

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

- Supplementary Material -

Jongmin Park
KAIST

jm.park@kaist.ac.kr

Jooyoung Lee
ETRI

leejy1003@etri.re.kr

Munchurl Kim*
KAIST

mkimee@kaist.ac.kr

A. Details for optimization

To optimize our COMPASS, we use the CLIC [15] dataset which includes 1,633 high-resolution images. As mentioned in the main paper, we separately pre-train the residual compression module and the LIFF module, and further train the overall architecture for ease of convergence. We pre-train the residual compression module with a three-layer scalability (one BL and two ELs) using randomly selected scale factors, while keeping the image compression module of the BL frozen. The training images are cropped into fixed patches of 512×512 size, resulting in 19,813 patches that serve as inputs. To generate multiple downsampled versions of the input patches, we apply different downscale factors to the original input patches of 512×512 size. Specifically, we produce three downsampled versions, denoted as I^0 , I^1 , and I^2 , where I^0 has a downscale factor of 0.25, and I^1 and I^2 have downscale factors randomly selected from a uniform distribution $\mathcal{U}(0.25, 0.5)$ and $\mathcal{U}(0.5, 1)$, respectively. It should be noted that during the pre-training of the residual compression module, we substitute the LIFF module in our COMPASS architecture with a simple bicubic interpolation function for generating the residual image \hat{I}_{res}^k as

$$\hat{I}_{res}^k = I^k - B_{\uparrow}(\hat{I}^{k-1}), \quad (1)$$

where $B_{\uparrow}(\cdot)$ refers to a bicubic interpolation function used to increase the resolution of \hat{I}^{k-1} to match I^k . We use this function to prevent the training of the residual compression module from being unstable by the insufficiently trained LIFF module in the early optimization stage. We set the batch size to 8 and the initial learning rate to 5×10^{-5} without any decay. We pre-train the residual compression module for 600 epochs, using the combined RD loss function in Eq. 5 of the main paper. To pre-train the LIFF module, we crop the training images into patches of size 128×128, resulting in a total of 269,296 patches. These

patches are then randomly downsampled by the scale factors of a uniform distribution $\mathcal{U}(0.5, 1)$. We set the batch size to 64, and the initial learning rate to 1×10^{-4} , while applying a step decay strategy with a decay factor of 0.5 every 50 epochs. We pre-train the LIFF module for 200 epochs, using an L1 loss function. To jointly train the overall COMPASS architecture with the two pre-trained modules, we set the total number of epochs to 300, and the initial learning rate to 5×10^{-5} with step decay every 100 epochs using a decay factor of 0.5. Other settings, such as the batch size, the loss function, and the image downscaling procedure, are the same as those used for the pre-training of the residual compression module. For all training procedures, we utilize the Adam [8] optimizer. We also clip the maximum gradient norm to 1.0 to prevent gradient explosions and ensure a stable optimization. We use four A100 GPUs to train our COMPASS architecture. Additionally, we train four different quality versions of our COMPASS by selecting the values of λ for the combined RD loss function with $\lambda = 0.0018, 0.0035, 0.0067$ and 0.013.

B. Additional experimental results

B.1. Coding efficiency of BL and EL-1 in three-layer scalability with a fixed scale factor of 2

In addition to the coding efficiency comparison for the final EL in the main paper, we further present the coding efficiency comparison results for both BL and EL-1 in a three-layer scalability with a fixed scale factor of 2. Table 1 shows the coding efficiency performance of our COMPASS for the BL in terms of BD-rate, compared to other methods. It should be noted that we use the same frozen image compression module as Simulcast (Mean-scale [12]) and Single-layer (Mean-scale [12]), which means that the coding efficiencies of these two methods are exactly the same as that of our BL. Figure 1 shows the rate-PSNR performance curves for Table 1. Table 2 shows the coding efficiency performance comparison for the EL-1, and Figure 2 shows the

*Corresponding author.

Methods	BD-rate↓
SHVC [2]	-24.97%
Simulcast (Factorized [1])	-12.96%
Simulcast (Mean-scale [12])	-
Mei <i>et al.</i> [11] (original)	-18.48%
Mei <i>et al.</i> [11] (enhanced)	-18.31%
Single-layer (Mean-scale [12])	-

Table 1: Coding efficiency comparison for a three-layer scalable coding with a fixed scale factor of 2. BD-rate gains of our COMPASS over the various methods are measured in the BL where the negative values indicate BD-rate gains of our COMPASS.

Methods	BD-rate↓
SHVC [2]	-35.87%
Simulcast (Factorized [1])	-31.89%
Simulcast (Mean-scale [12])	-16.29%
Mei <i>et al.</i> [11] (original)	-28.20%
Mei <i>et al.</i> [11] (enhanced)	-13.52%
Single-layer (Mean-scale [12])	8.80%

Table 2: Coding efficiency comparison for a three-layer scalable coding with a fixed scale factor of 2. BD-rate gains of our COMPASS over the various methods are measured in the EL-1 where the negative values indicate BD-rate gains of our COMPASS.

corresponding rate-PSNR performance curves. As shown in both Table 1 and Table 2, our COMPASS significantly outperforms the existing spatially scalable coding methods for both the BL and EL-1, as can be seen in the results obtained for the final EL in the main paper.

B.2. Rate-PSNR performance curves for arbitrary scale factors

We plot the rate-PSNR performance curves for the results in Tables 2 and 3 of the main paper. Figures 3 through 6 show the rate-PSNR performance curves for a two-layer scalability with four different scale factors of $1.2\times$, $1.6\times$, $2.4\times$ and $2.8\times$ between BL and EL-1, while Figures 7 through 10 show the rate-PSNR performance curves for a three-layer scalability with a fixed scale factor of 2 between BL and EL-1 and four different scale factors of $2.4\times$, $2.8\times$, $3.2\times$ and $3.6\times$ between BL and EL-2.

B.3. Coding efficiency on additional datasets

We additionally evaluate our method on additional datasets with higher resolutions than the Kodak Lossless True Color Image [6] dataset to demonstrate the versatility of our COMPASS. Table 3 presents the coding efficiency comparison for three-layer scalable coding with a fixed scale factor of 2 on the large Urban100 [7] dataset. This dataset is a widely used for super-resolution tasks and

Methods	BD-rate↓
Simulcast (Mean-scale [12])	-29.09%
Mei <i>et al.</i> [11] (enhanced)	-24.77%
Single-layer (Mean-scale [12])	-2.30%

Table 3: Coding efficiency comparison for a three-layer scalable coding with a fixed scale factor of 2 on the Urban100 [7] dataset. BD-rate gains of our COMPASS over the various methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS.

Methods	BD-rate↓
Simulcast (Mean-scale [12])	-26.50%
Mei <i>et al.</i> [11] (enhanced)	-23.10%
Single-layer (Mean-scale [12])	4.01%

Table 4: Coding efficiency comparison for a three-layer scalable coding with a fixed scale factor of 2 on the CLIC 2021 validation [15] dataset. BD-rate gains of our COMPASS over the various methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS.

contains multiple scales. Impressively, our COMPASS even outperforms the single-layer coding with -2.30% BD-rate gain, and shows significant superiority against Simulcast (Mean-scale [12]) and Mei *et al.* [11]’s enhanced version. Table 4 presents the coding efficiency comparison for three-layer coding with a fixed scale factor of 2 on the CLIC 2021 validation [15] dataset, which contains 2K resolution images. In this table, our COMPASS exhibits higher performance for higher resolution images when compared to its performance using the Kodak Lossless True Color Image [6] dataset. This emphasizes the enhanced capabilities of our approach for handling higher resolution scenarios.

B.4. Coding efficiency with out-of-distribution scale factors

Table 5 presents the coding efficiency comparison for three-layer scalable coding with larger scale factors than $4.0\times$ which are out of training distribution. For this table, we set the scale factor of the final EL (EL-2) relative to the BL to $4.5\times$, $5.0\times$, $5.5\times$, and $6.0\times$, respectively. We set the scale factor of the EL-1 relative to the BL to 2, equally.

