Learning Series-Parallel Lookup Tables for Efficient Image Super-Resolution Supplementary Material

Cheng Ma[1,2,*], Jingyi Zhang[1,2,*], Jie Zhou[1,2], and Jiwen Lu[1,2,†]

[1] Beijing National Research Center for Information Science and Technology, China
[2] Department of Automation, Tsinghua University, China
macheng17@tsinghua.org.cn; zhangjy20@mails.tsinghua.edu.cn
{jzhou, lujiwen}@tsinghua.edu.cn

A Details of Mapping Modules

In the training phase, we replace the LUTs with mapping modules, which are comprised of a convolutional layer with a kernel size of $k_h \times k_w$, activation layers of GELU [1] and 1×1 convolutional layers. The architecture is displayed in Fig. 1. All convolutional layers output feature maps with 64 channels except the last one. The output channel number C_{out} of the last convolutional layer is set to $C_{out} = C_f$ for the mapping modules that extract intermediate features and $C_{out} = C_{SR} = s^2$ for the mapping modules that produce the final SR results. s is the upscaling factor. The consecutive 1×1 convolutions followed by GELU layers strengthen the nonlinearity and representative ability of the mapping modules. In the last query block of each parallel branch, the mapping module contains an additional pixel-shuffle layer [3]. It maps the outputs vectors with s^2 channels to $s \times s$ output patches to get the final SR images. In the inference phase, we only care about the inputs and outputs of these mapping modules. Therefore, the detailed architecture of the mapping modules does NOT affect the computation complexity of LUTs.

B Details of SPLUT-S and SPLUT-L

We design three SPLUT models with different model sizes, namely SPLUT-S, SPLUT-M, and SPLUT-L. The three models have similar architectures. The difference between the three models lies in the channel number of intermediate features C_f and the details of query blocks. We have introduced the details of SPLUT-M in Section 3 of our main paper. SPLUT-S and SPLUT-L have the same overall framework as SPLUT-M but have different query blocks. Note that the three models have the same receptive field size. Here we describe the details of their query blocks, which are shown in Fig. 3.

* Equal contribution.
† Corresponding author.

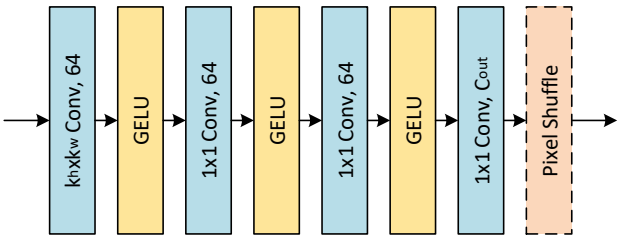


Fig. 1. Detailed architecture of the mapping module. Only the last mapping module of each branch contains the pixel shuffle layer.

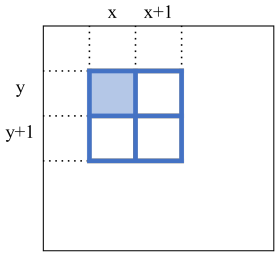


Fig. 2. The receptive field of $r = 4$ for the pixel colored in blue.

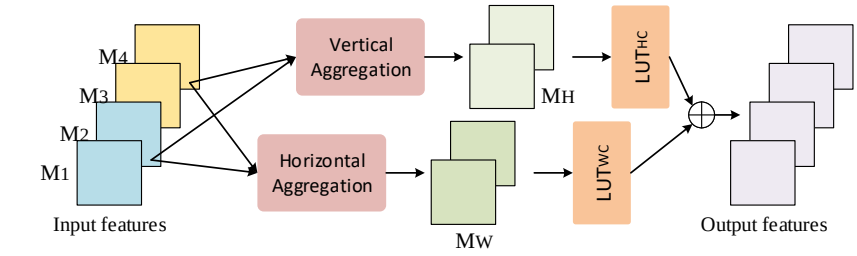
SPLUT-S is a small model with $C_f = 4$. In the query block of SPLUT-S, we split 4 intermediate feature maps into 2 groups, each adjacent two in one group. Following SPLUT-M, there is a horizontal aggregation module and a vertical aggregation module in each query block. The two modules both take the above two groups of features as inputs. The obtained features, M_H and M_W, are used for the following LUT retrievals. We get the output of the query block by adding the LUT results.

