AdaBM: On-the-Fly Adaptive Bit Mapping for Image Super-Resolution

Supplementary Material

Cheeun Hong[1] Kyoung Mu Lee[1,2]
[1] Dept. of ECE & ASRI, [2] IPAI, Seoul National University, Seoul, Korea
{cheeun914, kyoungmu}@snu.ac.kr

In this supplementary material, we present additional experimental results in Section A, additional analyses in Section B, additional ablation study in Section C, and additional qualitative results in Section D.

A. Additional experiments

A.1. Comparison on scale 2

In addition to the evaluations done in the main manuscript on SR networks of scale 4, we extend our evaluation to SR networks of scale 2. First, we compare our method with existing adaptive quantization methods for SR in Table S1. For a fair comparison, we apply quantization to the body module following previous methods. As shown in Table S1, our method achieves a similar trade-off to existing adaptive quantization methods but with a significantly shorter process time. We note that although PSNR/SSIM scores are lower on SRResNet ($\times$2), we incur lower computational costs (lower FAB). Furthermore, we compare our method with existing quantization methods without quantization-aware training (QAT) on scale 2 SR networks. The results in Table S2 demonstrate that our method achieves competitive results against existing static quantization methods without QAT; our method results in lower computational complexity (FAB) and higher accuracy (PSNR/SSIM).

Model	QAT	GT	Process Time	W / A	Urban100		Test2K		Test4K	
					FAB	PSNR / SSIM	FAB	PSNR / SSIM	FAB	PSNR / SSIM
EDSR ($\times$2)	-	-	-	32 / 32	32.0	31.98 / 0.927	32.0	32.76 / 0.928	32.0	34.37 / 0.944
EDSR-CADyQ	✓	✓	47 hrs	8 / 6MP	6.1	31.90 / 0.927	5.8	32.70 / 0.928	5.7	34.31 / 0.943
EDSR-CABM	✓	✓	82 hrs	8 / 6MP	5.8	31.89 / 0.927	5.4	32.72 / 0.927	5.4	34.33 / 0.943
EDSR-AdaBM (**Ours**)	✗	✗	**103 sec**	8 / 6MP	**5.8**	**31.86 / 0.927**	**5.5**	**32.73 / 0.928**	**5.4**	**34.33 / 0.943**
SRResNet ($\times$2)	-	-	-	32 / 32	32.0	31.60 / 0.923	32.0	32.60 / 0.927	32.0	34.20 / 0.942
SRResNet-CADyQ	✓	✓	51 hrs	8 / 6MP	6.5	31.53 / 0.922	6.5	32.55 / 0.925	5.4	34.16 / 0.942
SRResNet-CABM	✓	✓	89 hrs	8 / 6MP	5.8	31.52 / 0.922	5.5	32.55 / 0.925	5.4	34.16 / 0.942
SRResNet-AdaBM (**Ours**)	✗	✗	**123 sec**	8 / 6MP	**5.6**	**31.32 / 0.920**	**5.2**	**32.42 / 0.922**	**5.2**	**33.96 / 0.937**

Table S1. **Comparisons with adaptive quantization methods on SR networks of scale 2.**

A.2. Full quantization v.s. partial quantization

In this work, we fully quantize SR networks to compare with existing static quantization methods without QAT. However, most quantization methods on SR adopt partial quantization for the SR networks by only applying quantization to the body module of the network. Thus, we analyze the effect of fully quantizing the network in Table S3. Although partial quantization provides limited benefits in terms of cost reduction (*i.e.*, the overall computational cost for the network remains larger), it results in higher reconstruction accuracy. Overall, our method achieves higher accuracy with a lower computational cost in both partial and full quantization settings.

A.3. Comparison on CARN

In addition to the networks evaluated in the main manuscript, we present an evaluation on CARN, a more lightweight SR model. We compare our method with existing adaptive quantization methods for SR in Table S4. The results indicate that AdaBM achieves a similar trade-off with existing methods, while the processing time is substantially accelerated to the second level. Although CARN-AdaBM utilizes a higher average bit-width (FAB) compared to existing methods, it leads to improved reconstruction accuracy. Moreover, we compare our method with static quantization methods without QAT in Table S5. Our adaptive method consistently outperforms existing methods with a lower FAB.

