COMPASS: High-Efficiency Deep Image Compression with Arbitrary-scale Spatial Scalability

Jongmin Park
KAIST
jm.park@kaist.ac.kr

Jooyoung Lee
ETRI
leejy1003@etri.re.kr

Munchurl Kim*
KAIST
mkimee@kaist.ac.kr

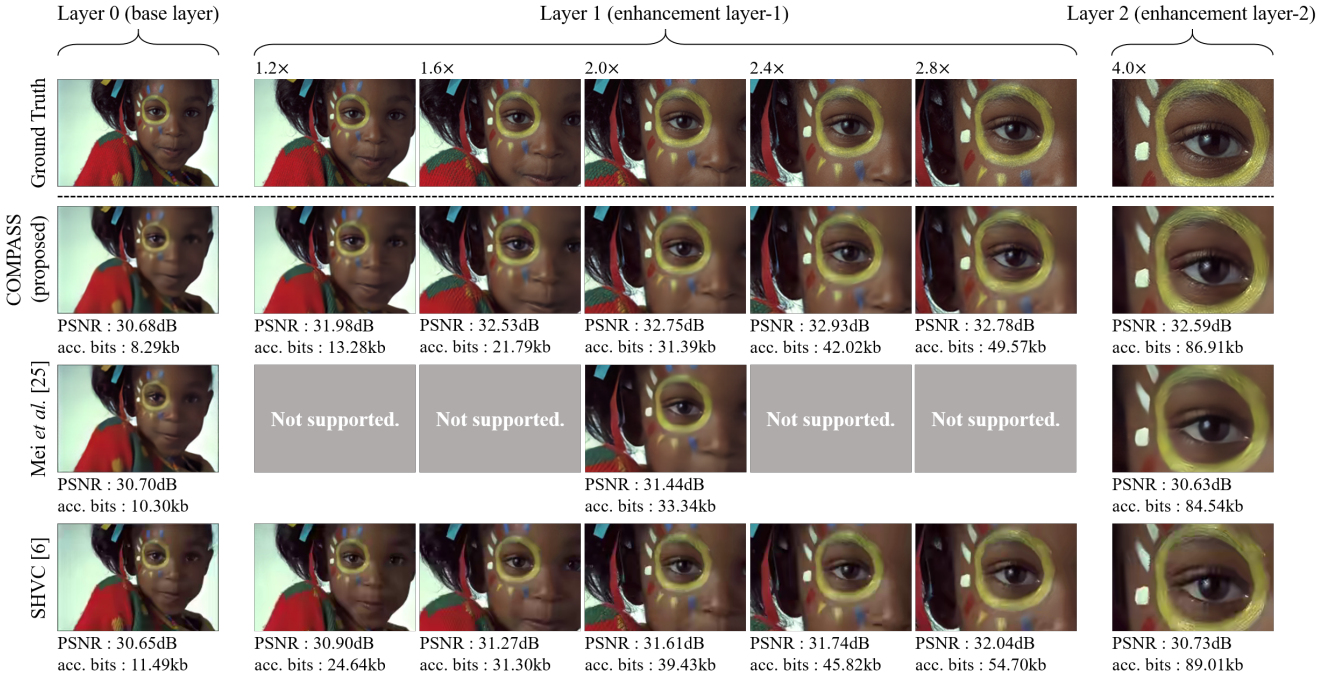

Figure 1: **Visual comparison results of spatially scalable image compression methods for arbitrary scale factors.** The image of Layer 2 is reconstructed from the image of Layer 1 with scale factor 2.0× for each method. The 'acc. bits' indicates the accumulated bits up to the corresponding layers.

Abstract

Recently, neural network (NN)-based image compression studies have actively been made and has shown impressive performance in comparison to traditional methods. However, most of the works have focused on non-scalable image compression (single-layer coding) while spatially scalable image compression has drawn less attention although it has many applications. In this paper, we propose a novel NN-based spatially scalable image compression method, called COMPASS, which supports arbitrary-scale spatial scalability. Our proposed COMPASS has a very flexible structure where the number of layers and their respective scale factors can be arbitrarily determined during inference. To reduce the spatial redundancy between adjacent layers for arbitrary scale factors, our COMPASS adopts an inter-layer arbitrary scale prediction method, called LIFF, based on implicit neural representation. We propose a combined RD loss function to effectively train multiple layers. Experimental results show that our COMPASS achieves BD-rate gain of -58.33% and -47.17% at maximum compared to SHVC and the state-of-the-art NN-based spatially scalable image compression method, respectively, for various combinations of scale factors. Our COMPASS also shows comparable or even better coding efficiency than the single-layer coding for various scale factors.

*Corresponding author.

1. Introduction

Recently, image compression has become increasingly important with the growth of multimedia applications. The exceptional performance of neural network (NN)-based methods in computer vision has led to active research on NN-based image compression methods [38, 3, 36, 4, 27, 18, 8, 19, 21, 28, 14, 2, 26], resulting in remarkable improvements in coding efficiency. However, although the same content is often consumed in various versions in multimedia systems, most existing NN-based image compression methods must separately compress an image into multiple bitstreams for their respective versions, thus leading to low coding efficiency. To resolve this issue, there have been a few recent studies [37, 34, 16, 43, 13, 24, 23, 25] on NN-based scalable image compression, where various versions of an image are encoded into a single bitstream in a hierarchical manner with multiple layers. Each layer is in charge of en/decoding one corresponding version of the image, and typically, redundancy between adjacent layers is reduced by a prediction method for higher coding efficiency.

The scalable coding methods are divided into two classes: quality scalable codecs for the images of different quality levels and spatially scalable codecs for the images of different sizes. In this paper, we focus on the spatially scalable coding that has not been actively studied compared with the quality scalable coding. Upon our best knowledge, only one previous study [25] deals with the spatially scalable coding in the recent deep NN-based approach.

