Joint Global and Local Hierarchical Priors for Learned Image Compression
– Supplementary Material –

Jun-Hyuk Kim¹  Byeongho Heo²  Jong-Seok Lee¹
¹School of Integrated Technology, Yonsei University  ²NAVER AI Lab
{junhyuk.kim, jong-seok.lee}@yonsei.ac.kr  bh.heo@navercorp.com

A. Effectiveness of Informer

In our main paper, we evaluated our entropy model (Informer) with the encoder-decoder structure in Minnen et al. [8]. Additionally, in this supplementary material, we evaluate Informer with another encoder-decoder structure in Cheng et al. [4], which consists of residual blocks and attention modules. For the baseline entropy model [8], we use the result values reported in the CompressAI Github page¹.

As shown in Fig. A.1, Informer shows superiority across all PSNR levels on Kodak [6] with performance gaps from 3.03% up to 5.25%. From this result, we observe that our proposed Informer effectively models remaining dependencies in the quantized latent representation regardless of the encoder-decoder structure.

Figure A.1. Performance of different image compression methods. Each curve means the rate savings (%) relative to BPG [3] at different PSNR levels. Larger values mean better performance. The results of the MSE-optimized methods are averaged over Kodak [6].

B. More implementation details

We illustrate the overall diagram of the learned image compression method with Informer in Fig. B.1. In addition, architectural details of Informer are summarized in Tab. B.1.

Figure B.1. Overall diagram of the learned image compression method with the proposed entropy model, Informer. The white blocks are data tensors, the blue blocks are learned models, the red blocks are entropy coding, and the gray circles are quantization operations. Informer can be combined with any transformation part (consisting of an encoder \( f_a \) and a decoder \( f_s \)). Informer jointly optimizes Context Model (capturing dependencies in the previously decoded local context \( \hat{y}_{<i} \)), Local Hyperprior Model (capturing inter-channel dependencies at each spatial location), Global Hyperprior Model (capturing global dependencies across the whole image region), and Parameter Model (predicting the distribution parameters \( \mu \) and \( \sigma \)). By utilizing the predicted \( \mu \) and \( \sigma \) by Informer, the quantized latent representation \( \hat{y} \) is encoded into a bitstream using an entropy encoder (EC) and the bitstream is decoded by an entropy decoder (ED). In addition to \( \hat{y} \), the proposed local hyperprior \( \hat{z}_l \) and global hyperprior \( \hat{z}_g \) are also encoded using entropy coding. For this, we employ the factorized entropy model [2], where all elements of \( \hat{z}_l \) with the same channel index are assumed to follow the same distribution and all elements of \( \hat{z}_g \) are assumed to follow different distributions.

<table>
<thead>
<tr>
<th>Context Model</th>
<th>Local Hyper Encoder</th>
<th>Local Hyper Decoder</th>
<th>Global Hyper Encoder</th>
<th>Global Hyper Decoder</th>
<th>Parameter Model (1)</th>
<th>Parameter Model (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Masked: 5×5 c384 s1</td>
<td>Conv: 1×1 c48 s1</td>
<td>Conv: 1×1 c12 s1</td>
<td>MHA: c192 h4</td>
<td>Conv: 1×1 c384 s1</td>
<td>MHA: c384 h8</td>
<td>Conv: 1×1 c640 s1</td>
</tr>
<tr>
<td>Leaky ReLU</td>
<td>Leaky ReLU</td>
<td>Leaky ReLU</td>
<td>Linear: c768</td>
<td>Linear: c384 s1</td>
<td>Linear: c1536</td>
<td>Linear ReLU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GELU</td>
<td></td>
<td>GELU</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Linear: c192</td>
<td></td>
<td>Linear: c384</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Conv: 1×1 c24 s1</td>
<td></td>
<td>Conv: 1×1 c512 s1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Leaky ReLU</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table B.1. Architectural details of Informer when \( C = 192 \) and \( N = 8 \). Convolutional layers are represented by the “Conv” prefix followed by the kernel size, number of channels and stride (e.g., the first layer of the local hyper encoder uses 1×1 kernels with 48 channels and a stride of one). The “Masked” prefix means the masked convolutional layer as in [9]. The “MHA” prefix means the multi-head attention blocks as in [5], followed by the number of channels and the number of heads. The “Linear” prefix means the linear layers followed by the number of channels.
C. Rate–distortion curves

Fig. C.1 shows rate–distortion curves corresponding to Fig. 6 in the main paper. For traditional methods, we use JPEG [12], JPEG2000 [11], and BPG [3]. For learned methods, we use those of Minnen et al. [8], Lee et al. [7], and Qian et al. [10].

![Rate–distortion curves](image)

Figure C.1. Rate–distortion curves. Top: MSE-optimized models on Kodak [6]. Middle: MSE-optimized models on Tecnick [1]. Bottom: MS-SSIM-optimized models on Kodak [6].
D. More qualitative results

Figs. D.1 to D.5 show qualitative performance comparison between our method and other traditional image codecs. As in the main paper, for fair comparisons, we encode the images under compression settings for as similar bpp values as possible. Overall, our methods exhibit better qualitative performance by restoring images more clearly, not reconstructing wrong content that does not exist in the original image, and producing patterns more similar to those of the original image.

Figure D.1. Visual comparison of the decoded images by our methods and the other image codecs on “Kodim03” from Kodak [6].

Figure D.2. Visual comparison of the decoded images by our methods and the other image codecs on “Kodim15” from Kodak [6].
Figure D.3. Visual comparison of the decoded images by our methods and the other image codecs on “Kodim04” from Kodak [6].

Figure D.4. Visual comparison of the decoded images by our methods and the other image codecs on “Kodim09” from Kodak [6].

Figure D.5. Visual comparison of the decoded images by our methods and the other image codecs on “Kodim17” from Kodak [6].
References


