SASIC: Stereo Image Compression with Latent Shifts and Stereo Attention

Matthias Wödlinger
TU Wien, Vienna, Austria
mwoedlinger@cvl.tuwien.ac.at

Jan Kotera
TU Wien

Jan Xu
Deep Render, London, UK

Robert Sablatnig
TU Wien

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 978-1-6654-6946-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/CVPR52688.2022.00074

Abstract

We propose a learned method for stereo image compression that leverages the similarity of the left and right images in a stereo pair due to overlapping fields of view. The left image is compressed by a learned compression method based on an autoencoder with a hyperprior entropy model. The right image uses this information from the previously encoded left image in both the encoding and decoding stages. In particular, for the right image, we encode only the residual of its latent representation to the optimally shifted latent of the left image. On top of that, we also employ a stereo attention module to connect left and right images during decoding. The performance of the proposed method is evaluated on two benchmark stereo image datasets (Cityscapes and InStereo2K) and outperforms previous stereo image compression methods while being significantly smaller in model size.

1. Introduction

Lossy image compression is a fundamental task in image processing that aims to preserve the visual image content while reducing the bitrate needed for storage or transmission. It is a long-studied problem and a very active field of research both in traditional hand-crafted approaches and newly emerging learned methods. The traditional image encoding and decoding pipeline ("codec") typically consist of partitioning the image into small blocks to be processed separately, a linear transform to decorrelate the image values, intra block prediction (motion search) and residual coding to exploit repetition and self-similarity of the image content and decrease the entropy of its representation, quantization to obtain a finite set of symbols to code, and an entropy coder to store the resulting representation most efficiently. The decoding pipeline performs analogous operations in reverse. In contrast, modern learned compression methods typically process the image as a whole, without partitioning, by models, based on variational autoencoders, in which the latent representation is quantised and entropy coded with a learned probability distribution. The parametrization of this distribution, the entropy model, along with the autoencoder, are trained to minimize the cross-entropy against the true latent distribution so that the whole codec can be trained by optimizing the weighted rate/distortion loss.

In stereo image compression, the primary objective is the same with the additional possibility to achieve better performance by exploiting the mutual information between left and right images caused by their overlapping fields of view (albeit from slightly different viewpoints). In traditional methods, this can be achieved by the same set of tools available for inter-frame or motion prediction. Motion vector search in learned compression methods is far less explicit and extending them to stereo compression is therefore paradoxically more difficult. In the current literature, there are two existing deep learning based methods explicitly targeting stereo image compression – the DSIC model by Liu et al. [21] and HESIC model by Deng et al. [13]. In DSIC, a dense warp field is estimated, and warped features from the left image are fed to the encoder and decoder of the right image. In HESIC, a rigid homography transform in the image space is used. In both cases, the models are augmented by additional modules for joint entropy modelling and image enhancement, and as a result, they are rather large, and their training far from straightforward.

In contrast, the proposed model is very lightweight (4% and 10% the size of DSIC and HESIC, respectively, in number of parameters), conceptually simple, and does not require any special training procedure without sacrificing performance. The "backbone" of our method is an adaption of [38], but in principle, any general single image compression encoder-decoder model can be used as the backbone. The left image is encoded normally. After the right image is processed by the encoder, we find optimal horizontal shifts (minimizing the mean-square error) of each channel of its latent representation to the corresponding channel of the left image latent and subtract the two shifted channels so that only the residual is encoded for the right latent. This is motivated by the observation that the dominant rigid transform between rectified images in a stereo pair is a horizontal shift and working in the latent space results in larger effective disparity range due to downsampling. To account for

978-1-6654-6946-3/22/$31.00 ©2022 IEEE
DOI 10.1109/CVPR52688.2022.00074

smaller local displacements caused by depth variation, we also connect the two image representations by a stereo attention module [40], proposed originally for stereo image superresolution. A full description of the method is given in Sec. 3.

To summarize, we propose a method for stereo image compression with the following highlights:

- The principle of the method mimics the same techniques that are used for stereo compression in traditional codecs but remains fully end-to-end learnable.
- The method outperforms existing stereo image compression state-of-the-art on two standard test datasets.
- The method is very lightweight and easy to train, and its code is publicly available.[1]

In the rest of the paper, we summarize the related previous work, give a full description of the method and present and discuss the experimental results.

2. Related work

The image compression literature can be broadly classified into traditional and learned methods. Traditional methods use image partitioning (tiling), hand-crafted transforms, and redundancy elimination by explicit intra prediction. In learned methods the image is typically processed as a whole and the transform operations are parametrized and learned from the training data by minimizing the rate-distortion loss

$$\mathcal{L}_{RD} = \mathcal{R} + \lambda \mathcal{D}, \tag{1}$$

where $\mathcal{D}$ denotes the distortion metric and $\mathcal{R}$ cross-entropy of the latent code under a learned entropy model and $\lambda \in \mathbb{R}^+$ is the trade-off parameter. Both of these approaches rely on the quantization of the transformed representation and subsequent entropy coding to obtain the final bitstream.

Traditional compression The Joint Photographic Experts Group (JPEG) method, initially proposed in [36] is based on fixed 8x8 block tiling, chroma subsampling, discrete cosine transform, and next-block intra prediction. Its successor JPEG2000 [31] uses discrete wavelet transform and multiresolution processing. Modern image compression methods are usually wrappers over intra-frame compression developed for video codecs, such as BPG [7] (based on HEVC [32]), AVIF (based on AV1), or VVC intra [9]. VVC intra, in particular, has very slow encoding times but arguably the best compression performance to date among traditional methods.

