Supplementary Material: Lightweight Image Super-Resolution with Scale-wise Network

Xiaole Zhao*
zxlation@foxmail.com
Xinkun Wu
wulisinerti@gmail.com

School of Computing and
Artificial Intelligence,
SWJTU, Chengdu, Sichuan, China

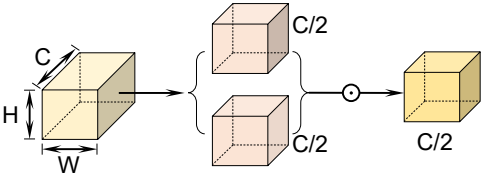

Figure 1: Diagram of ChunkGate unit (CG). The intermediate feature is evenly divided into 2 sub-features along the channel direction, which are fused together by simple element-wise product $\odot$. The operation halves the number of channels for the feature.

1 Nonlinear Activation Function

Nonlinear Activation Function While gaussian error linear unit (GeLU)[5] has gained popularity in computer vision, we aim to investigate whether its performance can be enhanced while retaining the same number of parameters, or if it can be simplified without compromising its performance. To answer these questions, we examine some of the latest state-of-the-art (SOTA) methods, such as [6, 8, 12, 13]. Our analysis reveals that all of these methods use gated linear units (GLU)[1].

The usual gated linear unit can be expressed as:

$$Gate(\mathbf{X}, f, g, \varphi) = f(\mathbf{X}) \odot \varphi(g(\mathbf{X})) \tag{1}$$

Where $\mathbf{X}$ represents feature mapping, which undergoes linear transformations, φ is a nonlinear activation function such as rectified linear unit (ReLU)[9] or Sigmoid, and $\odot$ denotes element-wise multiplication. While incorporating GLU into our network structure may enhance performance, it also leads to increased intra-block complexity, which is not desirable. To address this issue, we re-examine the activation function used in the network structure, specifically GeLU:

$$GeLU(x) = x\Phi(x) \tag{2}$$

*Corresponding author.

Table 1: Effects of skip connections (SCs) in ChunkGate blocks and groups (CGBs, CGGs) measured on Urban100 with $\times 3$. The best result is highlighted.

Config	1	2	3	4
SCs in CGBs	×	✓	×	✓
SCs in CGGs	×	×	✓	✓
#Params	**975K**	1023K	987K	1068K
PSNR	28.38	28.52	28.50	**28.63**

Where Φ is the cumulative distribution function of the standard normal distribution. And based on [5], it is suggested that the GeLU can be approximated and implemented by:

$$0.5x(1+tanh[\sqrt{\frac{2}{\pi}}(x+0.044715x^3)]) \tag{3}$$

From Eqn.1 and Eqn.2, it is evident that GeLU is a specific instance of GLU, where f and g are identity functions, and φ is replaced by Φ. This similarity leads us to speculate that GLU can be viewed as a generalization of the activation function and could potentially replace the nonlinear activation functions. Moreover, we observe that GLU itself contains nonlinearity and does not rely on φ, the formula $Gate(\mathbf{X}) = f(\mathbf{X}) \odot g(\mathbf{X})$ contains nonlinearity even if φ is removed. Based on these observations, we propose a simplified version of GLU, called ChunkGate, which directly divides the feature map into two parts along the channel dimension and multiplies them, as illustrated in Fig.1. This modification aims to reduce intra-block complexity and achieve better performance while maintaining the same number of parameters.

In contrast to the complex implementation of Eqn.3, our ChunkGate can be implemented through element-wise multiplication:

$$ChunkGate(\mathbf{X},\mathbf{Y}) = \mathbf{X} \odot \mathbf{Y} \tag{4}$$

Where $\mathbf{X}$ and $\mathbf{Y}$ are feature graphs of the same size.

By replacing GeLU in the baseline with the proposed ChunkGate, the performance of Single image super resolution (SISR) is significantly improved. The results clearly indicate that GeLU can be effectively replaced by our proposed ChunkGate. As a result, only a few types of nonlinear activations, such as Sigmoid and ReLU in the attention module, remain in the network.

2 Ablation Studies

The proposed method's performance behavior was investigated by analyzing the effects of skip connections within the locally dense and globally dense groups of SwiseNet and the effects of ChunkGate and Scale-wise upsample module (SUM).

2.1 Feature extraction.

Table 1 presents the ablation study on the impact of skip connections (SCs) in gated cell groups (CGG) and gated cell units (CGB) and their effect on the performance of the proposed

Table 2: Average PSNR to show the performance of SwiseNet with different interpolation methods. The test dataset is Set5. Best results are highlighted.

Experiment	Parameters	Scales		
		×2	×3	×4
Pixelshuffle	1035K	38.10	34.43	32.31
SUM-bilinear	1077K	38.16	34.59	32.39
SUM-bicubic	1077K	**38.18**	**34.61**	**32.43**

Table 3: Average PSNR to show the performance of SwiseNet across different scale groups. The test dataset is Set5. Best results are highlighted.

