





Supplementary Materials

Heng Xu¹, Bowen Hai¹, Yushun Tang¹, and Zhihai He^{1,2*}

¹ Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China

² Pengcheng Laboratory, Shenzhen, China

xuh2022@mail.sustech.edu.cn, haibw2022@mail.sustech.edu.cn,
tangys2022@mail.sustech.edu.cn, hezh@sustech.edu.cn

In this Supplementary Materials, we provide more details and experimental results for further understanding of the proposed Space-Channel Hybrid (SCH) framework with window-based channel attention and wavelet transform for learned image compression (LIC).

A Traditional Image Compression Codecs Settings

A.1 VTM-23.1

VVC Test Model (VTM) from Versatile Video Coding (VVC) [17] standard is available at https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM. We use the benchmark script from CompressAI [3] platform to evaluate VTM-23.1, and the command is listed as follows:

```
python -m compressai.utils.bench vtm [path of dataset]
-c [path of VTM]/cfg/encoder_intra_vtm.cfg
-b [path of VTM]/bin
-q [quantization step size].
```

We set the quantization step size as {24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44}.

A.2 BPG

Better Portable Graphics (BPG) [4] is available at <https://bellard.org/bpg/>, and the command using CompressAI is as follows:

```
python -m compressai.utils.bench bpg [path of dataset]
-encoder-path [path of bpg]/bpgenc
-decoder-path [path of bpg]/bpgdec
-q [quantization step size].
```

We set the quantization step size as {26, 28, 30, 32, 34, 36, 38, 40, 42, 44}.

A.3 JPEG2000

JPEG2000 [14] is integrated into CompressAI as follows:

```
python -m compressai.utils.bench jpeg2000 [path of dataset]
-q [quantization step size].
```

We set the quantization step size as {10, 15, 20, 25, 30, 35, 40, 45}.

* Corresponding author.

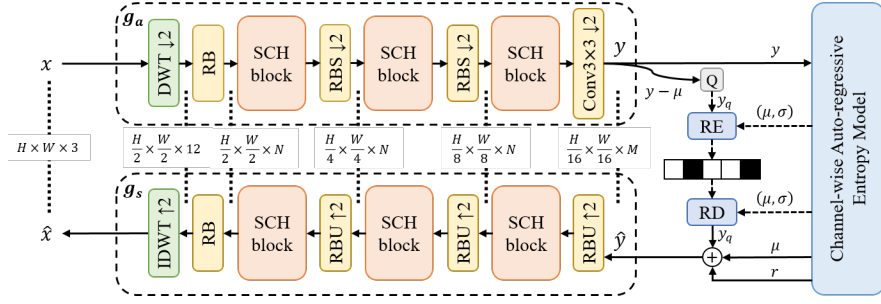


Fig. 1: The overall architecture of our model.

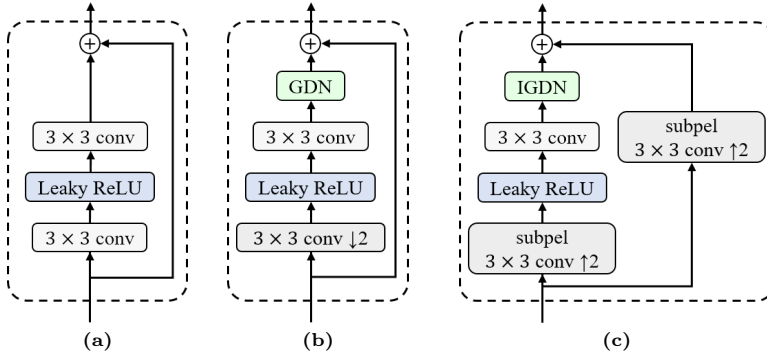


Fig. 2: (a) Residual Block (RB), (b) Residual Block with Stride (RBS), and (c) Residual Block Upsampling (RBU) from [5].

B Detailed Network Architecture

The architecture of our model is illustrated in Fig. 1. The numbers of heads of SCH blocks in g_a and g_s are $\{8, 16, 32, 32, 16, 8\}$. Space attention modules and channel attention modules share the same number of heads in the same level of blocks. Following [10], the number of channel slices in the channel-wise auto-regressive model is set as 5.