B.5. Coding efficiency with more combinations of arbitrary scale factors

Table 6 through 9 present the coding efficiency comparison in terms of BD-rate for three-layer scalable coding (one BL and two ELs) with more combinations of arbitrary scale factors. For each table, we set the scale factor of the EL-1 relative to the BL to $1.2\times$, $1.4\times$, $1.6\times$ and $1.8\times$, respectively. Additionally, we set the scale factor of the final EL to $2.4\times$, $2.8\times$, $3.2\times$, $3.6\times$ and $4.0\times$ with respect to the BL.

Methods	Scale Factors (vs. BL)			
	4.5×	5.0×	5.5×	6.0×
Simulcast (Mean-scale [12])	-21.24%	-19.94%	-18.52%	-17.85%
Single-layer (Mean-scale [12])	6.77%	7.77%	8.99%	8.76%

Table 5: Coding efficiency comparison for a three-layer scalable coding with out-of-distribution scale factors. BD-rate gains of our COMPASS over the methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS. We set the scale factor of the EL-1 relative to the BL to 2, equally.

B.6. Additional ablation study for the LIFF module

We compare our method to the ablated model which uses LIIF [3] rather than our LIFF module. The experiment is conducted for the three-layer scalability with a fixed scale factor of 2. Our method achieves a 26% reduction in FLOPs (811G) and a 9.5% reduction in average prediction time (0.019 sec.) compared to the ablated model (1.1T FLOPs and 0.021 sec.), achieving -0.64% BD-rate gain.

C. Additional visual results

C.1. Visual results for multi-layer scalable coding with more than three layers

Figures 11 and 12 show the visual results of our COMPASS for multi-layer scalable coding greater than three layers. We consider the total number of layers with three, four, five, six and nine, and compare the PSNR values and accumulated bits in each layer. As shown in Figures 11 and 12, our COMPASS shows faithful reconstructions over various layers.

C.2. Additional visual results for three-layer scalable coding

Figures 13 through 19 show the visual results of our COMPASS in comparison with SHVC [2], the simulcast coding, and Mei *et al.* [11] for a three-layer scalable coding with a fixed scale factor of 2 between adjacent layers. The images shown in Figure 13 through 19 are the largest-sized reconstructions obtained from the final EL. As shown in Figures 13 through 19, it is noted that our COMPASS can better reconstruct high-frequency details such as texture, edges and small image structures at the usage of similar bit amounts.

D. Limitations and future work

Due to the hierarchical architecture, the computational complexity of our COMPASS eventually becomes higher as the number of layers increases, which may yield a challenge

for real-time applications and devices with limited computational resources. It should be noted that this complexity issue is not the issue that only our COMPASS is facing, but the issue given to the research domain of scalable coding where almost all studies are based on the hierarchical coding architecture, so further study is required for a more consistent method in terms of complexity increase.

In addition, although our COMPASS successfully achieves more powerful spatial scalability in terms of the scale factors and the number of layers, a single COMPASS model (including the compression modules in the base layer and enhancement layer) is optimized for its designated target compress quality. Therefore, separately trained multiple COMPASS models are necessary to deal with the various target compression qualities, which leads to an increase in the number of total model parameters. Meanwhile, a few studies on variable-rate NN-based image compression [4, 5, 13, 14, 10, 9] show promising results, so we expect we can integrate those variable-rate techniques into our COMPASS. In the future, we'll further enhance our COMPASS based on the priors of the NN-based image compression research area from the perspective of practicality and flexibility, as well as improve its coding efficiency.

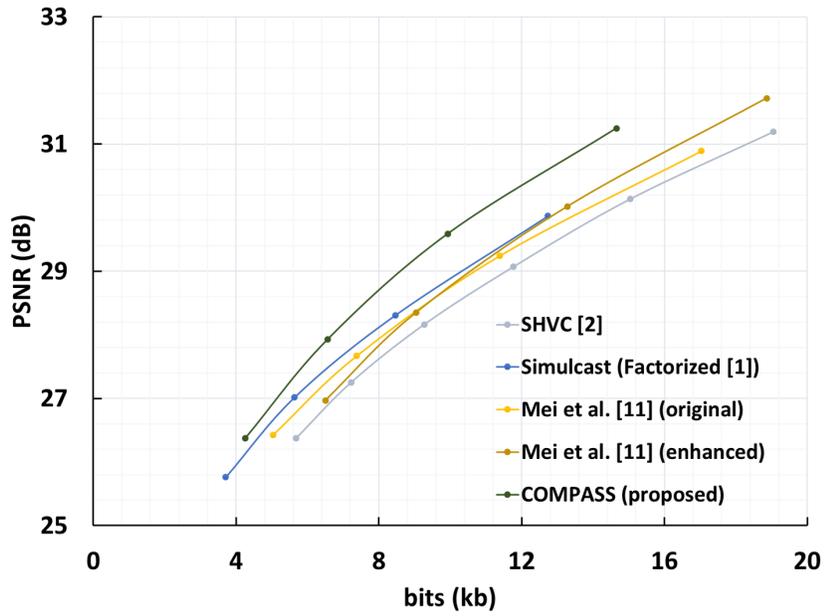


Figure 1: The rate-PSNR performance curves of the BLs for SHVC [2], the simulcast coding, Mei *et al.* [11], and our COMPASS. The experiment is conducted with a three-layer scalability (one BL and two ELs (EL-1: 2.0 \times and EL-2: 4.0 \times)). The ‘bits’ indicates the bits corresponding to the BL.

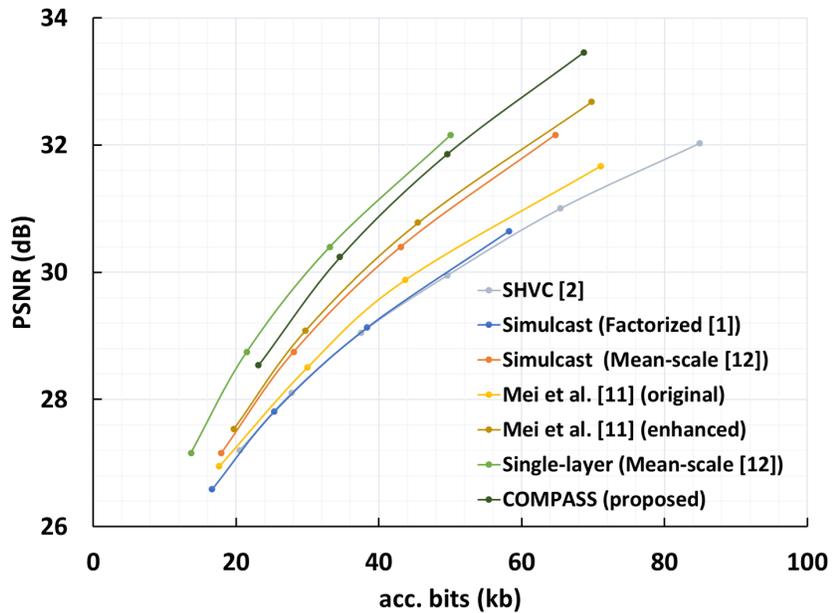


Figure 2: The rate-PSNR performance curves of the intermediate ELs for SHVC [2], the simulcast coding, Mei *et al.* [11], the single-layer coding, and our COMPASS. The experiment is conducted with a three-layer scalability (one BL and two ELs (EL-1: 2.0 \times and EL-2: 4.0 \times)). The ‘acc. bits’ indicates the accumulated bits up to the EL-1.

