Content Adaptive Latents and Decoder for Neural Image Compression

Guanbo Pan[1], Guo Lu[2], Zhihao Hu[1], and Dong Xu[3](✉)

[1] School of Software, Beihang University, Beijing, China
[2] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[3] Department of Computer Science, The University of Hong Kong, Hong Kong, China
dongxu@cs.hku.hk

Abstract. In recent years, neural image compression (NIC) algorithms have shown powerful coding performance. However, most of them are not adaptive to the image content. Although several content adaptive methods have been proposed by updating the encoder-side components, the adaptability of both latents and the decoder is not well exploited. In this work, we propose a new NIC framework that improves the content adaptability on both latents and the decoder. Specifically, to remove redundancy in the latents, our content adaptive channel dropping (CACD) method automatically selects the optimal quality levels for the latents spatially and drops the redundant channels. Additionally, we propose the content adaptive feature transformation (CAFT) method to improve decoder-side content adaptability by extracting the characteristic information of the image content, which is then used to transform the features in the decoder side. Experimental results demonstrate that our proposed methods with the encoder-side updating algorithm achieve the state-of-the-art performance.

Keywords: Neural Image Compression, Content Adaptive Coding

1 Introduction

Data compression has been studied for decades as an essential issue to alleviate data storage and transmission burden. The traditional codecs, such as JPEG [40], JPEG2000 [35], BPG [9] for image compression and H.264 [44], H.265 [37], H.266 [11] for video compression, still prevail nowadays. In recent years, neural image compression (NIC) has shown promising coding performance due to its powerful nonlinear transformation capability and end-to-end optimization strategy. The recent state-of-the-art NIC methods like [47] outperform the latest traditional compression standard Versatile Video Coding (VVC) [11] on various datasets including the Kodak [1] and Tecnick [4] datasets. These approaches generally reduce the redundancy of the images by using an autoencoder architecture, which learns a mapping between the RGB color space and the learned latent space. The latent representation of the image is then quantized into a discrete-valued version, which is further compressed by the lossless entropy coding methods.

Neural data compression methods learn a generalized model to ensure the coding performance during performance evaluation. However, domain shift between the training and testing data and lack of adaptability to the visual content degrade the performance when compressing unseen data samples. Therefore, some works [12,50,28,41] were proposed to improve the adaptability for neural image compression and neural video compression (NVC) by updating the encoder-side components. Those methods aim at generating more compressible latents and estimating more accurate entropy model parameters for each data instance by fine-tuning the latents [12,50], the encoder [28] or the input image [41]. However, such fine-tuning process is extremely time-consuming and the adaptability is still limited due to the fixed decoder.

To exploit the adaptability at the decoder, some full-model over-fitting methods [34] entropy encode and transmit the updates of the decoder parameters along with the quantized latents to the receiver side for better and consistent reconstruction. However, the design of additional model compression is quite complex and the updating approach is also time-consuming. Another limitation in NIC is that the number of channels of the latents is not adapted to the rate-distortion (RD) trade-offs and the image content. Most works train multiple models with the same network architecture based on different RD trade-offs for rate control, which generate the latents with the same channel number for different RD trade-offs and spatial locations. However, this leads to redundant elements in the latents.

In this work, we propose a content adaptive NIC framework to improve the adaptability on both latents and the decoder. To improve the adaptability of latent codes, we propose the content adaptive channel dropping (CACD) method, which selects the optimal quality level at each spatial location for the latents and drops redundant elements along the channel dimension. In order to improve decoder-side content adaptability, we propose the content adaptive feature transformation (CAFT) method for the decoder, which extracts characteristic information of the image content in the decoder side and utilizes it to adapt each upsampled feature to the image content by using the Spatial Feature Transform (SFT) [42] strategy.

The experiments demonstrate that our proposed methods improve the performance of the baseline framework [30] in terms of both latents and the decoder. Our proposed content adaptive methods are also complementary to those encoder-side updating methods. Experimental results on the Kodak dataset demonstrate that our framework equipped with the encoder-side updating method Stochastic Gumbel Annealing (SGA) [50] achieves comparable overall results to the recent state-of-the-art NIC methods [47,46] and outperforms them in terms of PSNR. Additionally, the experimental results also indicate that our methods are general and can be readily applied to NVC for better coding performance. The contributions of our work are summarized as follows:

- We propose the content adaptive channel dropping (CACD) method to improve the adaptability of RD trade-offs and the image content for latent

codes. Our CACD automatically selects the optimal quality level at each spatial location, and then drops redundant elements for bit-rate saving.
- To exploit the adaptability at the decoder side, our content adaptive feature transformation (CAFT) method modulates the output features at multiple levels by considering the characteristic information of the image content.
- Experimental results demonstrate that our methods improve the performance by adapting both latents and the decoder without any additional updating steps during performance evaluation, which are also complementary to the encoder-side updating methods.

2 Related Work

2.1 Neural Image Compression

In recent years, neural image compression (NIC) performance has been improved significantly, which are mostly based on recurrent neural networks (RNNs) [23,38,39], convolutional neural networks (CNNs) [6,7,30,15,46,13,51], or invertible neural networks (INNs) [47]. In most works, CNN-based autoencoder is selected as the basic framework. Ballé *et al.* [6] proposed an end-to-end optimized image compression framework based on nonlinear transformation, the additive noise quantization proxy and the fully factorized entropy model. Subsequently, the researchers focus more on improving the accuracy of the estimated entropy model using hyperprior [7], auto-regressive context model [30] and Gaussian Mixture Model (GMM) [15]. Different transformations are also proposed to enhance the expression capability of the latent space, such as residual blocks with attention module [15] and INN [47]. Some works [23,25] applied the spatially variant bit allocation strategy as a post-process [23] or by using importance map [25]. Our method is also based on the convolutional autoencoder approach, but we improve the content adaptability of the baseline method [30].