The structures of Residual Block (RB), Residual Block with Stride (RBS) and Residual Block Up-sampling (RBU) are shown in Fig. 2. Generalized Divisive Normalization (GDN) [2] is used in LIC instead of batch normalization [8] because of its spatial-adaptive learning, which is formulated as:

$$y_i(m, n) = x_i(m, n) \cdot \frac{1}{\sqrt{\beta_i + \sum_j \gamma_{ij}(x_j(m, n))^2}}, \quad (1)$$

where i is the output channel index, j is the input channel index, γ acts as the role of the 1×1 convolutional kernel, and β is the bias. This normalization is invertible, and we use GDN in the analysis transform g_a and IGDN in the

synthesis transform g_s . IGDN is formulated as follows:

$$y_i^{inv}(m, n) = x_i(m, n) \cdot \sqrt{\beta_i + \sum_j \gamma_{ij}(x_j(m, n))^2}. \quad (2)$$

In RBU, "subpel 3×3 conv" is 3×3 sub-pixel convolution for up-sampling, combining a 3×3 convolutional layer and a Pixel Shuffle module from PyTorch [13].

Note that we do not replace RBS and RBU between successive SCH blocks with the proposed wavelet transform module for down-sampling and up-sampling. This is because the channel size of intermediate features of g_a and g_s is a constant N with a large value of 256. For the $H \times W \times N$ input, the wavelet transform module transforms it into the shape of $\frac{H}{2} \times \frac{W}{2} \times 4N$. Therefore, we need to perform the transformation on the channel dimension to maintain the consistency of our LIC framework. One option is to use the residual block with input channel $4N$ and output channel N to reduce the channel size after wavelet transform, but it results in a large quantity of parameters in this residual block. The other option is to use the residual block with input channel N and output channel $\frac{N}{4}$ to reduce the channel size before wavelet transform, but it causes information loss in the network. As a result, we only use the wavelet transform module to process raw input images.

C Detailed Rate-Distortion Results

We provide detailed Rate-Distortion results in Figs. 3 to 6, corresponding to Rate-Distortion results in Section 4.2 of the main paper.

Kodak [9] dataset is available at <http://r0k.us/graphics/kodak>.

Tecnick [1] dataset is available at https://sourceforge.net/projects/testimages/files/OLD/OLD_SAMPLING/testimages.zip.

CLIC Professional Validation [15] dataset is available at https://data.vision.ee.ethz.ch/cvl/clic/professional_valid_2020.zip.

CLIC 2021 Test [16] dataset is available at https://storage.googleapis.com/lic2021_public/professional_test_2021.zip.

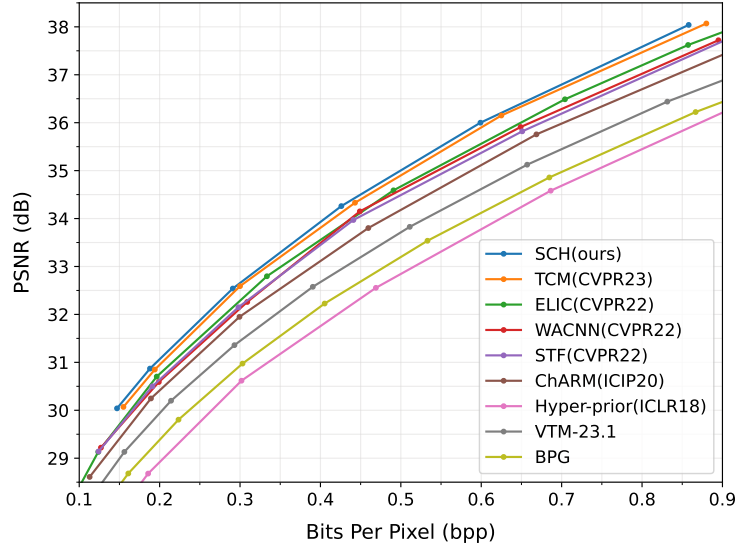


Fig. 3: Rate-Distortion results on Kodak [9] (24 images, 768×512 or 512×768).

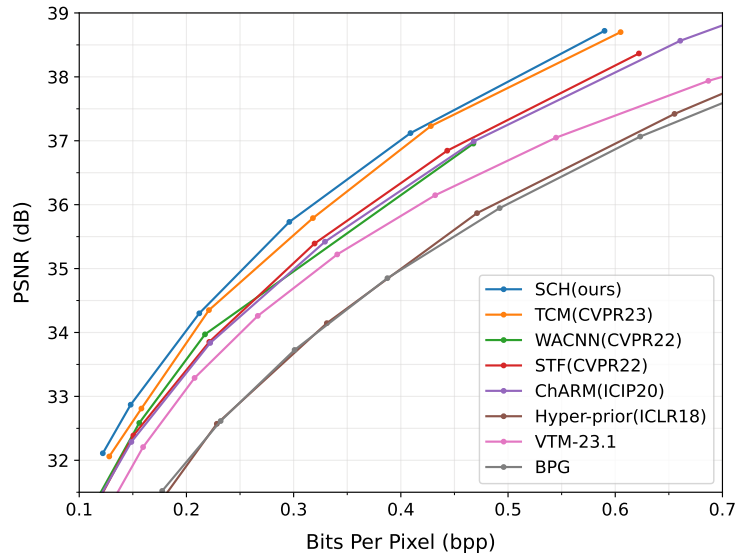


Fig. 4: Rate-Distortion results on Tecnick [1] (100 images, 1200×1200).