As for SPLUT-L with $C_f = 16$, there are four aggregation modules and four LUTs per query block to increase the capacity of the model. The inputs to the aggregation modules include 8 groups of intermediate features. The first four groups are fed into two vertical aggregation modules and the other four groups are fed into two horizontal modules. The outputs are M_{H1}, M_{H2}, M_{W1}, M_{W2}, respectively. We then feed M_{H1} and M_{H2} into two LUT_{HC} and feed M_{W1} and M_{W2} into two LUT_{WC} for further processing.

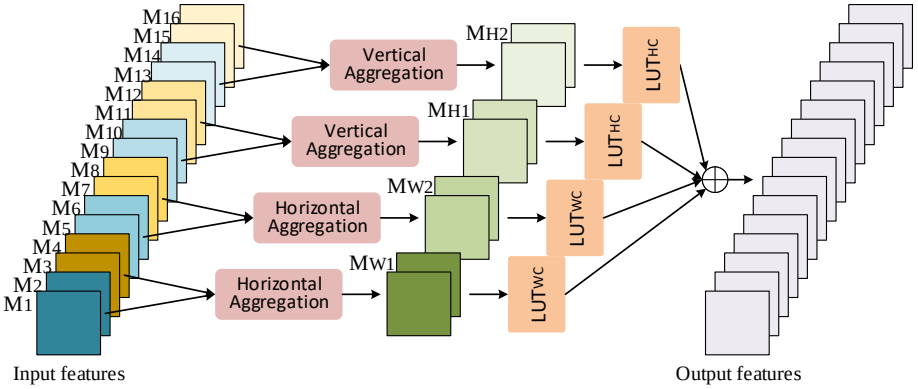
C Details of Two Interpolation Methods

Here we describe more implementation details of the ablation studies on the parallel network vs. interpolation methods. We first introduce the interpolation methods in SRLUT [2] and then provide the details of the two designed interpolation methods. SR approximates the output of an input pattern by interpolating the 4D LUT outputs of the nearest sampled points to the input pattern. For a position of (x, y) shown in Fig. 2, the input pixels of (x, y), $(x + 1, y)$, $(x, y + 1)$ and $(x + 1, y + 1)$ form an input pattern. SR-LUT implements 4-simplex interpolation for the 4D LUT by exploring the relation of $I_{LSB}^{(x,y)}$, $I_{LSB}^{(x+1,y)}$, $I_{LSB}^{(x,y+1)}$ and $I_{LSB}^{(x+1,y+1)}$. A weighted sum of the retrieval results for the 16 bounding vertices is computed as the final output since SR-LUT has only one layer of LUT. Specifically, the nearest sampled points $P_{ijkl}[x][y]$ at the position of (x, y) are calculated as follows:

$$P_{ijkl}[x][y] = ((I_{MSB}^{(x,y)} + i), (I_{MSB}^{(x+1,y)} + j), (I_{MSB}^{(x,y+1)} + k), (I_{MSB}^{(x+1,y+1)} + l)) \quad (1)$$



(a) SPLUT-S



(b) SPLUT-L

Fig. 3. Details of the proposed query block for (a) SPLUT-S and (b) SPLUT-L. The two models have the same receptive field size.

where i, j, k, l are 0 or 1. We can compute the indices for LUT retrieval by these points. SR-LUT selects the contributing bounding vertices by their distances to the input pattern and then use 4-simplex interpolation to compute a weighted sum of their retrieval results according to $I_{LSB}^{(x,y)}$, $I_{LSB}^{(x+1,y)}$, $I_{LSB}^{(x,y+1)}$ and $I_{LSB}^{(x+1,y+1)}$.

In our methods, we also utilize I_{LSB} for interpolation but we cannot get the final SR output by directly fusing the retrieval results of the first layer of spatial LUT since we still have other following LUTs for retrieval. Since the cascaded LUTs in our model bring a large RF size of r, it is intractable to consider all the bounding vertices and simply implement interpolations like SR-LUT due to the computational complexity of 2^r. To improve the SR accuracy of this model, we design two interpolation methods for the input images. In our first interpolation method, we concatenate the 16 nearest sampled points $P_{ijkl}[x][y]$ of the input patterns of all positions (x, y) to form 16 index maps $P_{0000}, P_{0001}, ..., P_{1111}$. We take $P_{0000}, P_{0001}, ..., P_{1111}$ as the inputs to the cascaded LUTs, which reduce the complexity of 2^r to 2^4. We can get 16 SR results through the whole network. We interpolate the 16 SR results by I_{LSB} using the 4-simplex method and get the final SR output. We call this method tail-layer interpolation. In our second

Table 1. Ablation study for Skip Connections.

method	SC1	SC2	SC3	Set5		Set14	
				PSNR	SSIM	PSNR	SSIM
SPLUT w/o SC3	✓	✓		29.95	0.8473	27.12	0.7366
SPLUT w/o SC2	✓		✓	30.16	0.8549	27.29	0.7452
SPLUT w/o SC1		✓	✓	30.18	0.8562	27.31	0.7458
SPLUT	✓	✓	✓	30.23	0.8567	27.32	0.7460

method, we feed the 16 index maps to the spatial lookup blocks and get 16 intermediate feature maps. We fuse the 16 feature maps by I_{LSB} and 4-simplex method to get one feature map. By feeding the feature map to the following layers, we get the final SR output. We call this method first-layer interpolation.

