Supplementary Material - Controlling Rate, Distortion, and Realism: Towards a Single Comprehensive Neural Image Compression Model

A. Detailed Discriminator Architecture

We show the detailed architectures of each discriminator design in Fig 1. We apply leaky ReLU after each convolution layer except in the final layer.

B. Algorithm of HRRGAN

Algorithm 1 shows the procedure for calculating the HRRGAN loss function. In the algorithm, NIC, D, and sg indicate the NIC model, discriminator, and stop-gradient operation, respectively. As shown in Algorithm 1, we use the original image $\boldsymbol{x}$ to calculate p_r when $q = Q-1$ because the NIC model cannot reconstruct an image with $q = Q$.

C. Model Size

Table 1 shows the number of parameters in our encoder and generator. As described in Sec 3.1, we adopt the encoder and generator in ELIC [4] as a base architecture and incorporate Interpolation Channel Attention (ICA) layers [6] and β-conditioning [2] to adjust rate and distortion-realism trade-off, respectively. ICA layers are used in the encoder and generator, while β-conditioning is used only in the generator. As shown in Table 1, using these two modules results in an approximate parameter increase of $2.7M$. It demonstrates that our approach significantly saves model storage costs compared to employing separate NIC models optimized for distinct rates and distortion-realism balances.

	ICA	β-cond	Encoder	Generator
Base			7.34M	10.72M
Ours w/o MR	✓		7.36M (+0.019M)	10.74M (+0.024M)
Ours full	✓	✓	7.36M (+0.019M)	13.38M (+2.66M)

Table 1. Parameter counts for the encoder and generator across various configurations. “ICA” stands for interpolation channel attention [6], while “β-cond” refers to β-conditioning [2]. *Base* represents the encoder and decoder (generator) used in ELIC [4]. The numbers in parentheses represent the increase in parameters compared to *Base*.

D. Additional Results

D.1. Results on different realism weights

We show the quantitative results on different realism weights $\beta = \{0, 1.28, 2.56, 3.84, 5.12\}$ in Fig 2. These results illustrate that different β results in different balances between distortion and realism. Specifically, smaller β achieves high PSNR, indicating higher pixel-level fidelity. On the other hand, larger β yields lower FID, indicating high realism. Although $\beta = 3.84$ consistently surpasses $\beta = 5.12$ in terms of PSNR, the difference in FID between $\beta = 3.84$ and $\beta = 5.12$ is marginal. Based on this observation, we selected $\beta = 3.84$ as our high-realism mode in our main experiments.

D.2. LPIPS evaluation on CLIC2020 dataset

Fig 3 shows the results of LPIPS [8] on CLIC2020 dataset. Our high-realism mode ($\beta = 3.84$) matches the performance of HiFiC [5] (single-rate model) and outperforms DIRAC [3] (variable-rate model).

D.3. Detailed quantitative results

To facilitate direct comparisons in future studies, we have compiled the quantitative results into Table 2.

D.4. Additional Qualitative Results

We show additional qualitative results in Fig 4,5,6. Since Kodak reconstructions of Multi-Realism [2] are not publically available, we compare our reconstructions with only HiFiC [5]. Overall, our reconstructions contain fewer artifacts than HiFiC [5] (e.g., the top figure in Fig 4), and the visual quality of our method is competitive with Multi-Realism [2].

References

[1] Kodak photodc dataset, 1991. https://r0k.us/graphics/kodak/. 2

[2] Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer. Multi-realism image compression with a conditional generator. In *Proceedings of the IEEE/CVF Confer-*

Table 2. Main results of our model on CLIC2020 [7] and Kodak [1]. Note that these results can be obtained with only a single model.