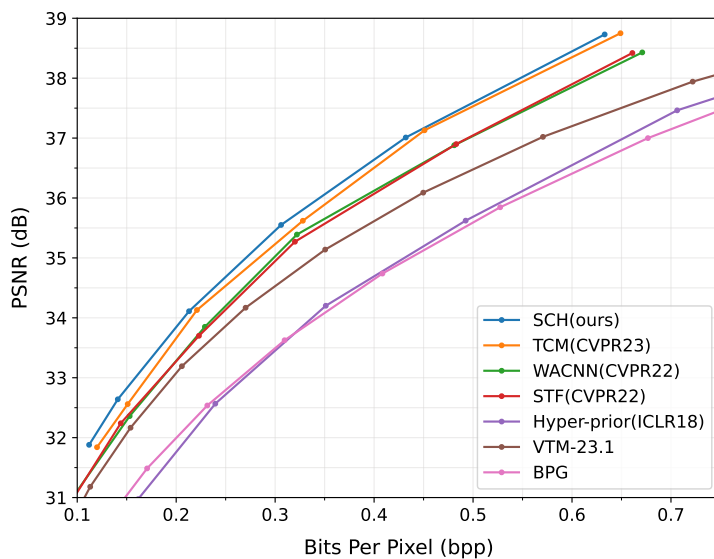


Fig. 5: Rate-Distortion results on CLIC Professional Validation [15] (41 images, from 512×384 to 2048×1370).

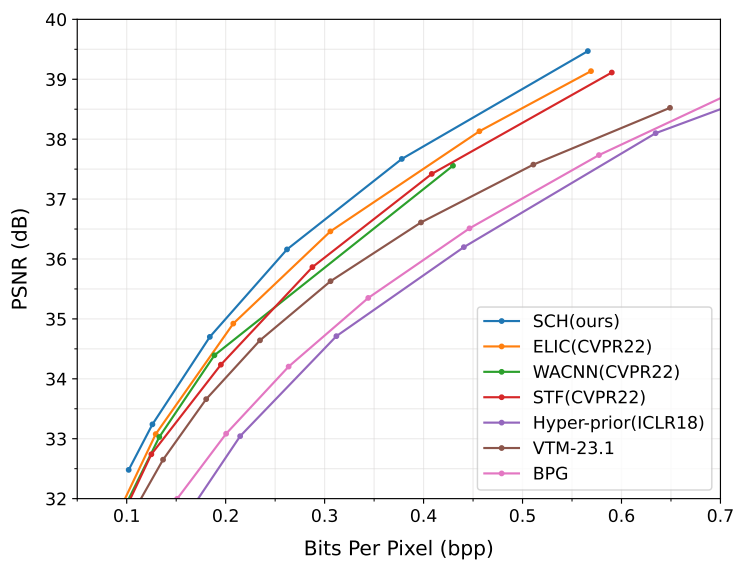


Fig. 6: Rate-Distortion results on CLIC 2021 Test [16] (60 images, from 751×500 to 2048×1415).

D Discussions on the Parameters of Attention Module

If we ignore parameters in bias and normalization, for the $L \times C$ input, a space attention module has $12C^2$ parameters. The self-attention module contains three input projection layers and one output projection layer, their matrices W_Q , W_K , W_V , and W_O have $4 \times (C \times C) = 4C^2$ parameters in total. Multilayer Perceptron (MLP) layer contains two linear projection layers, having $C \times 4C + 4C \times C = 8C^2$ parameters given the MLP ratio of 4. The sum of parameters in a space attention module is $4C^2 + 8C^2 = 12C^2$. After the space-channel dimension transposition, our window-based channel attention module contains $12(hw)^2$ parameters if the window size is $h \times w$.

