A Appendix

Different saliency predictor designs. In our method, the channel saliency predictor consists of two full-connected layers with ReLU in-between and Sigmoid at the output. This MLP-based predictor takes as input the feature-maps statistics computed by SI operator, and outputs the importance scores. For comparison, we introduce a CNN-based predictor, consisting of one 3 × 3 stride-2 convolution followed by global average pooling and one fully connected layer. The use of convolution in the predictor serves to replace the role of SI by learning to extract the feature-maps statistics. Moreover, we introduce a recurrent neural network (RNN) based predictor to enable parameter sharing and reuse of feature-maps information from previous layers. The RNN-based predictor consists of a global average pooling, a single layer LSTM [4], and one fully-connected layer. As shown in Table 1, the MLP-based predictor achieves the highest PSNR. The CNN predictor adds much more parameters and is prone to over-fitting, while the weight-sharing in RNN predictor neglects different impacts from different layers to the final reconstruction performance.

Table 1: Comparison of different saliency predictor designs, including recurrent (RNN) based, convolution (CNN) based, and the default multi-layer perceptrons (MLP). PSNR are evaluated with EDSR-baseline after 50% FLOPs reduction for x4 SISR.

Predictor	Set5	Set14	B100	Urban100
CNN	32.20	28.56	27.54	25.93
RNN	32.13	28.55	27.53	25.87
MLP (default)	32.25	28.63	27.59	26.04

Effect of two-stage training. In Table 2, we compare the PSNR of models obtained by our two-stage training schema versus training from scratch. Our two-stage training schema involves (1) pre-training stage that trains the plain SR network from scratch without routers and channel saliency predictors (2) searching stage that trains the pre-trained SR network together with routers and predictors using joint SR loss and sparsity loss for learning input-dependent compression policies. For scratch training, we train with doubled epoch numbers to match the total training steps of the two-stage training. As observed, two-stage training yields higher PSNR over four benchmarks. This suggests that supervised pre-training can provide a more effective initialization which facilitates the searching stage to excavate the model redundancy with negligible PSNR loss.

More comparisons with SOTA efficient SR. In Table 2 of main paper, we compared our extremely compressed models (Ours-L,M,S) with SOTA super-efficient SR methods (less than 10G FLOPs). Here in Table 3, we additionally compare our moderately compressed models with more lightweight SR methods for x4 SISR. We compress EDSR-baseline (114 GFLOPs) to obtain efficient SR models (Ours-XL) with 40 and 30 GFLOPs for comparison with other leading methods. As shown, our compressed models achieve competitive PSNR while with similar or fewer FLOPs.

Table 2: Effect of our two-stage training schema. PSNR are evaluated with EDSR-baseline after 50% FLOPs reduction for x4 SISR.

Training strategy	Set5	Set14	B100	Urban100
Scratch	32.04	28.48	27.50	25.83
Two-stage (default)	32.25	28.63	27.59	26.04

Table 3: Quantitative comparison with SOTA efficient SR methods. FLOPs are calculated as the number of multiply-adds needed to convert an image to 720p (1280 × 720) resolution. Best results are highlighted as Red.

Method	FLOPs	Set5	Set14	B100	Urban100
SRCNN [2]	52.7G	30.48	27.49	26.90	24.52
VDSR [6]	612.6G	31.35	28.01	27.29	25.18
LapSRN [7]	149.4G	31.54	28.19	27.32	25.21
DRRN [9]	6,796.9G	31.68	28.21	27.38	25.44
BTSRN [3]	165.2G	31.85	28.20	27.47	25.74
MemNet [10]	2,662.4G	31.74	28.26	27.40	25.50
SRResNet [8]	146.1G	32.05	28.49	27.58	25.90
CARN-M [1]	32.5G	31.92	28.42	27.44	25.62
CBPN-S [12]	63.1G	31.93	28.50	27.50	25.85
IMDN [5]	40.9G	32.21	28.58	27.56	26.04
PAN [11]	28.2G	32.13	28.61	27.59	26.11
Ours-XL	40G	32.21	28.62	27.60	26.11
	30G	32.17	28.61	27.59	26.07

Realistic accelerations of compressed models. The realistic accelerations of compressed SR models (in Table 1 of main paper) on DIV2K validation set are shown in Table 4, which is calculated by counting the average inference time for processing each image on CPU. The realistic acceleration is slightly less than the theoretical acceleration calculated by FLOPs reduction, which is due to practical factors such as I/O operations (e.g., accessing weights of networks), BLAS libraries and buffer switch, whose impact may be reduced by future engineering optimizations.

Table 4: Realistic speedup (inference run-time reduction) and theoretical speedup (FLOPs reduction) of compressed SR models on DIV2K validation.