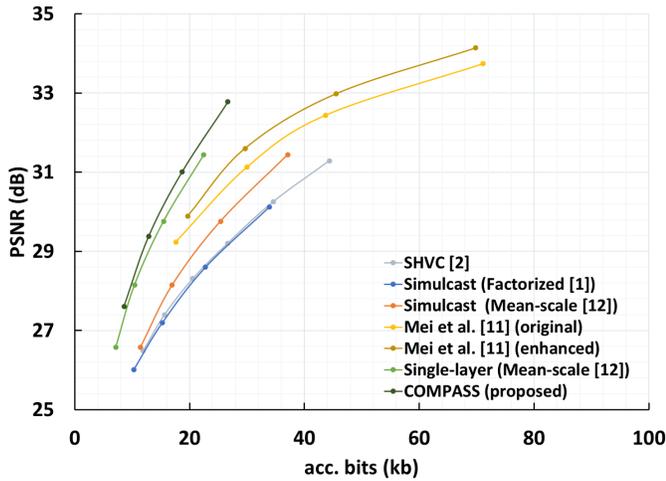


Figure 3: The rate-PSNR performance curves of the final ELs for SHVC [2], the simulcast coding, Mei *et al.* [11], the single-layer coding, and our COMPASS. The experiment is conducted with a two-layer scalability (one BL and one EL (EL-1: 1.2 \times)). The ‘acc. bits’ indicates the accumulated bits up to the final EL.

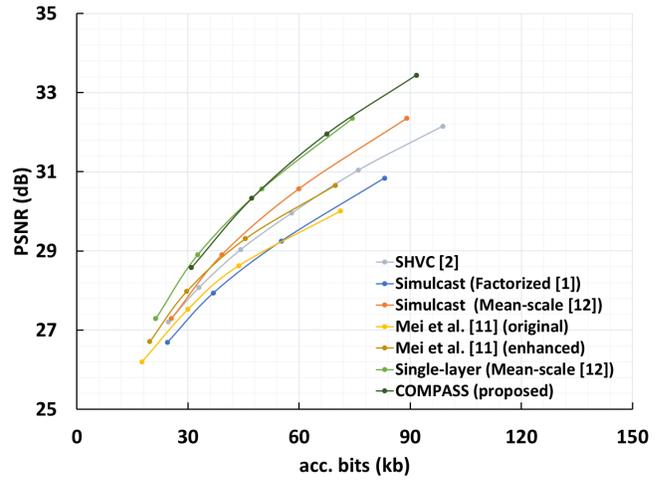


Figure 5: The rate-PSNR performance curves of the final ELs for SHVC [2], the simulcast coding, Mei *et al.* [11], the single-layer coding, and our COMPASS. The experiment is conducted with a two-layer scalability (one BL and one EL (EL-1: 2.4 \times)). The ‘acc. bits’ indicates the accumulated bits up to the final EL.

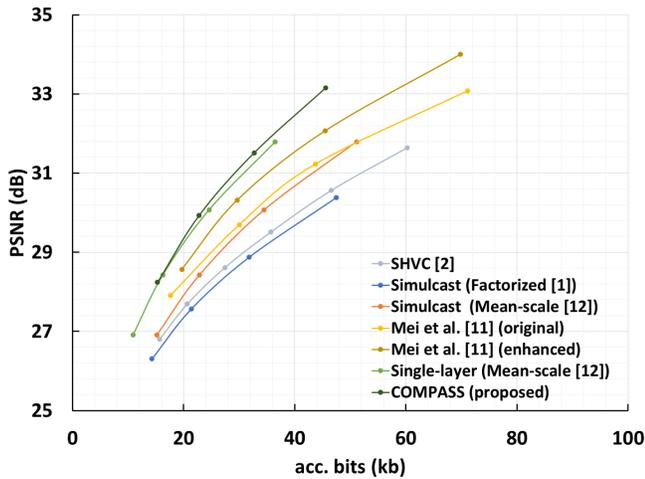


Figure 4: The rate-PSNR performance curves of the final ELs for SHVC [2], the simulcast coding, Mei *et al.* [11], the single-layer coding, and our COMPASS. The experiment is conducted with a two-layer scalability (one BL and one EL (EL-1: 1.6 \times)). The ‘acc. bits’ indicates the accumulated bits up to the final EL.

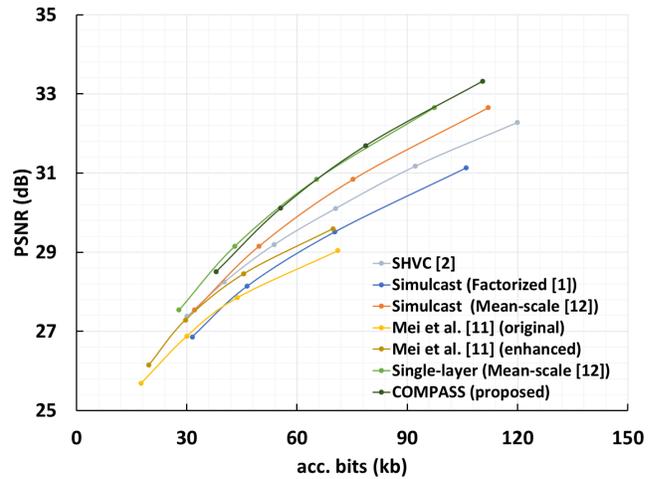


Figure 6: The rate-PSNR performance curves of the final ELs for SHVC [2], the simulcast coding, Mei *et al.* [11], the single-layer coding, and our COMPASS. The experiment is conducted with a two-layer scalability (one BL and one EL (EL-1: 2.8 \times)). The ‘acc. bits’ indicates the accumulated bits up to the final EL.

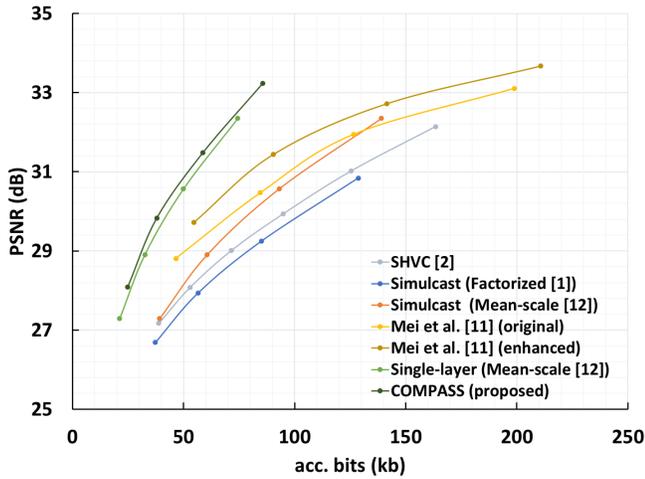


Figure 7: The rate-PSNR performance curves of the final ELs for SHVC [2], the simulcast coding, Mei *et al.* [11], the single-layer coding, and our COMPASS. The experiment is conducted with a three-layer scalability (one BL and two ELs (EL-1: 2.0 \times and EL-2: 2.4 \times)). The ‘acc. bits’ indicates the accumulated bits up to the final EL.

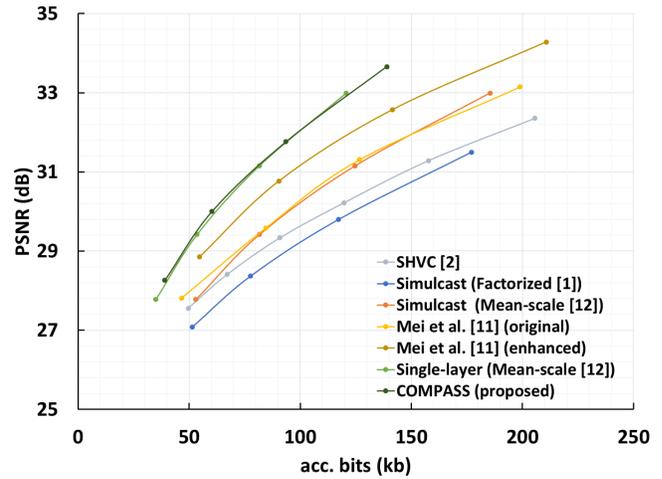


Figure 9: The rate-PSNR performance curves of the final ELs for SHVC [2], the simulcast coding, Mei *et al.* [11], the single-layer coding, and our COMPASS. The experiment is conducted with a three-layer scalability (one BL and two ELs (EL-1: 2.0 \times and EL-2: 3.2 \times)). The ‘acc. bits’ indicates the accumulated bits up to the final EL.