CLIC							Kodak				
	$\beta = 0.0$ (Low distortion)			$\beta = 3.84$ (High reality)				$\beta = 0.0$		$\beta = 3.84$	
bpp	PSNR	FID	LPIPS	PSNR	FID	LPIPS	bpp	PSNR	LPIPS	PSNR	LPIPS
0.080	30.846	41.496	0.245	30.128	5.714	0.084	0.109	28.132	0.264	27.445	0.096
0.092	31.349	39.072	0.230	30.655	5.199	0.076	0.128	28.578	0.243	27.946	0.087
0.106	31.849	36.339	0.215	31.152	4.667	0.068	0.150	29.095	0.220	28.485	0.077
0.122	32.344	33.483	0.200	31.612	4.125	0.061	0.175	29.641	0.197	29.014	0.068
0.140	32.833	30.750	0.184	32.010	3.594	0.055	0.204	30.231	0.172	29.523	0.059
0.164	33.436	27.672	0.171	32.656	3.123	0.049	0.243	30.822	0.153	30.199	0.052
0.192	34.029	24.661	0.159	33.243	2.658	0.043	0.288	31.489	0.134	30.880	0.044
0.225	34.625	21.920	0.146	33.784	2.211	0.038	0.341	32.177	0.117	31.530	0.039
0.264	35.212	19.288	0.133	34.261	1.829	0.034	0.402	32.939	0.099	32.162	0.033
0.308	35.780	16.797	0.119	34.946	1.635	0.030	0.469	33.536	0.087	32.899	0.029
0.360	36.350	14.557	0.106	35.547	1.447	0.026	0.547	34.204	0.074	33.589	0.025
0.421	37.007	12.611	0.093	36.158	1.257	0.022	0.636	35.004	0.063	34.323	0.021
0.489	37.586	10.949	0.078	36.647	1.108	0.019	0.733	35.790	0.052	34.958	0.017
0.540	38.017	10.102	0.072	37.103	1.020	0.018	0.804	36.230	0.048	35.454	0.016
0.596	38.428	9.266	0.065	37.520	0.974	0.016	0.880	36.711	0.044	35.932	0.014
0.657	38.821	8.433	0.059	37.879	0.993	0.015	0.962	37.162	0.040	36.337	0.013
0.725	39.182	7.601	0.052	38.065	1.008	0.015	1.050	37.638	0.037	36.606	0.012

ence on Computer Vision and Pattern Recognition (CVPR), June 2023. 1

[3] Noor Fathima Ghouse, Jens Petersen, Auke Wiggers, Tianlin Xu, and Guillaume Sautière. A residual diffusion model for high perceptual quality codec augmentation. arXiv preprint arXiv:2301.05489, 2023. 1

[4] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[5] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[6] Zhenhong Sun, Zhiyu Tan, Xiuyu Sun, Fangyi Zhang, Yichen Qian, Dongyang Li, and Hao Li. Interpolation variable rate image compression. In *Proceedings of ACM International Conference on Multimedia (ACMMM)*, 2021. 1, 6, 7

[7] George Toderici, Lucas Theis, Nick Johnston, Eirikur Agustsson, Fabian Mentzer, Johannes Balle, Wenzhe Shi, and Radu Timofte. CLIC 2020: Challenge on learned image compression, 2020. `https://www.tensorflow.org/datasets/catalog/clic`. 2