If we include parameters in bias and normalization, the space attention module has $12C^2 + 13C$ parameters. Biases of four linear projection layers in the self-attention module have $4C$ parameters. Biases of two linear projection layers in the MLP layer have $4C + C = 5C$ parameters. Layer normalization modules have two learnable parameters, γ and β , whose lengths are both C . Consequently, layer normalization modules in the self-attention module and MLP layer have $2C + 2C = 4C$ parameters. The sum of parameters in bias and normalization is $4C + 5C + 4C = 13C$. For our window-based channel attention module, it contains $12(hw)^2 + 13hw$ parameters. Since hw is generally smaller than C , our channel attention module is more parameter-efficient in both cases.

E More Ablation Studies

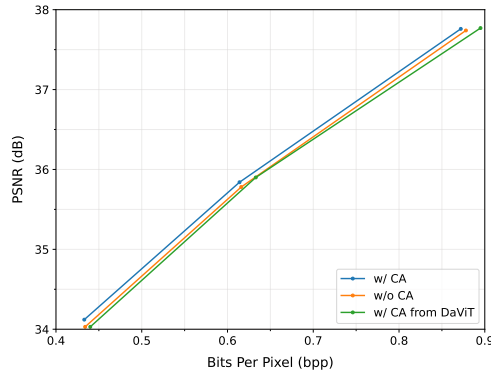
We provide ablation studies on the Kodak [9] dataset in Fig. 7. Our SCH framework benefits from the global information learning in our window-based channel attention module, so the performance gain on this lower-resolution dataset is reduced. These results still validate the effectiveness of the proposed channel attention module and wavelet transform module.

F More Visualization Examples

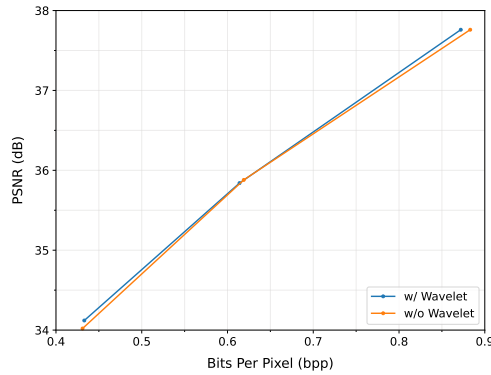
F.1 Effective Receptive Fields

We provide more examples in Figs. 8 and 9 to visualize Effective Receptive Fields (ERF) [12] of computing modules from our SCH, TCM [10] and DaViT [6]. ERF can be obtained as gradients of one point in the feature map to all pixels at the input, and we select the output feature from modules of the last block in g_a to generate it. Results are normalized and clipped by a threshold of 0.3 for appropriate visualization.

We compare our window-based channel attention with residual block [7], window attention [11], shifted-window attention [11] and channel group attention [6]. For Figs. 8a to 8c, 8e and 8g, and Figs. 9a to 9c, 9e and 9g, these modules work with wavelet transform, and it is obvious that the proposed window-based channel attention provides the largest ERF. Since ERF indicates the ability to



(a) Channel Attention Module



(b) Wavelet Transform Module

Fig. 7: Ablation studies on the Kodak [9] dataset (24 images, 768×512 or 512×768). (a) Channel attention modules (with or without our channel attention, with channel group attention [6]). (b) Wavelet transform module (with or without wavelet transform).

capture global dependencies in the image, our window-based channel attention module excels in global information learning compared with other modules from [6, 7, 10, 11]. For Figs. 8d, 8f and 8h, and Figs. 9d, 9f and 9h, these modules work without wavelet transform, and they offers smaller ERFs compared with modules with wavelet transform, validating the effectiveness of the wavelet transform module in enlarging ERF. It is worth noticing that shifted-window attention provides an ERF significantly smaller than residual block and window attention, confirming the restricted growth of receptive fields for shifted-window attention argued by [18]. Therefore, we replace the shifted-window attention with our window-based channel attention in our SCH block to capture global dependencies in image features for LIC.