Model	FT	W / A	Set5		Set14		BSD100		Urban100	
			FAB	PSNR / SSIM	FAB	PSNR / SSIM	FAB	PSNR / SSIM	FAB	PSNR / SSIM
EDSR ($\times 2$)	-	32 / 32	32.0	37.99 / 0.961	32.0	33.57 / 0.917	32.0	32.16 / 0.900	32.0	31.98 / 0.927
EDSR-MinMax	✗	4 / 4	4.0	32.87 / 0.850	4.0	30.48 / 0.818	4.0	29.55 / 0.799	4.0	28.92 / 0.821
EDSR-Percentile	✗	4 / 4	4.0	25.83 / 0.876	4.0	26.55 / 0.867	4.0	27.09 / 0.862	4.0	24.18 / 0.842
EDSR-MinMax+FT	✓	4 / 4	4.0	34.55 / 0.907	4.0	31.51 / 0.867	4.0	30.50 / 0.867	4.0	29.19 / 0.847
EDSR-Percentile+FT	✓	4 / 4	4.0	29.69 / 0.915	4.0	28.77 / 0.884	4.0	28.86 / 0.876	4.0	26.23 / 0.864
EDSR-PTQ4SR	✓	4 / 4	4.0	36.88 / 0.947	4.0	32.81 / 0.904	4.0	31.59 / 0.886	4.0	30.60 / 0.907
EDSR-AdaBM (**Ours**)	✓	4 / 4MP	**3.6**	**37.10 / 0.955**	**3.6**	**32.85 / 0.910**	**3.5**	**31.63 / 0.891**	**3.8**	**30.48 / 0.912**
RDN ($\times 2$)	-	32 / 32	32.0	38.05 / 0.961	32.0	33.59 / 0.918	32.0	32.20 / 0.900	32.0	32.12 / 0.929
RDN-MinMax	✗	4 / 4	4.0	24.44 / 0.549	4.0	23.16 / 0.525	4.0	23.29 / 0.527	4.0	22.38 / 0.549
RDN-Percentile	✗	4 / 4	4.0	23.33 / 0.918	4.0	23.39 / 0.757	4.0	24.86 / 0.859	4.0	21.47 / 0.848
RDN-MinMax+FT	✓	4 / 4	4.0	33.63 / 0.930	4.0	30.53 / 0.878	4.0	29.76 / 0.856	4.0	27.13 / 0.851
RDN-Percentile+FT	✓	4 / 4	4.0	27.64 / 0.928	4.0	27.11 / 0.878	4.0	27.42 / 0.861	4.0	24.36 / 0.853
RDN-PTQ4SR	✓	4 / 4	4.0	33.68 / 0.933	4.0	30.73 / 0.868	4.0	29.92 / 0.848	4.0	27.52 / 0.844
RDN-AdaBM (**Ours**)	✓	4 / 4MP	**3.8**	**34.90 / 0.932**	**3.7**	**31.42 / 0.885**	**3.6**	**30.37 / 0.863**	**3.8**	**28.34 / 0.864**

Table S2. **Comparisons with static quantization methods without QAT on SR networks of scale 2.**

Model	FQ	W / A	Set5		Set14		BSD100		Urban100	
			FAB	PSNR / SSIM	FAB	PSNR / SSIM	FAB	PSNR / SSIM	FAB	PSNR / SSIM
EDSR	-	32 / 32	32.0	32.10 / 0.894	32.0	28.58 / 0.781	32.0	27.56 / 0.736	32.0	26.04 / 0.785
EDSR-MinMax+FT	✗	4 / 4	4.0	30.10 / 0.821	4.0	27.37 / 0.722	4.0	26.67 / 0.679	4.0	24.56 / 0.698
EDSR-Percentile+FT	✗	4 / 4	4.0	31.15 / 0.876	4.0	27.96 / 0.769	4.0	27.21 / 0.727	4.0	25.12 / 0.757
EDSR-PTQ4SR	✗	4 / 4	4.0	31.23 / 0.864	4.0	28.02 / 0.757	4.0	27.17 / 0.713	4.0	25.28 / 0.746
EDSR-AdaBM (**Ours**)	✗	4 / 4MP	**4.0**	**31.43 / 0.875**	**3.8**	**28.17 / 0.764**	**3.7**	**27.20 / 0.717**	**3.9**	**25.46 / 0.757**
EDSR-MinMax+FT	✓	4 / 4	4.0	28.97 / 0.821	4.0	26.47 / 0.721	4.0	26.24 / 0.687	4.0	23.46 / 0.674
EDSR-Percentile+FT	✓	4 / 4	4.0	27.01 / 0.819	4.0	25.71 / 0.736	4.0	25.69 / 0.707	4.0	23.18 / 0.707
EDSR-PTQ4SR	✓	4 / 4	4.0	30.51 / 0.836	4.0	27.62 / 0.735	4.0	26.88 / 0.693	4.0	24.92 / 0.721
EDSR-AdaBM (**Ours**)	✓	4 / 4MP	**3.8**	**31.02 / 0.860**	**3.7**	**27.87 / 0.751**	**3.5**	**26.91 / 0.700**	**3.7**	**25.11 / 0.736**

