma_i)$, in an auto-regressive fashion, where the mean is computed over the previously decoded standard deviations. However, this approach does not work in practice. During training, for computational efficiency reasons, the auto-regressive computation of means (μ_i) and scales (that is, standard deviations σ_i) is to be done in parallel for all the latents using a masked convolution, the mask restricting the latents being used in the convolution to only those latents that were previously decoded. Since some of those latents would be skipped in the latent coding process, and the skip decision depends on the scales, this causes a chicken and egg situation in that the computation of the means and scales depends on the knowledge of the scales. If this is not taken care and all the latents are used in the computation of standard deviations (scales) and means without accounting for any latent skips, the model may show significant performance improvement during training, but the inference would show poor coding efficiency. This is due to some information leakage that happens in the training path of the network. The information leakage is due to not replacing the skipped latents with means in the computation of the output context parameters (127) in the context model (125). The training ends up using information not available during inference. During inference, the decoder does not have access to the skipped latents and hence it cannot reproduce the scales and means used during the encoding; furthermore, the skip decisions computed from the scales also cannot be retrieved. This causes a divergence between the encoder and the decoder.

[00033] In an embodiment, this problem is addressed by modifying the original neural network to first produce an initial set of means and scales from the hyperprior information alone. FIG. 2 depicts the block diagram of the system. Compared to FIG. 1, the new network includes an entropy skip encoder (205) and an entropy skip decoder (210). It also replaces the original Gaussian entropy parameters estimation block (130) with two similar blocks: an initial entropy parameters estimation block (215) and a refined entropy parameters estimation block (220). The initial entropy parameters estimation block (215) estimates initial means and scales (μ_i^0, σ_i^0) to be used in entropy skipping. The output of the hyper-decoder (115) is also applied

to a new refined-entropy parameters estimation block (220) to estimate refined means and scales (μ_i^r, σ_i^r) to be used in arithmetic encoding (AE) and arithmetic decoding (AD) of the quantized latents. In an embodiment, without limitation, the skip decisions are now made using the $mean(\sigma_i)$ method, wherein this scale-mean is obtained from the initial scales σ_i^0 (217), and the entropy-skipped latents are replaced with the initial means estimates μ_i^0 . The latent maps obtained from the entropy skip encoder (205) are used in context model block 125 for the computation of the context parameters, which are subsequently being used to compute in block 220 the refined means and scales. The refined means and scales are then used to get the probability distributions for coding the latents that are not skipped. At the decoder, the entropy skip decoder (210) provides latent maps (212) to the context modeler (125) to generate the context model during inference.

[00034] Experimental results confirmed the benefits of this approach. The training and inference paths give results that are in agreement, and the resulting codec is significantly more efficient than the codec without entropy skip. As discussed, the codec now has two entropy parameters networks, one that generates the initial means and scales and the other that generates refined means and scales. The network is trained by minimizing the rate-distortion loss from equation (3) where the rate R is the entropy of the autoencoder latents under the probability model using the refined mean and scale parameters. Since the initial means and scales are not directly affecting the rate-distortion loss, to ensure that the initial entropy parameters network is also properly trained, the rate R is modified to be a weighted average of the entropies of the autoencoder latents under the probability models using the refined and the initial mean and scale parameters. For example this weighting can be expressed as:

$$R = w * E_{x \sim p_x}[-\log_2 p_y^0(q(f(x)))] + (1-w) * E_{x \sim p_x}[-\log_2 p_y^r(q(f(x)))] \quad , (11)$$

where w denotes a weight in $[0,1]$, p_y^0 is an initial discrete entropy model and p_y^r is a refined entropy model. It was found that the performance of the codec is not particularly sensitive to the value of w , and a value of $w = 0.5$ was empirically found to work well.

[00035] During inference, the decoder first decodes the hyper latents and generates the initial means and scales using the initial entropy parameters network (215). The latent skip decisions are obtained from the initial scales. For example, using the model of equation (9), given initial means and scales (μ_i^0, σ_i^0) , every missing quantized latent in the stream of coded latents will be assigned a value of $\hat{y}_i = \mu_i^0$.

[00036] The bitstream contains only the coded latents which are not skipped by the encoder. The entropy skip decoder (210) takes in the arithmetic decoded latents from the bitstream and fills in the skipped latents with the corresponding initial means estimates to produce the output latents so far decoded, which is used by the context model (125) as the context input. The refined entropy parameter network (220) generates the refined means and scales using the output of the context model (125) along with the hyper prior parameters. Refined means and scales are then used to produce the entropy model used by the arithmetic decoder (AD) to decode subsequent latents from the bitstream in an autoregressive fashion.

10 Entropy skipping in multi-stage entropy models

[00037] Current state of the art neural network-based compression models (Refs.[4-5]) have opted to relax the autoregressive entropy coding approach due to run-time constraints. The autoregressive model, while offering the best performance, is not parallelizable. A popular alternative approach for entropy models is a multi-stage entropy model. These multi-stage models are like auto-regressive entropy models but with a slight difference. Instead of encoding each latent element in a sequential manner conditioned on the previous latent element, multi-stage models partition the latent into multiple blocks and are encoded in parallel.

[00038] This partitioning strategy is commonly known as two stage checkerboarding or four stage quadtree partitioning, although alternative partitioning trees with eight or more partitions can also be employed. The checkboard partitioning is simply dividing the latent tensor into odd and even spatial elements and then splitting the channels in half. This results in a partition with odd elements, with one half of the channels, and another partition with the remaining half of the channels with the even elements. The selected latents are then encoded with the predicted $\mu_{i,0}, \sigma_{i,0}$. Using the first stage entropy parameters $\mu_{i,0}, \sigma_{i,0}$ and the processed latent $\hat{y}_i$ are then used to compute the second stage entropy parameters $\mu_{i,1}, \sigma_{i,1}$ for the remaining latents that have yet to be encoded. The quadtree partitioning is similar to the checkerboard partition strategy but divides the channels into four partitions and partitions the latent spatially into a quadtree structure. More details can be found in Refs. [4,5] and will be discussed later in this section.

[00039] FIG. 3 depicts an example block diagram of such system (300). Compared to FIG. 1, the output 107 of the quantizer (Q) is now fed to an optional temporal feature model block (305) and then to a checkerboard or quadree entropy parameters block (310), replacing the

context model (125) and Gaussian entropy parameters (130) blocks, but also removing the need for a feedback from the decoder AD back to the context model (125).

[00040] The role of the temporal feature model block is to add temporal information (or modeling parameters), such as motion vector information, when encoding video sequences.

5 Thus, while for still images or the first frame in a video sequence this block is not used, its use in subsequent frames in a video sequence adds additional modeling information which can improve the estimation of the entropy parameters.