2.2 Content Adaptive Data Compression

The effectiveness of neural data compression relies on the generalization capability to unseen data in the evaluation process. However, domain shift between training and testing data and lack of adaptability may degrade the coding performance when compressing various types of testing data. To solve this issue, a straightforward idea is to over-fit the encoder-side components. In this way, the model can adapt to test samples during performance evaluation, and does not affect the reconstruction quality because the encoder is not involved in the decoding process. To this end, Campos *et al.* [12] refined the latents by directly back propagating them, and Yang *et al.* [50] further closed the discretization gap by replacing the differentiable approximation for quantization with Stochastic Gumbel Annealing (SGA) when refining the latents. Moreover, Lu *et al.* [28] updated the encoder on each test frame for neural video compression (NVC), which generates content adaptive latent codes by using the over-fitted encoder.

Recently, some full-model adaption methods for NVC have been proposed to adapt the decoder. The work in [34] updated both encoder and decoder when compressing I frames, and then transmitted the updates of the decoder parameters along with the compressed video sequences. These updating methods require hundreds or thousands of back propagation steps for each sample, which is extremely time-consuming. In summary, the encoder-side approaches do not utilize the adaptability of decoders and the full-model approach is often complex due to the additional model compression process.

Our proposed content adaptive methods adapt the latents and the decoder to the image content in a non-updating way. Our methods are also complementary to those encoder-side updating methods, which leads to a fully-adapted solution to address the issues of both domain shift and lack of adaptability.

2.3 Neural Video Compression

In recent years, significant progress has also been achieved for neural video compression (NVC). Increasing number of learning based approaches [45,29,3,19,26,16,17,21,22,14,20] have been proposed. Lu *et al.* [29] first proposed an end-to-end video compression framework DVC that follows the traditional hybrid coding framework and implements the key components with neural networks. Some subsequent works improved the motion compensation [3] or motion compression [19] for better optical flow based motion compensation. Recently, more works [16,17,21] were proposed to perform the operations in the feature space. Hu *et al.* [21] proposed the FVC framework where motion compensation and residual coding are performed in the feature space rather than the pixel space.

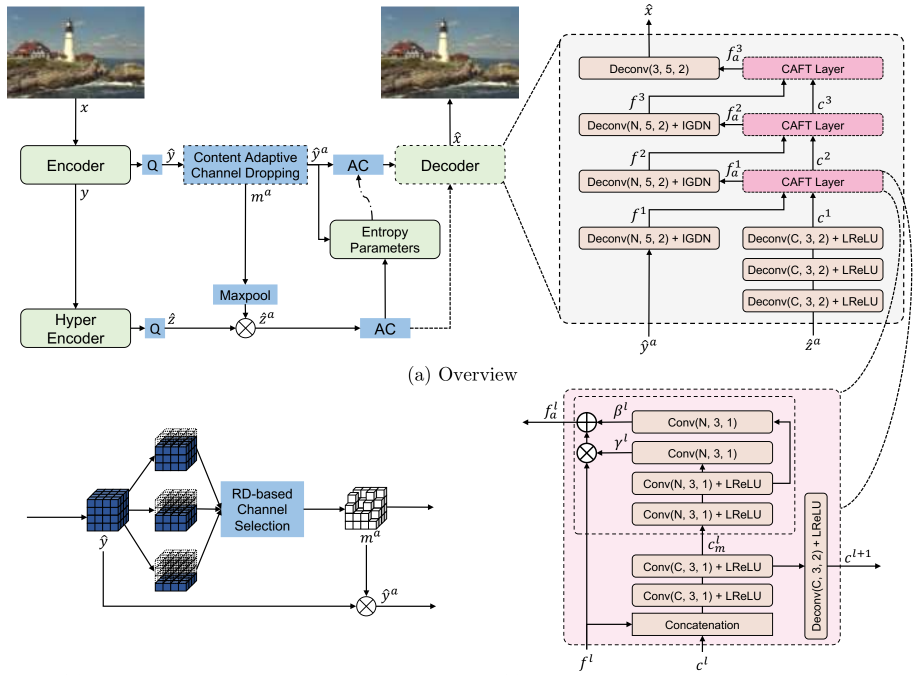
3 Proposed Method

3.1 Overall Architecture of Neural Image Compression

We use the state-of-the-art neural image compression (NIC) method [30] as our baseline method and apply our methods on top of both context version and the non-context version. The overview of the baseline framework is provided in Fig. 1(a). We also describe the details of the baseline method as follows.

At the encoder side, the input image x is first transformed into the latent representation y by using the encoder network, which consists of several convolution layers and uses the generalized divisive normalization (GDN) [5] layer as activation. The hyper-encoder captures the spatial dependencies of y and produces the hyperprior z. Then y and z are quantized into discrete-valued version $\hat{y}$ and $\hat{z}$ respectively by using the round operation, which is replaced by adding uniform noise [6] as an approximation during the training process. After that, the quantized features $\hat{y}$ and $\hat{z}$ are entropy coded into bit-stream. Each element in $\hat{z}$ is modeled as a factorized model $p_{\hat{z}}$ and each element in $\hat{y}$ is modeled as a Gaussian distribution $p_{\hat{y}|\hat{z}}$ conditioned on $\hat{z}$.

At the decoder side, the quantized hyperprior $\hat{z}$ is first entropy decoded and used to estimate the distribution of the quantized latent representation $\hat{y}$.