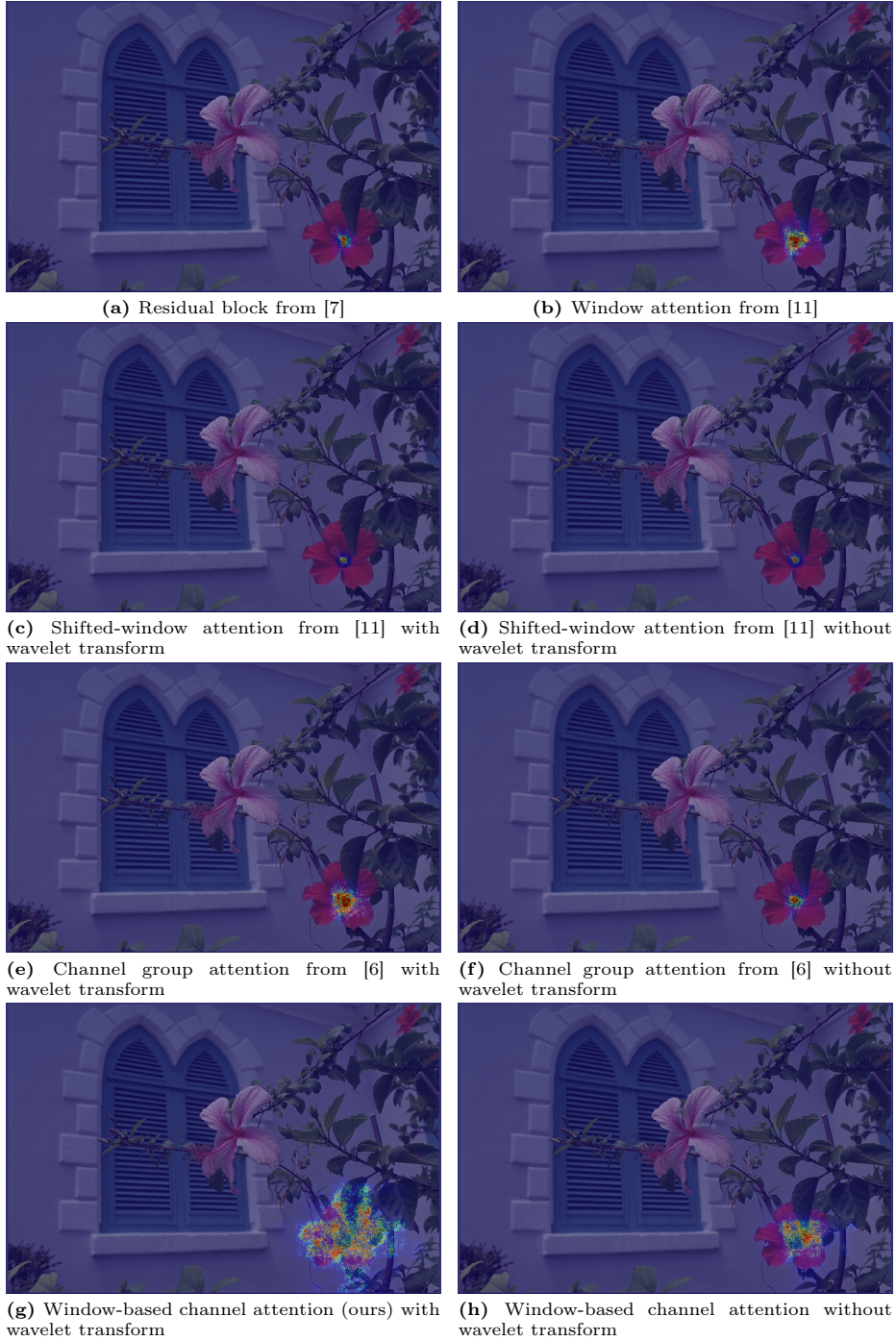
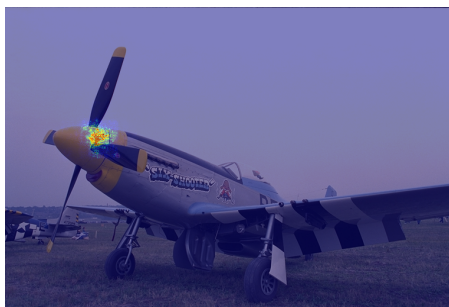


Fig. 8: Effective Receptive Fields (ERF) on *kodim07* from computational modules. The color changes from blue to red as the value increases.



(a) Residual block from [7]



(b) Window attention from [11]



(c) Shifted-window attention from [10] with wavelet transform



(d) Shifted-window attention from [10] without wavelet transform



(e) Channel group attention from [6] with wavelet transform



(f) Channel group attention from [6] without wavelet transform



(g) Window-based channel attention (ours) with wavelet transform



(h) Window-based channel attention (ours) without wavelet transform

Fig. 9: Effective Receptive Fields (ERF) on *kodim20* from computational modules. The color changes from blue to red as the value increases.



Fig. 10: Reconstructed images of *kodim07* from Kodak [9].

F.2 Qualitative Results

We provide more qualitative results in the Kodak [9] and the Tecnick [1] datasets to compare our SCH method with an available learn method STF [19] and traditional image codecs in Figs. 10 to 13. Our method reconstructs the clearest details while maintaining low bit rates in the following results.