Table S3. **Comparisons between fully quantized networks and partially quantized networks.** FQ denotes full quantization and the evaluation is done on EDSR of scale 4 that consists of 16 residual blocks (64 channels).

Model	QAT	GT	Process Time	W / A	Urban100		Test2K		Test4K	
					FAB	PSNR / SSIM	FAB	PSNR / SSIM	FAB	PSNR / SSIM
CARN	-	-	-	32 / 32	32.0	26.07 / 0.784	32.0	27.70 / 0.782	32.0	28.77 / 0.814
CARN-CADyQ	✓	✓	23 hrs	8 / 6MP	5.2	25.90 / 0.780	4.5	27.64 / 0.781	4.5	28.72 / 0.812
CARN-CABM	✓	✓	41 hrs	8 / 6MP	4.4	25.83 / 0.778	4.2	27.60 / 0.780	4.2	28.67 / 0.811
CARN-AdaBM (**Ours**)	✗	✗	**49 sec**	8 / 6MP	**5.6**	**25.98 / 0.781**	**5.3**	**27.68 / 0.781**	**5.2**	**28.77 / 0.813**

Table S4. **Comparisons with adaptive quantization methods on CARN ($\times 4$).**

B. Analysis

B.1. Data sampling

To obtain the calibration data, we randomly sampled data with a fixed random seed for our main manuscript experiments. However, we found that different random seeds for data sampling yield different performances of the quantized model. Here, we investigate the different sampling schemes for building the calibration dataset. For example, we implement a stratified sampling scheme based on image complexity. Images are divided into N sub-groups based on the image gradient. Then, random sampling is done for each sub-group. As shown in Table S6, such sampling gives additional gain but at the cost of additional processing time from forming the sub-groups.

Model	FT	W / A	Set5		Set14		BSD100		Urban100	
			FAB	PSNR / SSIM	FAB	PSNR / SSIM	FAB	PSNR / SSIM	FAB	PSNR / SSIM
CARN (×4)	-	32 / 32	32.0	32.14 / 0.893	32.0	28.61 / 0.781	32.0	27.58 / 0.736	32.0	26.07 / 0.784
CARN-MinMax	✗	4 / 4	4.0	30.94 / 0.874	4.0	27.82 / 0.760	4.0	27.01 / 0.715	4.0	25.06 / 0.749
CARN-Percentile	✗	4 / 4	4.0	26.55 / 0.806	4.0	25.75 / 0.729	4.0	25.78 / 0.696	4.0	23.42 / 0.703
CARN-MinMax+FT	✓	4 / 4	4.0	31.36 / 0.881	4.0	28.01 / 0.766	4.0	27.21 / 0.723	4.0	25.15 / 0.753
CARN-Percentile+FT	✓	4 / 4	4.0	30.75 / 0.870	4.0	27.73 / 0.759	4.0	26.95 / 0.715	4.0	24.67 / 0.733
CARN-PTQ4SR	✓	4 / 4	4.0	31.41 / 0.881	4.0	28.03 / 0.766	4.0	27.19 / 0.722	4.0	25.22 / 0.755
CARN-AdaBM (**Ours**)	✓	4 / 4MP	**3.7**	**31.68 / 0.885**	**3.6**	**28.23 / 0.771**	**3.4**	**27.30 / 0.726**	**3.6**	**25.45 / 0.762**