(a) Overview

(b) Content Adaptive Channel Dropping for the latents

(c) Content Adaptive Feature Transformation for the decoder

Fig. 1. Overview of our proposed framework based on [30] (a), the details in our content adaptive channel dropping (CACD) method for the latents (b) and the network architecture of our content adaptive feature transformation (CAFT) method for the decoder (c). For simplicity, the hyper-decoder and auto-regressive context model are denoted as "Entropy Parameters" and AC denotes arithmetic coding in the pipeline (a). The operation and modules with dashed container (*i.e.*, the CACD, the CAFT and the decoder in (a)) along with the dashed data flow are our newly proposed modules. In CACD (b), the features with different channel widths are first generated from the quantized latent representation $\hat{y}$. Then the rate-distortion (RD) based selection technique is applied to select the optimal channel number for each spatial location, which is stored in a binary mask m^a. Channel dropping is then completed by element-wise multiplication of the latents $\hat{y}$ and the mask m^a, which is also used to generate $\hat{z}^a$ (see section 3.2 for more details). In CAFT (c), we adapt each upsampled feature with the transformation parameters generated by using the Spatial Feature Transform (SFT) layer conditioned on the characteristic information of the image content. The characteristic information is first generated by using the hyperprior $\hat{z}^a$ in the decoder and then mixed with intermediate features and upsampled in CAFT layers (see section 3.3 for more details). Conv(C, K, S) denotes the convolution layer with the output channel C, the kernel size $K \times K$ and the stride S. LReLU denotes the LeakyReLU activation for simplicity.

In the non-context version of [30], $\hat{z}$ is fed into the hyper-decoder to estimate the mean and standard deviation of $\hat{y}$. While in the context version, an auto-regressive context model is added to utilize the entropy-decoded parts of $\hat{y}$ for more accurate entropy parameter estimation. Finally, the decoder takes $\hat{y}$ as the input to generate the reconstructed image $\hat{x}$ by using several deconvolution layers and inverse generalized divisive normalization (IGDN) layers.

During the training process of NIC, a rate-distortion optimization (RDO) problem is formulated to minimize the bit-rate cost and the distortion between the original image x and its reconstruction image $\hat{x}$. A Lagrange multiplier λ is used to control the trade-off between the bit-rate cost and the distortion. The loss function is formulated as follows:

$$R + \lambda D = H(\hat{y}) + H(\hat{z}) + \lambda d(x, \hat{x}) \tag{1}$$

where $H(\hat{y})$ and $H(\hat{z})$ denote the bit costs to compress $\hat{y}$ and $\hat{z}$, $d(x, \hat{x})$ denotes the distortion between the reconstructed image and the input image, where mean squared error (MSE) is usually used.

In our approach, we propose new operations and modules for the latents and the decoder. The channel dropping algorithm selects the optimal quality level at each spatial location for the latents $\hat{y}$ by minimizing the rate-distortion (RD) value. Then the latents $\hat{y}$ and the hyperprior $\hat{z}$ are replaced with their channel-adapted version $\hat{y}^a$ and $\hat{z}^a$ before entropy coding for bit-rate saving, where the exceeding channels are dropped (see section 3.2 for more details). In the decoder, we modulate the upsampled features after each IGDN layer by using the Spatial Feature Transform (SFT) [42] layer conditioned on characteristic information of the image content, which is first extracted from the hyperprior $\hat{z}^a$ (see section 3.3 for more details).

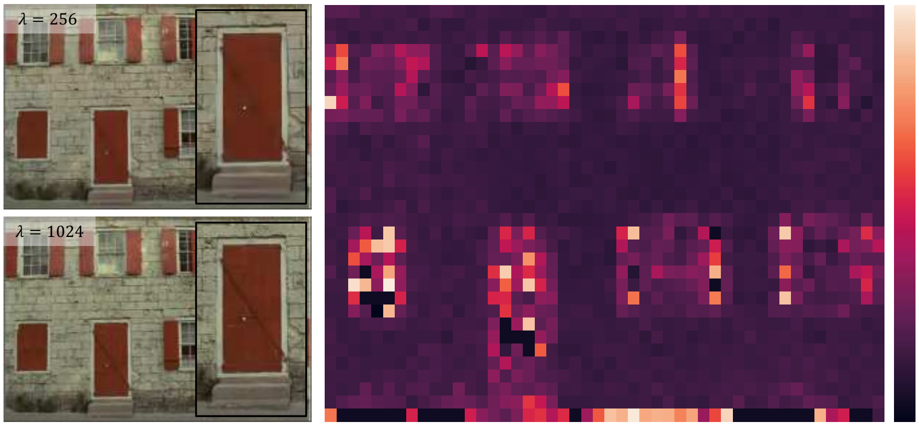
3.2 Content Adaptive Channel Dropping for the Latents

In neural image compression, rate control is implemented by training the models with different trade-offs (*i.e.*, different λ values) between bit-rate cost and reconstruction distortion. It is well-known that the more bits we use, the better reconstruction quality we can achieve. We also observe that the ability of converting extra bits to reconstruction quality (*i.e.*, the quality gain when assigning similar additional bits) is different among image blocks. To this end, we quantify this ability as “bit conversion ratio”, which is formulated as follows:

$$\eta(x, \lambda^l, \lambda^h)_i = \frac{PSNR(x, \lambda^h)_i - PSNR(x, \lambda^l)_i}{R(x, \lambda^h)_i - R(x, \lambda^l)_i} \tag{2}$$

where $PSNR(x, \lambda)$ denotes the peak signal-to-noise ratio (PSNR) between the input image x and its reconstructed image produced by the model trained with λ, $R(x, \lambda)$ denotes the bit-rate cost of the latents and the hyperprior generated by the model trained with λ, λ^l and λ^h denote the relatively lower and higher λ values respectively, and i denotes the ith spatial block of the image.

In Fig. 2, we provide a visualization example about bit conversion ratio on an image from the Kodak dataset [1]. Fig. 2(a) visualizes two images decoded by [30] trained with two different λ values. It is observed that the grains of both



(a) Decoded images from different λ values.

(b) Bit conversion ratio calculated from (a).