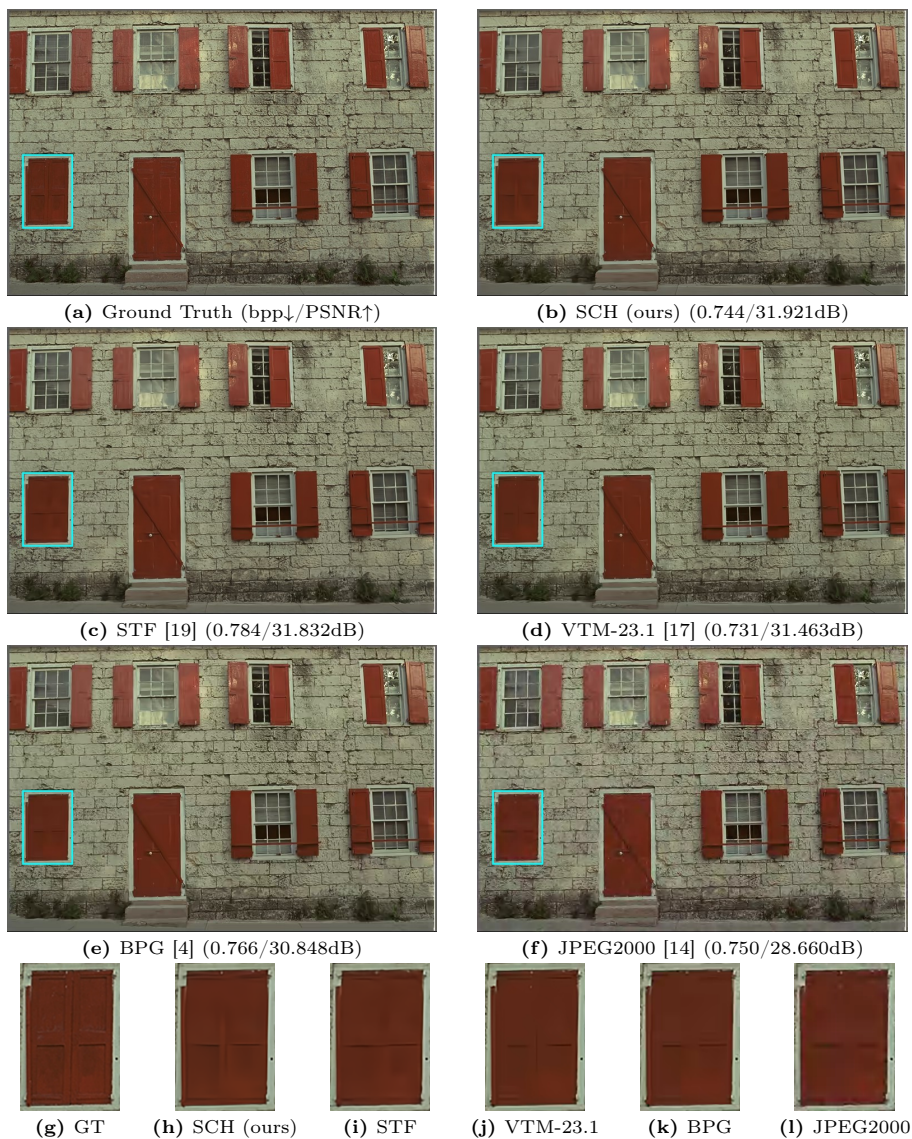


Fig. 11: Reconstructed images of *kodim01* from Kodak [9].