[8] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 4

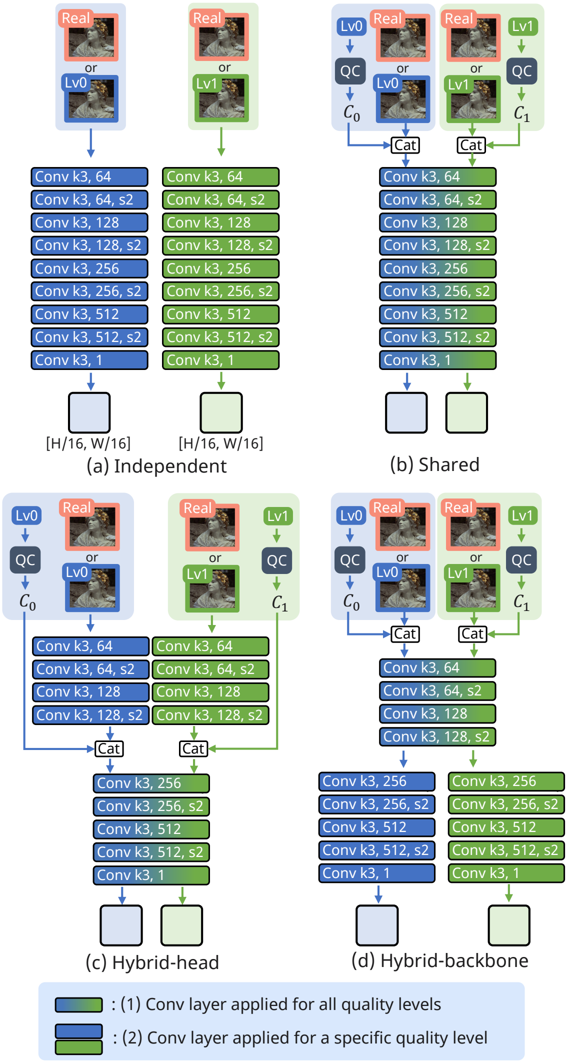

Figure 1. The detail of the discriminator architecture. Numbers within the convolution layers represent the kernel size, output channel, and stride, respectively. For instance, ”Conv k3, 256, s2” denotes a convolution layer with a kernel size of 3, an output channel of 256, and a stride of 2. Each discriminator’s output dimensions are $(H/16, W/16)$, where H and W are the height and width of the image, respectively.

Algorithm 1 HRRGAN loss function

```
Input: Original image $\boldsymbol{x}$
Uniformly sample realism weight $\beta \in [0, \beta_{max}]$
Uniformly sample quality level $q \in \{0, 1, \cdots, Q-1\}$
$\hat{\boldsymbol{x}}_q \leftarrow \mathrm{NIC}(\boldsymbol{x}, q, \beta)$
if $q < Q - 1$ then
    $\hat{\boldsymbol{x}}_{q+1} \leftarrow \mathrm{NIC}(\boldsymbol{x}, q+1, \beta)$
    $p_r \leftarrow \mathrm{sigmoid}(D(\hat{\boldsymbol{x}}_q) - \mathrm{sg}(D(\hat{\boldsymbol{x}}_{q+1})))$
else
    $p_r \leftarrow \mathrm{sigmoid}(D(\hat{\boldsymbol{x}}_q) - \mathrm{sg}(D(\boldsymbol{x})))$
end if
$p_f \leftarrow \mathrm{sigmoid}(D(\boldsymbol{x}) - D(\hat{\boldsymbol{x}}_q))$
$\mathcal{L}^G_{HRRGAN} \leftarrow -\log p_r$
$\mathcal{L}^D_{HRRGAN} \leftarrow -\log p_f$
```