Fig. 2. An example of bit conversion ratio calculation on an image from the Kodak dataset based on the existing method [30].

woodworks (*i.e.*, the wooden door and windows) and the bricks are constructed with more details in the bottom image with high bit-rate. In Fig. 2(b), we observe that the bit conversion ratio of the wooden areas is much higher than that of the brick areas. To achieve better rate-distortion (RD) performance, it is therefore reasonable to assign more bits for the areas with higher bit conversion ratio. To this end, we aim at compressing each image block with a suitable quality level, at which the RD cost is minimal among all the alternative quality levels.

Before selecting the quality level at each spatial location for the latents $\hat{y}$, our content adaptive channel dropping (CACD) method needs to enable multiple quality levels in one single model. For each λ value, we first decide the corresponding maximum channel number $g(\lambda)$ (also called the optimal channel width in this work), where the RD performance saturates at this channel width even if more channels are allowed for this λ value [49]. Then we train our model with multiple rate-distortion optimization (MRDO) loss [49]. Note that we only set the additional elements in the channel dimension as zero instead of directly reducing the number of channels as in the slimmable implementation [49]. Specifically, for the original target λ value, we have K λ values (*i.e.*, K quality levels) including its original λ value and $K-1$ smaller λ values (K is set as 3 in this work). A mask $m^{g(\lambda)}$ is generated by setting the value to zero for the channel locations exceeding the channel width $g(\lambda)$, and one otherwise. The latent representation with level λ is generated by the element-wise multiplication operation between the latents $\hat{y}$ and the corresponding mask $m^{g(\lambda)}$ (*i.e.*, $\hat{y}^{g(\lambda)} \leftarrow \hat{y} \odot m^{g(\lambda)}$), and the hyperprior is also mapped in the same way (*i.e.*, $\hat{z}^{g(\lambda)} \leftarrow \hat{z} \odot Maxpool(m^{g(\lambda)})$). The MRDO loss is then formulated as follows,

$$\sum_{\lambda \in \Lambda} R(\hat{y}^{g(\lambda)}, \hat{z}^{g(\lambda)}) + \lambda D(\hat{y}^{g(\lambda)}) \tag{3}$$

where Λ denotes the set of K λ values, R and D denote the rate cost and the distortion in Eq.(1) respectively, and they are calculated by using the features with different quality levels.

As the model can compress the image with K quality levels, we adopt the block-based RD selection strategy, which selects the optimal channel width among alternatives for the smallest RD value at each spatial location. Specifically, at each spatial location, we calculate K RD values by using the features with the channel widths $g(\lambda)$ among alternative quality levels and store the channel width corresponding to the smallest RD value in the channel allocation matrix a. We further generate the adaptation mask m^a by setting the value to zero for the channel locations exceeding the allocated channel width, and one otherwise. Then the adapted features are generated by the element-wise multiplication operation with the adaption mask m^a (*i.e.*, $\hat{y}^a \leftarrow \hat{y} \odot m^a, \hat{z}^a \leftarrow \hat{z} \odot Maxpool(m^a)$). Therefore, our CACD method for the latents can automatically drop redundant elements at each spatial location and thus reduce the bit-rate cost.

3.3 Content Adaptive Feature Transformation for the Decoder

Domain shift between the training and testing data is a common problem for learning-based algorithms. Different from most tasks, the ground truth in neural image compression is exactly the same as the input image. Thus the model can be fine-tuned with the whole target domain dataset or even a target sample. Generally, only the encoder-side components are adapted because the change in the decoder will result in inconsistent reconstruction at the receiver side, which can not exploit the adaptability in the decoder. Although some works [34] synchronize the decoder to the receiver by transmitting the parameter changes, it is a non-trivial task to compress such parameter changes.

Recently, Spatial Feature Transform (SFT) [42] has shown efficient spatial adaptability for various vision tasks including image super-resolution [42], semantic image synthesis [33] and variable-rate image compression [36]. Inspired by these works, we propose the content adaptive feature transformation (CAFT) method for the decoder, which uses the SFT layers conditioned on the relatively high-level characteristic information to adapt the decoder to the image content.

Fig. 1(a) shows the architecture of the decoder network with our proposed CAFT layers. In the decoder, we first extract the characteristic information of the image content from the hyperprior $\hat{z}$ by using the image characteristic extractor, which consists of 3 transposed convolution layers (C is set as 192 in this work). To adapt the features in the decoder, we append our proposed CAFT layer after each IGDN layer, which considers the characteristic information c^l as the condition of the SFT layer to adapts the up-sampled feature f^l and produces the characteristic information c^{l+1} at next level, where $l = 1...L$ (L is set as 3 in this work).

The architecture of the CAFT layer is shown in Fig. 1(c). In the CAFT layer at level l, the characteristic information c^l and the feature to be adapted f^l are first concatenated in the channel dimension and mixed by using several convolution layers. The mixed characteristic information c_m^l then forwards twofold. On one hand, it is up-sampled to c^{l+1} if l is not the last level. On the other hand, it is input into the conditioned SFT layer, which generates the affine transformation parameters (γ^l, β^l) by learning the mapping function $\Psi(c^l) \mapsto (\gamma^l, \beta^l)$. The input feature f^l is then transformed by using the learned parameters (γ^l, β^l) to produce the content adapted feature f_a^l:

$$f_a^l = f^l \odot \gamma^l + \beta^l \tag{4}$$

where $\odot$ denotes the element-wise multiplication operation.

Our CAFT modulates the features by using the conditioned SFT layer whose condition is the relatively high-level characteristic information of the image content to improve decoder-side content adaptability.

4 Experiments

4.1 Experimental Setup

Datasets. We adopt the Flicker 2W dataset from [27] as our training dataset, which consists of 20,745 images. Each image is randomly cropped into 256×256 patches for data augmentation. The rate-distortion performance of our method is evaluated on the Kodak [1] and Tecnick [4] datasets.

Implementation Details. We apply our proposed content adaptive methods on both [30] and its non-context version. We train our models with seven λ values (*i.e.*, $\lambda = 128$, 256, 512, 1024, 2048, 4096 and 6144). We use N=M=192 for the three lower λ values and N=M=320 for the four higher values. We first train two models with higher λ values ($\lambda = 1024$ for low bit-rates and $\lambda = 8192$ for high bit-rates). Other models are then fine-tuned from their corresponding pretrained model with their λ values.