Learned compression Initial work on end-to-end learned image compression started with the pioneering work by Toderici et al. [33] where a recurrent neural network is proposed for variable rate image compression. Another line of research in learned image compression was proposed by Ballé et al. [4] where an autoencoder based model with a parametrized distribution as prior for the latent is trained with rate-distortion loss for a fixed target bitrate. The autoencoder uses generalized divisive normalization [3] as nonlinearities and channel-wise piecewise-linear functions for the entropy model. The latter, however, does not allow for spatial adaptation and was later replaced by a per-pixel fully factorized zero-mean Gaussian with scale determined by a hyperprior [5]. In subsequent works, this model was further extended by allowing Gaussians with non-zero mean [25] or Gaussian mixtures [11]. Using an autoregressive network as a non-factorized conditional entropy model was proposed by Mentzer et al. [22] and Minnen et al. [25], which significantly improved performance at the expense of decoding complexity. A faster channel-based version appeared in [26].

Many different approaches and architectures were proposed, such as multiscale processing [28], dense blocks and content weighting [20], non-local attention modules [10, 11, 15], or asymmetric encoder-decoder setup [39]. For a more detailed overview, see [17]. A separate avenue of research is employing generative models, in particular GANs, in the decoders [1, 23, 34]. Such methods are capable of achieving high perceptual quality at very low bitrates, but the reconstructed image may lose semantic fidelity to the original.

Attention Since the seminal work of Vaswani et al. [35] where the transformer, a self-attention based model for machine translation, has been introduced, several ideas have been proposed for the use of self-attention in vision. Due to the quadratic growth in complexity, a naive application to image data is often prohibitive. In [30] this is circumvented by restricting attention to a local neighborhood, and in [14] self-attention is applied between image patches instead. For rectified stereo images, in particular, parallax stereo attention has been proposed in [37, 40], where attention at a certain location in the left (or right) image is limited to the corresponding epipolar line in the other image.

Stereo image compression Methods for compression of stereo image pairs work by saving bitrate through the exploitation of the mutual information between the left and right image. From traditional methods, MV-HEVC [24] is an extension of the HEVC video codec, which, on top of intra-frame prediction, also leverages prediction between multiple views. It performs very well, but the official implementation lacks support for several important features, such as operation in higher bit-depths or 4:4:4 chroma mode, and as a result, MV-HEVC is not very competitive against state-of-the-art single image compression methods. Learnable lossless stereo compression was recently proposed by

[1] https://github.com/mwoedlinger/sasic

Huang et al. [18], consisting of multiscale transforms, disparity estimation, and warping. A "distributed" compression method, which assumes that one image of the stereo pair is available for the decoder, was proposed by Ayzik and Avidan [2] and more recently by Mital et al. [27].

The list of learned lossy stereo compression methods is rather short and, to our best knowledge, includes the DSIC model by Liu et al. [21] from 2019 and HESIC model by Deng et al. [13] from 2021. The DSIC method uses skip modules that feed disparity-warped features from the encoded first image to the second and conditional entropy model to capture the dependence of image codes. The disparity map is used implicitly and is not transmitted in the bitstream. In the HESIC model, the second image is warped by an estimated homography, and only the residual is encoded. In addition, context-based entropy model and final quality-enhancement module are used to decrease bitrate and increase quality. Both methods reportedly outperform single-image compression methods by a solid margin. However, these methods are quite large and difficult to train.

3. Proposed method

Fig. 1 shows an overview of the proposed method. Our method compresses a stereo image pair in two streams that are connected in the latent, the entropy model and in the decoder. We use a hyperprior model to estimate the parameters of our latent entropy model. For a given stereo image pair $\boldsymbol{x}_1, \boldsymbol{x}_2$ in the first step the left image is encoded independently from the right image. Then the right image is processed by encoder module E and the optimal channel-wise horizontal shift for the quantised left latent $\hat{\boldsymbol{y}}_1$ is computed such that the MSE to the right latent $\boldsymbol{y}_2$ is minimal. For each channel c in the latent representations y_2 we find the optimal shift $s_c = \operatorname{argmin}_s \operatorname{MSE}\left(\boldsymbol{y}_2^{(c)} - \operatorname{shift}_s(\boldsymbol{y}_1^{(c)})\right)$ where $\operatorname{shift}_s(\boldsymbol{y})$ is defined as a tensor of the same size as $\boldsymbol{y}$ but horizontally (with respect to the original image) shifted by s pixels (zero-padded where necessary). Instead of $\boldsymbol{y}_2$ we then encode only the residual defined for each channel as $\boldsymbol{y}_{\text{res}}^{(c)} = \boldsymbol{y}_2^{(c)} - \operatorname{shift}_c(\boldsymbol{y}_1^{(c)})$. The search range for s_c is in our experiments limited to 64 pixels (in the downsampled latent representation) in one direction only (stereo disparity has only one polarity). The optimal shift can be found efficiently using a convolution the horizontal direction (achieved by corresponding padding) and element-wise operations to compute the MSE. It is therefore not significantly more demanding than other common operations in CNNs. The residual between the right latent and the shifted left quantised latent

$$\boldsymbol{y}_{\text{res}} \coloneqq E(\boldsymbol{x}_2) - \operatorname{shift}(\hat{\boldsymbol{y}}_1) \tag{2}$$

is then encoded. During decoding we decode the left latent first and add the shifted left quantised latent $\operatorname{shift}(\hat{y}_1)$ to the quantised residual $\hat{y}_{\text{res}}$ to obtain the right latent

$$\hat{\boldsymbol{y}}_2 \coloneqq \hat{\boldsymbol{y}}_{\text{res}} + \operatorname{shift}(\hat{\boldsymbol{y}}_1). \tag{3}$$

In the final step, $\hat{\boldsymbol{y}}_1$ and $\hat{\boldsymbol{y}}_2$ are processed jointly in the decoder modules D_1, D_2.