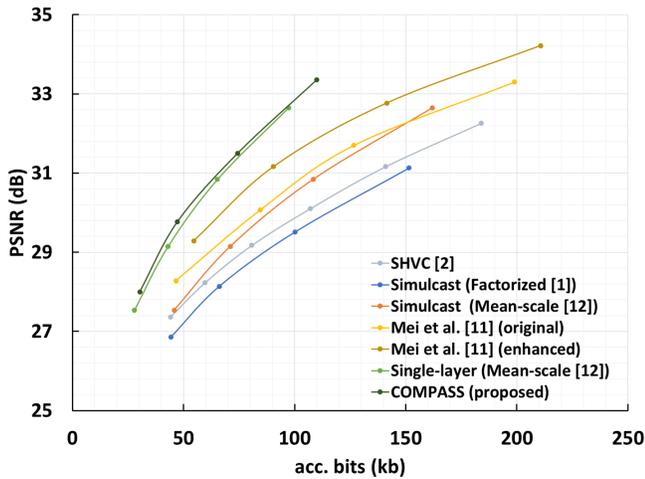


Figure 8: The rate-PSNR performance curves of the final ELs for SHVC [2], the simulcast coding, Mei *et al.* [11], the single-layer coding, and our COMPASS. The experiment is conducted with a three-layer scalability (one BL and two ELs (EL-1: 2.0 \times and EL-2: 2.8 \times)). The ‘acc. bits’ indicates the accumulated bits up to the final EL.

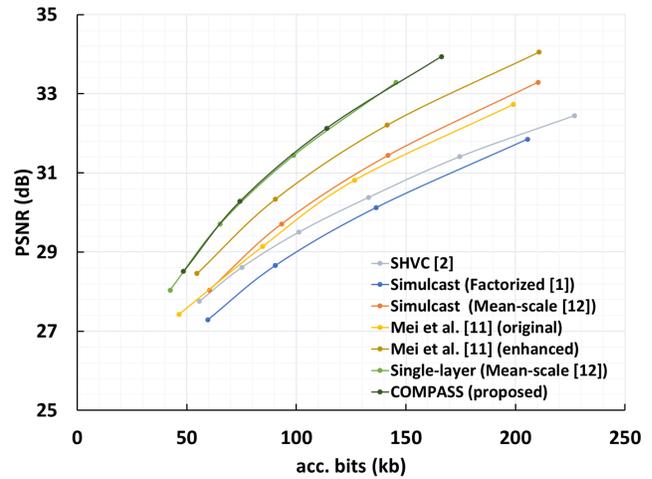


Figure 10: The rate-PSNR performance curves of the final ELs for SHVC [2], the simulcast coding, Mei *et al.* [11], the single-layer coding, and our COMPASS. The experiment is conducted with a three-layer scalability (one BL and two ELs (EL-1: 2.0 \times and EL-2: 3.6 \times)). The ‘acc. bits’ indicates the accumulated bits up to the final EL.

Methods	Scale Factors (vs. BL)				
	2.4×	2.8×	3.2×	3.6×	4.0×
SHVC [2]	-41.91%	-36.68%	-33.47%	-30.79%	-21.63%
Simulcast (Factorized [1])	-48.75%	-44.69%	-41.91%	-39.42%	-34.01%
Simulcast (Mean-scale [12])	-34.75%	-29.16%	-24.11%	-20.03%	-12.23%
Mei <i>et al.</i> [11] (original)	-44.56%	-37.14%	-33.86%	-32.63%	-31.03%
Mei <i>et al.</i> [11] (enhanced)	-35.12%	-25.14%	-19.13%	-15.88%	-12.83%
Single-layer (Mean-scale [12])	-1.32%	-1.49%	-0.34%	0.64%	6.47%

Table 6: Coding efficiency comparison for a three-layer scalable coding with arbitrary scale factors. BD-rate gains of our COMPASS over the various methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS. We set the scale factor of the EL-1 relative to the BL to 1.2×, equally.

Methods	Scale Factors (vs. BL)				
	2.4×	2.8×	3.2×	3.6×	4.0×
SHVC [2]	-45.51%	-39.90%	-36.72%	-33.99%	-24.78%
Simulcast (Factorized [1])	-51.88%	-47.31%	-44.40%	-41.86%	-36.15%
Simulcast (Mean-scale [12])	-38.69%	-32.53%	-27.45%	-23.35%	-15.44%
Mei <i>et al.</i> [11] (original)	-44.81%	-37.27%	-34.14%	-33.16%	-31.42%
Mei <i>et al.</i> [11] (enhanced)	-35.29%	-25.26%	-19.59%	-16.58%	-13.43%
Single-layer (Mean-scale [12])	-1.88%	-1.67%	-0.83%	-0.14%	5.68%

Table 7: Coding efficiency comparison for a three-layer scalable coding with arbitrary scale factors. BD-rate gains of our COMPASS over the various methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS. We set the scale factor of the EL-1 relative to the BL to 1.4×, equally.

Methods	Scale Factors (vs. BL)				
	2.4×	2.8×	3.2×	3.6×	4.0×
SHVC [2]	-49.05%	-43.39%	-39.80%	-36.93%	-27.38%
Simulcast (Factorized [1])	-55.84%	-50.56%	-47.26%	-44.41%	-38.96%
Simulcast (Mean-scale [12])	-42.84%	-36.22%	-30.74%	-26.43%	-18.61%
Mei <i>et al.</i> [11] (original)	-45.16%	-37.73%	-34.48%	-33.52%	-32.02%
Mei <i>et al.</i> [11] (enhanced)	-35.41%	-25.78%	-19.99%	-17.03%	-14.10%
Single-layer (Mean-scale [12])	-3.13%	-2.44%	-1.33%	-0.66%	4.90%

Table 8: Coding efficiency comparison for a three-layer scalable coding with arbitrary scale factors. BD-rate gains of our COMPASS over the various methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS. We set the scale factor of the EL-1 relative to the BL to 1.6×, equally.

Methods	Scale Factors (vs. BL)				
	2.4×	2.8×	3.2×	3.6×	4.0×
SHVC [2]	-53.17%	-46.56%	-42.78%	-39.72%	-29.96%
Simulcast (Factorized [1])	-59.49%	-53.63%	-49.64%	-46.51%	-41.06%
Simulcast (Mean-scale [12])	-47.71%	-40.15%	-34.24%	-29.65%	-21.82%
Mei <i>et al.</i> [11] (original)	-46.28%	-38.15%	-34.64%	-33.69%	-32.21%
Mei <i>et al.</i> [11] (enhanced)	-36.63%	-25.81%	-20.21%	-17.24%	-14.39%
Single-layer (Mean-scale [12])	-5.34%	-3.18%	-1.65%	-0.90%	4.53%

Table 9: Coding efficiency comparison for a three-layer scalable coding with arbitrary scale factors. BD-rate gains of our COMPASS over the various methods are measured in the final EL where the negative values indicate BD-rate gains of our COMPASS. We set the scale factor of the EL-1 relative to the BL to 1.8×, equally.