To train our model with content adaptive channel dropping (CACD), we first use the multi rate-distortion optimization (MRDO) technology (Eq.(3)) to achieve its original performance with multiple quality levels. Then the CACD module is activated to select the optimal channel width in the subsequent fine-tuning iterations.

We use the Adam [24] optimizer and set the batch size as 4. The initial learning rate is set as $5e-5$. Each fine-tuning step requires 1,000,000 iterations, which uses the initial learning rate for the first 600,000 iterations and $5e-6$ for the remaining iterations. For MS-SSIM [43] based rate-distortion performance evaluation, we further fine-tune our model with the learning rate of $5e-6$ for 500,000 iterations by using MS-SSIM as the distortion loss.

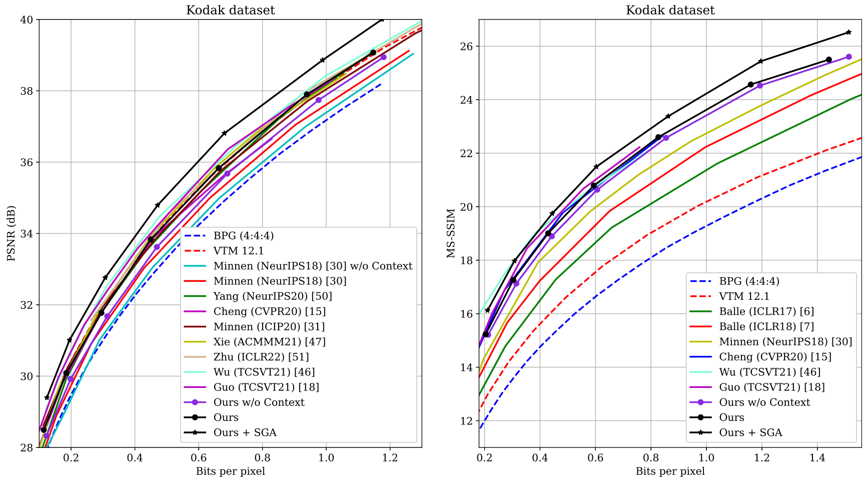


Fig. 3. Rate-distortion performance evaluation results on the Kodak dataset.

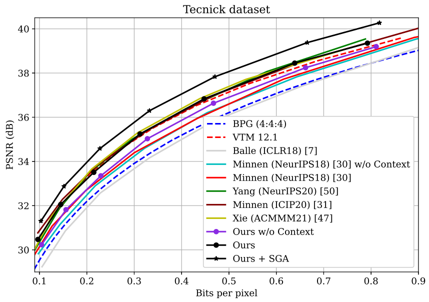


Fig. 4. Rate-distortion performance evaluation results on the Tecnick dataset.

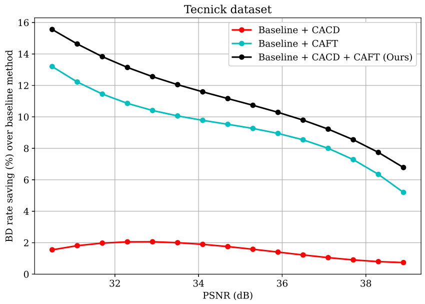


Fig. 5. BD rate saving (%) of our CACD and CAFT methods on the Tecnick dataset. We use [30] without the auto-regressive context model as our baseline method.

4.2 Rate-Distortion Performance

In Fig. 3, we report the performance of traditional image codecs [9,32], the state-of-the-art image compression methods [30,15,50,31,47,51,18,46] and our proposed methods (denoted as "Ours") on the Kodak dataset. VVC is evaluated by VTM-12.1 [2] on the CompressAI [8] evaluation platform. We evaluate our methods on both [30] and its non-context version (denoted as a suffix of "w/o Context"). We observe that our methods improve the rate-distortion performance on both versions of the baseline method in terms of both PSNR and MS-SSIM [43]. It is worth mentioning that our method is compatible with the state-of-the-art updating-based adaption method Stochastic Gumbel An-

Table 1. BDBR(%) results of our proposed methods on the Tecnick dataset. Negative values indicate bit-rate saving. We use [30] without the auto-regressive context model as the baseline method to calculate the BDBR results.

Methods	BDBR(%)
Baseline + CACD	-1.51
Baseline + CAFT	-9.47
Baseline + CACD + CAFT (Ours)	-11.08

nealing (SGA) [50]. We also report the fully-adapted result by combining our methods and SGA, which is denoted as "Ours + SGA". It is obvious that our fully-adapted method outperforms recent state-of-the-art methods [46,18,51] in terms of PSNR. For example, our fully-adapted method achieves 0.4dB improvement at 1.0bpp when compared with the current state-of-the-art methods Xie (ACMMM21) [47] and Wu (TCSVT21) [46].

