Dynamic Dual Trainable Bounds for Ultra-low Precision Super-Resolution Networks (Supplementary Material)

Yunshan Zhong[1,2], Mingbao Lin[3], Xunchao Li[2], Ke Li[3], Yunhang Shen[3], Fei Chao[1,2], Yongjian Wu[3], Rongrong Ji[1,2⋆]

[1]Institute of Artificial Intelligence, Xiamen University.
[2]MAC Lab, School of Informatics, Xiamen University. [3]Tencent Youtu Lab.
zhongyunshan@stu.xmu.edu.cn, linmb001@outlook.com,
lixunchao@stu.xmu.edu.cn,
{tristanli.sh, shenyunhang01}@gmail.com,
fchao@xmu.edu.cn, littlekenwu@tencent.com, rrji@xmu.edu.cn

Table S1. Complexity analysis. The number in brackets indicates the parameters in the high-level feature extraction module. We compute BOPs by generating a 1920×1080 image (upscaling factor ×4).

Model	Bit	Params	Gate Params(*ratio*)	BOPs	Gate BOPs(*ratio*)
EDSR	32	1.52M	0	532T	0
EDSR_DDTB	2	0.41M(0.08M)	0.6%	219T	0.0000013%
EDSR_DDTB	3	0.45M(0.11M)	0.6%	220T	0.0000013%
EDSR_DDTB	4	0.49M(0.15M)	0.5%	222T	0.0000013%
RDN	32	22.3M	0	6038T	0
RDN_DDTB	2	1.76M(1.42M)	2.8%	239T	0.0000066%
RDN_DDTB	3	2.44M(2.10M)	2%	267T	0.0000059%
RDN_DDTB	4	3.13M(2.79M)	1.6%	307T	0.0000051%
SRResNet	32	1.543M	0	591T	0
SRResNet_DDTB	2	0.44M(0.07M)	0.1%	278T	0.0000002%
SRResNet_DDTB	3	0.47M(0.11M)	0.1%	280T	0.0000002%
SRResNet_DDTB	4	0.51M(0.15M)	0.1%	282T	0.0000002%

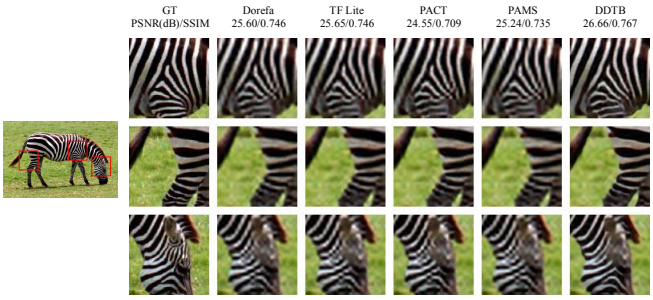
1 Model Complexity

Table S1 provides the complexity analyses of 4, 3, 2-bit SR models. The extra overhead of the dynamic gate controller is negligible.

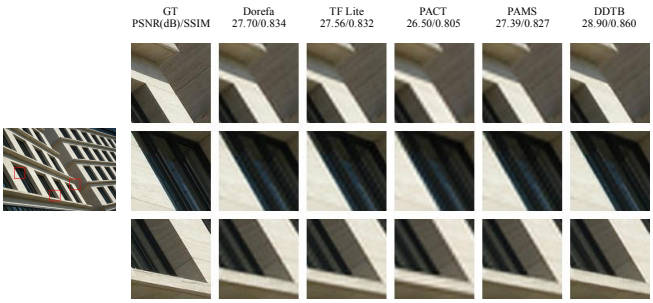
2 Visualization

Fig. 1(a), Fig. 1(b), and Fig. 1(c) exhibit the reconstructed results of the 2-bit EDSR×4, 2-bit RDN×4, and 2-bit SRResNet×4, respectively. The reported PSNR/SSIM are measured by the displayed image. It can be seen that our method obtains the best visualization results compared with other methods. This demonstrates the superiority of our DDTB.

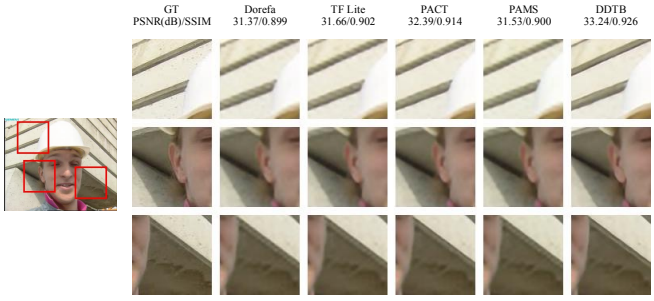
⋆ Corresponding Author



(a) 2-bit EDSR×4.