Applying a channel-wise shift is computationally cheap and requires almost no additional side information. Because the encoder performs $4\times$ downsampling, a maximal shift of 64 pixels in the latent corresponds to a shift of 256 in the original image. This equals to 72 bits of side information (6 bits times 12 latent channels), which for a 512×512 input image corresponds to only ≈ 0.00027 bits/pp overhead. Furthermore, a simple shift is also theoretically motivated by the fact that for a rectified stereo image pair, a shift is the transformation between the two image planes.

3.1. Encoding modules and quantization

The encoder/decoder architecture is loosely based on the single image compression method proposed in [38]. The encoder module E and hyperprior encoder h_E^1 and h_E^{res} each consist of four convolutional layers with Parametrized Rectified Linear Units (PReLUs) [16] as non-linearities. The structure of the encoder modules is shown in the top row of Fig.2. In both cases we downsample in the second and third convolution, which results in a $4\times$ downsampling for the latent and $16\times$ downsampling for the hyperlatent compared to the size of the inputs $\boldsymbol{x}_1, \boldsymbol{x}_2$. We use the same encoder module E (i.e. shared weights) for left and right images and the same architecture for h_E^1 and h_E^{res} (with separate weights).

Motivated by the discussion in [29], during training we use the noise approximation of quantization [4] for the rate loss and a straight-through-estimation (STE) quantization for the distortion loss.

3.2. Decoding

The architectures of the hyperprior decoders follow the same general structure of four convolutional layers with PReLUs as non-linearities; see the bottom row of Fig. 2. The hyperprior decoder for the left image h_D^1 gets the quantised hyperlatent $\hat{\boldsymbol{z}}_1$ as input and performs nearest neighbor upsampling after the second and third convolutional layers. The hyperprior decoder for the residual h_D^{res} gets both the $4\times$ upsampled quantised hyperlatent $\hat{\boldsymbol{z}}_{res}$ as well the shifted $\hat{\boldsymbol{y}}_1$ as input. There is no additional upsampling after the convolutional layers in h_D^{res}.

The final decoder modules D_1 and D_2 consist again of four convolutional layers with PReLU activation functions and upsampling after the second and third convolution but with stereo attention modules (SAM) from [40] before the first three convolutional layers that connect the left and right decoder stream; see overview in Fig. 3. SAM works by computing an attention mask between left and right input,

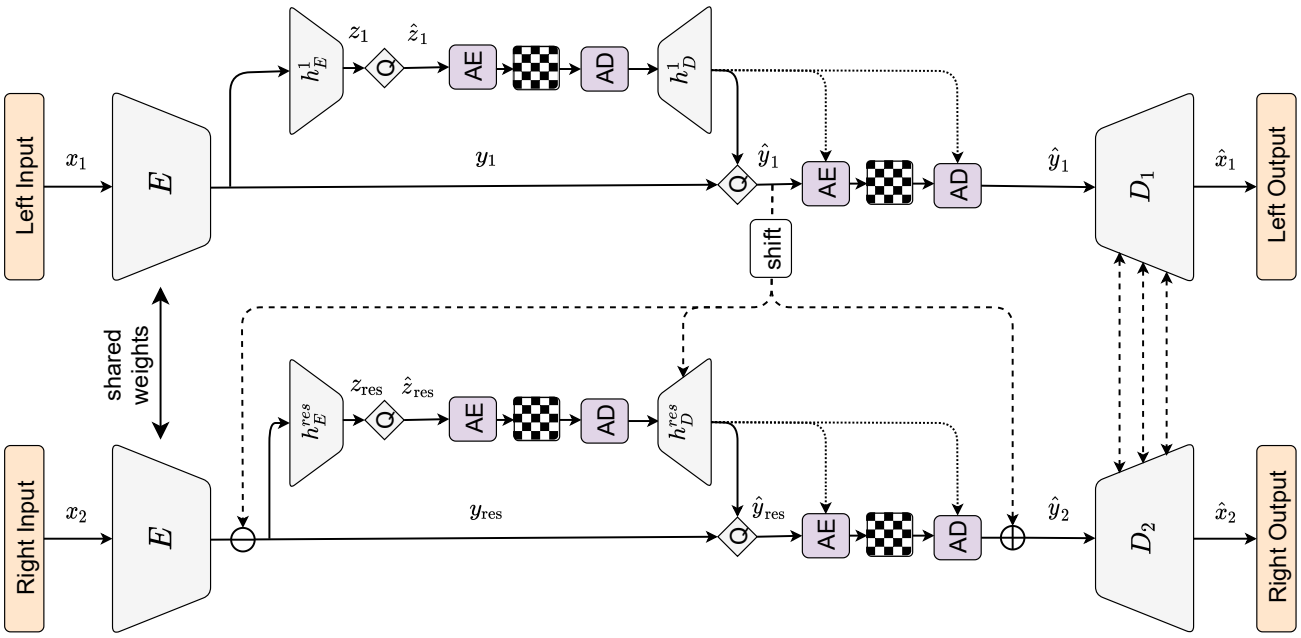

Figure 1. The full architecture of our proposed method. Layer structure of submodules are shown in Fig. 2 and Fig. 3. The arithmetic encoder *AE* and the arithmetic decoder *AD* are not relevant during training. The bitstreams are pictured with a checkerboard pattern. Dotted lines are connections not relevant during training and dashed lines show connections between the left and the right side.