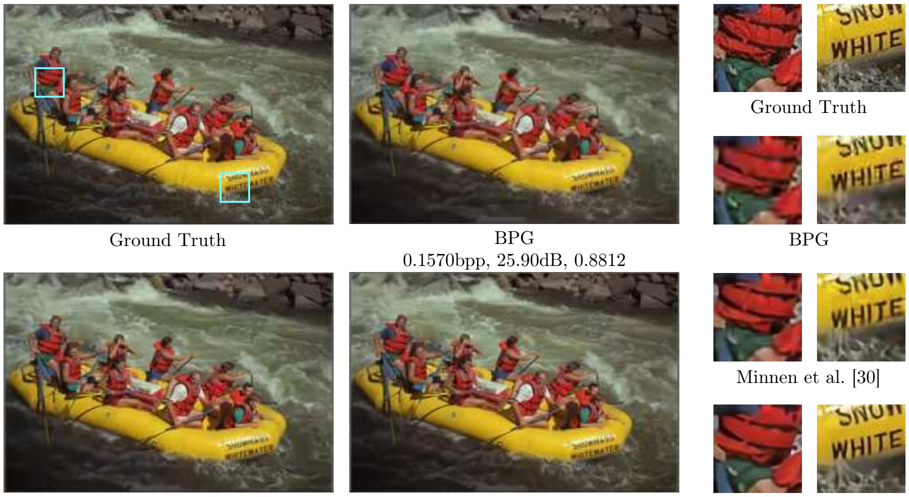
In Fig. 4, we also report the coding performance of different methods on the Tecnick dataset. We have similar observations as on the Kodak dataset that our fully-adapted method achieves the state-of-the-art performance at all bit-rates and achieves 0.6dB improvement at 0.5bpp when compared with current state-of-the-art methods Minnen (ICIP20) [31] and Xie (ACMMM21) [47]. The experimental results clearly demonstrate the effectiveness of our proposed fully-adapted method.

4.3 Ablation Study and Model Analysis

Effectiveness of the Proposed Methods. To demonstrate the effectiveness of our proposed content adaptive methods for the latents and the decoder, we conduct ablation study on the Tecnick dataset. To fairly compare our work with the updating-based adaption method SGA [50], we take the non-context version of [30] as the baseline method. We provide the BD rate saving result of our proposed methods over the baseline method based on the piecewise BDBR [10] results. As shown in Fig. 5, the alternative method equipped with our content adaptive feature transformation (*i.e.*, Baseline + CAFT) outperforms the baseline method with the BD rate saving from 5% to 13% at all bit-rates. Additionally, the alternative method equipped with our content adaptive channel dropping strategy (*i.e.*, Baseline + CACD) generally achieves better performance than the baseline method. Our method equipped with both CAFT and CACD achieves the best performance and outperforms all other methods, which saves about 14% bit-rate in low PSNR range. We also provide the BDBR [10] results compared with the baseline method in Table 1, which clearly demonstrates the improvement of our proposed methods over the baseline method. The ablation study results demonstrate that our overall framework is able to adapt to the image content on both latents and the decoder for better compression performance.

Table 2. BDBR(%) results about the compatibility of our method and SGA [50] on different datasets. Negative values indicate bit-rate saving. We use [30] without the auto-regressive context model as the anchor method to calculate the BDBR results.

Methods	Kodak	Tecnick
SGA	-15.17	-18.72
Ours w/o Context	-11.51	-11.08
Ours w/o Context + SGA	-26.52	-28.51



Ground Truth

BPG
0.1570bpp, 25.90dB, 0.8812

Ground Truth

BPG

Minnen et al. [30]

Minnen et al. [30]
0.1558bpp, 26.33dB, 0.9005

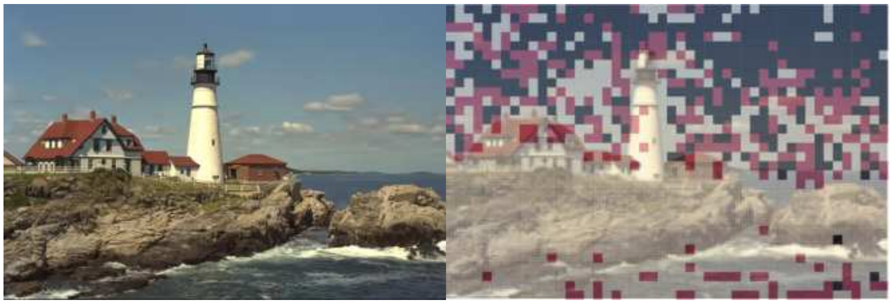
Ours
0.1499bpp, 26.86dB, 0.9046

Ours

Fig. 6. Qualitative comparison results of the traditional codes BPG [9], neural image compression method Minnen et al. [30] and our method.

Compatibility with Updating-based Method in the Encoder Side. Our methods adapt to the image content on both latents and the decoder, which is also compatible with the updating-based adaption method SGA [50]. To demonstrate the compatibility, we provide the BDBR [10] results on the Kodak and the Tecnick datasets in Table 2. Although our method (*i.e.*,“Ours w/o Context”) saves less bit-rates than “SGA”, our method in combination with SGA (*i.e.*,“Ours w/o Context + SGA”) outperforms “SGA”, which indicates that our content adaptive approach is complementary to SGA.

Qualitative Results. As shown in Fig. 6, we provide the visualization results of the reconstructed image *kodim14* from the Kodak dataset for qualitative comparison. It is observed that our method clearly improves the reconstruction quality over the baseline method [30] and achieves better performance than BPG. Our method preserves more details of the image content. For example, the artifacts can be clearly observed in both Minnen et al. [30] and BPG on the red life jacket, which are less obvious in our method. Additionally, the letters in front



(a) Reconstructed image *kodim21* from the Kodak dataset.

(b) Visualization of the channel allocation results. Lighter color indicates more channels.

Fig. 7. Visualization of the channel width selection results by using our method CACD for the latents. In (b), the white, red and black colors represent the quality levels from the highest level (*i.e.*, the most channel number) to the lowest level (*i.e.*, the least channel number).

of the boat reconstructed by our proposed method are more clear than those reconstructed by other baseline algorithms with similar bit-rates.

Visualization of Content Adaptive Channel Dropping. In Fig. 7, we visualize the allocated channel number selected by our method CACD. Fig. 7(a) is the reconstructed image of *kodim21* from the Kodak dataset and Fig. 7(b) visualizes the quality level selection results for the latents. The white, red and black colors represent three quality levels from high to low. It is observed that fewer channels are allocated in the sky area because the sky area is smooth and needs less bits for reconstruction, while full channels are allocated to preserve more details in the sharp areas like the rocks, houses and the lighthouse.