[00041] In traditional, “single frame” entropy encoding, given a sequence of quantized latents (subsequently, for simplification, to be referred to as just latents) (405), one computes
10 the entropy parameters μ and σ for the whole “frame,” to generate a coded bitstream. FIG. 4A depicts an example of checker-board partitioning which allows for parallizing the entropy encoding process across “white” (407) and “black” (409) areas of the latents of frame (405). The depiction of only four latents in frame 405 is simply an example and not limiting. As depicted in FIG. 4A, a first stage bitstream (410) can be generated based only on the white
15 latents (407) and their corresponding μ and σ stage 1 entropy parameters (412). A second stage bitstream (430) can be generated based only on the black latents (409); however, the second stage-entropy parameters (422) are conditioned based on the first-stage entropy parameters (412), the hyper latents, temporal context (if available), and previous latents (black or white).

[00042] FIG. 4B depicts an example of a two-stage entropy coding model using entropy
20 skipping. As depicted, for stage one, given the entropy parameters (μ , σ) (412) and the corresponding thresholds (e.g., see equation (9)) one computes a Boolean choice-mask (435) in the black and white latents determining which of the white latents can be entropy skipped.

[00043] In general, each latent has an associated mean and scale parameter. In an embodiment, for each set of stage latents (e.g. in FIG. 4B, stage 1: latents A and D, stage 2:
25 latents B and C) one takes the average scale (variance) of all latents, either in that stage or across all latents. Next, if a latent scale in the stage is above the average scale, then one encodes the latent otherwise it is entropy-skipped. Typically, one never entropy-skips all latents in a stage, but one is often skipping a large majority of the elements in each stage. As an example, in FIG. 4B, based on choice mask 435, for stage 1, one entropy-skips latent D and entropy
30 encodes latent A. For stage 2, based on choice mask 440, one only encodes latent C and entropy skips latent B.

[00044] Choice masks identify to the encoder and the decoder what values are kept and what values are skipped and need to be recomputed. In general, in each stage, a choice mask value for a specific latent is set to zero when its scale value is less than the computed average of all

variances of the elements in that stage, otherwise it is set to 1, signifying that it is not entropy-skipped. Note that choice masks don't need to be sent from an encoder to the decoder. A decoder has all information needed to compute the mean and scale entropy parameters, and corresponding masks, based on the hyper-latents and optional temporal context and past decoded latents. A decoder will simply replace skipped latents with the estimated mean of each stage.

[00045] Expanding the methodology to a quadtree partitioning, FIG. 5A depicts a division of the latent space (505) into a quadtree representation with quantized latent areas (or stages) 1, 2, 3, and 4. In such a scenario, one would start by entropy encoding the "1" latents, use the updated context information to entropy encode the "2" latents, use the updated context information to encode the "3" latents, and finally encode the "4" latents. Again, the depiction of only 16 latents (505) is an example and is not limiting. Once all four latent areas are encoded, one can transmit the four bitstreams. When using entropy skipping, the Boolean choice masks (e.g., 435, 440) need to be adapted to support the new quadtree structure. An example of such a choice mask is depicted in FIG. 5B, where, without limitation, one selects to entropy encode two latents in the "1" stage and skip the other two, or in a (row, column) representation: encode latents (1,1) and (3,3) and skip encoding latents (1,3) and (3,1). For example, after encoding the "1" stage, one continues with the stage "2" latents, e.g., encoding latents (1,2), (3,2), and (3,4) (skipping latent (1,4)), the stage "3" latents, e.g., encoding latent (2,1) (and skipping latents (2,3), (4,1), and (4,3), and encoding all stage "4" latents, e.g., latents (2,2), (2,4), (4,2) and (4,4) (with no entropy skips).

[00046] In summary, in this type of neural network model, the application of the entropy-skipping scheme of equation (9) requires computing the entropy skipping threshold, e.g., $mean(\sigma_i)$, for each stage in the partitioning strategy. This is due to the gradual refinement of the entropy parameters which are refined at each stage and are conditioned on the previous stage. With these new scale parameters, a new entropy-skipping threshold needs to be computed otherwise there will be a mismatch between the actual entropy parameters and the initial entropy parameters in which the choice mask was computed for. This process is summarized in FIG. 6.

However, the processing of the latents is the same as previously described models, that is, for elements which are skipped, $\hat{y}_i = \mu_{i,m}$.

[00047] As depicted in FIG. 6, given a multi-stage entropy skipping model, the entropy skipping process starts in step 605 with input the quantized latents and hyper-latents and optional temporal-context (if applicable). As an example, the first stage is denoted as stage $j =$

0 (out of K total stages). For example, for a checkerboard division $K=2$, and for a quadtree division $K = 4$.

[00048] For stage $j = 0$, in step 610, one computes the corresponding entropy parameters $(\mu_{i,0}, \sigma_{i,0})$ and the entropy-skipping threshold. Then, in step 615, based on the computed thresholds, one generates a choice mask to select which latents are to be entropy skipped and to generate a coded bitstream for this stage based on the non-skipped latents.

[00049] Subsequently, after the first stage, one detects whether the whole latent space (all K stages) has been encoded. If not, stage-specific bitstreams are generated as follows:

- compute corresponding entropy parameters $(\mu_{i,j}, \sigma_{i,j})$ based on the hyper-latents and $(\mu_{i,j-1}, \sigma_{i,j-1})$, and optionally based on past latents and temporal context (if available) (step 620); then in step (625)
 - computed updated entropy skipping threshold for stage j
 - compute choice-masks to define latents that can be entropy skipped
 - generate an entropy-coded bitstream for stage j using the non-skipped latents

Steps 620 and 625 are repeated until all K stages are coded.

[00050] In a decoder, the process is identical, except that the computed choice masks will be used to regenerate skipped quantized latent elements by simply assigning to them the mean latent value for each stage. Then, the received decoded bitstream and the reconstructed missing elements will be combined to form the final, decoded, bitstream.

[00051] Experimental results have shown that training with multi-stage entropy-skipping results in an increase in the efficiency of the codec. The network is trained by minimizing the rate-distortion loss from equation (3) with a slight difference. Prior to computing the rate term defined in equation (2) the elements which are skipped need to be masked out in order to avoid overestimating the rate. This mask M is formed from the choice mask at each stage merged into a single mask by simply adding them together. The rate term is then computed as the following.

$$R = E_{x \sim p_x} [M * -\log_2 p_{\hat{y}}(*q(f(x)))] \quad (12)$$

References

Each one of the references listed herein is incorporated by reference in its entirety.

- [1] D. Minnen, J. Balle', and G. Toderici. "Joint autoregressive and hierarchical priors for learned image compression." 32nd Conf. on Neural Information Processing Systems (NeurIPS 2018), Montreal, Canada, 2018.
- 5 [2] D. Liu, Y. Li, J. Lin, H. Li, F. Wu, "Deep learning-based video coding: A review and a case study." arXiv:1904.12462v1, April 29, 2019.
- [3] Y. Shi, Y. Ge, J. Wang, J. Mao, "AlphaVC: High-Performance and Efficient Learned Video Compression," arXiv:2207.14678, July 29, 2022.
- [4] Li, Jiahao, Bin Li, and Yan Lu. "Hybrid spatial-temporal entropy modelling for neural video compression." *Proceedings of the 30th ACM International Conference on Multimedia*. 2022.
- 10 [5] Li, Jiahao, Bin Li, and Yan Lu. "Neural video compression with diverse contexts." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

15

EXAMPLE COMPUTER SYSTEM IMPLEMENTATION

[00052] Embodiments of the present invention may be implemented with a computer system, systems configured in electronic circuitry and components, an integrated circuit (IC) device such as a microcontroller, a field programmable gate array (FPGA), or another configurable or

20 programmable logic device (PLD), a discrete time or digital signal processor (DSP), an application specific IC (ASIC), and/or apparatus that includes one or more of such systems, devices or components. The computer and/or IC may perform, control, or execute instructions relating to skipping entropy coding of latent features in image and video coding, such as those described herein. The computer and/or IC may compute any of a variety of parameters or

25 values that relate to skipping entropy coding of latent features in image and video coding described herein. The image and video embodiments may be implemented in hardware, software, firmware and various combinations thereof.