Scale-wise factor	Parameters	Scales		
		×2	×3	×4
×(N, N+1, N+2)	1102K	38.13	34.57	32.37
×(N-1, N, N+1)	1086K	38.12	34.56	32.38
×(N, N+1)	1063K	38.16	34.59	32.39
×(N, N+1, 1/2)	1077K	**38.18**	**34.61**	**32.43**

method. In this study, SCs are composed of conjunctions and 1×1 convolution, and the exclusion of SCs using 1×1 convolution has resulted in a slight variation in the number of parameters among the columns.

The results demonstrate that using SCs only in CGG outperforms the model that does not use SCs. This is because the short connections within CGG efficiently carry information from the middle to the high level, enabling the model to better utilize multilevel representations by collecting all features before upgrading the module.

However, it has been suggested in [4] that multiplication operations on shortcut connections, such as 1×1 convolution, may hinder information propagation and complicate optimization. Therefore, when all SCs are deactivated, performance degradation can be expected. This is because SCs simplify information propagation while learning local connections. Therefore, the study found that SwiseNet performs better than all three models when it uses SCs in both CGB and CGG.

In addition, the use of SCs in CGB provides a more effective way to learn local features by enhancing the flow of information between adjacent blocks. This enables the model to capture more fine-grained features, which is particularly useful in tasks that require high-resolution images. Therefore, the combination of SCs in both CGB and CGG in SwiseNet allows for the efficient propagation of information both globally and locally, leading to improved performance in image super-resolution tasks.

The use of SCs in CGG enables the transmission of information globally, while the flow of information in CGB is combined with the flow of information from global connections. This approach facilitates the transmission of information through multiple shortcuts, which mitigates the issue of disappearing gradients. Specifically, in CGG, SCs take advantage of the multi-layered representation to facilitate the spread of information to higher levels.

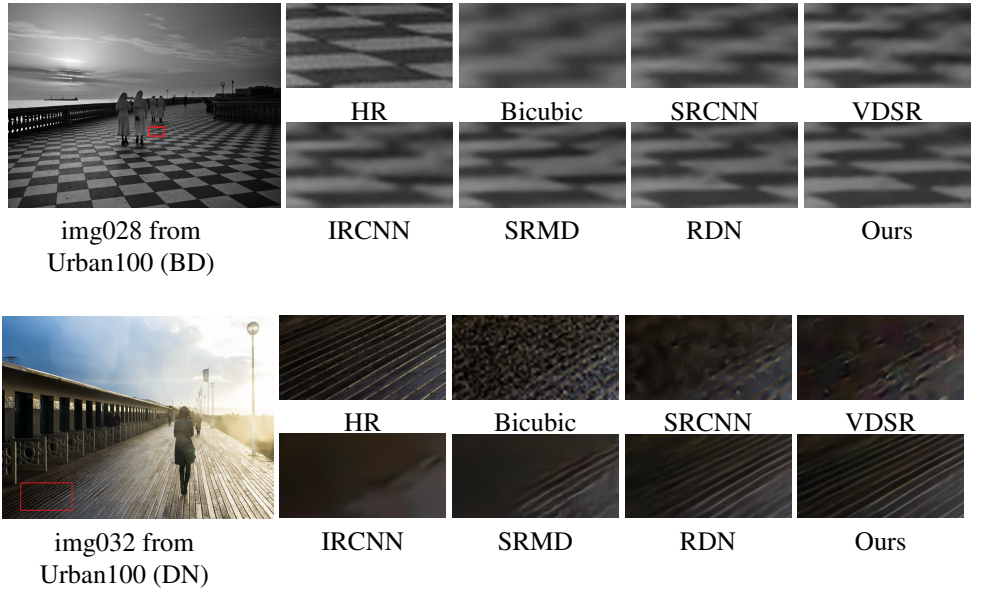

Figure 2: Visual results of BD and DN degradation model with scale factor $\times 3$.

2.2 SUM cross-scale effect.

This paper analyzes the advantages of introducing SUM module and discusses the influence of different interpolation methods on reconstruction. We carried out the following experiments :(i) generate images directly using pixelshuffle without multi-scale feature mapping; (ii) Using bilinear interpolation to upscale and downscale multi-scale feature images generated by pixelshuffle; (iii) Perform the same operation as (ii) using bicubic interpolation. As shown in Table 2, the multi-scale method proposed in this paper is applied and good results are obtained. These experiments show that, contrary to the usual practice in the field, the addition of multi-scale modules greatly improves the reconstruction accuracy. Compared with bilinear interpolation, the best result can be obtained by combining SUM with bicubic interpolation.

2.3 Multi-scale generalization.

The purpose of this section is to investigate the generalization ability of our architecture across different scales. In our architecture, the multi-scale factors are always N+1, N, and 1/2 when the maximum scale is N, as explained in Section 3.2 of the main text. To test the generalization ability, we trained models for N $\in$ 2,3,4 and evaluated them on different scales. The results in Table 3 show that models trained for multi-scale produced better PSNR scores for all scale factors, indicating that our architecture can generalize across scales without the need for specialized models for each scale.