4.4 Experiments for Neural Video Compression

Datasets. We train our methods on the Vimeo-90k [48] dataset, which is used as the training dataset in DVC [29]. For performance evaluation, we use the video sequences from the HEVC Class B and Class C [37] datasets.

Implementation Details. We use an enhanced version of DVC [29] called "DVC*" as our baseline method, where the entropy models of both motion vector (MV) feature and residual feature are modeled by the mean-scale hyperprior. We train the models in a similar way as in neural image compression. We first pretrain a model with the λ value of 2048. The learning rate is set as 1e-4 for the first 1,800,000 steps and 1e-5 for the following 200,000 steps. Then we fine-tune the pretrained model with other λ values (*i.e.*, 256, 512 and 1024) for 500,000 steps. For the adapted model with our proposed content adaptive methods, we

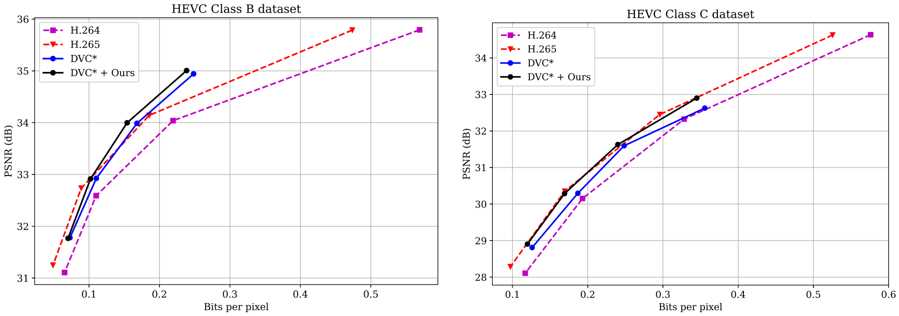


Fig. 8. Rate-distortion performance evaluation results on the HEVC Class B and Class C test sequences.

fine-tune the baseline model for 400,000 steps with the learning rate as 5e-5 and 100,000 steps with the learning rate as 5e-6 by using different λ values. We use the Adam [24] optimizer and set the batch size as 4 for all the training procedures.

Rate-Distortion Performance. Fig. 8 compares the rate-distortion performance between our methods and the baseline method DVC*. It is observed that our method improves the PSNR by about 0.1 dB at the middle bit-rate and by about 0.3 dB at other bit-rates on the HEVC Class C test sequence. Improvement can also be observed on the HEVC Class B test sequence, which has larger resolution than the HEVC Class C test sequence. The experimental results demonstrate that our methods are general and can be readily used for neural video compression.

5 Conclusions

In this work, we have proposed the content adaptive methods for both latents and the decoder to improve the content adaptability for neural image compression. Our newly proposed content adaptive channel dropping (CACD) method is able to adaptively compress different locations with different quality levels by dropping redundant channels for better bit-rate saving. Our newly proposed content adaptive feature transformation (CAFT) method at the decoder side can extract the characteristic information of the image content, which can be further regarded as the condition to transform the features in the decoder. Experimental results demonstrate that our content adaptive methods are general to different compression pipelines and are also complementary to the encoder-side updating-based content adaptive methods.

Acknowledgement

This work was supported by the National Key Research and Development Project of China (No. 2018AAA0101900).

References

1. Kodak lossless true color image suite. URLhttp://r0k.us/graphics/kodak/
2. VVC Official Test Model VTM. URLhttps://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-12.1
3. Agustsson, E., Minnen, D., Johnston, N., Ballé, J., Hwang, S.J., Toderici, G.: Scale-space flow for end-to-end optimized video compression. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 8500–8509. Computer Vision Foundation / IEEE (2020)
4. Asuni, N., Giachetti, A.: TESTIMAGES: a large-scale archive for testing visual devices and basic image processing algorithms. In: Giachetti, A. (ed.) Italian Chapter Conference 2014 - Smart Tools and Apps in computer Graphics, STAG 2014, Cagliari, Italy, September 22-23, 2014. pp. 63–70. Eurographics (2014)
5. Ballé, J., Laparra, V., Simoncelli, E.P.: Density modeling of images using a generalized normalization transformation. In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings (2016)
6. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017)
7. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018)
8. Bégaint, J., Racapé, F., Feltman, S., Pushparaja, A.: Compressai: a pytorch library and evaluation platform for end-to-end compression research. arXiv preprint arXiv:2011.03029 (2020)
9. Bellard, F.: Bpg image format. URL https://bellard.org/bpg (2015)
10. Bjontegaard, G.: Calculation of average psnr differences between rd-curves. ITU-T VCEG-M33, April, 2001 (2001)
11. Bross, B., Wang, Y., Ye, Y., Liu, S., Chen, J., Sullivan, G.J., Ohm, J.: Overview of the versatile video coding (VVC) standard and its applications. IEEE Trans. Circuits Syst. Video Technol. **31**(10), 3736–3764 (2021)
12. Campos, J., Meierhans, S., Djelouah, A., Schroers, C.: Content adaptive optimization for neural image compression. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019. p. 0. Computer Vision Foundation / IEEE (2019)
13. Chen, Z., Gu, S., Lu, G., Xu, D.: Exploiting intra-slice and inter-slice redundancy for learning-based lossless volumetric image compression. IEEE Transactions on Image Processing **31**, 1697–1707 (2022)
14. Chen, Z., Lu, G., Hu, Z., Liu, S., Jiang, W., Xu, D.: Lsvc: A learning-based stereo video compression framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6073–6082 (2022)
15. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 7936–7945. Computer Vision Foundation / IEEE (2020)