Model	Theoretical speedup	Realistic speedup
EDSR-baseline	50%	38%
CARN	50%	35%
RDN	47%	32%

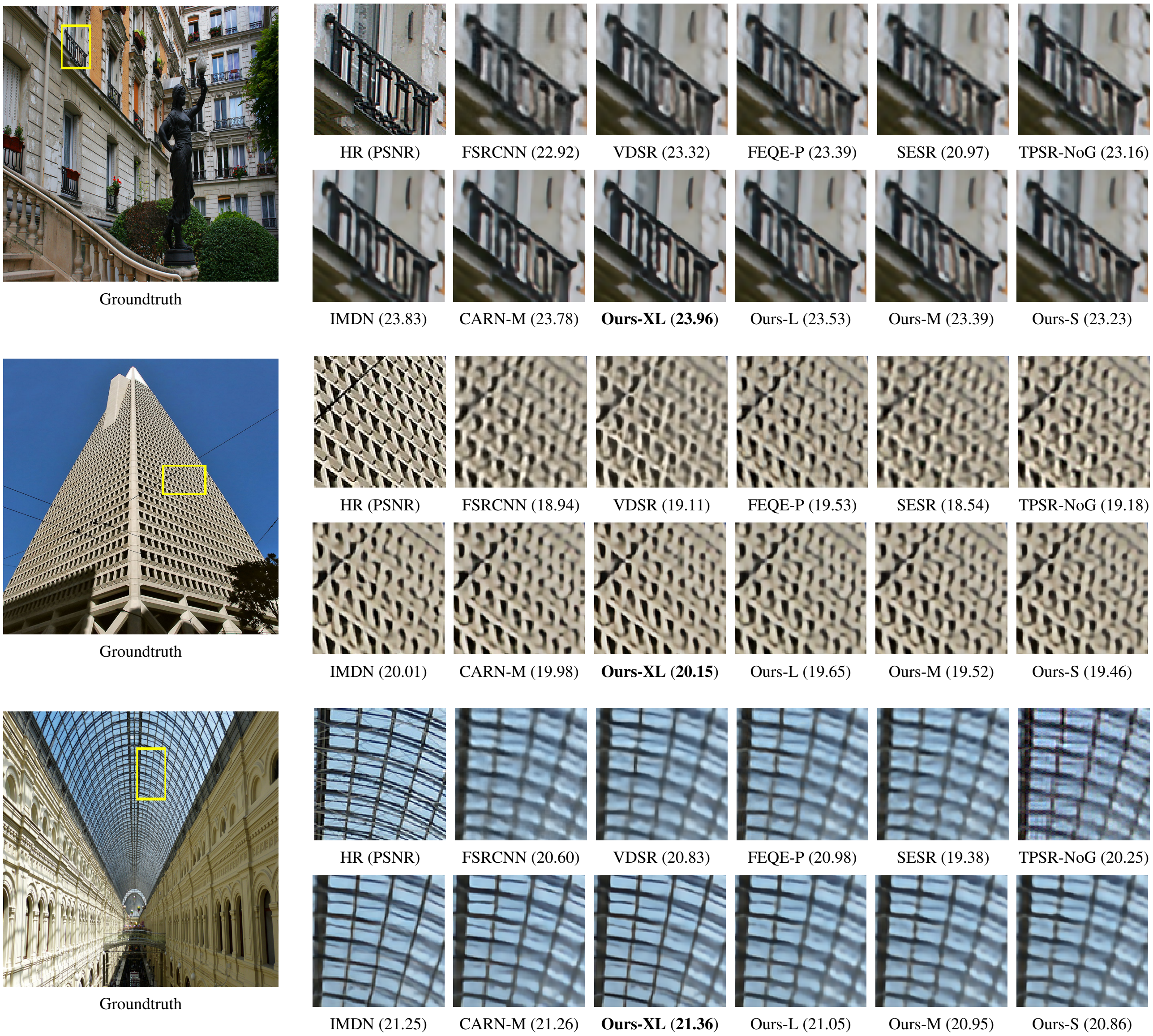

Figure 1: More visual comparisons with SOTA efficient SR methods for x4 SISR. Our compressed models demonstrate better image quality while requiring similar or fewer FLOPs than other manually designed or NAS-based methods.

More qualitative results. Fig.1 provides more qualitative comparisons with the state-of-the-art efficient SR methods.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 252–268, 2018.

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[3] Yuchen Fan, Honghui Shi, Jiahui Yu, Ding Liu, Wei Han, Haichao Yu, Zhangyang Wang, Xinchao Wang, and Thomas S Huang. Balanced two-stage residual networks for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 161–168, 2017.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2024–2032, 2019.

[6] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[7] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.

[8] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[9] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017.

[10] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.

[11] Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong. Efficient image super-resolution using pixel attention. *arXiv preprint arXiv:2010.01073*, 2020.

[12] Feiyang Zhu and Qijun Zhao. Efficient single image super-resolution via hybrid residual feature learning with compact back-projection network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.