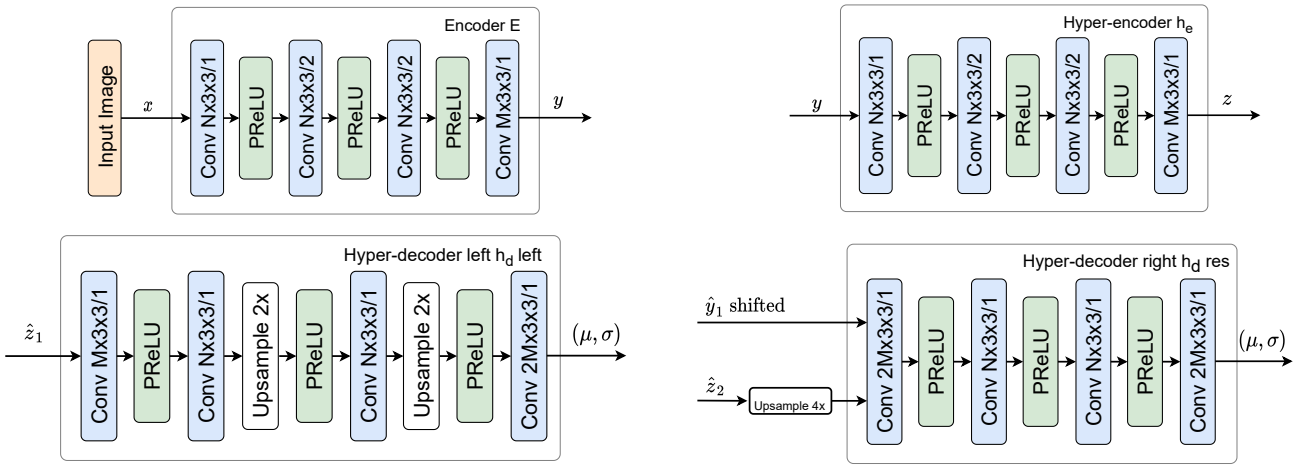

Figure 2. The top row shows the architecture of encoder E and hyper-encoder h_E. The bottom row shows the decoder of the hyperprior with the decoder for the left image bottom left and the decoder for the right image bottom right. We set $N = 192$ and $M = 12$

which is then used to warp left to right and vice versa. The input is then stacked with the warped image in the channel dimension and processed by the next convolutional layer. The attention is only computed between positions on the same epipolar line (we are assuming the images are rectified), which circumvents the issue of the quadratic complexity in sequence length of the attention mechanism.

3.3. Entropy estimation

Optimal entropy estimation is essential for the rate loss term during training and correct bitrate allocation during testing. For stereo image compression, where a pair of images with mutual information $H(\boldsymbol{x}_1, \boldsymbol{x}_2) > 0$ is compressed jointly, using the left latent as side information in the entropy model of the residual allows in principle to re-

duce the bitrate even further.

As discussed in 3.1 during training we mimic the quantization of the latent and hyperlatent with a noisy versions $\tilde{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{\epsilon}$ and $\tilde{\boldsymbol{z}} = \boldsymbol{z} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{U}(-0.5, 0.5)$. During testing we replace $\tilde{\boldsymbol{y}}$ and $\tilde{\boldsymbol{z}}$ with their integer quantised equivalents $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$.

Similar to [5] we include side information as a hyperprior $\tilde{\boldsymbol{z}}_n, n \in \{1, 2\}$ to reduce the entropy of the latents $\tilde{\boldsymbol{y}}_n, n \in \{1, 2\}$. We start by discussing the entropy model for the hyperpriors $\tilde{\boldsymbol{z_n}}$. Following the discussion in [5] we model the probability of the hyperprior $\tilde{\boldsymbol{z}}_n$ as a convolution of a parametric probability function $q_{\tilde{\boldsymbol{z}}_n}$ and a uniform distribution u

$$p_{\tilde{\boldsymbol{z}}_n}(\tilde{\boldsymbol{z}}_n \mid \boldsymbol{\theta}_{\tilde{\boldsymbol{z}}_n}) = (q_{\tilde{\boldsymbol{z}}_n} * u)(\tilde{\boldsymbol{z}}_n) \tag{4}$$

where $\boldsymbol{\theta}_{\tilde{\boldsymbol{z}}_n}$ denotes the parameters of $q_{\tilde{\boldsymbol{z}}_n}$ and $u(\tau) = \mathbb{1}_{[-0.5,0.5]}(\tau)$. $p_{\tilde{\boldsymbol{z}}_n}(\tilde{\boldsymbol{z}}_n \mid \boldsymbol{\theta}_{\tilde{\boldsymbol{z}}_n})$ can then be expressed via the cumulative density function $F_{\tilde{\boldsymbol{z}}_n}$ of $q_{\tilde{\boldsymbol{z}}_n}$:

$$\begin{aligned} p_{\tilde{\boldsymbol{z}}_n}(\tilde{\boldsymbol{z}}_n \mid \boldsymbol{\theta}_{\tilde{\boldsymbol{z}}_n}) &= \int_{-\infty}^{\infty} q_{\tilde{\boldsymbol{z}}_n}(\tau \mid \theta_{\tilde{\boldsymbol{z}}_n}) \mathbb{1}_{[-0.5,0.5]}(\tilde{\boldsymbol{z}}_n - \tau) d\tau \\ &= F_{\tilde{\boldsymbol{z}}_n}(\tilde{\boldsymbol{z}}_n + 0.5 | \theta_{\tilde{\boldsymbol{z}}_n}) - F_{\tilde{\boldsymbol{z}}_n}(\tilde{\boldsymbol{z}}_n - 0.5 | \theta_{\tilde{\boldsymbol{z}}_n}) \end{aligned}$$

We model the probability density function of $q_{\tilde{\boldsymbol{z}}_n}$ as a fully factorized Laplacian distribution

$$\text{Lap}_{\boldsymbol{\mu}_n, \boldsymbol{b}_n}(\tilde{\boldsymbol{z}}_n) = \prod_i \frac{1}{2b_{n;i}} \exp\left(-\frac{|\tilde{z}_{n;i} - \mu_{n;i}|}{b_{n;i}}\right), \tag{5}$$

where i denotes the pixel index and the parameters $\mu_{n;i} \in \mathbb{R}, b_{n;i} \in \mathbb{R}^+$ are shared between all positions in a channel. We denote the set of parameters $(\boldsymbol{\mu}_n, \boldsymbol{b}_n)$ by $\boldsymbol{\theta}_{\tilde{\boldsymbol{z}}_n}$.