16. Djelouah, A., Campos, J., Schaub-Meyer, S., Schroers, C.: Neural inter-frame compression for video coding. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 6420–6428. IEEE (2019)
17. Feng, R., Wu, Y., Guo, Z., Zhang, Z., Chen, Z.: Learned video compression with feature-level residuals. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020. pp. 529–532. Computer Vision Foundation / IEEE (2020)
18. Guo, Z., Zhang, Z., Feng, R., Chen, Z.: Causal contextual prediction for learned image compression. IEEE Trans. Circuits Syst. Video Technol. **32**(4), 2329–2341 (2022)
19. Hu, Z., Chen, Z., Xu, D., Lu, G., Ouyang, W., Gu, S.: Improving deep video compression by resolution-adaptive flow coding. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12347, pp. 193–209. Springer (2020)
20. Hu, Z., Lu, G., Guo, J., Liu, S., Jiang, W., Xu, D.: Coarse-to-fine deep video coding with hyperprior-guided mode prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5921–5930 (2022)
21. Hu, Z., Lu, G., Xu, D.: FVC: A new framework towards deep video compression in feature space. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 1502–1511. Computer Vision Foundation / IEEE (2021)
22. Hu, Z., Xu, D., Lu, G., Jiang, W., Wang, W., Liu, S.: Fvc: An end-to-end framework towards deep video compression in feature space. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
23. Johnston, N., Vincent, D., Minnen, D., Covell, M., Singh, S., Chinen, T.T., Hwang, S.J., Shor, J., Toderici, G.: Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 4385–4393. Computer Vision Foundation / IEEE Computer Society (2018)
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
25. Li, M., Zuo, W., Gu, S., You, J., Zhang, D.: Learning content-weighted deep image compression. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3446–3461 (2021)
26. Lin, J., Liu, D., Li, H., Wu, F.: M-LVC: multiple frames prediction for learned video compression. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 3543–3551. Computer Vision Foundation / IEEE (2020)
27. Liu, J., Lu, G., Hu, Z., Xu, D.: A unified end-to-end framework for efficient deep image compression. arXiv preprint arXiv:2002.03370 (2020)
28. Lu, G., Cai, C., Zhang, X., Chen, L., Ouyang, W., Xu, D., Gao, Z.: Content adaptive and error propagation aware deep video compression. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12347, pp. 456–472. Springer (2020)

29. Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., Gao, Z.: DVC: an end-to-end deep video compression framework. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 11006–11015. Computer Vision Foundation / IEEE (2019)
30. Minnen, D., Ballé, J., Toderici, G.: Joint autoregressive and hierarchical priors for learned image compression. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. pp. 10794–10803 (2018)
31. Minnen, D., Singh, S.: Channel-wise autoregressive entropy models for learned image compression. In: IEEE International Conference on Image Processing, ICIP 2020, Abu Dhabi, United Arab Emirates, October 25-28, 2020. pp. 3339–3343. IEEE (2020)
32. Ohm, J.R., Sullivan, G.J.: Versatile video coding–towards the next generation of video compression. In: Picture Coding Symposium. vol. 2018 (2018)
33. Park, T., Liu, M., Wang, T., Zhu, J.: Semantic image synthesis with spatially-adaptive normalization. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 2337–2346. Computer Vision Foundation / IEEE (2019)
34. van Rozendaal, T., Huijben, I.A.M., Cohen, T.: Overfitting for fun and profit: Instance-adaptive data compression. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021)
35. Skodras, A., Christopoulos, C.A., Ebrahimi, T.: The JPEG 2000 still image compression standard. IEEE Signal Process. Mag. **18**(5), 36–58 (2001)
36. Song, M., Choi, J., Han, B.: Variable-rate deep image compression through spatially-adaptive feature transform. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. pp. 2360–2369. IEEE (2021)
37. Sullivan, G.J., Ohm, J., Han, W., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
38. Toderici, G., O'Malley, S.M., Hwang, S.J., Vincent, D., Minnen, D., Baluja, S., Covell, M., Sukthankar, R.: Variable rate image compression with recurrent neural networks. In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings (2016)
39. Toderici, G., Vincent, D., Johnston, N., Hwang, S.J., Minnen, D., Shor, J., Covell, M.: Full resolution image compression with recurrent neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 5435–5443. IEEE Computer Society (2017)
40. Wallace, G.K.: The JPEG still picture compression standard. Commun. ACM **34**(4), 30–44 (1991)
41. Wang, X., Jiang, W., Wang, W., Liu, S., Kulis, B., Chin, P.: Substitutional neural image compression. CoRR **abs/2105.07512** (2021), `https://arxiv.org/abs/2105.07512`
42. Wang, X., Yu, K., Dong, C., Loy, C.C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June

18-22, 2018. pp. 606–615. Computer Vision Foundation / IEEE Computer Society (2018)

43. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. vol. 2, pp. 1398–1402. Ieee (2003)
44. Wiegand, T., Sullivan, G.J., Bjøntegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 560–576 (2003)
45. Wu, C., Singhal, N., Krähenbühl, P.: Video compression through image interpolation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII. Lecture Notes in Computer Science, vol. 11212, pp. 425–440. Springer (2018)
46. Wu, Y., Li, X., Zhang, Z., Jin, X., Chen, Z.: Learned block-based hybrid image compression. IEEE Trans. Circuits Syst. Video Technol. **32**(6), 3978–3990 (2022)
47. Xie, Y., Cheng, K.L., Chen, Q.: Enhanced invertible encoding for learned image compression. In: Shen, H.T., Zhuang, Y., Smith, J.R., Yang, Y., Cesar, P., Metze, F., Prabhakaran, B. (eds.) MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021. pp. 162–170. ACM (2021)
48. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. International Journal of Computer Vision **127**(8), 1106–1125 (2019)
49. Yang, F., Herranz, L., Cheng, Y., Mozerov, M.G.: Slimmable compressive autoencoders for practical neural image compression. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 4998–5007. Computer Vision Foundation / IEEE (2021)
50. Yang, Y., Bamler, R., Mandt, S.: Improving inference for neural image compression. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
51. Zhu, Y., Yang, Y., Cohen, T.: Transformer-based transform coding. In: International Conference on Learning Representations (2022)