In conventional tool-based scalable coding, SVC [31] and SHVC [6] have been standardized by MPEG [17] for video coding standards, as extensions to H.264/AVC [40] and H.265/HEVC [35], respectively. Despite of significant coding efficiency improvement compared with separate single-layer compression of different versions (simulcast coding), the scalable coding has not yet been widely adopted for real-world applications [6, 32]. One reason may be lower coding efficiency of the accumulated bitstream for the larger version compared with the single-layer coding of the same size. The scalable coding often yields lower coding efficiency due to its insufficient redundancy removal capability between the layers.

In addition, for the existing NN-based method [25], only one fixed scale factor 2 is used between adjacent layers as shown in Figure 1. This limitation makes it not practical for real-world applications that require a variety of scale combinations. For example, an image of 4,000×2,000 size needs to be encoded into SD (720×480), HD (1,280×720) and FHD (1,920×1,080) versions which are not in powers of 2 scales compared to the input size. Therefore, in order to support for the one-source-multiple-use (OSMU) with spatially scalable image compression, it is worthwhile for spatially scalable image compression to support arbitrary scale factors between the different layers.

To address the aforementioned issues, we propose a novel NN-based image COMPression network with Arbitrary-scale Spatial Scalability, called COMPASS. Our COMPASS supports spatially scalable image compression that encodes multiple arbitrarily scaled versions of an image into a single bitstream in which each version of the image is encoded with its corresponding layer. Inspired by LIIF [7] and Meta-SR [15], we adopt an inter-layer arbitrary scale prediction method in the COMPASS, called Local Implicit Filter Function (LIFF), based on implicit neural representation that can effectively reduce the redundancy between adjacent layers and also supports arbitrary scale factors. In addition, it should be noted that our COMPASS exploits only one shared prediction/compression module for all the enhancement layers, thus it effectively provides the extensibility in terms of the number of layers and also reduces the number of model parameters. For effective and stable optimization of the hierarchically recursive architecture of COMPASS, we introduce a combined RD loss function.

Based on its superior inter-layer prediction capability, our COMPASS significantly improves the coding efficiency compared to the existing scalable coding methods [6, 25], and achieves comparable or even better coding efficiency compared to the single-layer coding for various scale factors. Note that the coding efficiency of the single-layer coding has been regarded as the upper bound of the scalable coding efficiency. Furthermore, to the best of our knowledge, our COMPASS is the first NN-based spatially scalable image compression method that supports arbitrary scale factors with high coding efficiency. Our contributions are summarized as:

- The COMPASS is the first NN-based spatially scalable image compression method for *arbitrary* scale factors.
- The COMPASS adopts an inter-layer arbitrary scale prediction, called LIFF, which is based on implicit neural representation to reduce redundancy effectively as well as to support the arbitrary scale factors. Additionally, we propose a combined RD loss function to effectively train multiple layers.
- Our COMPASS significantly outperforms the existing spatially scalable coding methods [6, 25]. Furthermore, to the best of our knowledge, the COMPASS is the first work that shows comparable or even better performance in terms of coding efficiency than the single-layer coding for various scale factors, based on a same image compression backbone.

2. Related Work

Neural Network-based Image Compression. Recently, there have been proposals to optimize neural network (NN)-based image compression methods in an end-to-end

manner. Toderici *et al*. [38] first proposed a deep convolutional NN-based image compression method, while Ballé *et al*. [3] and Theis *et al*. [36] adopted the entropy model-based approaches that jointly minimize the rate and distortion terms in the optimization phase. Subsequent models, such as hyperprior [4], auto-regressive models [27, 18], Gaussian Mixture Models [8, 19], non-local attention modules [21], channel-wise auto-regressive entropy models [28] and the checkerboard context model [14], have improved coding efficiency. There are also a few generative model-based studies [2, 26] for human perception-oriented compression. Recently, several NN-based variable-rate compression models [9, 10, 30, 33, 22, 20] have been studied to support the multiple compression quality levels with a single trained model. Despite the significant improvements in coding efficiency and functionality brought about by the NN-based image compression networks, there remains an issue with coding efficiency when encoding different versions of an image as described in Sec. 1.

Spatially Scalable Image Compression. For OSMU applications that supports various-sized display devices, images often need to be compressed and transmitted to target devices with appropriate spatial sizes. To meet this requirement, the scalable extensions of traditional coding standards, H.264/AVC [40] and H.265/HEVC [35] have been developed as SVC [31] and SHVC [6], respectively. Recently, NN-based approaches for scalable image compression [37, 34, 16, 43, 13, 24, 23, 25] have also been proposed. However, most of these works focus on quality scalability and only Mei *et al*. [25] deals with spatial scalability. Mei *et al*. [25] proposed a hierarchical architecture which outperforms the simulcast coding and SVC [31], and shows comparable performance with SHVC [6] in terms of coding efficiency. However, it can only support fixed integer scale factors with powers of 2. Moreover, they didn't provide any experimental evidence on the extended multiple enhancement layers more than 2, although they proposed the layer extension concept.

Arbitrary Scale Super-Resolution. With the advancement of neural networks, several recent works have proposed super-resolution with arbitrary scale factors, such as [15, 7, 12, 39, 41]. Hu *et al*. [15] introduced Meta-SR, the first neural network-based method for super-resolution with arbitrary scales. In Meta-SR, the Meta-Upscale module takes the relative coordinate and scale factor as input to dynamically predict the upscaling filters. Wang *et al*. [39] proposed an asymmetric super-resolution method using conditional convolution. Cheng *et al*. [7] presented a continuous image representation method with Local Implicit Image Function (LIIF), and achieved outstanding performance for large scale ($\times 30$) super-resolution which is out of training distribution. Xu *et al*. [41] used periodic encoding with the implicit function. Inspired by these arbitrary scale super-resolution methods [7, 15], we adopt them for the inter-layer arbitrary scale prediction in our COMPASS. We refer to this method as the Local Implicit Filter Function (LIFF), which can effectively reduce redundancy between adjacent layers with arbitrary scale factors.