We proceed similarly for the latent distributions and model them as convolutions of a parametric probability function $q_{\tilde{\boldsymbol{y}}_n}$, with $n \in \{1, \text{res}\}$, and a uniform distribution u.

$$p_{\tilde{\boldsymbol{y}}_1}(\tilde{\boldsymbol{y}}_1 \mid \tilde{\boldsymbol{z}}_1, \boldsymbol{\theta}_{\tilde{\boldsymbol{y}}_1}) = (q_{\tilde{\boldsymbol{y}}_1} * u)(\tilde{\boldsymbol{y}}_1) \tag{6}$$

$$p_{\tilde{\boldsymbol{y}}_{\text{res}}}(\tilde{\boldsymbol{y}}_{\text{res}} \mid \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{z}}_{\text{res}}, \boldsymbol{\theta}_{\tilde{\boldsymbol{y}}_{\text{res}}}) = (q_{\tilde{\boldsymbol{y}}_{\text{res}}} * u)(\tilde{\boldsymbol{y}}_{\text{res}}) \tag{7}$$

The distributions are conditioned on the hyperpriors $\tilde{\boldsymbol{z}}_n$ and the parameters of the hyperprior decoder $\boldsymbol{\theta}_{\tilde{\boldsymbol{y}}_n}$. For the residual latent $\tilde{\boldsymbol{y}}_{\text{res}}$ we additionally condition the probability distribution on $\tilde{\boldsymbol{y}}_1$ by using the shifted $\tilde{\boldsymbol{y}}_1$ as an additional input to the decoder of the hyperprior h_D^{res}. We use a fully factorized Laplacian distribution for $q_{\tilde{\boldsymbol{y}}_1}$ and, contrary to the hyperlatents where we have a separate set of learned parameters for every channel, we predict a different set of parameters for every position and channel with

$$\boldsymbol{\theta}_{\tilde{\boldsymbol{y}}_1} = h_D^1(\hat{\boldsymbol{z}}_1) \tag{8}$$

$$\boldsymbol{\theta}_{\tilde{\boldsymbol{y}}_{res}} = h_D^{res}(\hat{\boldsymbol{z}}_{res}, \text{shift}(\hat{\boldsymbol{y}}_1)). \tag{9}$$

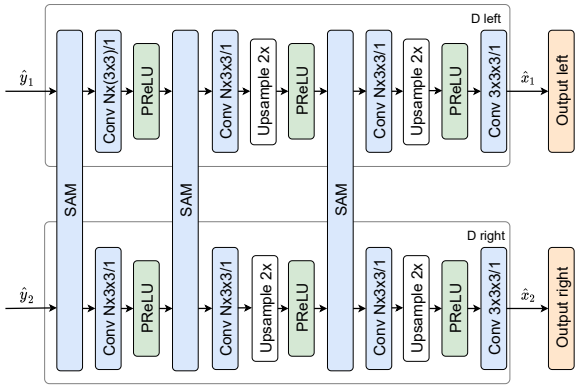

Figure 3. The decoder architecture. The SAM blocks contain the stereo attention module proposed in [40]. We set $N = 192$

The total rate is then the sum of the cross entropies for $\tilde{\boldsymbol{z}}_1, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{z}}_{\text{res}}, \tilde{\boldsymbol{y}}_{\text{res}}$:

$$\begin{aligned} \mathcal{R} = \mathbb{E}_{\boldsymbol{x_1}, \boldsymbol{x_2} \sim p_{\boldsymbol{x}}} \big[&- \log_2 p(\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{z}}_1 \mid \boldsymbol{\theta}_{\tilde{\boldsymbol{z}}_1}, \boldsymbol{\theta}_{\tilde{\boldsymbol{y}}_1}) \\ &- \log_2 p(\tilde{\boldsymbol{y}}_{res}, \tilde{\boldsymbol{z}}_{res} \mid \tilde{\boldsymbol{y}}_1, \boldsymbol{\theta}_{\tilde{\boldsymbol{z}}_{res}}, \boldsymbol{\theta}_{\tilde{\boldsymbol{y}}_{res}}) \big], \end{aligned} \tag{10}$$

where $p_{\boldsymbol{x}}$ denotes the true distribution of the input data.

3.4. Training

We train our model with the rate distortion loss

$$\mathcal{L} = \mathcal{R} + \lambda \mathcal{D}, \tag{11}$$

where $\mathcal{R}$ denotes the rate term from eq. (10) and $\mathcal{D}$ denotes the distortion metric, which in our is the sum of the MSE values for left and right images between the inputs $\boldsymbol{x}_1, \boldsymbol{x}_2$ and predictions $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2$,

$$\mathcal{D} = \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \sim p_{\boldsymbol{x}}} \big[\, \|\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1\|^2 + \|\boldsymbol{x}_2 - \hat{\boldsymbol{x}}_2\|^2 \, \big]. \tag{12}$$

4. Experiments

We will start with a brief overview of the datasets, followed by details about the implementation and benchmarks and finish with the discussion of the results and an ablations study.

4.1. Datasets

We evaluate our method on two datasets, Cityscapes [12] for distant views, and outdoor scenes and InStereo2K [6] for near views and indoor scenes. The datasets were also chosen to match recent works on stereo image compression.