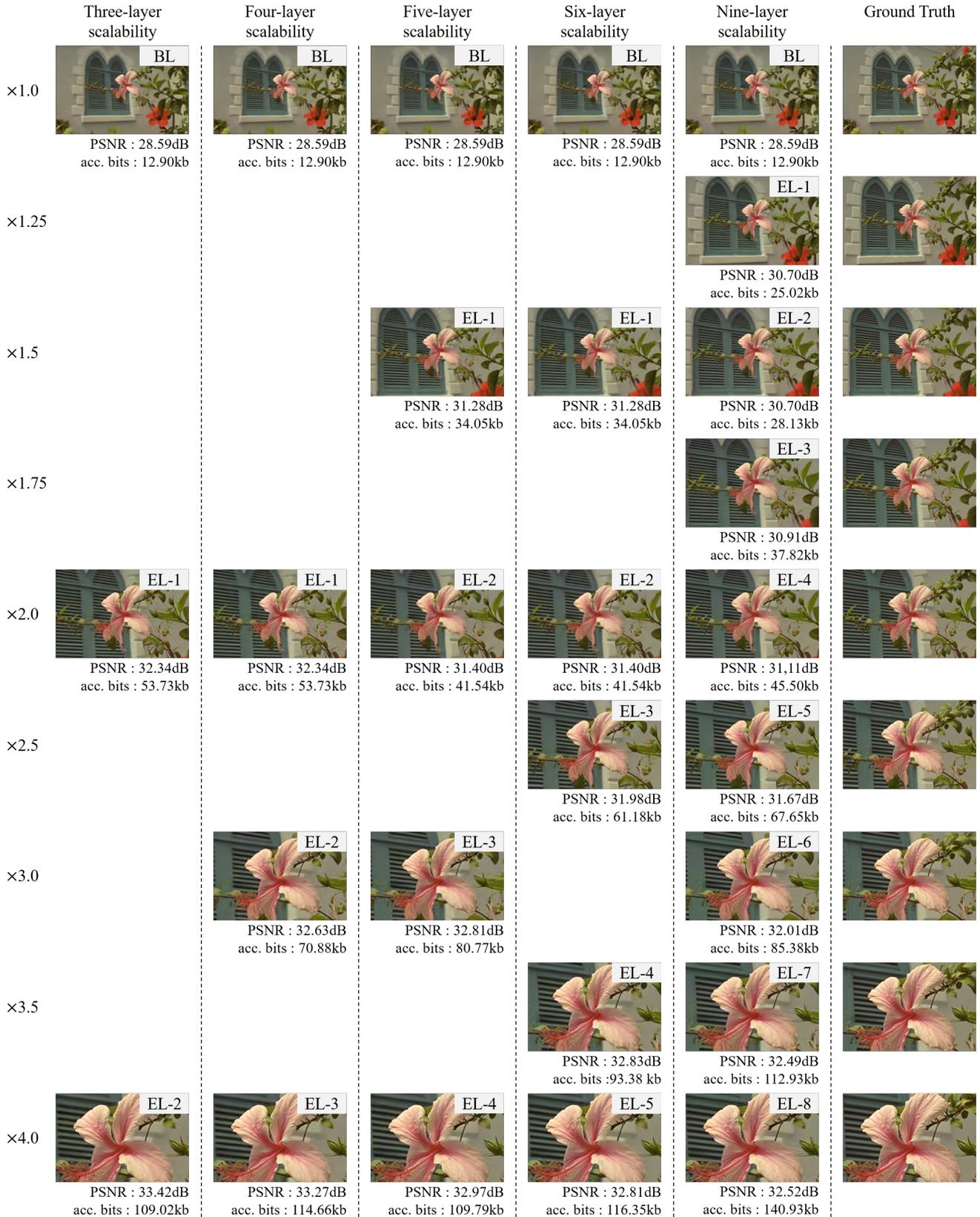


Figure 11: Visual results for multi-layer scalable coding for *kodim07.png* image in Kodak Lossless True Color Image dataset [6] (best viewed in digital format). The ‘acc. bits’ indicates the accumulated bits up to the corresponding layer.

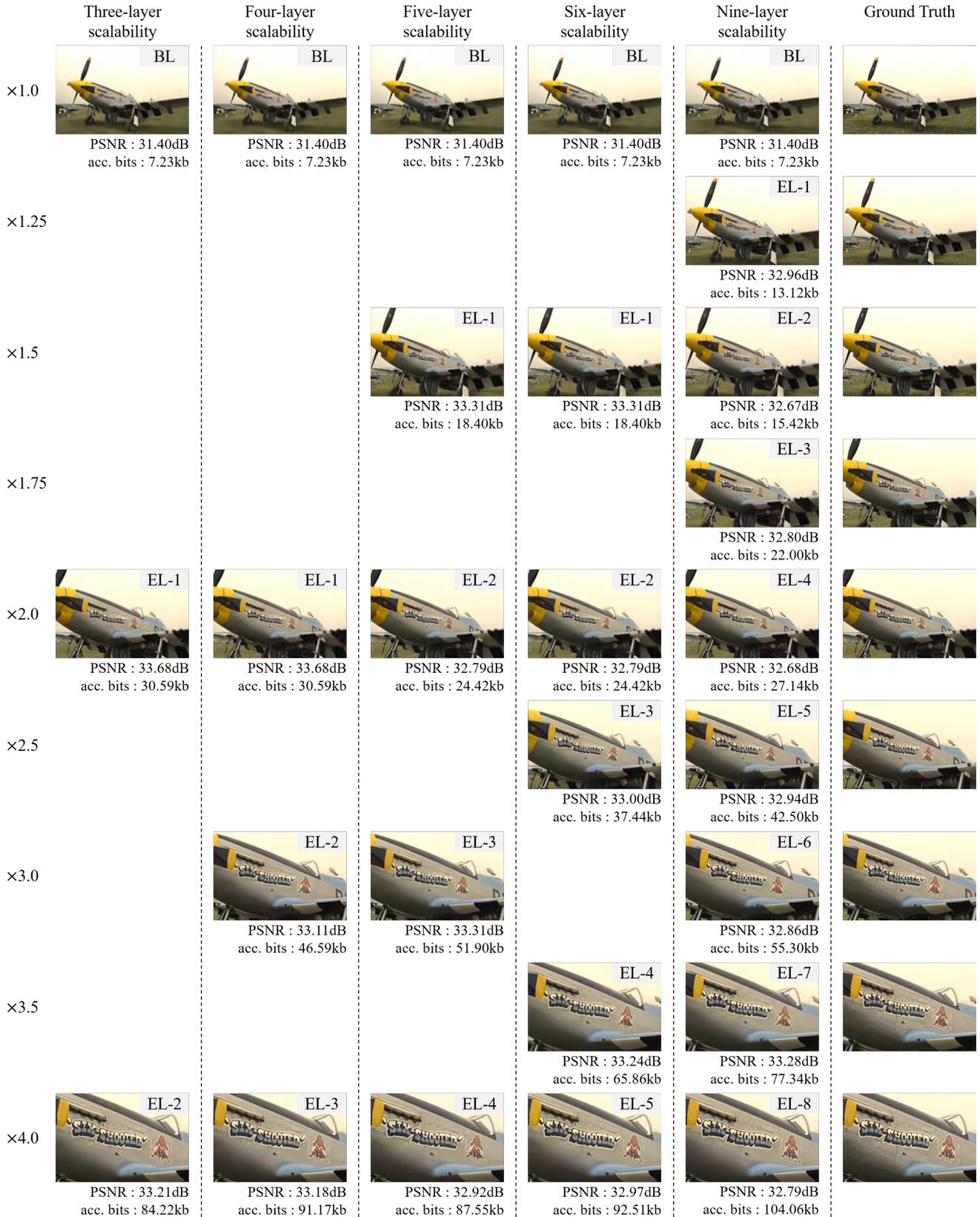


Figure 12: Visual results for multi-layer scalable coding for *kodim20.png* image in Kodak Lossless True Color Image dataset [6] (best viewed in digital format). The ‘acc. bits’ indicates the accumulated bits up to the corresponding layer.