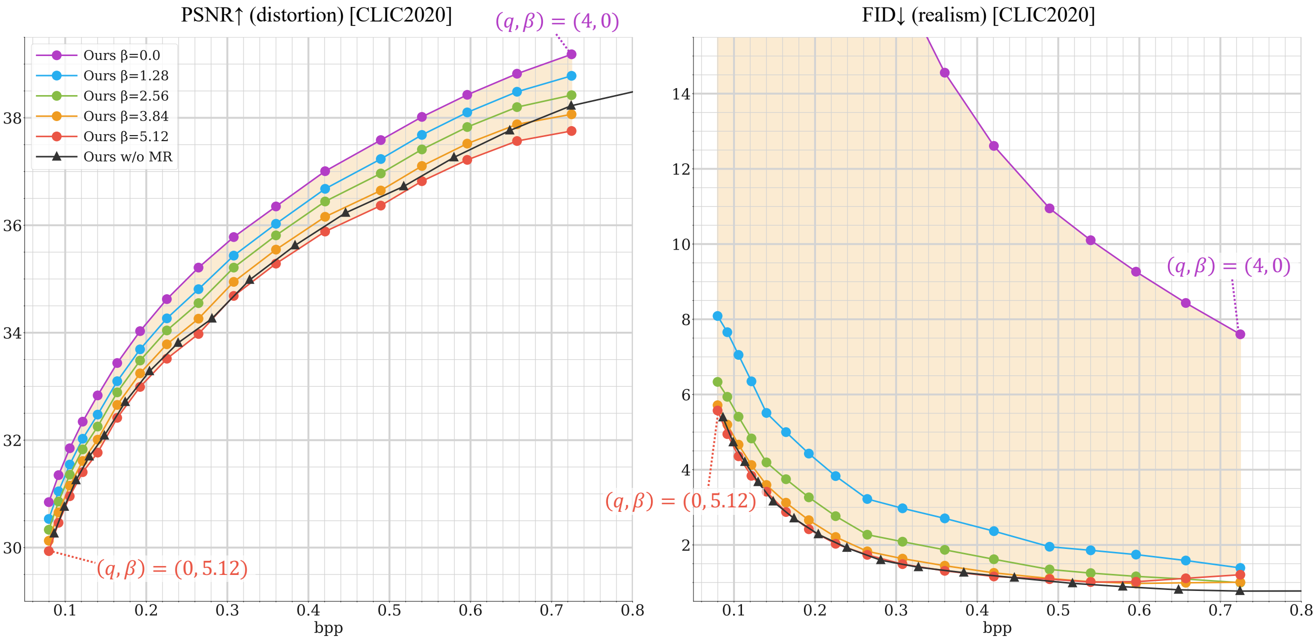

Figure 2. Quantitative comparison of different input realism weights β on CLIC2020 test dataset. *Ours w/o MR* represents a baseline model trained with fixed $\beta = 2.56$. These results demonstrate that our model effectively balances the distortion-realism trade-off by adjusting input β.

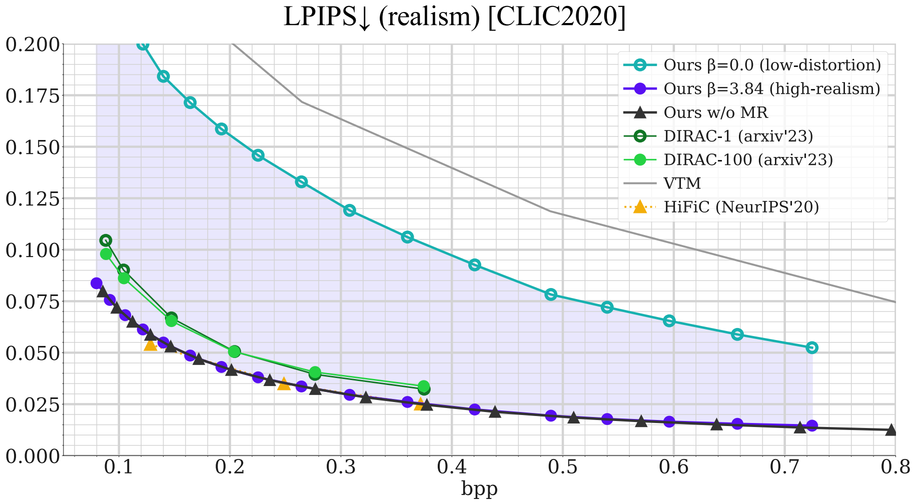

Figure 3. LPIPS [8] results on CLIC2020 test dataset.