Cityscapes The Cityscapes dataset [12] contains 5000 stereo image pairs of size 2048×1024 with maximum disparity of about 128 pixels. The set is split into 2975 training,

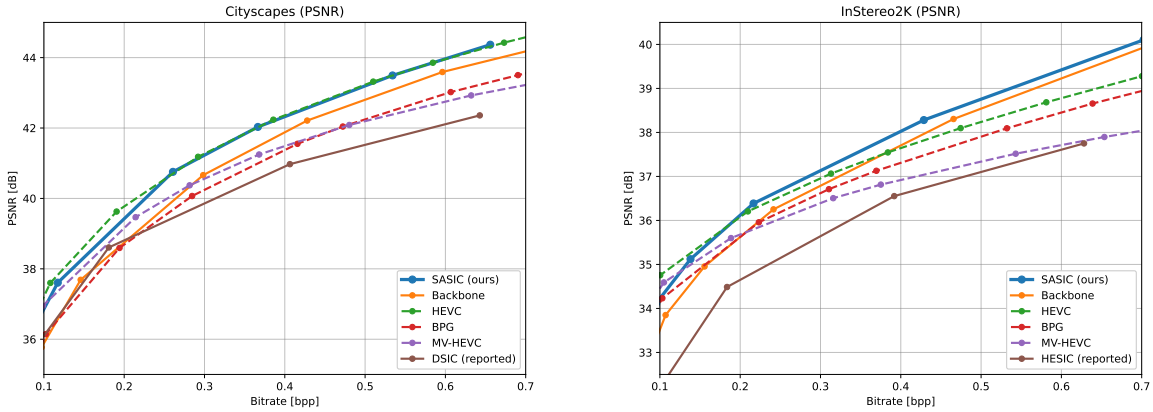

Figure 4. Rate distortion curves for our method against various compression baselines for Cityscapes (left column) and InStereo2K (right column) measured by PSNR

500 validation and 1525 test pairs. The images show street scenes from 50 German cities taken from a car while driving. For every image, we crop 64, 256, and 128 pixels from the top, bottom, and sides, respectively, to remove the car hood (repeated in each image) and rectification artefacts. This matches the transforms used in [21].

InStereo2K The InStereo2K dataset [6] contains 2060 stereo image pairs of indoor scenes of size 1080×860 and a maximum disparity of approx. 256 pixels. The set is split into 2010 image pairs for training and 50 for testing. The images are cropped minimally such that height and width are multiples of 16.

4.2. Implementation Details

We train our method for different values of $\lambda \in \{1e{-}3, \dots, 4e{-}1\}$ to achieve different desired target bitrates. For each bitrate, we train our models from scratch for 300 epochs on Cityscapes and 400 epochs on InStereo2K. We set the initial learning rate to 10^{-4} and drop the learning rate by a factor of 10 after 400k steps. We use the Adam optimizer [19] and a batch size of 1 for all runs. We train on random crops of size 256×256. The testing is done on full images with the exception of the cropping of Cityscapes specified in Sec. 4.1 and minimal required crop such that the input image size is divisible by 16. We train with the mean square error as the distortion metric $\mathcal{D}$ in eq. (11) and report results in terms of the PSNR metric.

4.3. Benchmark methods

We compare our method directly with the backbone alone (i.e. the backbone compression method is used to compress the left and right image independently) to highlight the improvements due to stereo handling. Apart from that, we show results of several state-of-the-art traditional or learned compression methods used either to compress both images of the stereo pair together or each of its images independently, depending on the capabilities of respective methods. With **HEVC** [32] we disable chroma subsampling and compress the stereo pair as a two-frame video sequence (left frame is an I frame and right frame is a P frame with the corresponding prediction). **MV-HEVC** [24] is used in its default config for stereo compression (two-view intra mode), but unfortunately supports only 4:2:0 chroma mode, which incurs a huge penalty in higher bitrates (we use bicubic upsampling for chroma channels). **BPG** [7] is used without chroma subsampling, and each image is compressed independently (essentially equivalent to HEVC intra, without stereo prediction). For **DSIC** [21] and **HESIC** [13] we quote their reported results on the respective datasets. We point out that comparison of the proposed method with other single-image learned methods (of similar size) is not critical as any one of such methods can potentially be used as the backbone of the proposed method and inherit its performance.

Method	# of params
Backbone	2.8M
SASIC (proposed)	6.6M
HESIC	66.2M
DSIC	159.6M

Table 1. Overview of the size of learned stereo compression methods in number of parameters.

4.4. Results

The experimental results of our method as well as the benchmarks on the test datasets are in Fig. 4, with the results for Cityscapes on the left and the results for InStereo2K on the right. The Cityscapes dataset with its many homogeneous regions is well suited for traditional compression, and

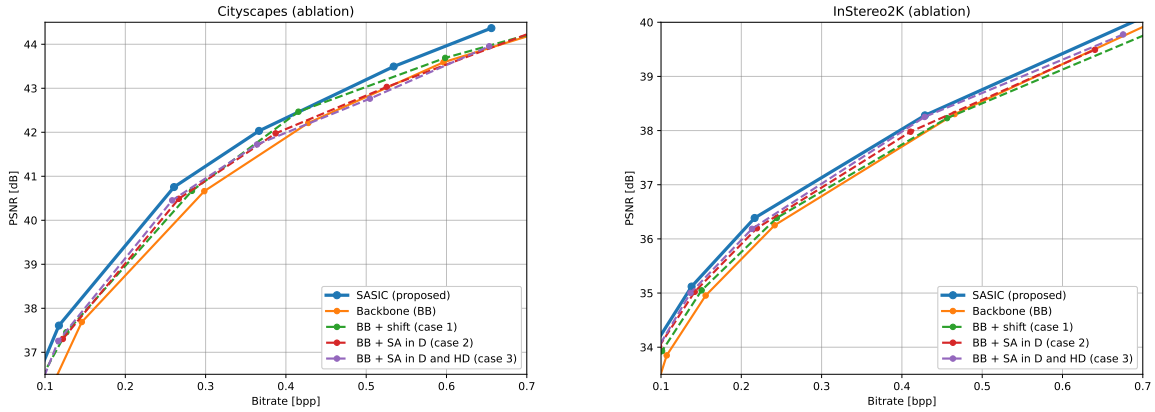

Figure 5. Ablation study: Comparison of the effects of the latent shift residual coding (green) and the stereo attention sub-modules (SA) on the rate-distortion performance. SA is used either only in the image decoder (red) or also in the hyperprior decoder (purple). The full proposed method (blue) and the original backbone (orange) are shown for reference.