Table S5. **Comparisons with static quantization methods without QAT on CARN** (×4)**.**

Sampling Method	FAB$_\downarrow$	PSNR$_\uparrow$	SSIM$_\uparrow$	Processing Time
Random	3.80 ± 0.12	30.79 ± 0.21	0.857 ± 0.004	76 sec
Stratified (N=4)	3.68	30.80	0.853	85 sec
Stratified (N=8)	3.78	30.94	0.858	86 sec

Table S6. **Sampling methods** for 4-bit EDSR (×4) on Set5. For random sampling, we average the result of different seeds.

B.2. On-device latency

Along with the speedup of time to obtain the quantized network, our framework also achieves speedup in inference time. In Table S7, we report the latency of our quantized model on x86 and ARM CPUs. Since only INT4/8 bits are supported for acceleration on current existing inference libraries, we upcast intermediate bits to INT8. The results show that our framework is beneficial in terms of inference time. We anticipate further speedup gain via acceleration on intermediate bits.

Method	EDSR	EDSR-CADyQ	EDSR-**AdaBM**
x86 CPU	4.002 sec	0.974 sec (×4.108)	0.742 sec (×5.391)
ARM CPU	3.998 sec	1.880 sec (×2.126)	1.746 sec (×2.290)

Table S7. **Average latency** for EDSR (×4) on DIV2K validation set.

C. Ablations

We investigate the effect of hyperparameters used in our work: weight for bit loss (λ_{bit}), the percentile for calibrating the image-to-bit mapping module (p_I), and for calibrating the layer-to-bit mapping module (p_L). As shown in Table S8a, the weight of bit loss controls the trade-off between accuracy and computational complexity. Reducing the bit loss weight can cause the bit mapping modules to select overall higher bit-widths, prioritizing minimal reconstruction loss. Consequently, a smaller λ_{bit} results in higher PSNR/SSIM but uses more computational costs (*i.e.*, larger FAB). However, employing a large λ_{bit} strictly restricts the average bit-width from increasing, resulting in a model with smaller computational cost but lower PSNR/SSIM. Our framework can achieve varying levels of trade-off by controlling λ_{bit}, but we fix $\lambda_{bit} = 50$ in our experiments. Additionally, the results in Table S8b and Table S8c justify our choice of hyperparameters.

λ_{bit}	FAB$_\downarrow$	PSNR$_\uparrow$ / SSIM$_\uparrow$
1	4.78	31.38 / 0.872
10	4.08	31.10 / 0.865
50	3.78	31.02 / 0.860
100	3.72	30.89 / 0.858

(a) Ablation on λ_{bit}

p_I	FAB$_\downarrow$	PSNR$_\uparrow$ / SSIM$_\uparrow$
5	3.99	31.13 / 0.864
10	3.78	31.02 / 0.860
20	3.85	30.93 / 0.860
30	3.99	30.79 / 0.858

(b) Ablation on p_I

p_L	FAB$_\downarrow$	PSNR$_\uparrow$ / SSIM$_\uparrow$
5	3.84	31.00 / 0.858
10	3.96	31.06 / 0.860
20	3.84	31.05 / 0.860
30	3.78	31.02 / 0.860

(c) Ablation on p_L

Table S8. **Effect of hyperparameters** evaluated on Set5 with 4-bit EDSR (×4).