[00053] Certain implementations of the invention comprise computer processors which execute software instructions which cause the processors to perform a method of the invention.

30 For example, one or more processors in a display, an encoder, a set top box, a transcoder, or the like may implement methods related to skipping entropy coding of latent features in image and video coding as described above by executing software instructions in a program memory accessible to the processors. Embodiments of the invention may also be provided in the form of a program product. The program product may comprise any non-transitory and tangible

medium which carries a set of computer-readable signals comprising instructions which, when executed by a data processor, cause the data processor to execute a method of the invention. Program products according to the invention may be in any of a wide variety of non-transitory and tangible forms. The program product may comprise, for example, physical media such as magnetic data storage media including floppy diskettes, hard disk drives, optical data storage media including CD ROMs, DVDs, electronic data storage media including ROMs, flash RAM, or the like. The computer-readable signals on the program product may optionally be compressed or encrypted. Where a component (e.g. a software module, processor, assembly, device, circuit, etc.) is referred to above, unless otherwise indicated, reference to that component (including a reference to a "means") should be interpreted as including as equivalents of that component any component which performs the function of the described component (e.g., that is functionally equivalent), including components which are not structurally equivalent to the disclosed structure which performs the function in the illustrated example embodiments of the invention.

15

EQUIVALENTS, EXTENSIONS, ALTERNATIVES AND MISCELLANEOUS

[00054] Example embodiments that relate to skipping entropy coding of latent features in image and video coding are thus described. In the foregoing specification, embodiments of the present invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is the invention, and what is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions expressly set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

25

[00055] Various aspects of the present disclosure may be appreciated from the following Enumerated Example Embodiments (EEEs):

30

EEE1. In a neural network for video coding, a method for skipping entropy encoding of quantized latents, the method comprising:

inputting video content into a neural-network encoder to generate latents (y_i);
quantizing the latents to generate quantized latents ($\hat{y}_i$);

estimating mean values (μ_i) and standard deviations (σ_i) of the latents; and
 applying entropy encoding to the quantized latents, wherein for a quantized
 latent ($\hat{y}_i$)

- if $\sigma_i < \text{mean}(\sigma_i)$, then $\hat{y}_i = \mu_i$, and its entropy encoding is skipped,
 5 else the quantized latent ($\hat{y}_i$) is entropy encoded to generate an entropy encoded
 latent, wherein $\text{mean}(\sigma_i)$ denotes the average of all σ_i values, for $i = 1, 2, \dots, N$, where N
 denotes the total number of generated latents.

EEE2. In a neural network for video coding, a method for skipping entropy encoding of
 10 quantized latents, the method comprising:

- inputting video content into a neural-network encoder to generate latents (y_i);
 quantizing the latents to generate quantized latents ($\hat{y}_i$);
 estimating mean values (μ_i) of the latents; and
 applying entropy encoding to the quantized latents, wherein for a latent y_i ,
 15 if $|\mu_i| < \tau$, then $\hat{y}_i = \varepsilon * \text{sign}(\mu_i)$ and its entropy encoding is skipped,
 else the quantized latent ($\hat{y}_i$) is entropy encoded to generate an entropy encoded
 latent, wherein τ denotes a threshold, and ε denotes a small positive value larger than zero and
 smaller than 0.001.

20 EEE3. In an autoregressive neural network for video coding, a method for skipping entropy
 encoding of quantized latents, the method comprising:

- inputting video content into a neural-network encoder to generate latents (y_i);
 quantizing the latents to generate quantized latents ($\hat{y}_i$);
 25 estimating initial mean values (μ_i^0) and standard deviations (σ_i^0) of the latents;
 applying the initial mean values (μ_i^0) and initial standard deviations (σ_i^0) of the
 latents to an entropy-skip encoder to determine updated quantized latent values and
 whether a quantized latent value will be entropy encoded or entropy encoding for the
 quantized latent will be skipped;
 30 based at least on the entropy-skipped and non-entropy skipped latents,
 estimating refined mean values (μ_i^r) and standard deviations (σ_i^r) of the latents; and
 applying the refined mean values (μ_i^r) and standard deviations (σ_i^r) of the
 latents for the arithmetic encoding and arithmetic decoding of the latents to be entropy

coded.

EEE4. The method of EEE3, wherein determining updated quantized latent values and whether a quantized latent value will be entropy encoded or entropy encoding will be skipped, comprises:

if $\sigma_i^0 < \text{mean}(\sigma_i^0)$, then $\hat{y}_i = \mu_i^0$, and its entropy encoding is skipped,
 else $\hat{y}_i$ is entropy encoded to generate an entropy encoded latent,
 wherein $\text{mean}(\sigma_i^0)$ denotes the average of all σ_i^0 values, for $i = 1, 2, \dots, N$,
 where N denotes the total number of generated latents.

EEE5. The method of EEE3, wherein determining updated quantized latent values and whether a quantized latent value will be entropy encoded or entropy encoding will be skipped, comprises:

for a latent y_i ,

if $|\mu_i^0| < \tau$, then $\hat{y}_i = \varepsilon * \text{sign}(\mu_i^0)$ and its entropy encoding is skipped,

else the quantized latent ($\hat{y}_i$) is entropy encoded to generate an entropy encoded latent, wherein τ denotes a threshold, and ε denotes a small positive value larger than zero and smaller than 0.001.

EEE6. A method for decoding pictures using a neural network, the method comprising:

receiving a bitstream comprising coded latents and coded hyper-latents;

decoding the hyper-latents to generate hyperprior parameters and initial entropy parameters comprising initial mean values and initial scales;

applying the initial scales into an entropy-skip decoder to identify skipped output latents, wherein the skipped output latents comprise latents that were not entropy-coded by an encoder of the bitstream;

determining values of the skipped output latents based on the initial mean values;

applying the skipped output latents and the hyperprior parameters to a context model to generate refined entropy parameters comprising refined mean values and refined scales;

applying the refined entropy parameters to decode the coded latents and generate non-skipped output latents;

and generating an output decoded picture based at least on the values of the skipped output latents and the non-skipped output latents.