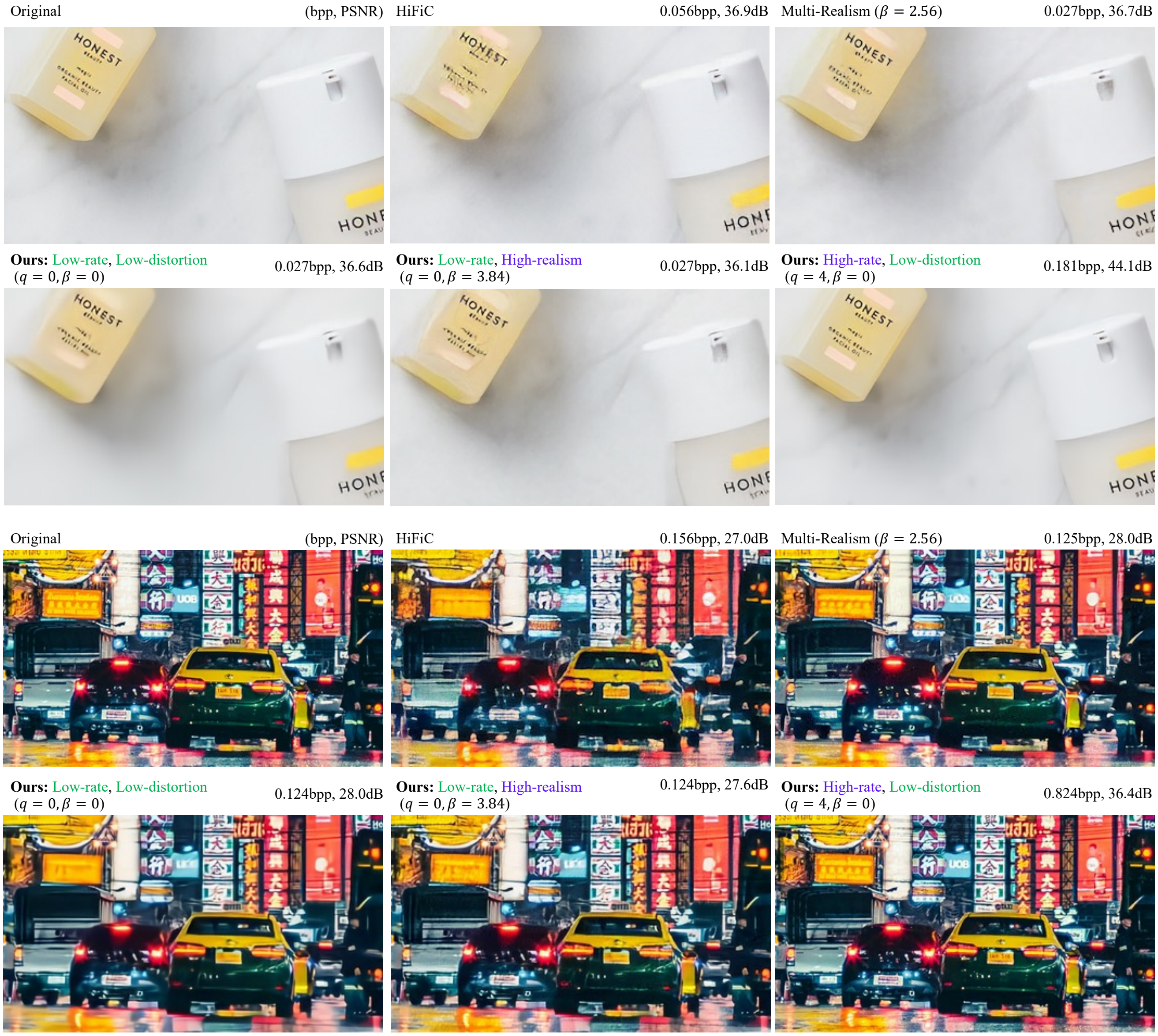
Original (bpp, PSNR)
HiFiC 0.056bpp, 36.9dB
Multi-Realism ($\beta = 2.56$) 0.027bpp, 36.7dB
Ours: Low-rate, Low-distortion ($q = 0, \beta = 0$) 0.027bpp, 36.6dB
Ours: Low-rate, High-realism ($q = 0, \beta = 3.84$) 0.027bpp, 36.1dB
Ours: High-rate, Low-distortion ($q = 4, \beta = 0$) 0.181bpp, 44.1dB
Original (bpp, PSNR)
HiFiC 0.156bpp, 27.0dB
Multi-Realism ($\beta = 2.56$) 0.125bpp, 28.0dB
Ours: Low-rate, Low-distortion ($q = 0, \beta = 0$) 0.124bpp, 28.0dB
Ours: Low-rate, High-realism ($q = 0, \beta = 3.84$) 0.124bpp, 27.6dB
Ours: High-rate, Low-distortion ($q = 4, \beta = 0$) 0.824bpp, 36.4dB

Figure 4. Qualitative comparison on CLIC2020 dataset.