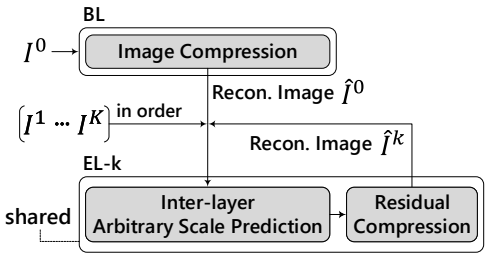

Figure 2: The COMPASS supports spatially scalable coding of K+1 arbitrary scaled versions of an image using a base layer (BL) and one or more enhancement layers (ELs). The EL-k ($1 \leq k \leq K$) exploits a shared subnetwork that the Inter-layer Arbitraty Scale Prediction and Residual Compression modules. I_0 indicates the smallest-sized input image in the BL. $I_1, ..., I_K$ are the input images in the ELs in an increasing order of scale factors where I_K is the largest-sized input image. Note that the scale factor between two adjacent layers can be any arbitrarily positive value.

3. Proposed Method

3.1. Overall architecture

Figure 2 depicts a flow diagram of our COMPASS. The COMPASS comprises of two types of layers: a base layer (BL) that encodes the lowest resolution image, and one or more enhancement layers (ELs) that sequentially encode multiple higher resolution images of arbitrary scales. For spatially scalable coding of (K+1)-scaled images $\{I^0, ..., I^K\}$ of gradually increasing sizes with arbitrary scale factors, the COMPASS operates with multiple coding in the BL and K ELs, each of which encodes the correspondingly scaled input image. It should be noted that the COMPASS exploits the shared modules for all the ELs, each of which recursively operates as depicted in Figure 2. In the BL, the smallest-sized input image I^0 is fed into a CNN-based image compression module to reconstruct $\hat{I}^0$. In the EL-k, the corresponding input image I^k and the reconstructed image $\hat{I}^{k-1}$ of the previous layer are fed into the current enhancement layer to reconstruct $\hat{I}^k$. Specifically, in the EL-k, the inter-layer arbitrary scale prediction module can effectively estimate and reduce the spatial redundancy between $\hat{I}^{k-1}$ and I^k for arbitrary scale factor. Therefore, the residual compression module only encodes the resulting essential residues in reconstructing $\hat{I}^k$ with high coding efficiency. Figure 3 shows the overall architec-

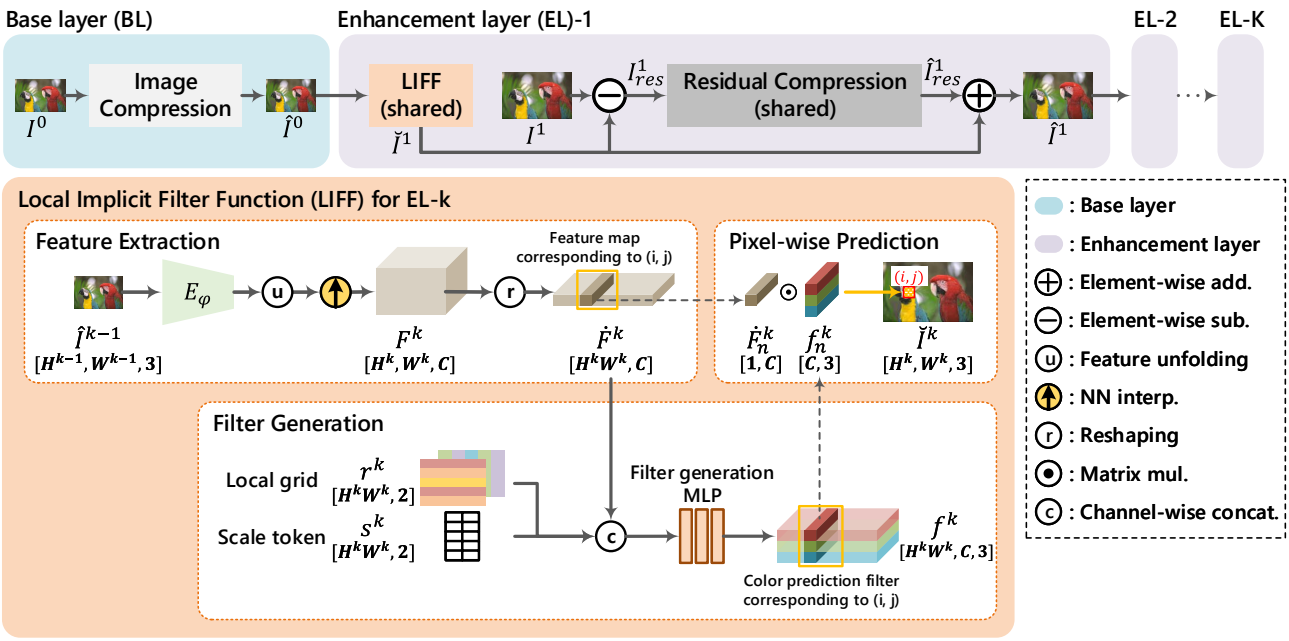

Figure 3: **Overall architecture of our COMPASS.** It consists of a base layer (BL) depicted in the sky blue box and one or more enhancement layers (ELs) depicted in the light purple boxes which operate in an iterative manner. Note that we exploit the shared modules (LIFF and residual compression) for multiple ELs.

ture of our COMPASS. We describe the operation of COMPASS with K+1 layers as:

$$\hat{I}^k = \begin{cases} IC(I^k), & \text{if } k = 0 \quad \text{(BL)} \\ \check{I}^k + \hat{I}^k_{res}, & \text{if } k > 0 \quad \text{(EL-}k\text{)} \end{cases} \tag{1}$$

where, for $k > 0$,

$$\check{I}^k = \psi(\hat{I}^{k-1}, \boldsymbol{s}^k, \boldsymbol{r}^k) \;\text{ and }\; \hat{I}^k_{res} = RC(I^k_{res}), \tag{2}$$