EEE7. In a neural network for video coding, a method for skipping entropy encoding of quantized latents, the method comprising:

- 5 inputting video content into a neural-network encoder to generate latents (y_i);
- quantizing the latents to generate quantized latents ($\hat{y}_i$);
- dividing the quantized latents into K non-overlapping groups of quantized latents;
- estimating mean values (μ_i^0) and standard deviations (σ_i^0) of the quantized latents in group $j=0$ of the K groups based on hyper-latents;
- applying the mean values (μ_i^0) and standard deviations (σ_i^0) of the latents to an
- 10 entropy-skip encoder to determine output latent values for the group $j=0$;
- for quantized latents in group j for groups $j=1$ to $K-1$ of the K groups:
 - estimating mean values (μ_i^j) and standard deviations (σ_i^j) of the quantized latents in the group j based on hyper-latents and the (μ_i^{j-1} , σ_i^{j-1}) values;
 - 15 applying the mean values (μ_i^j) and standard deviations (σ_i^j) of the latents to the entropy-skip encoder to determine output coded latent values for the group j ; and
 - generating a bitstream of encoded quantized latents based on the output coded latent values for the K groups.
- 20

EEE8. The method of EEE7, wherein $K = 2$ or 4 .

EEE9. The method of EEE7 or EEE8, wherein determining output latent values and whether a quantized latent value will be entropy encoded or entropy encoding will be skipped, comprises:

- 25 if $\sigma_i^j < \text{mean}(\sigma_i^j)$, then $\hat{y}_i = \mu_i^k$, and its entropy encoding is skipped,
- else $\hat{y}_i$ is entropy encoded to generate an entropy encoded output latent,
- wherein $\text{mean}(\sigma_i^j)$ denotes the average of all σ_i^j values for quantized latents in group j .

- 30 EEE10. A method for decoding pictures using a neural network, the method comprising:
 - receiving a bitstream comprising coded latents and coded hyper-latents, wherein the coded latents represent quantized latents divided into K non-overlapping groups;

for group $j=0$ of the coded latents:

decoding the hyper-latents to generate hyperprior parameters and initial entropy parameters comprising initial mean values (μ_i^0) and initial scales (σ_i^0);

5 applying the initial scales into an entropy-skip decoder to identify skipped output latents, wherein the skipped output latents comprise latents that were not entropy-coded by an encoder of the bitstream; and

determining values of the skipped output latents for group $j = 0$ based on the initial mean values;

for group $j, j=1$ to $K-1$, of the coded latents:

10 estimating mean values (μ_i^j) and standard deviations (σ_i^j) of the quantized latents in the group j based on the hyper-latents and the ($\mu_i^{j-1}, \sigma_i^{j-1}$) values; and

applying the estimated mean values (μ_i^j) and the standard deviations (σ_i^j) to decode the coded latents and generate skipped and non-skipped output
15 latents.

and generating an output decoded picture based at least on values of the skipped output latents and the non-skipped output latents for all K groups.

EEE11. The method of EEE10, wherein determining decoded skipped and non-skipped
20 quantized latents comprises:

if $\sigma_i^j < \text{mean}(\sigma_i^j)$, then entropy encoding of $\hat{y}_i$ was skipped and $\hat{y}_i = \mu_i^k$,

else $\hat{y}_i$ was entropy encoded and its value is generated by an entropy decoder,

wherein $\text{mean}(\sigma_i^j)$ denotes the average of all σ_i^j values for quantized latents in
25 group j .

EEE12. A non-transitory computer-readable storage medium having stored thereon computer-executable instructions for executing with one or more processors a method in accordance with any one of EEE1 to EEE11.

30 EEE13. An apparatus comprising a processor and configured to perform a method in accordance with any one of EEE1 to EEE11.

CLAIMS

- 5 1. In a neural network for video coding, a method for skipping entropy encoding of quantized latents, the method comprising:
- inputting video content into a neural-network encoder to generate latents (y_i);
- quantizing the latents to generate quantized latents ($\hat{y}_i$);
- estimating mean values (μ_i) and standard deviations (σ_i) of the latents; and
- 10 applying entropy encoding to the quantized latents, wherein for a quantized latent ($\hat{y}_i$)
- if $\sigma_i < \text{mean}(\sigma_i)$, then $\hat{y}_i = \mu_i$, and its entropy encoding is skipped,
- else the quantized latent ($\hat{y}_i$) is entropy encoded to generate an entropy encoded latent, wherein $\text{mean}(\sigma_i)$ denotes the average of all σ_i values, for $i = 1, 2, \dots, N$, where N
- 15 denotes the total number of generated latents.
2. In a neural network for video coding, a method for skipping entropy encoding of quantized latents, the method comprising:
- inputting video content into a neural-network encoder to generate latents (y_i);
- 20 quantizing the latents to generate quantized latents ($\hat{y}_i$);
- estimating mean values (μ_i) of the latents; and
- applying entropy encoding to the quantized latents, wherein for a latent y_i ,
- if $|\mu_i| < \tau$, then $\hat{y}_i = \varepsilon * \text{sign}(\mu_i)$ and its entropy encoding is skipped,
- else the quantized latent ($\hat{y}_i$) is entropy encoded to generate an entropy encoded
- 25 latent, wherein τ denotes a threshold, and ε denotes a small positive value larger than zero and smaller than 0.001.
3. In an autoregressive neural network for video coding, a method for skipping entropy encoding of quantized latents, the method comprising:
- 30 inputting video content into a neural-network encoder to generate latents (y_i);
- quantizing the latents to generate quantized latents ($\hat{y}_i$);
- estimating initial mean values (μ_i^0) and standard deviations (σ_i^0) of the latents;
- applying the initial mean values (μ_i^0) and initial standard deviations (σ_i^0) of the
- 35 latents to an entropy-skip encoder to determine updated quantized latent values and

whether a quantized latent value will be entropy encoded or entropy encoding for the quantized latent will be skipped;

based at least on the entropy-skipped and non-entropy skipped latents, estimating refined mean values (μ_i^r) and standard deviations (σ_i^r) of the latents; and

5 applying the refined mean values (μ_i^r) and standard deviations (σ_i^r) of the latents for the arithmetic encoding and arithmetic decoding of the latents to be entropy coded.

4. The method of claim 3, wherein determining updated quantized latent values and whether a quantized latent value will be entropy encoded or entropy encoding will be skipped, comprises:

if $\sigma_i^0 < \text{mean}(\sigma_i^0)$, then $\hat{y}_i = \mu_i^0$, and its entropy encoding is skipped,

else $\hat{y}_i$ is entropy encoded to generate an entropy encoded latent,

wherein $\text{mean}(\sigma_i^0)$ denotes the average of all σ_i^0 values, for $i = 1, 2, \dots, N$,

where N denotes the total number of generated latents.

15

5. The method of claim 3, wherein determining updated quantized latent values and whether a quantized latent value will be entropy encoded or entropy encoding will be skipped, comprises:

for a latent y_i ,

20 if $|\mu_i^0| < \tau$, then $\hat{y}_i = \varepsilon * \text{sign}(\mu_i^0)$ and its entropy encoding is skipped,

else the quantized latent ($\hat{y}_i$) is entropy encoded to generate an entropy encoded latent, wherein τ denotes a threshold, and ε denotes a small positive value larger than zero and smaller than 0.001.