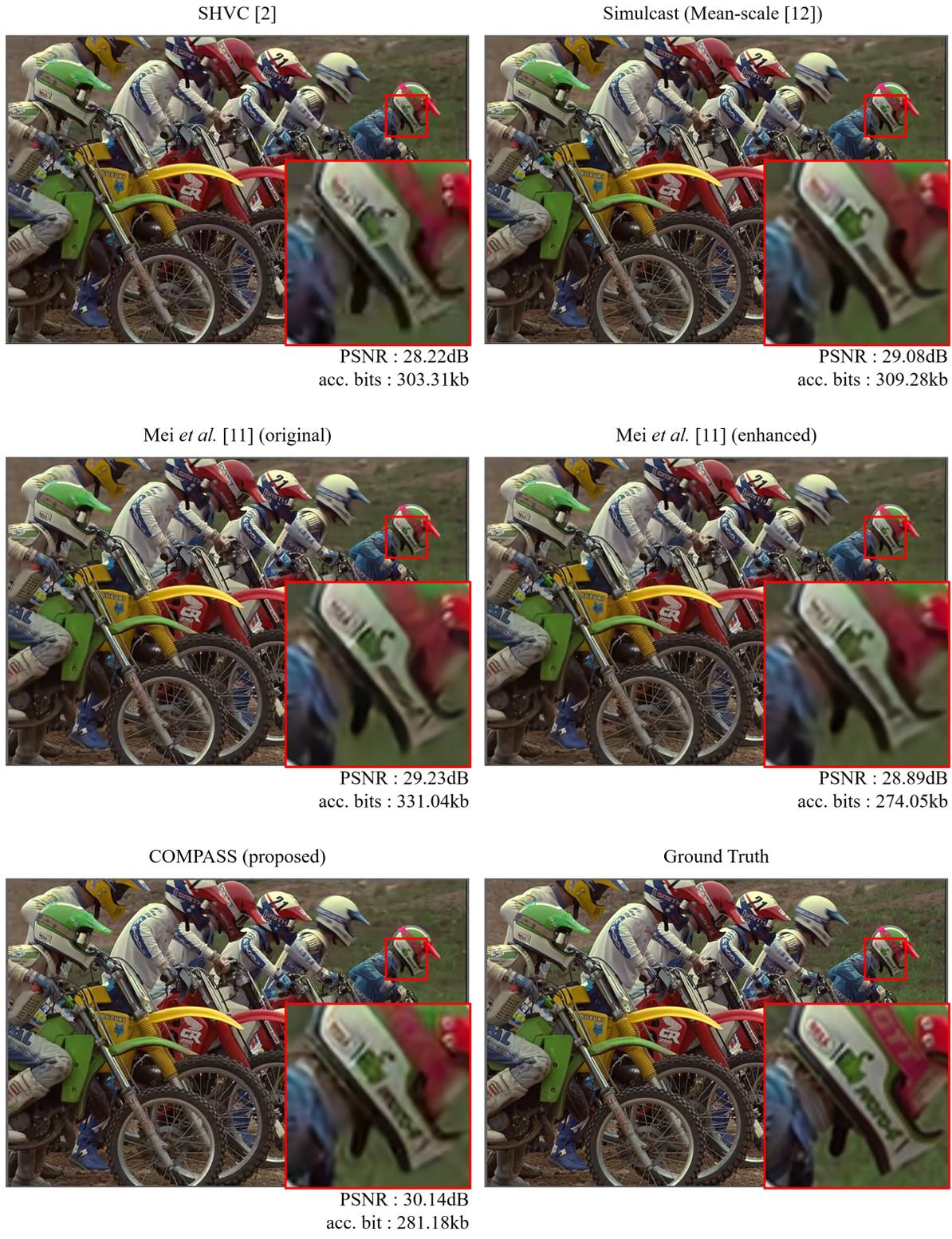


Figure 13: Visual comparison results for *kodim05.png* image in Kodak Lossless True Color Image dataset [6] (best viewed in digital format). The ‘acc. bits’ indicates the accumulated bits up to the final EL. We match the accumulated bits among the compared methods as much as possible.

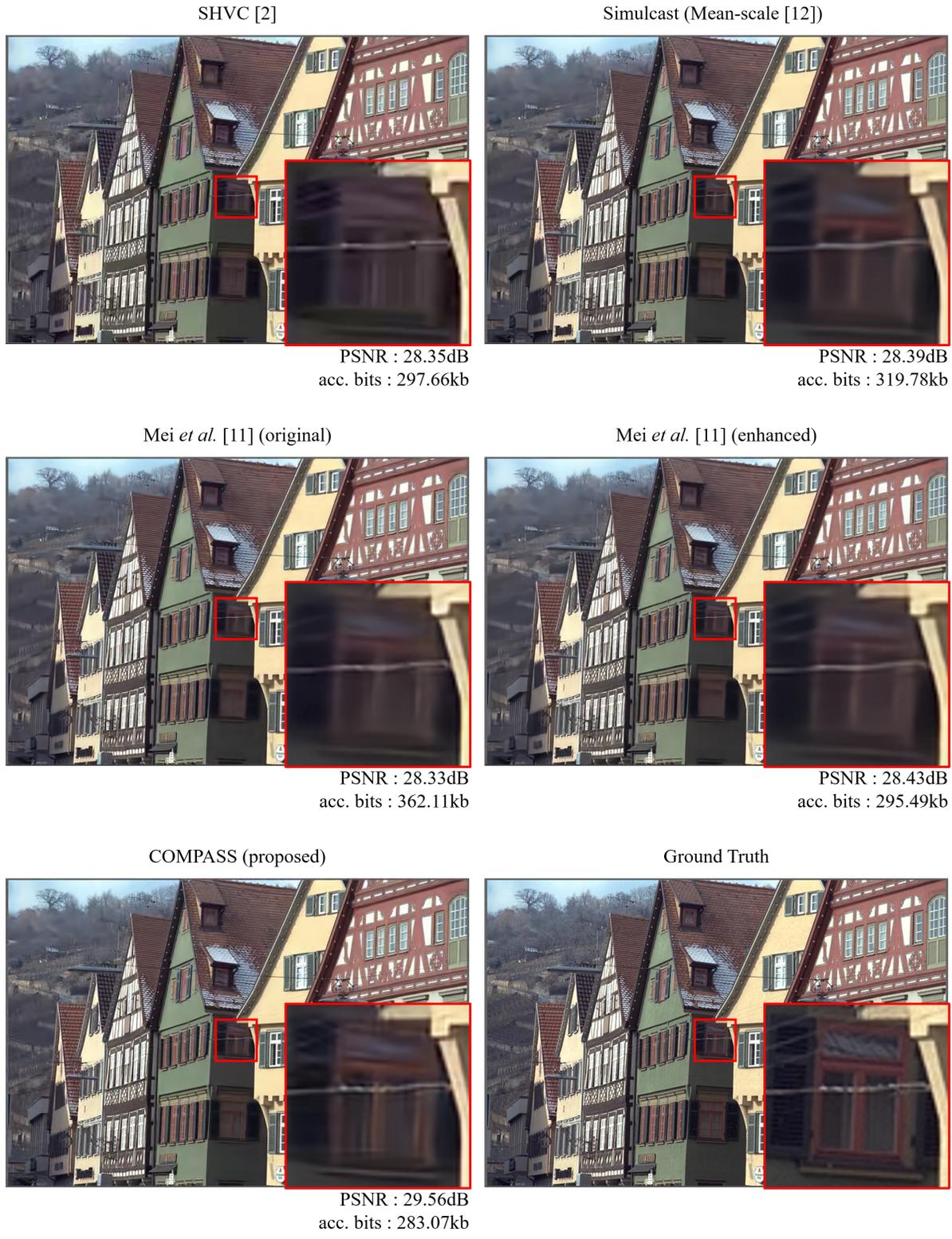


Figure 14: Visual comparison results for *kodim08.png* image in Kodak Lossless True Color Image dataset [6] (best viewed in digital format). The ‘acc. bits’ indicates the accumulated bits up to the final EL. We match the accumulated bits among the compared methods as much as possible.