where $IC(\cdot)$ refers to an image compression module of the BL and $RC(\cdot)$ refers to a residual compression module of the EL-k, as shown in Figure 3. We adopt the Mean-scale [27] architecture for both the compression modules. $\check{I}^k \in \mathbb{R}^{H^k \times W^k \times 3}$ refers to an arbitrarily upscaled prediction for the EL-k, and it is predicted from the smaller reconstruction $\hat{I}^{k-1}$ by the LIFF module which is denoted as $\psi(\cdot)$, and I^k_{res} indicates a residual image between I^k and $\check{I}^k$. The LIFF module takes a local grid $\boldsymbol{r}^k \in \mathbb{R}^{H^k W^k \times 2}$ and a scale token $\boldsymbol{s}^k \in \mathbb{R}^{H^k W^k \times 2}$ as additional inputs, which are described in details in Sec. 3.2. Since the output of convolutional layers are progressively reduced in half due to the convolution of stride 2 in the encoder part, the input to the encoder part is often padded into the size of a power of 2 in a lump at the beginning. This actually deteriorates the coding efficiency in our image compression with arbitrary scale factors. Therefore, we adopt a convolutional-layer-wise padding scheme where (i) a replicate padding with the padding size of 1 is performed if the width or height size of the input is an odd number in each convolutional layer of the encoder part of the residual compression module; and (ii) we crop out the padded region for the output of the corresponding convolutional layer of the decoder part.

3.2. LIFF: Inter-layer arbitrary scale prediction

To achieve high coding efficiency with the COMPASS, it is essential to effectively reduce the redundancy between adjacent layers. For this, we adopt an inter-layer arbitrary scale prediction method using a local implicit filter function (LIFF) which is based on the local implicit image function (LIIF) [7] and Meta-SR [15]. Our LIFF module first transforms the reconstruction $\hat{I}^{k-1}$ of the previous layer into the feature domain and then increases its resolution to match the arbitrarily upscaled prediction $\check{I}^k$ through a simple interpolation. Our LIFF module also generates the color prediction filter for each pixel coordinate and then estimate the RGB color pixel-wise by applying the generated filter to the extracted feature slice corresponding to the target pixel coordinate. The procedure of the LIFF module is divided into 3 stages: 1) Feature Extraction, 2) Filter Generation, 3) Pixel-wise Prediction, as illustrated in the orange box of Figure 3.

Feature Extraction. We extract feature information from the reconstruction $\hat{I}^{k-1}$ of the previous layer through an RDN-like feature extractor E_φ [42], and apply feature unfolding [7] and nearest-neighbor upsampling to generate the feature map $\boldsymbol{F}^k \in \mathbb{R}^{H^k \times W^k \times C}$.

Filter Generation. We generate the color prediction filter

(a) $\hat{I}^{k-1}$ (b) $\breve{I}^{k}$ (c) I^{k} (d) I^{k}_{res}

Figure 4: A predicted image via the LIFF module. (a) the reconstruction of the previous layer k-1, (b) the output (predicted image) of the LIFF module, (c) the input image of the current layer k, (d) the residual image as the input of the residual compression module.

$\boldsymbol{f}^k \in \mathbb{R}^{H^k W^k \times C \times 3}$ using a filter generation MLP as

$$\boldsymbol{f}^k = \phi(\lceil \dot{\boldsymbol{F}}^k, \boldsymbol{r}^k, \boldsymbol{s}^k \rfloor; \theta), \tag{3}$$

where $\dot{\boldsymbol{F}}^k \in \mathbb{R}^{H^k W^k \times C}$ refers to the flattened feature map, $\phi(\cdot)$ refers to the filter generation MLP with parameters θ, and $\lceil \cdot \rfloor$ refers to channel-wise concatenation. The local grid $\boldsymbol{r}^k \in \mathbb{R}^{H^k W^k \times 2}$ and the scale token $\boldsymbol{s}^k \in \mathbb{R}^{H^k W^k \times 2}$ follow the same process as in the LIIF [7]. The local grid $\boldsymbol{r}^k$ is a normalized relative coordinate between the reconstruction $\hat{I}^{k-1}$ of the previous layer and the upscaled prediction $\breve{I}^k$, which is formulated as $\boldsymbol{r}^k(i,j) = \boldsymbol{p}^k(i,j) - \boldsymbol{p}^{k-1}(i',j')$. $\boldsymbol{p}^k(i,j)$ refers to a normalized coordinate of the upscaled prediction $\breve{I}^k$ at pixel coordinate (i,j), and $\boldsymbol{p}^{k-1}(i',j')$ indicates a corresponding normalized coordinate of the reconstruction $\hat{I}^{k-1}$ of the previous layer at pixel coordinate (i',j'). We adopt the nearest-neighbor to find the pixel correspondence. The normalized coordinate is calculated as $\boldsymbol{p}^l(i,j) = [-1 + (2i+1)/H^l, -1 + (2j+1)/W^l]$, where $i \in [0, H^l - 1]$ and $j \in [0, W^l - 1]$. The scale token $\boldsymbol{s}^k$ indicates the height/width ratio between $\hat{I}^{k-1}$ and $\breve{I}^k$. $\boldsymbol{s}^k$ then contains all the same ratio values of $(2 \cdot H^{k-1}/H^k, 2 \cdot W^{k-1}/W^k)$.

Pixel-wise Prediction. To determine the RGB color of the arbitrarily upscaled prediction $\breve{I}^k$ at pixel coordinate (i,j), we apply the color prediction filter $\boldsymbol{f}^k_n$ for the generated feature map $\dot{\boldsymbol{F}}^k_n$ by a simple matrix multiplication as

$$\breve{I}^k(i,j) = \dot{\boldsymbol{F}}^k_n \odot \boldsymbol{f}^k_n, \tag{4}$$

where $n \in [0, H^k W^k - 1]$ indicates the batch index number which is corresponding to the pixel coordinate (i,j) of the prediction $\breve{I}^k$ via $n = i + j \cdot H^k$. Note that the LIFF module can calculate this pixel-wise prediction for all coordinates in parallel as $\breve{I}^k = \dot{\boldsymbol{F}}^k \odot \boldsymbol{f}^k$.