25 6. A method for decoding pictures using a neural network, the method comprising:

receiving a bitstream comprising coded latents and coded hyper-latents;

decoding the hyper-latents to generate hyperprior parameters and initial entropy parameters comprising initial mean values and initial scales;

30 applying the initial scales into an entropy-skip decoder to identify skipped output latents, wherein the skipped output latents comprise latents that were not entropy-coded by an encoder of the bitstream;

determining values of the skipped output latents based on the initial mean values;

applying the skipped output latents and the hyperprior parameters to a context model to

generate refined entropy parameters comprising refined mean values and refined scales;
 applying the refined entropy parameters to decode the coded latents and generate non-
 skipped output latents;
 and generating an output decoded picture based at least on the values of the skipped
 5 output latents and the non-skipped output latents.

7. In a neural network for video coding, a method for skipping entropy encoding of quantized latents, the method comprising:

inputting video content into a neural-network encoder to generate latents (y_i);
 10 quantizing the latents to generate quantized latents ($\hat{y}_i$);
 dividing the quantized latents into K non-overlapping groups of quantized latents;
 estimating mean values (μ_i^0) and standard deviations (σ_i^0) of the quantized
 latents in group $j=0$ of the K groups based on hyper-latents;
 applying the mean values (μ_i^0) and standard deviations (σ_i^0) of the latents to an
 15 entropy-skip encoder to determine output latent values for the group $j=0$;
 for quantized latents in group j for groups $j=1$ to $K-1$ of the K groups:
 estimating mean values (μ_i^j) and standard deviations (σ_i^j) of the
 quantized latents in the group j based on hyper-latents and the (μ_i^{j-1} , σ_i^{j-1})
 values;
 20 applying the mean values (μ_i^j) and standard deviations (σ_i^j) of the
 latents to the entropy-skip encoder to determine output coded latent values for
 the group j ; and
 generating a bitstream of encoded quantized latents based on the output coded
 latent values for the K groups.

25

8. The method of claim 7, wherein $K = 2$ or 4.

9. The method of claim 7 or 8, wherein determining output latent values and whether a
 quantized latent value will be entropy encoded or entropy encoding will be skipped, comprises:

30 if $\sigma_i^j < \text{mean}(\sigma_i^j)$, then $\hat{y}_i = \mu_i^k$, and its entropy encoding is skipped,
 else $\hat{y}_i$ is entropy encoded to generate an entropy encoded output latent,
 wherein $\text{mean}(\sigma_i^j)$ denotes the average of all σ_i^j values for quantized latents in

group j .

10. A method for decoding pictures using a neural network, the method comprising:

receiving a bitstream comprising coded latents and coded hyper-latents, wherein the
5 coded latents represent quantized latents divided into K non-overlapping groups;

for group $j=0$ of the coded latents:

decoding the hyper-latents to generate hyperprior parameters and initial
entropy parameters comprising initial mean values (μ_i^0) and initial scales (σ_i^0);

applying the initial scales into an entropy-skip decoder to identify skipped
10 output latents, wherein the skipped output latents comprise latents that were not entropy-coded
by an encoder of the bitstream; and

determining values of the skipped output latents for group $j = 0$ based on the
initial mean values;

for group $j, j=1$ to $K-1$, of the coded latents:

15 estimating mean values (μ_i^j) and standard deviations (σ_i^j) of the
quantized latents in the group j based on the hyper-latents and the ($\mu_i^{j-1}, \sigma_i^{j-1}$)
values; and

applying the estimated mean values (μ_i^j) and the standard deviations (σ_i^j)
20 to decode the coded latents and generate skipped and non-skipped output
latents.

and generating an output decoded picture based at least on values of the skipped output
latents and the non-skipped output latents for all K groups.

11. The method of claim 10, wherein determining decoded skipped and non-skipped quantized
25 latents comprises:

if $\sigma_i^j < \text{mean}(\sigma_i^j)$, then entropy encoding of $\hat{y}_i$ was skipped and $\hat{y}_i = \mu_i^k$,

else $\hat{y}_i$ was entropy encoded and its value is generated by an entropy decoder,

wherein $\text{mean}(\sigma_i^j)$ denotes the average of all σ_i^j values for quantized latents in

group j .

30

12. A non-transitory computer-readable storage medium having stored thereon computer-executable instructions for executing with one or more processors a method in accordance with any one of claims 1-11.
- 5 13. An apparatus comprising a processor and configured to perform a method in accordance with any one of claims 1-11.

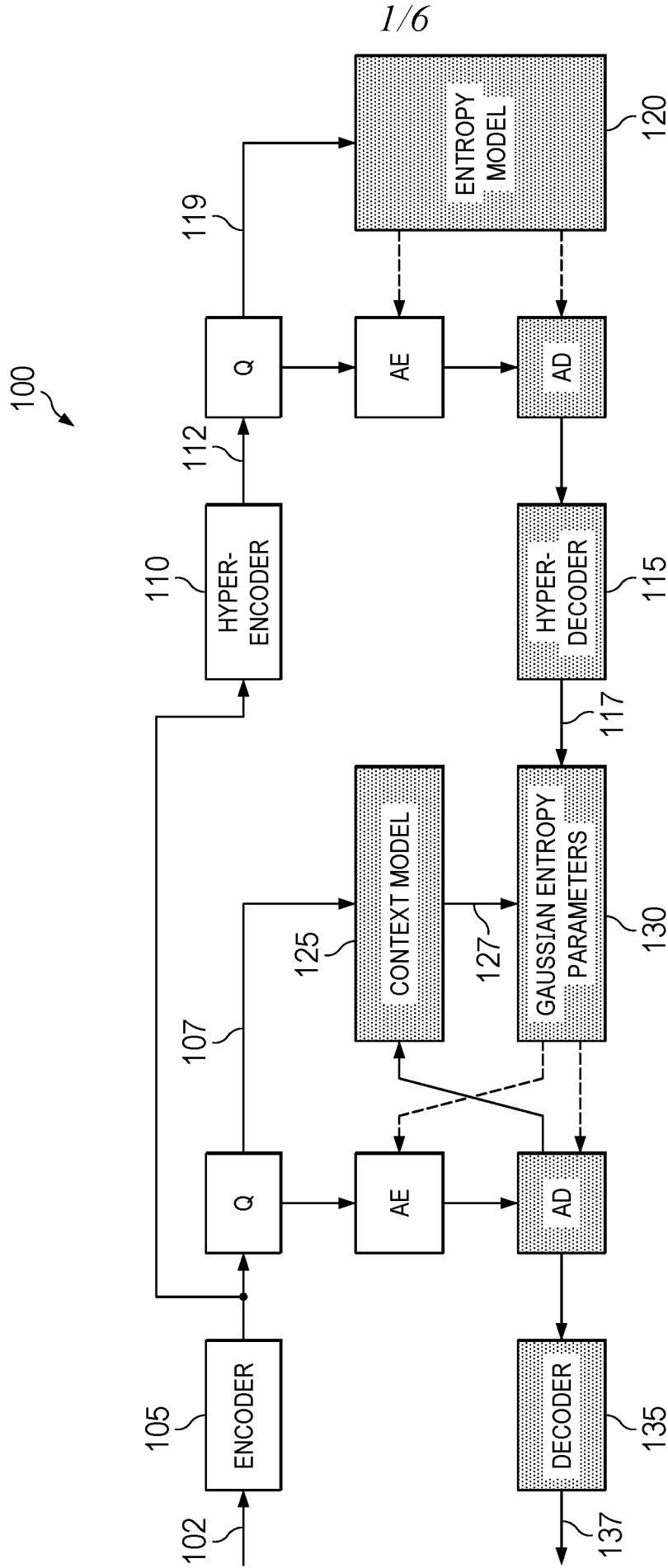


FIG. 1
(PRIOR ART)