D Ablation Study for Skip Connections.

The overall framework of our method is shown in Fig. 2 (a) of the main paper. The first and second skip connections are between the input and output of the spatial LUT block and the first query block. We call the two skip connections "SC1" and "SC2", respectively. The third skip connection "SC3" is between the input image and the output of the last query block. We perform ablation studies to verify the effect of the skip connections. As shown in Table 1, it can be seen that removing the third skip connection has the greatest impact on SR performance (-0.28dB for Set5). The model with this identity mapping can make the lookup tables pay more attention to the reconstruction of residual information and reduce the difficulty of super-resolution. After removing "SC3", the LUTs have to store the information contained in LR inputs, which affects the SR ability of SPLUT. The model of SPLUT w/o SC1 has a similar performance to SPLUT w/o SC2. They both have a performance degradation compared to SPLUT. This indicates that the skip connections can help maintain the feature precision and improve inference accuracy. In practice, it is very efficient to implement skip connections with simple addition operations.

E Ablation Study for Training Strategy

To analyze the effectiveness of the jointly training strategy and measure the function of each branch, we conduct experiments on multiple models with different training strategies. We remove the MSB branch and train the LSB branch with I_{LSB} as inputs from the scratch to get the SPLUT-LSB model. In this model, I_{MSB} is upsampled by nearest-neighbor interpolation and is added to the network output to form the SR results. We remove the LSB branch and train the MSB branch with I_{MSB} as inputs to get the SPLUT-MSB model. We show the corresponding results in Table 2. In the table, "Interpolation" represents the results of implementing nearest-neighbor interpolation on I_{MSB}. We see the results

Table 2. Comparison of different training strategies. The results show that jointly training the two parallel branches achieves the best SR performance.

Layer num	Set5		Set14		BSDS100		Urban100		Manga109	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Interpolation	26.26	0.7380	24.75	0.6551	25.04	0.6299	22.17	0.6160	23.43	0.7415
SPLUT-LSB	26.26	0.7384	24.80	0.6568	25.04	0.6305	22.19	0.6175	23.49	0.7438
SPLUT-MSB	29.54	0.8356	26.85	0.7217	26.34	0.6793	23.90	0.6907	26.56	0.8316
SPLUT-ST	29.54	0.8379	26.92	0.7296	26.49	0.6912	23.92	0.6907	26.64	0.8288
SPLUT	**30.23**	**0.8567**	**27.32**	**0.7460**	**26.74**	**0.7044**	**24.21**	**0.7094**	**27.20**	**0.8478**

of SPLUT-LSB are very close to the nearest-neighbor interpolation. With only I_{LSB} and the interpolated I_{MSB}, the network cannot obtain local semantic information and can only generate some meaningless details. SPLUT-MSB performs much better than SPLUT-LSB and nearest-neighbor interpolation. However, it still has a significant accuracy degradation compared to the original SPLUT model. The reason may be that the information from I_{LSB} is lost and there are fewer input patterns compared to the full-precision LR images, which aggravates the one-to-many ill-posed nature of SR. Furthermore, we design a SPLUT-ST model whose LSB branch and MSB branch are separately trained (ST). We first train a model of SPLUT-MSB. Then we fix the parameters of the MSB branch and train the LSB branch. The final results show that the performance of SPLUT-ST is only slightly better than that of SPLUT-MSB, which is similar to the comparison between nearest-neighbor interpolation and SPLUT-LSB. This indicates that the separate training of the LSB branch based on the interpolated I_{MSB} and the pre-trained model of SPLUT-MSB both fail to utilize the information from I_{LSB}. The comparisons demonstrate that only the jointly training strategy can fully exploit the information of the two branches and boost reconstruction accuracy.

References

1. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016) 1
2. Jo, Y., Kim, S.J.: Practical single-image super-resolution using look-up table. In: CVPR. pp. 691–700 (2021) 2
3. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR. pp. 1874–1883 (2016) 1