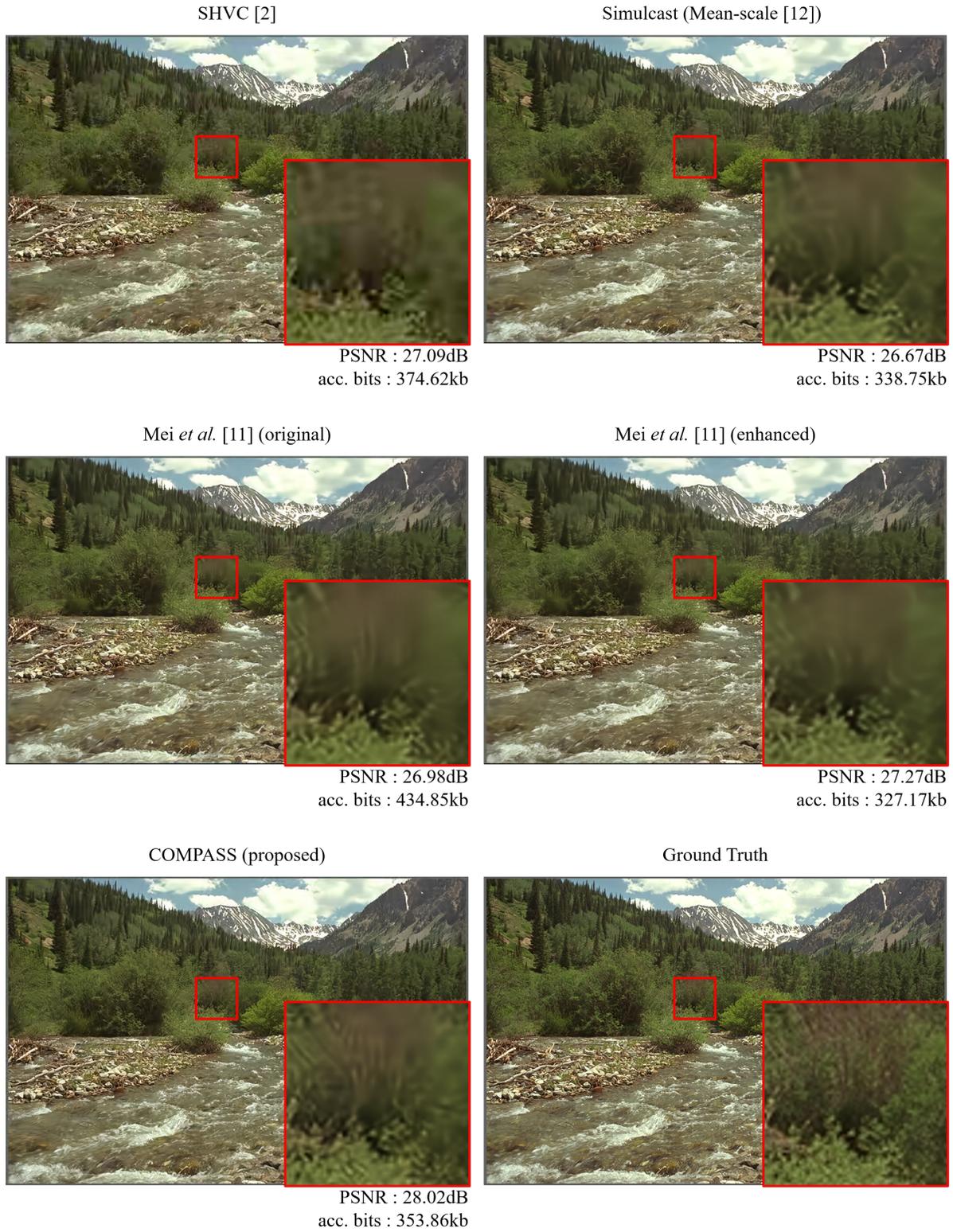


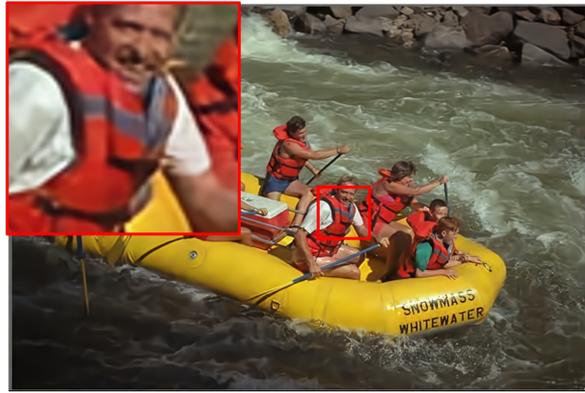
Figure 15: Visual comparison results for *kodim13.png* image in Kodak Lossless True Color Image dataset [6] (best viewed in digital format). The ‘acc. bits’ indicates the accumulated bits up to the final EL. We match the accumulated bits among the compared methods as much as possible.

SHVC [2]



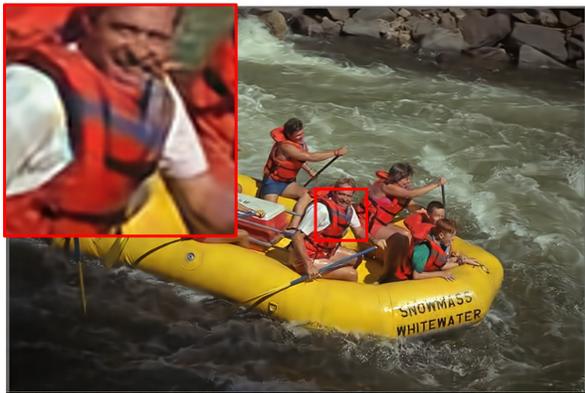
PSNR : 28.60dB
acc. bits : 205.98kb

Simulcast (Mean-scale [12])



PSNR : 29.54dB
acc. bits : 213.50kb

Mei *et al.* [11] (original)



PSNR : 29.62dB
acc. bits : 242.62kb

Mei *et al.* [11] (enhanced)



PSNR : 29.19dB
acc. bits : 180.03kb

COMPASS (proposed)



PSNR : 30.34dB
acc. bits : 199.71kb

Ground Truth



Figure 16: Visual comparison results for *kodim14.png* image in Kodak Lossless True Color Image dataset [6] (best viewed in digital format). The ‘acc. bits’ indicates the accumulated bits up to the final EL. We match the accumulated bits among the compared methods as much as possible.