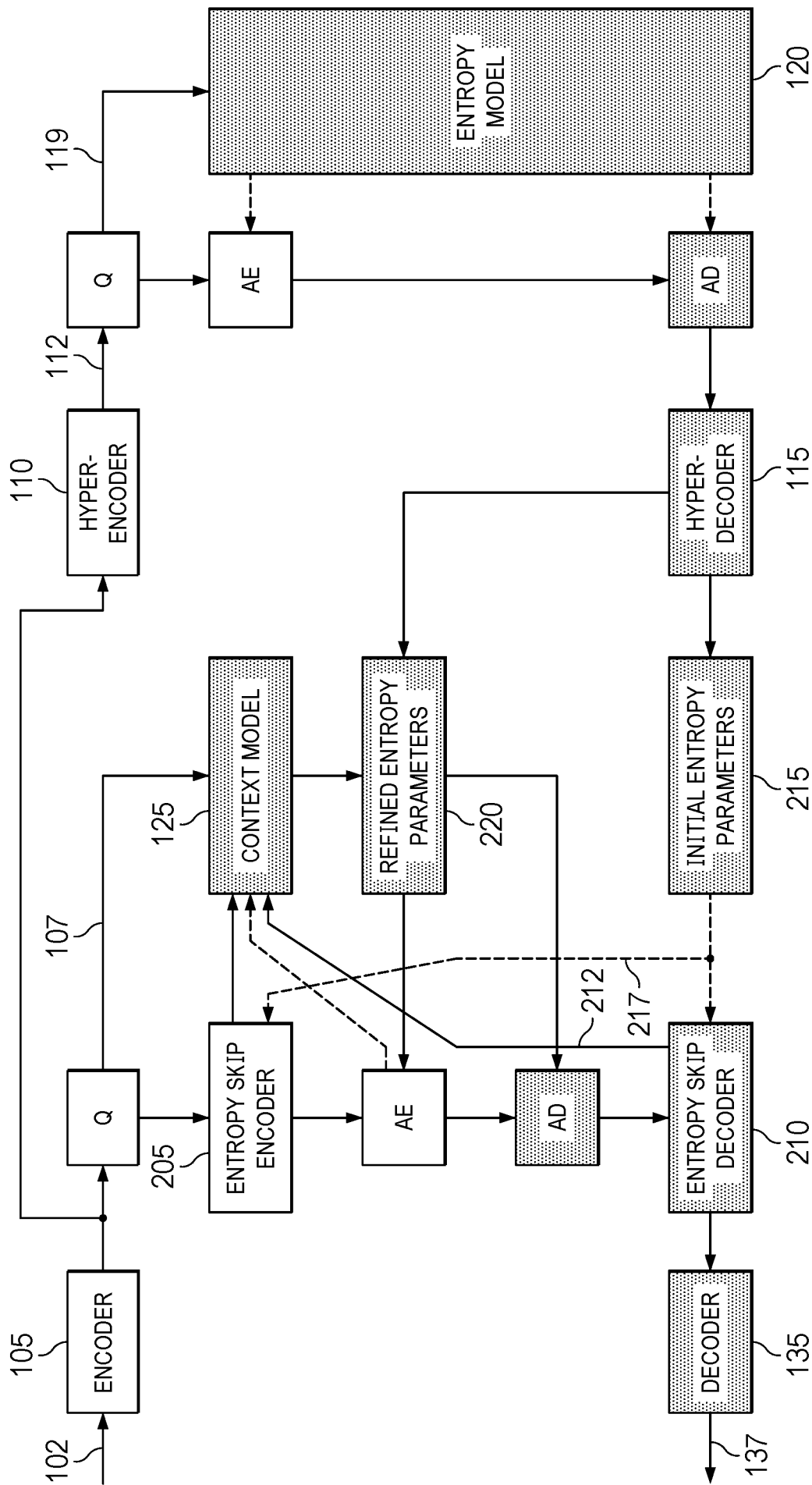


FIG. 2

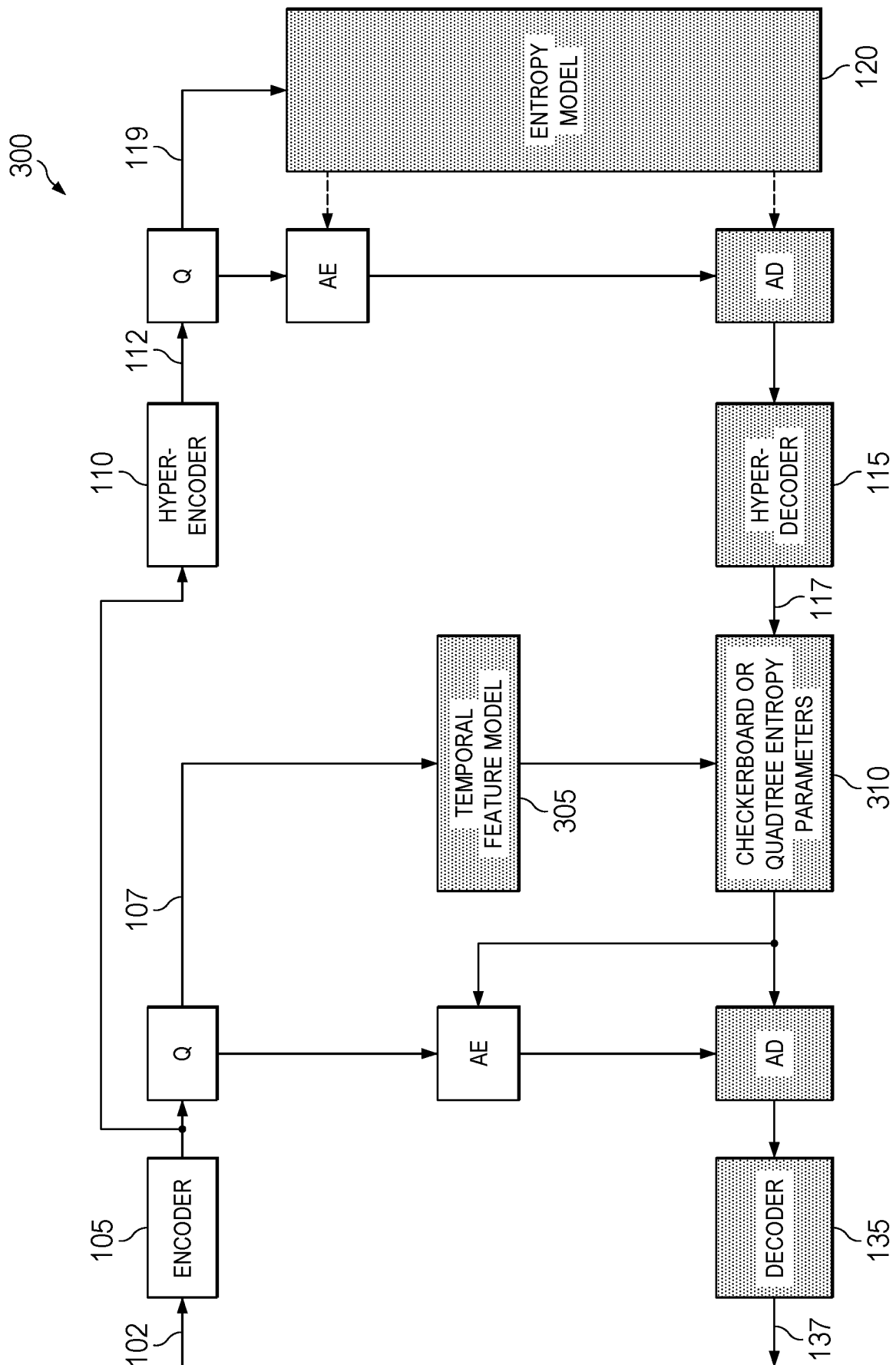


FIG. 3

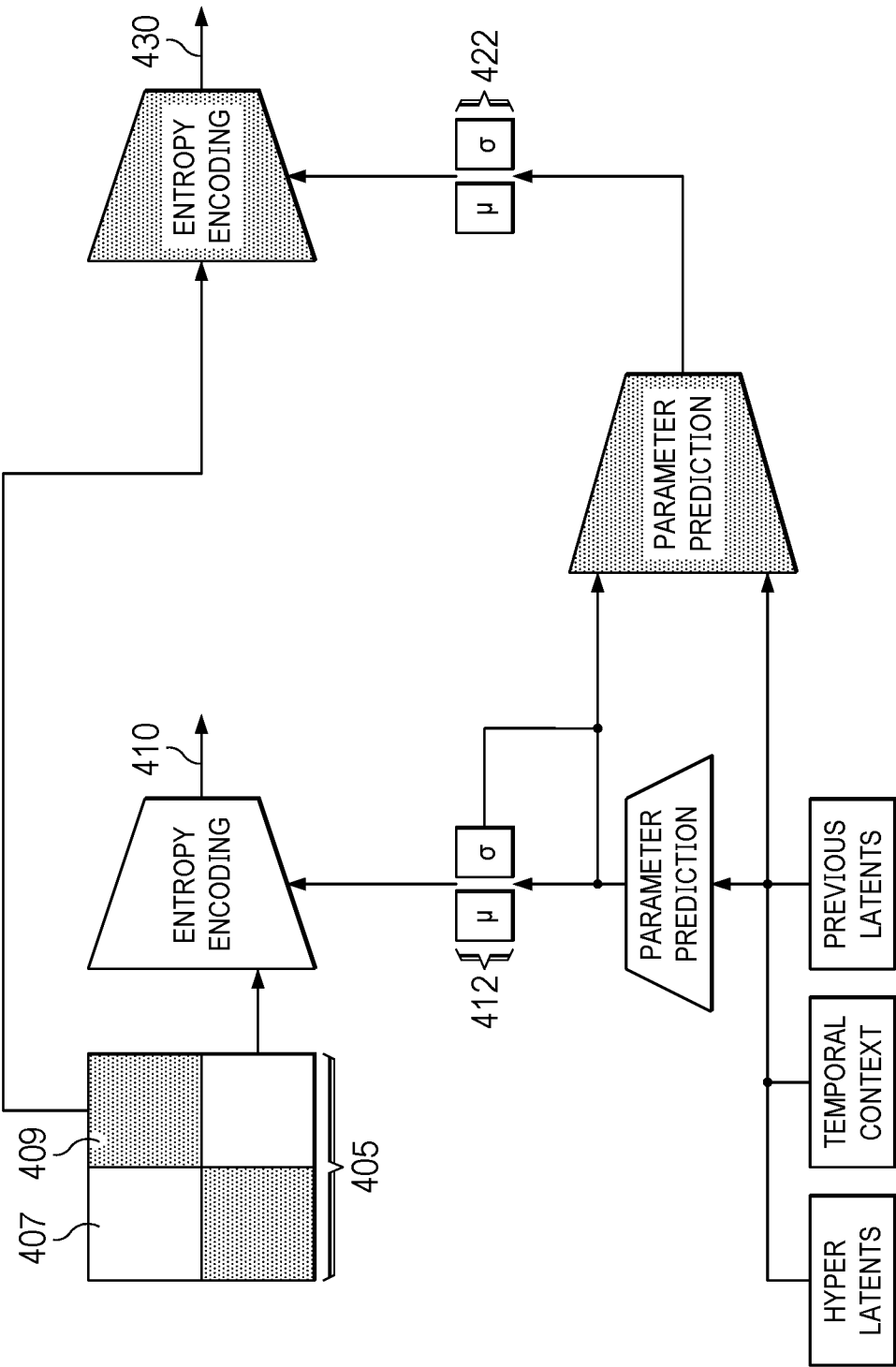


FIG. 4A

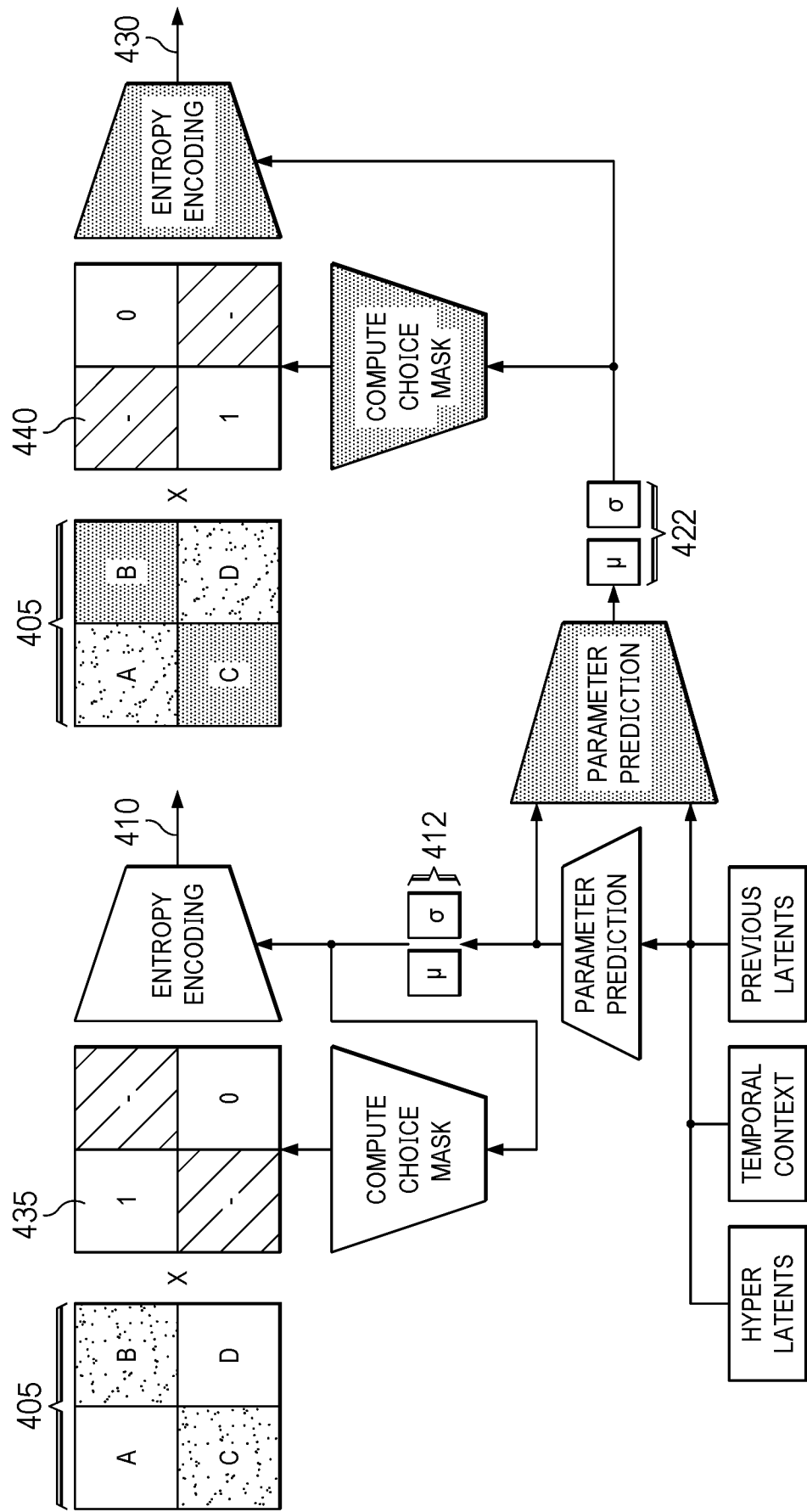


FIG. 4B

505

1	2	1	2
3	4	3	4
1	2	1	2
3	4	3	4

FIG. 5A

1	-	0	-
-	-	-	-
0	-	1	-
-	-	-	-

FIG. 5B

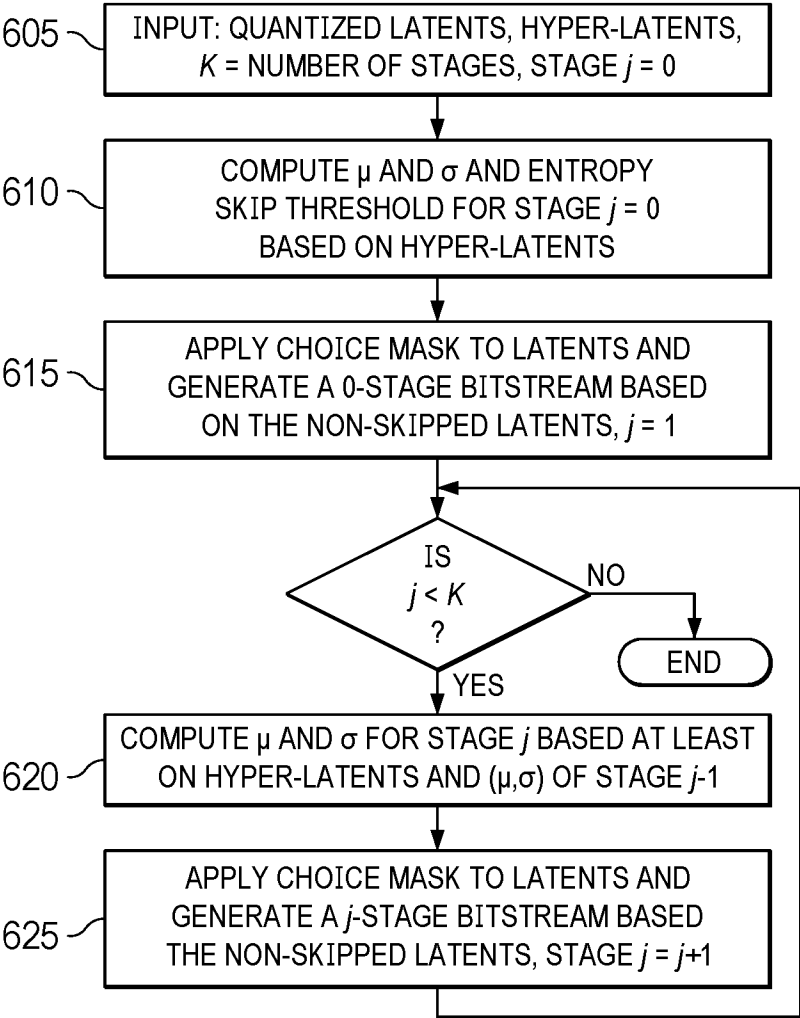


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2024/031854

A. CLASSIFICATION OF SUBJECT MATTER INV. H04N19/132 H04N19/91 ADD.		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) H04N		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal, WPI Data		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	WO 2023/165599 A1 (BEIJING BYTEDANCE NETWORK TECH CO LTD [CN]; BYTEDANCE INC [US]) 7 September 2023 (2023-09-07) abstract; figure 10 Section 4.3	1,12,13
Y	----- YIBO SHI ET AL: "AlphaVC: High-Performance and Efficient Learned Video Compression", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 29 July 2022 (2022-07-29), XP091283036, cited in the application	7,8,10, 12,13