beating traditional methods at PSNR is notoriously hard. HEVC performs best, with the proposed method (SASIC) trailing only slightly in low bitrates. Our method significantly outperforms the DSIC method and shows a consistent improvement over the backbone for the whole bitrate range.

The InStereo2K dataset offers a larger variation in image content than the Cityscapes. At higher bitrates, the learned methods work better, while at very low bitrates, the traditional methods keep their advantage. The proposed method (SASIC) is clearly on top for most of the bitrate range. The reported performance of HESIC is rather bad at PSNR. We were unable to reproduce the results from the HESIC paper, even when using the model from their official codebase[2], which is why we report the scores stated in their paper. The relatively poor results of MV-HEVC on both datasets are due to colour subsampling, which is not competitive in objective evaluations.

The difference between the proposed method (SASIC) and its backbone illustrates the gains due to stereo handling. This gain is most noticeable for low and medium bitrates, achieving approximately 15% and 18% bitrate reduction for the second image for Cityscapes and InStereo2K, respectively, at the middle $\mathrm{bpp} = 0.4$. For higher bitrates, the (absolute) reduction gain decreases, which could be intuitively explained as follows: If high reconstruction quality is demanded at the cost of higher bitrate, then it may be optimal to decrease the reliance on predictions and pay the bitrate rather than risk compromising the quality. However, by looking and the results of BPG and HEVC (analogous pair of compression methods, one for single images and the other for stereo), we can see that HEVC maintains its edge throughout the whole bitrate range, which suggests that further improvements in bitrate reduction are possible. A qualitative comparison on an image from the InStereo2K test set can be seen in Fig. 6. Additional examples are available in the supplemental material.

As can be seen from Tab. 1 our model is significantly smaller than existing methods for learned stereo image compression, with a model size of our SASIC model of only 4% and 10% of the DSIC and HESIC models respectively. A comparison and discussion of runtimes can be found in the supplementary material.

4.5. Ablation Study

In this paragraph, we compare the modules in our method and how they affect the overall rate-distortion curves. A comparison of these cases can be seen in Fig. 5. Furthermore we provide Bjøntegaard Delta PSNR (BD-PSNR) [8] and BD-Rate values in Table 2. BD-PSNR approximates quality increase for equivalent bitrate (higher is better) and BD-Rate approximates bitrate savings percentage for equivalent quality (negative and lower is better).

Backbone: For the backbone network, both images are compressed independently of each other, with the model used to compress the left image in the SASIC model.

Backbone + shift (case 1): For this case we remove the stereo attention modules that connect D_1 and D_2 in Fig. 1 and only keep the connections between $\hat{\boldsymbol{y}}_1$ and $\boldsymbol{y}_2$ as well as h_D^{res}. After training, we can see from the RD-curves in Fig. 5 that the model performs significantly worse than the full model, indicating that the stereo attention in the decoder helps to reduce the bitrate further. The BD-rate in Tab. 2 indicates that the remaining connection still leads to a substantial improvement for Cityscapes when compared to

[2]https://github.com/ywz978020607/HESIC. The authors did not release any training code. The model converged but we were unable to reach better results than stated in Fig. 4

Figure 6. A qualitative comparison on an image from the InStereo2K test set.

the backbone model where no such connections are present. For InStereo2K, the improvements are smaller.

Backbone + stereo attention in decoder (case 2): In this case we remove the connection between $\hat{\boldsymbol{y}}_1$ and $\boldsymbol{y}_2$ as well as h_D^{res}. We also replace the decoder of the hyperprior model for the right image with that of the left model (Fig. 2 bottom left). We keep the stereo attention connection between D_1 and D_2. The resulting model performs significantly better than the backbone; however still worse than the full SASIC model.

Backbone + stereo attention in decoder and hyperprior decoder (case 3): To show the effectiveness of the connections in the latent, not present in the *case 2*, we also investigate an architecture that is solely based on a stereo attention connection. For this case, we add three connections between the hyperprior decoders in the *case 2*, similar to the decoder connections. From Tab. 2 we see that, indeed, stereo attention alone is inferior in performance to our full SASIC model. In fact, the resulting model performs even worse than the simpler *case 2* model.

SASIC The SASIC model combines all improvements, i.e. Backbone + shift + stereo attention in decoder and hyperprior decoder. It is identical to the SASIC model in Fig. 4.

5. Conclusion

We have presented a new method for stereo image compression nicknamed SASIC. The method extends a general single image compression backbone model by two additions: a global shift and subtraction in the latent domain so that only the residual is encoded for the right image, and stereo attention modules in the decoder to account for finer local displacements between images. We have shown in the ablation study that the two proposed extensions in combination make a greater positive impact on the compression performance than individually. The resulting model is very lightweight, fast to train and during encoding and decoding, and yet outperforms the existing state-of-the-art in learned stereo image compression. Comparison to traditional methods shows that these perform slightly better at low bitrates; at mid to high bitrates, the proposed method is on par or superior and is much faster at encoding. The experimental results further show that the performance gains due to stereo handling diminish for higher bitrates. This is understandable, yet apparently not unavoidable, as the comparison with traditional compression shows, so there remains room for improvement in future work. Our model code is publicly available[3].