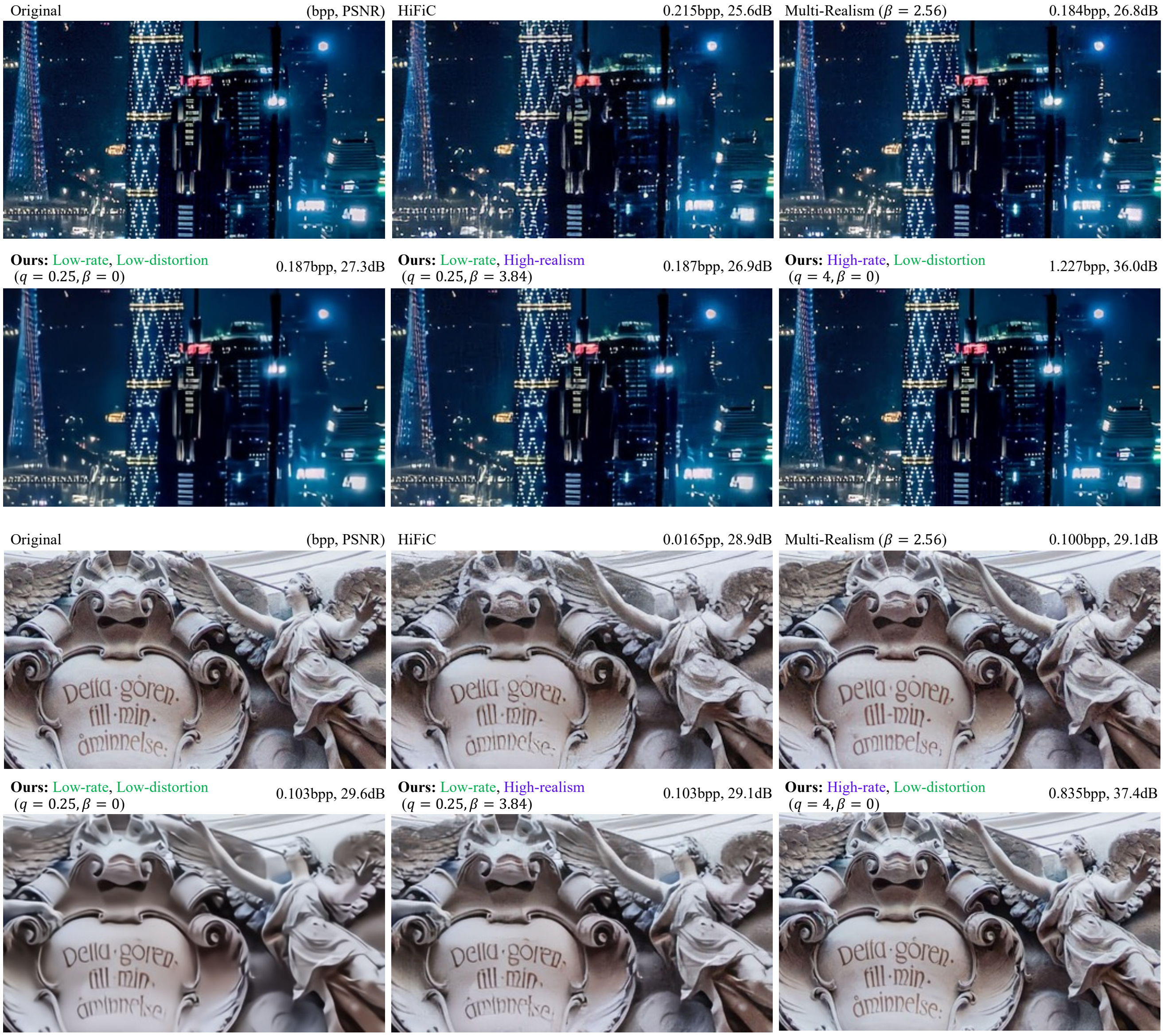
Original
(bpp, PSNR)
HiFiC
0.215bpp, 25.6dB
Multi-Realism ($\beta = 2.56$)
0.184bpp, 26.8dB
Ours: Low-rate, Low-distortion ($q = 0.25, \beta = 0$)
0.187bpp, 27.3dB
Ours: Low-rate, High-realism ($q = 0.25, \beta = 3.84$)
0.187bpp, 26.9dB
Ours: High-rate, Low-distortion ($q = 4, \beta = 0$)
1.227bpp, 36.0dB
Original
(bpp, PSNR)
HiFiC
0.0165pp, 28.9dB
Multi-Realism ($\beta = 2.56$)
0.100bpp, 29.1dB
Ours: Low-rate, Low-distortion ($q = 0.25, \beta = 0$)
0.103bpp, 29.6dB
Ours: Low-rate, High-realism ($q = 0.25, \beta = 3.84$)
0.103bpp, 29.1dB
Ours: High-rate, Low-distortion ($q = 4, \beta = 0$)
0.835bpp, 37.4dB

Figure 5. Qualitative comparison on CLIC2020 dataset. In *Ours*, non-integer q indicates that we interpolated the scaling vectors for fine rate control as in [6].