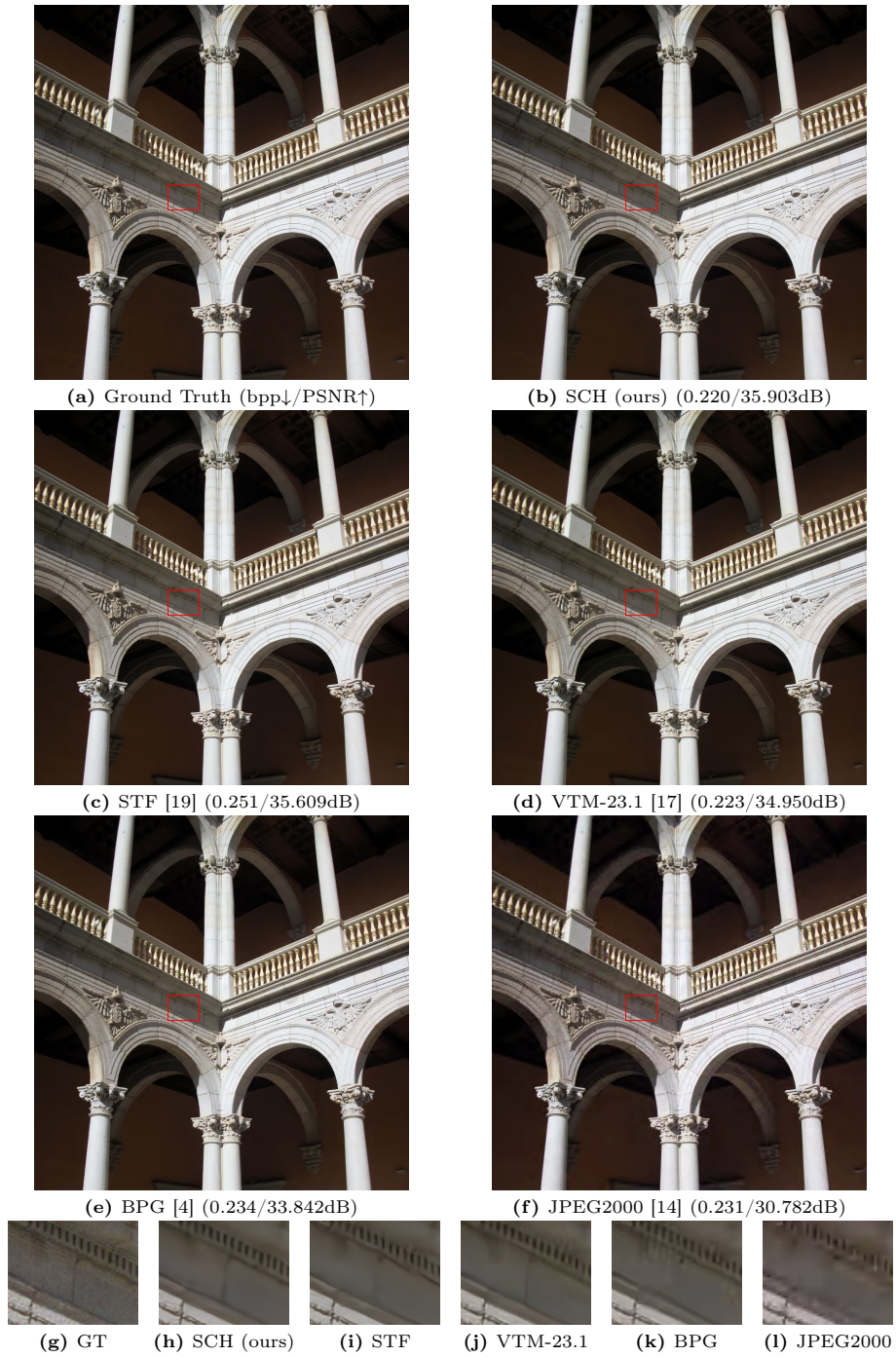


Fig. 12: Reconstructed images of *RGB_OR_1200x1200_003* from Tecnick [1].