Method	Cityscapes BD-Rate	BD-PSNR
SASIC	-23.42	1.05
BB + Shift	-14.58	0.67
BB + SA in D	-19.70	0.80
BB + SA in D and HD	-17.78	0.73
Method	**InStereo2K BD-Rate**	**BD-PSNR**
SASIC	-11.28	0.38
BB + Shift	-2.28	0.07
BB + SA in D	-10.6	0.31
BB + SA in D and HD	-8.97	0.28

Table 2. Comparison of BD-Rate (lower is better) and BD-PSNR (higher is better) between the backbone model an each of the cases.

6. Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 965502. We thank Christian Besenbruch and the Deep Render team for valuable discussions.

References

[1] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In

[3] https://github.com/mwoedlinger/sasic

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 221–231, 2019. 2

[2] Sharon Ayzik and Shai Avidan. Deep image compression using decoder side information. In *European Conference on Computer Vision*, pages 699–714. Springer, 2020. 3

[3] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. Density modeling of images using a generalized normalization transformation. *CoRR*, abs/1511.06281, 2016. 2

[4] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In *5th International Conference on Learning Representations, ICLR 2017*, 2017. 2, 3

[5] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018. 2, 5

[6] Wei Bao, Wei Wang, Yuhua Xu, Yulan Guo, Siyu Hong, and Xiaohu Zhang. Instereo2k: a large real dataset for stereo matching in indoor scenes. *Science China Information Sciences*, 63(11):1–11, 2020. 5, 6

[7] Fabrice Bellard. BPG Image format. `https://bellard.org/bpg`. Accessed: 2021-09-24. 2, 6

[8] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *VCEG-M33*, 2001. 7

[9] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the Versatile Video Coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2021. 2

[10] Tong Chen, Haojie Liu, Zhan Ma, Qiu Shen, Xun Cao, and Yao Wang. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Transactions on Image Processing*, 30:3179–3191, 2021. 2

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7936–7945, 2020. 2

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5

[13] Xin Deng, Wenzhe Yang, Ren Yang, Mai Xu, Enpeng Liu, Qianhan Feng, and Radu Timofte. Deep homography for efficient stereo image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1501, June 2021. 1, 3, 6

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[15] Zongyu Guo, Zhizheng Zhang, Runsen Feng, and Zhibo Chen. Causal contextual prediction for learned image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2021. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 3

[17] Yueyu Hu, Wenhan Yang, Zhan Ma, and Jiaying Liu. Learning end-to-end lossy image compression: A benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 2

[18] Zihao Huang, Zhe Sun, Feng Duan, Andrzej Cichocki, Peiying Ruan, and Chao Li. L3c-stereo: Lossless compression for stereo images. *arXiv preprint arXiv:2108.09422*, 2021. 3

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[20] Mu Li, Wangmeng Zuo, Shuhang Gu, Jane You, and David Zhang. Learning content-weighted deep image compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3446–3461, 2021. 2

[21] Jerry Liu, Shenlong Wang, and R. Urtasun. Dsic: Deep stereo image compression. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3136–3145, 2019. 1, 3, 6

[22] Fabian Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Gool. Conditional probability models for deep image compression. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4394–4402, 2018. 2

[23] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11913–11924. Curran Associates, Inc., 2020. 2

[24] Philipp Merkle, Karsten Muller, Aljoscha Smolic, and Thomas Wiegand. Efficient compression of multi-view video exploiting inter-view dependencies based on h. 264/mpeg4-avc. In *2006 IEEE International Conference on Multimedia and Expo*, pages 1717–1720. IEEE, 2006. 2, 6

[25] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in Neural Information Processing Systems*, 31:10771–10780, 2018. 2

[26] 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, 2020. 2

[27] Nitish Mital, Ezgi Ozyilkan, Ali Garjani, and Deniz Gunduz. Neural distributed image compression using common information, 2021. 3

[28] Ken M. Nakanishi, Shin-ichi Maeda, Takeru Miyato, and Daisuke Okanohara. Neural multi-scale image compression. In C.V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision – ACCV 2018*, pages 718–732, Cham, 2019. Springer International Publishing. 2

[29] Shi Pan, Chris Finlay, Chri Besenbruch, and William Knottenbelt. Three gaps for quantisation in learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 720–726, 2021. 3

[30] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Standalone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 2

[31] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, 2001. 2

[32] 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. 2, 6

[33] G. Toderici, Sean M. O'Malley, S. J. Hwang, Damien Vincent, David C. Minnen, S. Baluja, Michele Covell, and R. Sukthankar. Variable rate image compression with recurrent neural networks. *CoRR*, abs/1511.06085, 2016. 2

[34] Michael Tschannen, Eirikur Agustsson, and Mario Lucic. Deep generative models for distribution-preserving lossy compression. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[36] G.K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. 2

[37] Longguang Wang, Yingqian Wang, Zhengfa Liang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning parallax attention for stereo image super-resolution. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12242–12251, 2019. 2

[38] Jan Xu, Alexander Lytchier, Ciro Cursio, Dimitrios Kollias, Christian Besenbruch, and Arsalan Zafar. Efficient context-aware lossy image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 130–131, 2020. 1, 3

[39] Yibo Yang, Robert Bamler, and Stephan Mandt. Improving inference for neural image compression. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 573–584. Curran Associates, Inc., 2020. 2

[40] Xinyi Ying, Yingqian Wang, Longguang Wang, Weidong Sheng, Wei An, and Yulan Guo. A stereo attention module for stereo image super-resolution. *IEEE Signal Processing Letters*, 27:496–500, 2020. 2, 3, 5