Figure 4 shows a predicted image $\breve{I}^k$ via the LIFF module and its associated residual image I^k_{res} to be compressed for the given reconstructed image $\hat{I}^{k-1}$ of the previous layer k-1, and an uncompressed input image I^k (ground truth) in the current layer k. Compared to $\hat{I}^{k-1}$ in Figure 4-(a), $\breve{I}^k$ in Figure 4-(b) shows much closer result to I^k in Figure 4-(c), thus leading to a smaller amount of residues I^k_{res} in Figure 4-(d).

3.3. Optimization

We train the whole elements of our COMPASS in an end-to-end manner with the frozen pre-trained image compression module of the BL. To boost up the training, we use the separately pre-trained LIFF and residual compression modules. To train the COMPASS architecture, we use a combined RD loss function as:

$$L = \sum_{k=1}^{K} R^k + \lambda \cdot D^k, \tag{5}$$

where R^k and D^k represent a rate term and a distortion term for the EL-k, respectively. As in other NN-based image compression methods [3, 4, 27, 18], the rate and distortion are jointly optimized, but we use the summation of those for the K ELs. It should be noted that we use the same λ value for the K ELs to maintain the R-D balance over the whole layers. The rate term R^k is the estimated rate amount for the EL-k. Specifically, it is determined as the summation of cross-entropy values for latent representations $\boldsymbol{y}^k$ and $\boldsymbol{z}^k$. $\boldsymbol{y}^k$ is the latent representation transformed from an input residual image I^k_{res} via the encoder network of the residual compression module, and $\boldsymbol{z}^k$ is the latent representation transformed from the representation $\boldsymbol{y}^k$ via the hyper-encoder network of the residual compression module, as in the previous hyperprior-based models [4, 27, 18]. The rate term R^k is represented as $R^k = H^k(\tilde{\boldsymbol{y}}^k|\tilde{\boldsymbol{z}}^k) + H^k(\tilde{\boldsymbol{z}}^k)$, where $H^k(\tilde{\boldsymbol{y}}^k|\tilde{\boldsymbol{z}}^k)$ and $H^k(\tilde{\boldsymbol{z}}^k)$ are the cross-entropy terms for noisy latent representations $\tilde{\boldsymbol{y}}^k$ and $\tilde{\boldsymbol{z}}^k$ for the EL-k, respectively. The cross-entropy values are calculated based on the Gaussian entropy model used in the Mean-scale model [27]. As in other NN-based image compression methods [3, 4, 27, 18], we sample the noisy latent representations $\tilde{\boldsymbol{y}}^k$ and $\tilde{\boldsymbol{z}}^k$ for the EL-k with the additive uniform noise to fit the samples to the approximate probability mass function (PMF) $P(\cdot)$ of the discretized representations $\boldsymbol{y}^k$ and $\boldsymbol{z}^k$ for the EL-k. D^k is a mean squared error (MSE) between the reconstructed image $\hat{I}^k$ and input image I^k for the EL-k. $\hat{I}^k$ is represented as $\hat{I}^k = \breve{I}^k + \hat{I}^k_{res}$, where $\hat{I}^k_{res} = D^{RC}(\hat{\boldsymbol{y}}^k)$ and $\hat{\boldsymbol{y}}^k = Q(E^{RC}(I^k_{res}))$. Note that $E^{RC}(\cdot)$ and $D^{RC}(\cdot)$ refer to the encoder and decoder networks of the residual compression module $RC(\cdot)$, respectively, and $Q(\cdot)$ is a rounding function.

Here, we use a rounded latent representation $\hat{\boldsymbol{y}}^k$ rather than the noisy representation $\tilde{\boldsymbol{y}}^k$ for calculating the distortion term. We first used the noisy representation $\tilde{\boldsymbol{y}}^k$ as the input into the decoder network D^{RC}, but we obtained very poor optimization results. On the other hand, when we feed the rounded representations $\hat{\boldsymbol{y}}^k$ instead of $\tilde{\boldsymbol{y}}^k$, the coding efficiency is much improved. The suboptimal performance of the COMPASS with noisy representations could be attributed to the propagation of small errors in the reconstructions, caused by the additive uniform noise, to the following ELs. This propagation of errors could result in a sig-

Methods	BD-rate↓	Params.
SHVC [6]	-33.34%	-
Simulcast (Factorized [3])	-41.71%	-
Simulcast (Mean-scale [27])	-22.90%	-
Mei *et al*. [25] (original)	-32.12%	40.7M
Mei *et al*. [25] (enhanced)	-14.23%	52.8M
Single-layer (Mean-scale [27])	4.74%	-
COMPASS (proposed)	-	15.5M

Table 1: Coding efficiency and model size comparison. BD-rate gains of our COMPASS over the various methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS. The 'Params.' indicates the total number of parameters.

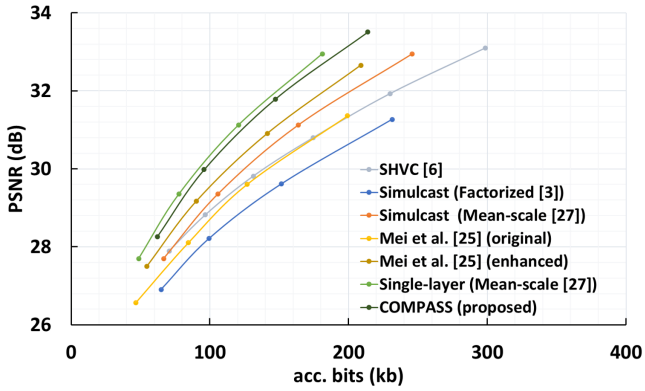

Figure 5: The rate-PSNR performance curves of the final ELs for SHVC [6], the simulcast coding, Mei *et al*. [25], the single-layer coding, and our COMPASS. The 'acc. bits' indicates the accumulated bits up to the final EL.

nificant discrepancy between the training and inference, ultimately leading to poor results. The hierarchical and recursive operation of the COMPASS may also contribute to this issue. Whereas, using the rounded representation can certainly prevent the mentioned error propagation because it does not cause any discrepancy between the training and inference phases at all. To deal with the discontinuity due to the rounding operations, we just bypass the gradients backward. In contrast to the distortion term, it should be noted that our COMPASS still uses the noisy representation $\tilde{\boldsymbol{y}}^k$ to calculate the rate term R^k to fit the samples to the approximate PMF $P(\cdot)$. Further details in optimization are described in Appendix A.