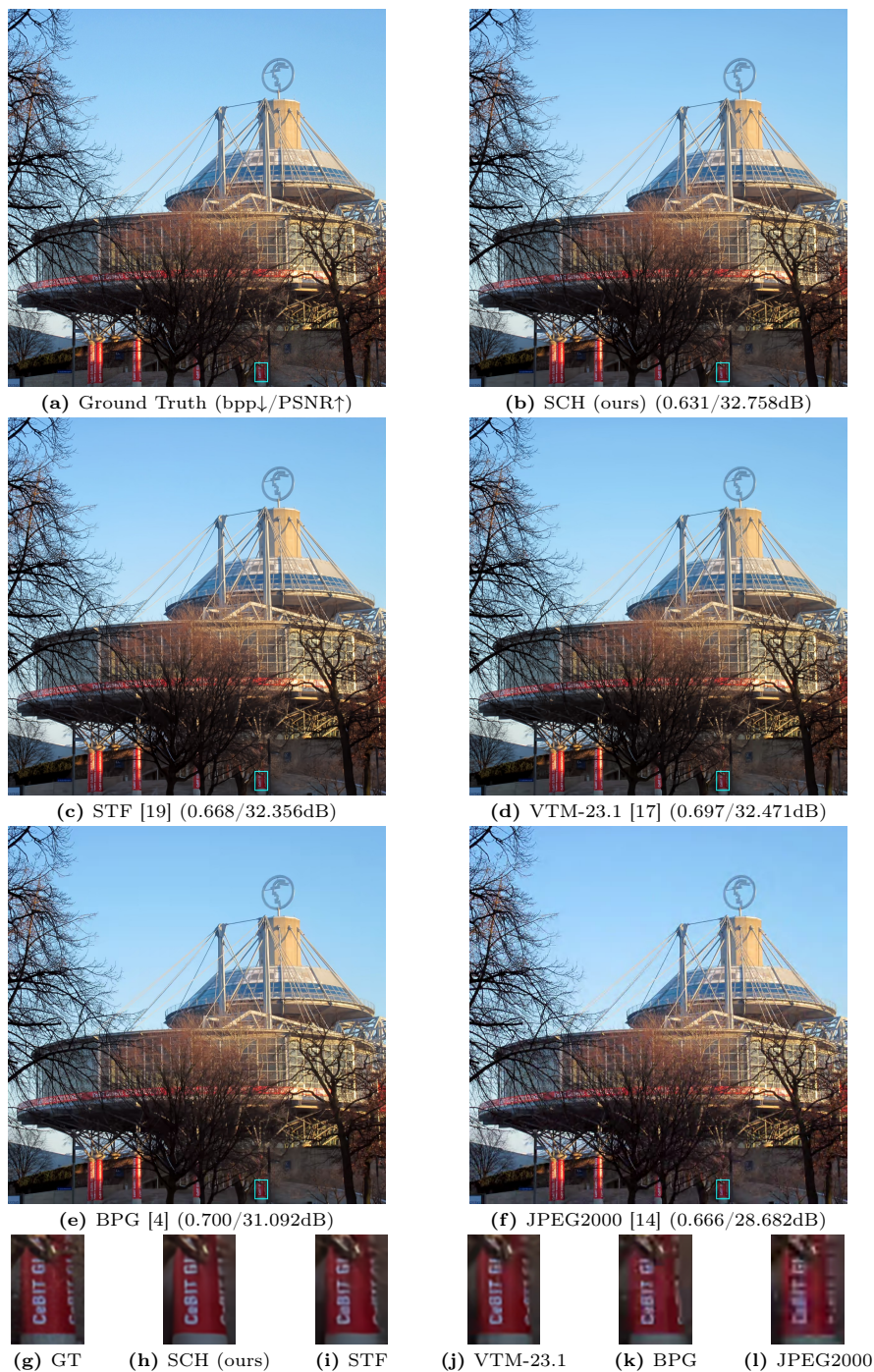


Fig. 13: Reconstructed images of *RGB_OR_1200x1200_026* from Tecnick [1].

References

1. Asuni, N., Giachetti, A.: Testimages: a large-scale archive for testing visual devices and basic image processing algorithms. In: STAG. pp. 63–70 (2014)
2. Ballé, J., Laparra, V., Simoncelli, E.P.: Density modeling of images using a generalized normalization transformation. In: 4th International Conference on Learning Representations, ICLR 2016 (2016)
3. Bégin, 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)
4. Bellard, F.: Bpg (better portable graphics) image format. <http://bellard.org/bpg/> (2014)
5. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7939–7948 (2020)
6. Ding, M., Xiao, B., Codella, N., Luo, P., Wang, J., Yuan, L.: Davit: Dual attention vision transformers. In: European Conference on Computer Vision. pp. 74–92. Springer (2022)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
8. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015)
9. Kodak, E.: Kodak lossless true color image suite (photocd pcd0992) (1993), <http://r0k.us/graphics/kodak>
10. Liu, J., Sun, H., Katto, J.: Learned image compression with mixed transformer-cnn architectures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14388–14397 (2023)
11. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
12. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems* **29** (2016)
13. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019)
14. Taubman, D.S., Marcellin, M.W.: Jpeg2000: Standard for interactive imaging. *Proceedings of the IEEE* **90**(8), 1336–1357 (2002)
15. Toderici, G., Shi, W., Timofte, R., Theis, L., Balle, J., Agustsson, E., Johnston, N., Mentzer, F.: Workshop and challenge on learned image compression (clic2020). In: CVPR (2020)
16. Toderici, G., Shi, W., Timofte, R., Theis, L., Balle, J., Agustsson, E., Johnston, N., Mentzer, F.: Workshop and challenge on learned image compression (clic2021). In: CVPR (2021)

17. Wien, M., Bross, B.: Versatile video coding—algorithms and specification. In: 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP). pp. 1–3. IEEE (2020)
18. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4794–4803 (2022)
19. Zou, R., Song, C., Zhang, Z.: The devil is in the details: Window-based attention for image compression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17492–17501 (2022)