(b) 2-bit RDN×4.



(c) 2-bit SRResNet×4.

Fig. S1. Reconstructed results of 2-bit EDSR×4, 2-bit RDN×4, and 2-bit SRResNet×4.

3 Results of Fully Quantized Models

In this section, we provide the comparisons between the existing fully quantized method FQSR [3] and our DDTB. Following FQSR [3], all layers and skip-connections of SR models are quantized. As shown in Table S2, DDTB outperforms FQSR by a large margin when performing 4-bit quantization. For instance, DDTB obtains performance gains by 0.98dB, 0.58dB, 0.37dB, and 0.77dB on Set5, Set14, BSD100, and Urban100, respectively.

Table S2. PSNR/SSIM comparisons between the existing fully quantized method and our DDTB. "SC" indicates the bit-width of skip-connections.

Model	Bit	SC	Methods	Set5	Set14	BSD100	Urban100
EDSR ×4	4	8	FQSR	30.93/0.870	27.82/0.761	27.07/0.715	24.93/0.744
			DDTB(Ours)	**31.91/0.889**	**28.40/0.777**	**27.44/0.732**	**25.70/0.775**
EDSR ×2	4	8	FQSR	37.04/0.951	32.84/0.908	31.67/0.889	30.65/0.911
			DDTB(Ours)	**37.83/0.960**	**33.44/0.916**	**32.07/0.898**	**31.60/0.924**
SRGAN ×4	4	8	FQSR	30.96/0.872	27.85/0.759	27.08/0.713	24.93/0.742
			DDTB(Ours)	**31.46/0.879**	**28.10/0.766**	**27.26/0.721**	**25.33/0.757**
SRGAN ×2	4	8	FQSR	36.69/0.950	32.64/0.906	31.57/0.888	30.37/0.908
			DDTB(Ours)	**36.84/0.950**	**32.79/0.907**	**31.69/0.889**	**30.70/0.910**
SRResNet ×4	4	8	FQSR	31.04/0.874	27.86/0.761	27.09/0.714	24.95/0.744
			DDTB(Ours)	**31.51/0.881**	**28.17/0.767**	**27.31/0.722**	**25.39/0.760**
SRResNet ×2	4	8	FQSR	36.34/0.945	32.40/0.901	31.37/0.882	29.98/0.899
			DDTB(Ours)	**36.85/0.951**	**32.73/0.907**	**31.63/0.890**	**30.63/0.910**

4 Differences with LSQ and ReActNet

Though both LSQ [1] and DDTB adopt a linear quantizer, they differ in: 1) LSQ focuses on the classification task while DDTB focuses on the SR task; 2) LSQ uses the symmetric quantizer for signed data which is not suitable for highly asymmetric activations in the SR task. In contrast, DDTB uses an asymmetric quantizer; 3) LSQ trains the scaling factor while DDTB trains the upper bound and the lower bound; 4) To stabilize training, LSQ adjusts the gradient of the scaling factor by multiplying a delicate selected constant. However, DDTB only uses an initializer without the need to manipulate the gradient.

Both ReActNet [2] and DDTB discover the importance of activation distribution, but they fundamentally differ in: 1) ReActNet performs binary quantization for high-level vision while DDTB focuses on low-bit quantization for low-level vision; 2) ReactNet modifies the activation function and network architecture which were retained in DDTB.

5 Discussion about DDTB Initializer

The accumulation of quantization error and random initial weight of the dynamic gate result in improper initial $\alpha_u, \alpha_l, \beta_u, \beta_l$, further causing inferior performance. These two problems exist regardless of the target dataset, thus DDTB Initializer is always essential. It provides performance improvement (see Tab. 5) as well as stabilizes the training in particular in the case of 3-bit RDN.

References

1. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2020)
2. Liu, Z., Shen, Z., Savvides, M., Cheng, K.T.: Reactnet: Towards precise binary neural network with generalized activation functions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 143–159. Springer (2020)
3. Wang, H., Chen, P., Zhuang, B., Shen, C.: Fully quantized image super-resolution networks. In: Proceedings of the 29th ACM International Conference on Multimedia (ACM MM). pp. 639–647 (2021)