4. Experiments

We first describe the experimental setup in terms of two aspects: coding efficiency with a fixed scale factor of 2 and coding efficiency with arbitrary scale factors. Then we present the corresponding experimental results.

Methods	Scale Factors (vs. BL)				
	1.2×	1.6×	2.0×	2.4×	2.8×
SHVC [6]	-53.88%	-42.58%	-35.87%	-26.24%	-22.34%
Simulcast (Factorized [3])	-53.04%	-44.87%	-31.89%	-34.62%	-31.59%
Simulcast (Mean-scale [27])	-44.04%	-31.40%	-16.29%	-16.35%	-12.38%
Mei *et al*. [25] (original)	-36.26%	-29.44%	-28.20%	-33.19%	-36.31%
Mei *et al*. [25] (enhanced)	-29.09%	-17.45%	-13.52%	-20.63%	-23.94%
Single-layer (Mean-scale [27])	-8.19%	-3.70%	8.80%	0.31%	0.94%

Table 2: Coding efficiency comparison for the two-layer scalable coding with arbitrary scale factors. BD-rate gains of our COMPASS over the various methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS.

Methods	Scale Factors (vs. BL)				
	2.4×	2.8×	3.2×	3.6×	4.0×
SHVC [6]	-58.33%	-51.51%	-46.72%	-43.65%	-33.34%
Simulcast (Factorized [3])	-61.04%	-55.09%	-50.42%	-47.11%	-41.71%
Simulcast (Mean-scale [27])	-49.85%	-42.20%	-35.33%	-30.81%	-22.90%
Mei *et al*. [25] (original)	-47.17%	-38.73%	-34.34%	-33.56%	-32.12%
Mei *et al*. [25] (enhanced)	-38.23%	-26.46%	-19.70%	-17.08%	-14.23%
Single-layer (Mean-scale [27])	-6.60%	-4.23%	-1.25%	-0.74%	4.74%

Table 3: Coding efficiency comparison for the three-layer scalable coding with arbitrary scale factors. BD-rate gains of our COMPASS over the various methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS. We set the scale factor of the EL-1 relative to the BL to 2, equally.

4.1. Coding efficiency with a fixed scale factor of 2

Experimental setup. To validate the coding efficiency of our COMPASS, we compare it with SHVC [6], the simulcast coding, Mei *et al*. [25], and the single-layer coding with the Mean-scale model [27], in terms of BD-rate. We conduct the coding efficiency comparison of each method for three-layer scalability (one BL and two ELs) with a fixed scale factor of 2 between adjacent layers. For a fair comparison, we used both Mei *et al*. [25]'s original version with the Factorized model [3] and its enhanced one with the Mean-scale model [27] as the same image compression model as ours. Note that we set the same numbers of channels for all image compression modules with $N = 128$ and $M = 192$

*. Following the typical validation procedures [6, 32] for the scalable coding, we measure the BD-rate performance for the final ELs with the largest scales (spatial sizes) of reconstructions, but we also provide the BD-rate results for the BL and intermediate ELs in Appendix B. For the simulcast and single-layer coding, we use the pre-trained models from CompressAI-Pytorch [5]. The BD-rate performance of SHVC [6] is measured using the All-Intra Mode of the SHVC reference software (SHM-12.4) [1] with the QP values of 30, 32, 34, 36, 38 and 40. The RGB inputs are converted into the YUV420 format and the reconstructions are converted back to the RGB format to achieve the best possible performance of SHVC [6]. More specifically, we use the Kodak Lossless True Color Image dataset [11] that consists of 24 768×512-sized (or 512×768-sized) images. For the downscaling of the input images, we use the bicubic interpolation function implemented in Pytorch [29]. For the feature extractor of the LIFF module in our COMPASS, we set the number of residual dense blocks (RDBs) and convolutional layers of each RDB to 4. The number of output channels and the growth rate of each RDB are set to 64 and 32, respectively. For the filter generation MLP of the LIFF module, we set the number of hidden layers to 5, each of which has 256 output channels. The COMPASS is built using the open-source CompressAI Pytorch library [5].

Experimental results. Table 1 shows the coding efficiency performance in terms of BD-rate for our COMPASS against the compared methods. It should be noted in Table 1 that each compared method becomes an anchor for comparison against our COMPASS. Therefore, the negative percentage values indicate that our COMPASS outperforms the corresponding methods in BD-rate coding efficiency by those amounts while the positive values imply under-performance of the COMPASS. Figure 5 shows the rate-PSNR performance curves for Table 1. As shown in Table 1 and Figure 5, our COMPASS significantly outperforms all the spatially scalable coding methods except the Single-layer (Mean-scale [27]). It is worthy to note that, although that Mei *et al*. [25]'s enhanced version uses the same image compression module (Mean-scale [27]) as the COMPASS and focuses only on the fixed scale factor of 2, -14.23% of BD-rate gain is achieved by our COMPASS that supports various different scale factors, which will be discussed in Sec. 4.2. It is noted in Table 1 that our COMPASS has a less number of parameters, compared to Mei *et al*. [25]'s method. Furthermore, impressively, our COMPASS achieves comparable results to the single-layer coding with the Mean-scale model [27], while the existing scalable coding methods [6] show considerably lower coding efficiency compared with their single-layer coding as described in [6, 32] due to low inter-layer prediction accuracy. In addition, our COMPASS shows the BD-rate gains of -24.97% and -35.87% compared to SHVC [6] at the BL and the EL-1, respectively. Further comparisons for the other layers are provided in Appendix B.

*N and M refer to the output channels of the encoder network and hyper-encoder network of the image compression modules, respectively.