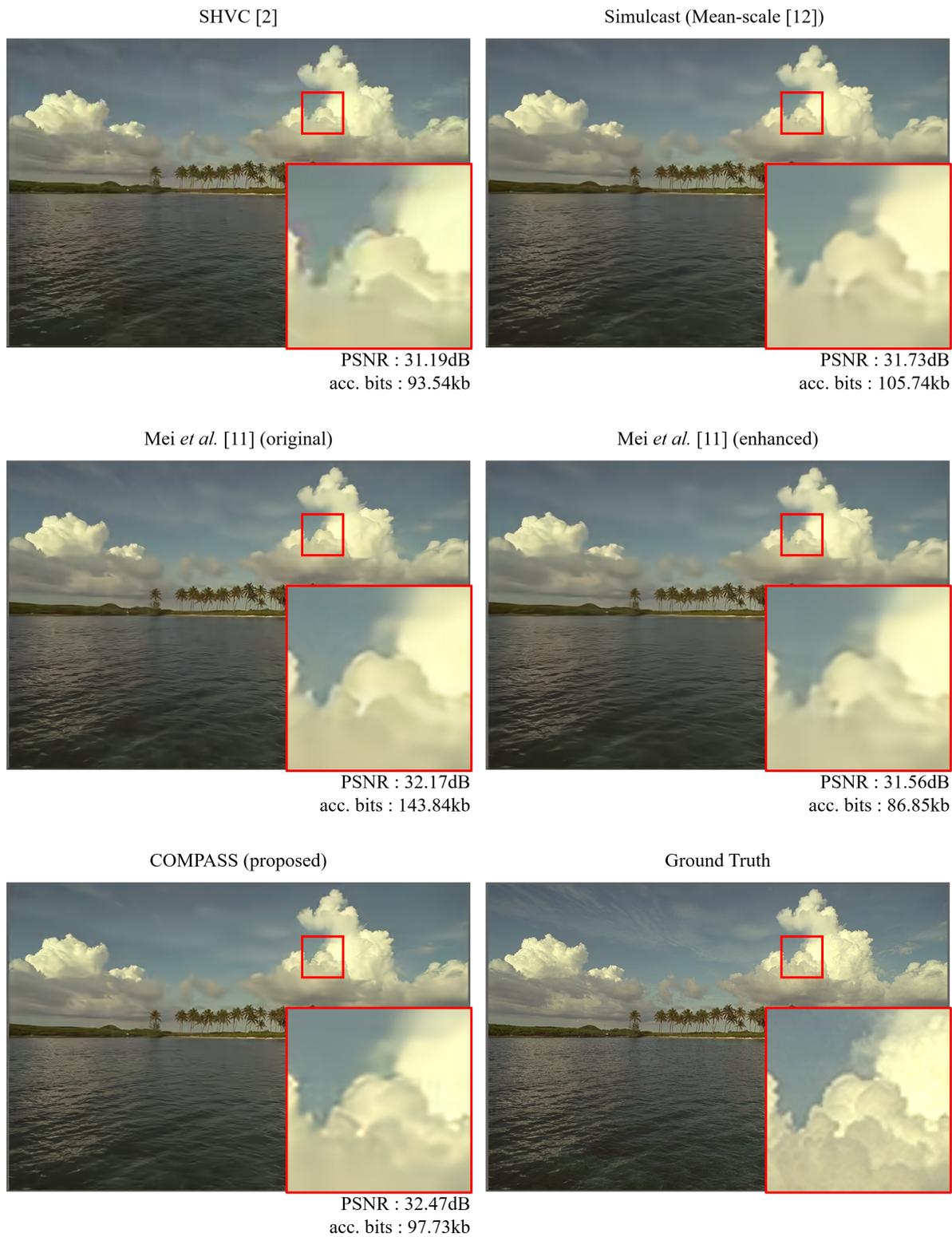


Figure 17: Visual comparison results for *kodim16.png* image in Kodak Lossless True Color Image dataset [6] (best viewed in digital format). The ‘acc. bits’ indicates the accumulated bits up to the final EL. We match the accumulated bits among the compared methods as much as possible.

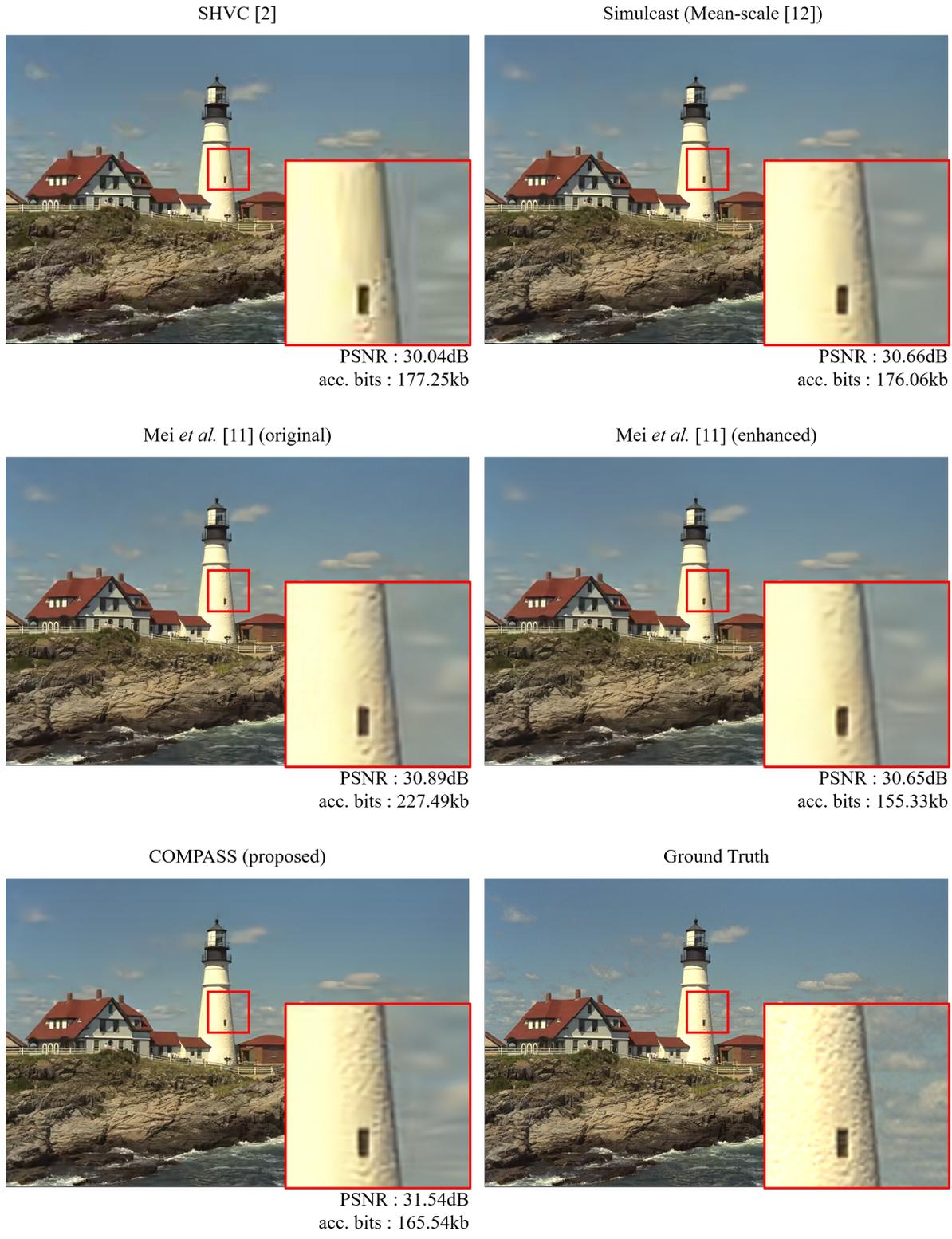


Figure 18: Visual comparison results for *kodim21.png* image in Kodak Lossless True Color Image dataset [6] (best viewed in digital format). The ‘acc. bits’ indicates the accumulated bits up to the final EL. We match the accumulated bits among the compared methods as much as possible.



Figure 19: Visual comparison results for *kodim24.png* image in Kodak Lossless True Color Image dataset [6] (best viewed in digital format). The ‘acc. bits’ indicates the accumulated bits up to the final EL. We match the accumulated bits among the compared methods as much as possible.

References

- [1] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.
- [2] Jill M Boyce, Yan Ye, Jianle Chen, and Adarsh K Ramasubramanian. Overview of shvc: Scalable extensions of the high efficiency video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(1):20–34, 2015.
- [3] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021.
- [4] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Variable rate deep image compression with a conditional autoencoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3146–3154, 2019.
- [5] Ze Cui, Jing Wang, Bo Bai, Tiansheng Guo, and Yihui Feng. G-VAE: A continuously variable rate deep image compression framework, 2020.
- [6] Rich Franzen. Kodak lossless true color image suite. *source: <http://r0k.us/graphics/kodak>*, 4(2), 1999.
- [7] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015.
- [8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [9] Jooyoung Lee, Seyoon Jeong, and Munchurl Kim. Selective compression learning of latent representations for variable-rate image compression. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [10] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. An end-to-end learning framework for video compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3292–3308, 2021.
- [11] Yixin Mei, Li Li, Zhu Li, and Fan Li. Learning-based scalable image compression with latent-feature reuse and prediction. *IEEE Transactions on Multimedia*, 2021.
- [12] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018.
- [13] Oren Rippel, Alexander G Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir Bourdev. Elf-vc: Efficient learned flexible-rate video coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14479–14488, 2021.
- [14] Myungseo Song, Jinyoung Choi, and Bohyung Han. Variable-rate deep image compression through spatially-adaptive feature transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2380–2389, 2021.
- [15] George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2018), 2018.