Additionally, the cost of additional parameters is low, since $\times 4$ and $\times 8$ consist of multiple consecutive $\times 2$ operations, resulting in fewer parameters introduced. Although selecting factors at multiple scales to higher scales slightly improves PSNR, it comes at the cost of

Table 4: Average PSNR/SSIM for models with the same order of magnitude of parameters (RDN included as a high-capacity reference model). Scores shown for scale factor ×3 using BD and DN degradation models. The best results are highlighted in red color and the second best is in blue.

Dataset	Model	Bicubic	SPMSR[22]	SRCNN[2]	FSRCNN[3]	VDSR[7]	SRMD(NF)[23]	RDN[24]	SwiseNet
Set5	**BD**	28.34/0.8161	32.21/0.9001	31.75/0.8988	26.25/0.8130	33.78/0.9198	34.09/0.9242	34.58/0.9280	34.68/0.9285
	DN	24.14/0.5445	-	28.10/0.7783	24.24/0.6992	27.81/0.7901	27.74/0.8026	28.47/0.8151	28.52/0.8168
Set14	**BD**	26.12/0.7106	28.97/0.8205	28.72/0.8024	25.63/0.7312	29.90/0.8369	30.11/0.8304	30.53/0.8447	30.53/0.8442
	DN	23.14/0.4828	-	25.55/0.6610	23.10/0.5869	25.92/0.6786	26.13/0.6974	26.60/0.7101	26.64/0.7118
BSD100	**BD**	26.02/0.6733	28.13/0.7740	27.97/0.7921	24.88/0.6850	28.70/0.8003	28.98/0.8009	29.23/0.8079	29.24/0.8081
	DN	22.94/0.4461	-	25.31/0.6351	23.70/0.5856	25.60/0.6455	25.64/0.6495	25.93/0.6573	25.96/0.6612
Urban100	**BD**	23.20/0.6661	25.84/0.7856	25.50/0.7812	22.14/0.6815	26.80/0.8191	27.50/0.8370	28.46/0.8582	28.49/0.8580
	DN	21.63/0.4701	-	23.40/0.6590	21.15/0.5682	24.01/0.6802	24.28/0.7092	24.92/0.7364	25.01/0.7424

more computation. Therefore, for the remaining experiments, we will extend to N+1 and 1/2, as it still provides a significant improvement at a slightly higher computational cost.

2.4 Results with BD and DN degradation models

To comprehensively evaluate the effectiveness of the proposed method, we utilized three degradation models to simulate LR images, as described in [14, 15, 16]. The first model, denoted as BI, generates LR images by bicubic downsampling the ground truth HR images with scaling factors of ×2, ×3, and ×4. This has already been discussed in the main text. The second model, denoted as **BD**, involves bicubic downsampling of HR images by a factor of ×3, followed by blurring the images using a Gaussian kernel with a size of 7×7 and a standard deviation of 1.6. Lastly, the third challenging model, denoted as **DN**, produces LR images by performing bicubic downsampling followed by the addition of additive Gaussian noise with a noise level of 30.

Following the methodology of Li et al. [16], we present the evaluation results obtained by applying BD and DN degradation models and compare them with six existing SR methods [2, 3, 7, 10, 11, 16]. Additionally, we include the RDN [16] model as a reference. To ensure a fair comparison, we retrained the SRCNN [2], FSRCNN [3], and VDSR [7] models for BD and DN degradations to consider the mismatches in degradation settings. The evaluation metrics, including PSNR and SSIM scores, along with the number of model parameters and Multi-Adds operations, are summarized in Table 4. We note that SwiseNet performs worse than RDN in some BD datasets, but better in DN datasets due to the benefits of the Scale-wise Upsample Module (SUM) in SwiseNet. SUM helps to reduce DN degradation and obtain better results than RDN with only 1.1M parameters, whereas RDN has 22M parameters.

In Fig.2, we present two sets of visual results obtained using the **BD** and **DN** degradation models from standard benchmark datasets. With the **BD** degradation model, other methods failed to eliminate blurring artifacts. In contrast, SwiseNet successfully mitigated distortions and generated SR images with improved accuracy and finer details. As for the **DN** degradation model, we observed that recovering details was challenging for other methods. However, our proposed method demonstrated good performance by effectively reducing noise and enhancing details in the reconstructed images.

References

[1] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017.

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014.

[3] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 391–407. Springer, 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[5] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[6] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In *International Conference on Machine Learning*, pages 9099–9117. PMLR, 2022.

[7] 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.

[8] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022.

[9] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[10] Tomer Peleg and Michael Elad. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE transactions on image processing*, 23(6):2569–2582, 2014.

[11] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE international conference on computer vision*, pages 4799–4807, 2017.

[12] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5769–5780, 2022.

[13] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022.

[14] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

[15] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3262–3271, 2018.

[16] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.