4.2. Coding efficiency with arbitrary scale factors

Experimental setup. We show the effective coding efficiency of our COMPASS at arbitrary scales by comparing it with the six coding methods. For this, five scale factors are used for each of two experiments. The first experiment is conducted with five two-layer scalabilities (one BL and one EL (EL-1: 1.2×, 1.6×, 2.0×, 2.4×, and 2.8×)) while the second one is with five three-layer scalabilities (one BL and two ELs (EL-1: 2.0× and EL-2: 2.4×, 2.8×, 3.2×, 3.6×, and 4.0×)). More experiment results are provided for more combinations of scale factors in Appendix B. For Mei *et al*. [25] which only supports a fixed scale factor of 2 between adjacent layers, we upscale or downscale the output images using bicubic interpolation to match with other scale factors. The other experimental conditions such as datasets and channel numbers are the same as those in Sec. 4.1.

Experimental results. Table 2 shows the comparison results of the two-layer scalable coding. As shown, our COMPASS significantly outperforms all the spatially scalable coding methods over the entire range of the different scale factors from 1.2× to 2.8×. Furthermore, it achieves comparable results to the single-layer coding with the Mean-scale model [27] over the entire range. Surprisingly, our COMPASS even outperforms it for the scale factors of 1.2× and 1.6×. This is because the input images for the single-layer coding need to be padded to a multiple of 64 in order to be processed into the CNN architecture of the image compression network, which can lead to lower the coding efficiency. In contrast, our LIFF module is effective at handling the arbitrary scale factors. For the three-layer scalable coding, as shown in Table 3, our COMPASS also outperforms all the spatially scalable coding methods as the two-layer scalable coding, and even exhibits a superiority to the single-layer coding with the Mean-scale model [27] over the whole scale factors only except scale factor 4.0×. The superiority of our COMPASS stems from the fact that the LIFF module can well perform the inter-layer prediction for arbitrary scale factors.

4.3. Visual comparison

Figure 6 shows the visual comparison results of our COMPASS with SHVC [6], the simulcast coding, and Mei *et al*. [25] for the three-layer scalable coding with a scale factor of 2 between adjacent layers. The images shown in Figure 6 are the largest reconstructions obtained from the final EL. We set the accumulated bits as close as possible between the methods. As shown, the subjective qualities of the reconstructions by our COMPASS are significantly

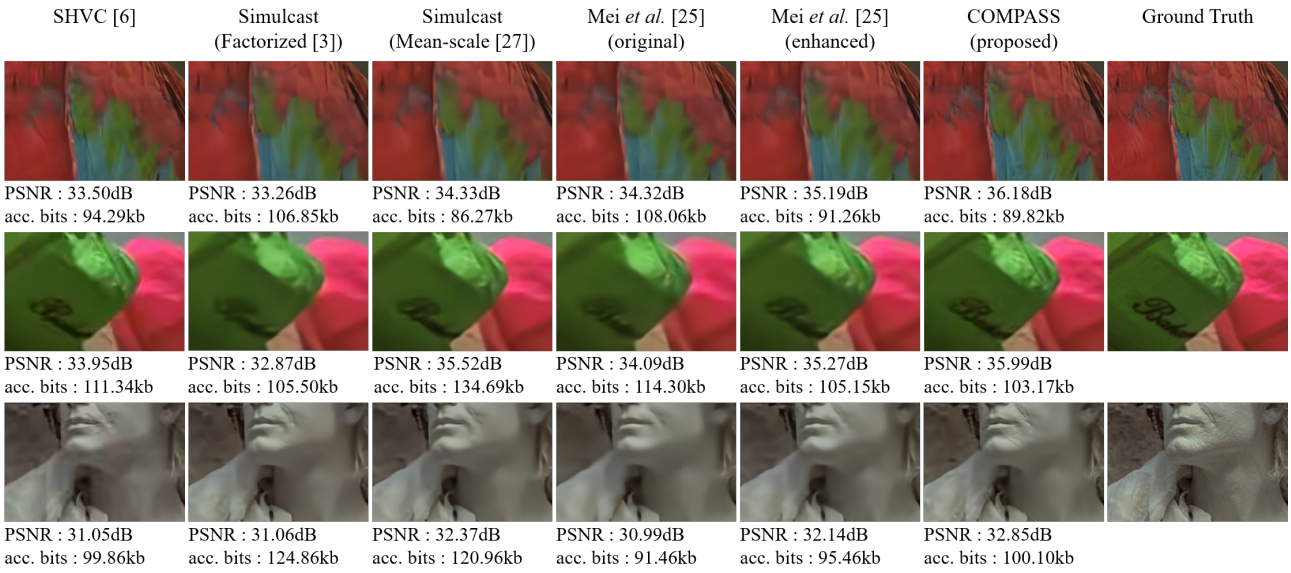

Figure 6: Visual comparison results for *kodim23.png*, *kodim03.png*, *kodim17.png* images in Kodak Lossless True Color Image dataset [11] (best viewed in digital format). The 'acc. bits' indicates the accumulated bits up to the final EL. We match the accumulated bits among the compared methods as much as possible. **Zoom for better visual comparison**.

better than those from the other methods, and especially, the high-frequency components such as edges and textures are more clearly reconstructed. More visual comparison results are provided in Appendix C, where we also provide the reconstructions with the multi-layer configuration greater than three layers to verify the extensibility of our COMPASS in terms of the number of layers.

4.4. Ablation study

To verify the effectiveness of the proposed elements (or optimization strategy) in our COMPASS, we measure the coding efficiency of the ablated models and compare the results with those of the full model of our COMPASS in terms of BD-rate. In the comparison, the ablated elements are the LIFF module described in Sec. 3.2, the convolutional-layer-wise padding described in Sec. 3.1, and the adoption of rounded representations replacing the noisy representations for calculation of the distortion term in training described in Sec. 3.3. It should be noted for the ablated model without the LIFF module that a bicubic interpolation is used instead. For the ablated model without the convolutional-layer-wise padding, we pad the input images to match their sizes to the multiple of 64 in both vertical and horizontal axes. For the ablated model without using the rounded representations in training, we use the noisy representations instead of the rounded representations. As shown in Table 4, our full COMPASS model significantly outperforms all the ablated models, which represents that each proposed element of the COMPASS model effectively contributes to the enhancement of coding efficiency.