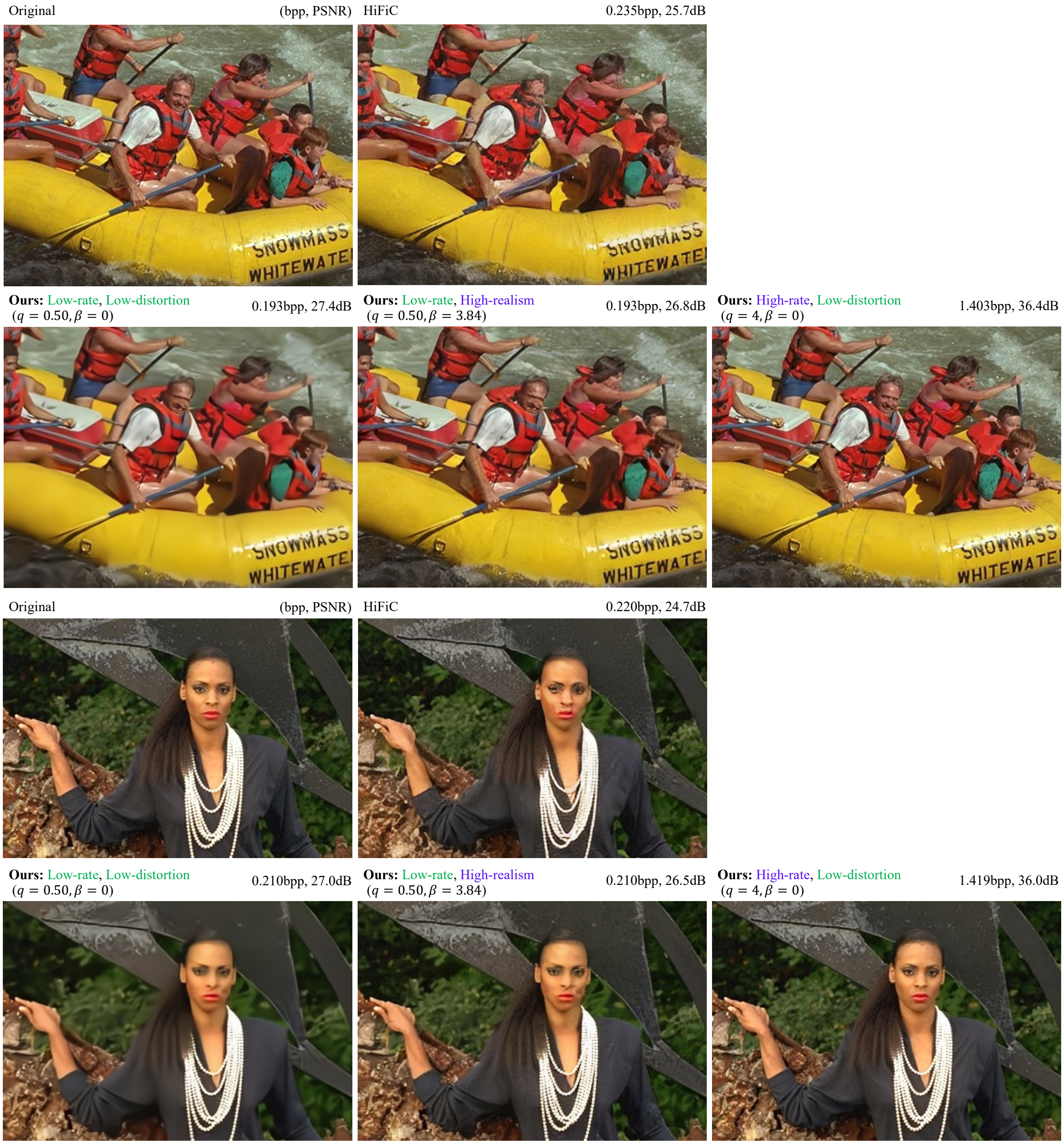

Figure 6. Qualitative comparison on Kodak dataset. In *Ours*, non-integer q indicates that we use interpolated channel attention [6] for fine rate control.