A	abstract; figure 2 page 2 - page 3 Section 3.3 ----- -/-	1,9,11
☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 31 October 2024		Date of mailing of the international search report 11/11/2024
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Streich, Sebastian

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2024/031854

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JOOYOUNG LEE ET AL: "Selective compression learning of latent representations for variable-rate image compression", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 8 November 2022 (2022-11-08), XP091363499, abstract; figure 1 Section 3.2 -----	1, 7-13
Y	JIAHAO LI ET AL: "Neural Video Compression with Diverse Contexts", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 10 March 2023 (2023-03-10), XP091455826, abstract; figures 2, 5, 8, 10 Sections 3.4 and Appendix A -----	7, 8, 10, 12, 13
A		9, 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2024/031854

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
3. ☒ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
1, 7-11 (completely); 12, 13 (partially)
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims;; it is covered by claims Nos.:

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☒ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1(completely); 12, 13(partially)

entropy skipping threshold

2. claims: 2(completely); 12, 13(partially)

entropy skipping replacement value

3. claims: 3-6(completely); 12, 13(partially)

entropy skipping parameters in autoregressive NN

4. claims: 7-11(completely); 12, 13(partially)

entropy skipping parameters in multi-stage models

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2024/031854

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2023165599	A1	07-09-2023	NONE