Inter-layer prediction	Conv. layer padding	Rounded latent rep.	BD-rate↓ (vs. full model)
Bicubic	✓	✓	9.27%
LIFF	✗	✓	18.95%
LIFF	✓	✗	13.00%

Table 4: Ablation study results about the proposed elements or optimization strategies in our COMPASS.

5. Conclusion

In this paper, we propose a new NN-based spatially scalable image compression method, called COMPASS, which can achieve high coding efficiency while supporting arbitrary scale factors between adjacent layers, not limited to doubling the scales. To be an effective architecture, all the enhancement layers share the LIFF module and the residual compression module, which are recursively performed into higher scale factors. The LIFF module is adopted for the inter-layer arbitrary scale prediction, which can effectively reduce the spatial redundancy between layers for arbitrary scale factors. We also propose the combined RD loss function to effectively train multiple layers. Experimental results show that our COMPASS significantly outperforms SHVC [6], the simulcast coding, and the existing NN-based spatially scalable coding method [25] in terms of BD-rate for all combinations of scale factors. Our COMPASS also uses a smaller number of parameters than the existing NN-based spatially scalable coding method [25]. To the best of our knowledge, the COMPASS is the first work that shows comparable or even better performance in terms of coding efficiency than the single-layer coding for various scale factors, based on a same image compression backbone.

Acknowledgement

This work was supported by internal fund/grant of Electronics and Telecommunications Research Institute (ETRI). [23YC1100, Technology Development for Strengthening Competitiveness in Standard IPR for communication and media]

References

[1] Shm-12.4 software package, 2017.

[2] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 221–231, 2019.

[3] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

[4] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

[5] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.

[6] Jill M Boyce, Yan Ye, Jianle Chen, and Adarsh K Ramasubramonian. Overview of shvc: Scalable extensions of the high efficiency video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(1):20–34, 2015.

[7] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021.

[8] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7939–7948, 2020.

[9] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Variable rate deep image compression with a conditional autoencoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3146–3154, 2019.

[10] Ze Cui, Jing Wang, Bo Bai, Tiansheng Guo, and Yihui Feng. G-VAE: A continuously variable rate deep image compression framework, 2020.

[11] Rich Franzen. Kodak lossless true color image suite. *source: http://r0k. us/graphics/kodak*, 4(2), 1999.

[12] Ying Fu, Jian Chen, Tao Zhang, and Yonggang Lin. Residual scale attention network for arbitrary scale image super-resolution. *Neurocomputing*, 427:201–211, 2021.

[13] Zongyu Guo, Zhizheng Zhang, and Zhibo Chen. Deep scalable image compression via hierarchical feature decorrelation. In *2019 Picture Coding Symposium (PCS)*, pages 1–5. IEEE, 2019.

[14] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14771–14780, 2021.

[15] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1575–1584, 2019.

[16] Chuanmin Jia, Zhaoyi Liu, Yao Wang, Siwei Ma, and Wen Gao. Layered image compression using scalable auto-encoder. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 431–436. IEEE, 2019.

[17] Didier Le Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, 1991.

[18] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018.

[19] Jooyoung Lee, Seunghyun Cho, and Munchurl Kim. An end-to-end joint learning scheme of image compression and quality enhancement with improved entropy minimization. *arXiv preprint arXiv:1912.12817*, 2019.

[20] Jooyoung Lee, Seyoon Jeong, and Munchurl Kim. Selective compression learning of latent representations for variable-rate image compression. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[21] Haojie Liu, Tong Chen, Peiyao Guo, Qiu Shen, Xun Cao, Yao Wang, and Zhan Ma. Non-local attention optimized deep image compression. *arXiv preprint arXiv:1904.09757*, 2019.

[22] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. An end-to-end learning framework for video compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3292–3308, 2021.

[23] Yadong Lu, Yinhao Zhu, Yang Yang, Amir Said, and Taco S Cohen. Progressive neural image compression with nested quantization and latent ordering. In *The IEEE International Conference on Image Processing (ICIP)*, 2021.

[24] Yi Ma, Yongqi Zhai, and Ronggang Wang. Deepfgs: Fine-grained scalable coding for learned image compression. *arXiv preprint arXiv:2201.01173*, 2022.

[25] Yixin Mei, Li Li, Zhu Li, and Fan Li. Learning-based scalable image compression with latent-feature reuse and prediction. *IEEE Transactions on Multimedia*, 2021.

[26] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33:11913–11924, 2020.

[27] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018.

[28] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020*

IEEE International Conference on Image Processing (ICIP), pages 3339–3343. IEEE, 2020.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[30] Oren Rippel, Alexander G Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir Bourdev. Elf-vc: Efficient learned flexible-rate video coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14479–14488, 2021.

[31] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. Overview of the scalable video coding extension of the h. 264/avc standard. *IEEE Transactions on circuits and systems for video technology*, 17(9):1103–1120, 2007.

[32] Vadim Seregin and Yong He. Common shm test conditions and software reference configurations. *Document JCTVCQ1009*, pages 1–4, 2014.

[33] Myungseo Song, Jinyoung Choi, and Bohyung Han. Variable-rate deep image compression through spatially-adaptive feature transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2380–2389, 2021.

[34] Rige Su, Zhengxue Cheng, Heming Sun, and Jiro Katto. Scalable learned image compression with a recurrent neural networks-based hyperprior. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3369–3373. IEEE, 2020.

[35] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[36] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

[37] George Toderici, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[38] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5306–5314, 2017.

[39] Longguang Wang, Yingqian Wang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning a single network for scale-arbitrary super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4801–4810, 2021.

[40] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

[41] Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Ultrasr: Spatial encoding is a missing key for implicit image function-based arbitrary-scale super-resolution. *arXiv preprint arXiv:2103.12716*, 2021.

[42] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[43] Zhizheng Zhang, Zhibo Chen, Jianxin Lin, and Weiping Li. Learned scalable image compression with bidirectional context disentanglement network. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1438–1443. IEEE, 2019.