D. Additional qualitative results

D.1. Qualitative comparison

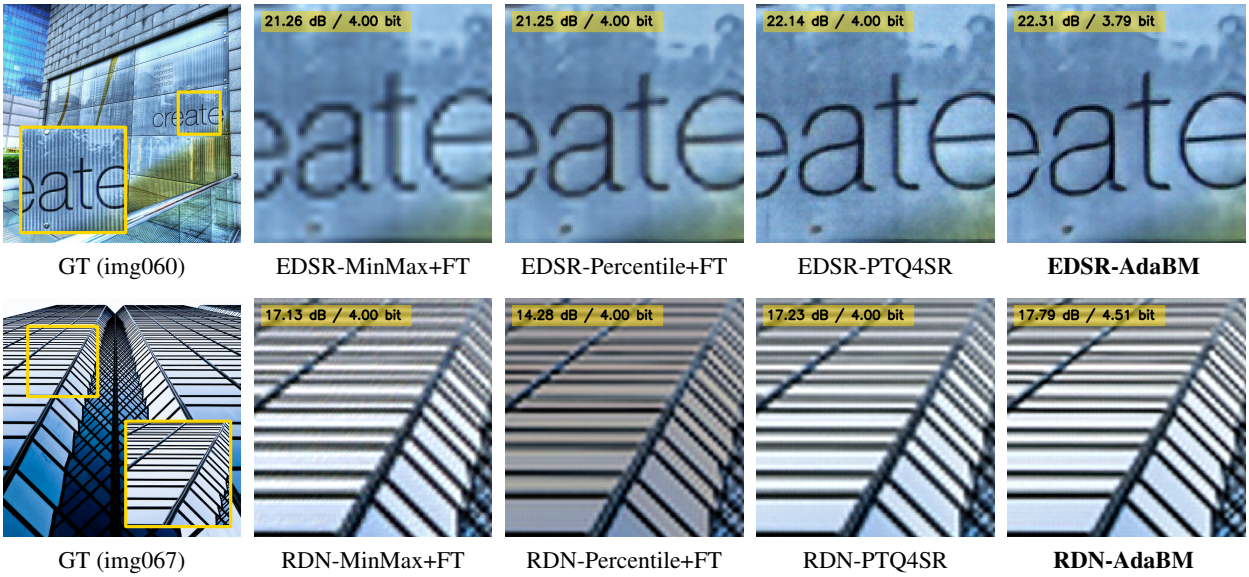
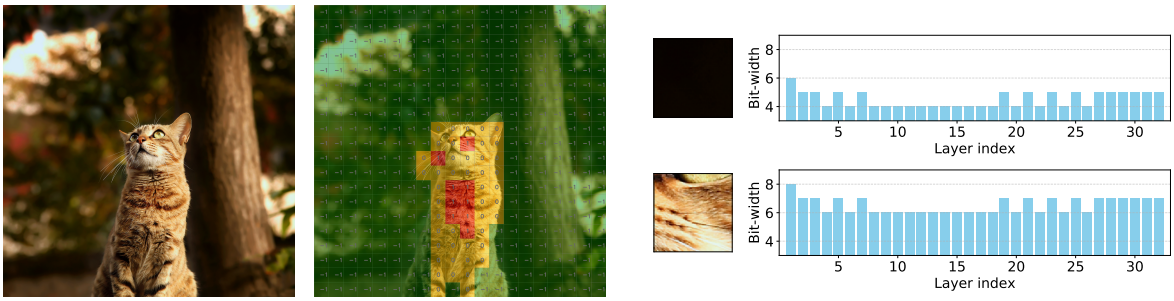
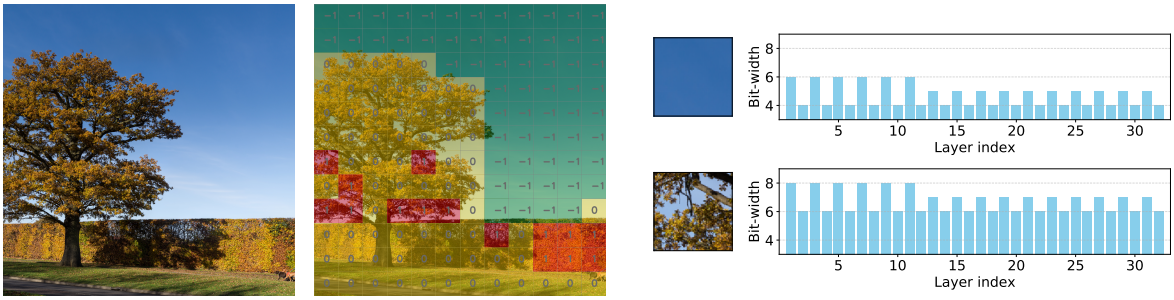


Figure S1. **Qualitative results** on 4-bit SR networks of scale 4. The networks are fully quantized.

D.2. Visualization



(a) EDSR-AdaBM



(b) SRResNet-AdaBM

Figure S2. **Visualization of adaptive bit-mapping of AdaBM** on large inputs. Evaluation done on SR networks of scale 4.