SL^{2}A-INR: Single-Layer Learnable Activation for Implicit Neural Representation

Supplementary Material

This supplementary material presents further details, including the activation function visualization, the specific hyperparameter values, the reasoning behind the proposed method, an analysis of the computational complexity of SL^{2}A, and additional applications of our approach to single image super-resolution.

1. Activation Function Visualization

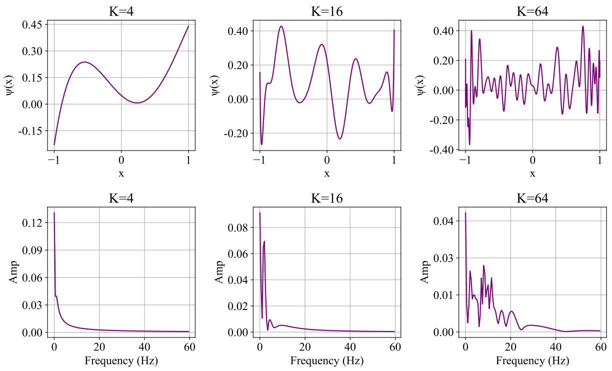

Figure 1. Frequency spectrum of our learnable activation block $\Psi(\mathbf{x})$.

Figure 1 shows the frequency spectrum of learnable activation block ($\Psi(\mathbf{x})$) across different values of K. The top row illustrates the function $\psi(x)$ in the time domain, while the bottom row shows its amplitude spectrum in the frequency domain. As K increases, the basis functions are capable of covering a wider frequency and bandwidth range and preserving high-frequency components, which is crucial for accurate INRs. By expanding the frequency coverage, SL^{2}A enhances the capacity to learn complex, high-frequency patterns effectively, addressing spectral bias problem.

2. SL^{2}A Design Choice

2.1. Chebyshev Polynomial.

Our architecture leverages Chebyshev polynomials to parameterize learnable functions due to their well-established advantages in numerical approximation. These polynomials exhibit superior convergence properties, numerical stability, and orthogonality, making them highly effective for function approximation [5]. One key motivation for using Chebyshev polynomials over B-splines is their ability to efficiently approximate activation functions such as ReLU. The minimax property of Chebyshev polynomials ensures that they minimize the maximum error in polynomial approximations, leading to higher accuracy with fewer parameters [4, 5]. Additionally, Chebyshev polynomials provide strong spectral approximation capabilities, making them well-suited for capturing high-frequency components of functions [1]. Compared to the B-splines used in KANs, which rely on grid-based representations and suffer from inefficiencies in deeper layers, Chebyshev polynomials offer a more flexible and efficient alternative. Prior research [6, 8], has demonstrated that integrating Chebyshev polynomials into KANs enhances efficiency, and our experiments further confirm that using them in KAN architecture leads to improved performance, as evidenced in Tab. 1. While integrating Chebyshev polynomials into KANs improves efficiency and performance, the result is still suboptimal. Our approach mitigates the inefficiencies of using KANs in all layers by leveraging Chebyshev-based activation learning in earlier layers while propagating rich representations through skip connections. This hybrid approach enhances overall performance compared to standard KAN architectures. Thus, the decision to use Chebyshev polynomials instead of B-splines (or other polynomials) is driven by their superior approximation properties, stability, and efficiency, ultimately leading to improved performance in our proposed architecture.

2.2. ReLU Layer.

We analyze the impact of the ReLU activation in our architecture by comparing models with and without ReLU across different ranks of linear layers and polynomial degree configurations. Specifically, alternative formulations of Equation (4) in the paper could remove ReLU, leading to:

$$\begin{aligned} \mathbf{z}_1 &= \Psi(\mathbf{x}), \\ \mathbf{z}_l &= \mathbf{W}_l(\mathbf{z}_{l-1} \odot \mathbf{z}_1) + \mathbf{b}_l, \quad l = 2, 3, \ldots, L-1, \\ f_\theta(\mathbf{x}) &= \mathbf{W}_L(\mathbf{z}_{L-1} \odot \mathbf{z}_1) + \mathbf{b}_L. \end{aligned} \tag{1}$$

Table 1. Comparison of PSNR of SL^{2}A and KAN methods for image approximation (image 00).

Method	#Params (M)	Time (min.)	Size (MB)	PSNR	SSIM
KAN (B-Spline)	0.329	210.1	0.93	25.40	0.722
KAN (Chebyshev)	**0.203**	4.27	**0.78**	30.50	0.845
SL^{2}A	0.330	**0.77**	0.93	**33.40**	**0.892**

As shown in Fig. 2, removing ReLU results in a consistent PSNR drop, with more severe degradation observed in lower-degree Chebyshev expansions, highlighting the critical role of ReLU in enhancing the model's expressive power to capture complex signal representations. The largest performance drop is 6.22 dB in lower-degree configurations, demonstrating its critical importance, Even in higher-degree settings, ReLU continues to provide improvements, reinforcing its necessity for expressivity. These results confirm that the modulation of layers by learnable activation $\Psi(\mathbf{x})$ could be insufficient and introducing nonlinearity such as ReLU improves performance and expressive power.

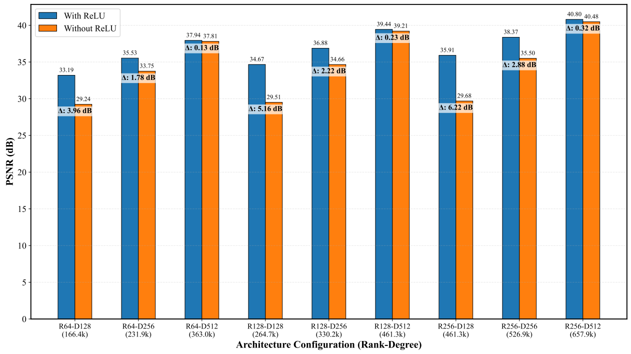

Figure 2. Comparison of PSNR (dB) across different rank-degree configurations with and without ReLU activation. Removing ReLU results in a consistent performance drop, particularly in lower-degree Chebyshev expansions, highlighting its importance in maintaining stability and expressivity.

3. Chebyshev Degree/ Linear layer rank Efeect

Figures 3 and 4 illustrate the impact of rank of linear layers and the polynomial degree on PSNR for image representation tasks. As shown Fig. 3, increasing both the Chebyshev polynomial degree and rank of linear layers leads to higher PSNR values, demonstrating the effectiveness of higher-degree representations in capturing finer details. More importantly, the results reveal a critical trade-off: under similar model size or parameter constraints, increasing the polynomial degree in the initial layer generally yields better accuracy than using higher-rank configurations with a lower degree in the image representation task. For instance, the blue point on the right (363k parameters) achieves superior PSNR than the orange point in the middle (330k parameters) despite having fewer rank, and similar trends hold across other configurations. This validates our architectural choice of employing lower-rank linear layers in deeper layers while allowing a higher polynomial degree in the initial layer, enabling efficient representations without excessive parameter growth. Figure 4 further supports this intuition by presenting the same analysis without ReLU activations. The observed trends remain consistent, reinforcing that altering the rank of linear layers alone is less effective in improving performance when ReLU is removed. This highlights the complementary role of non-linear activation functions and low-rank layers in achieving a balance between efficiency and expressivity in our proposed architecture.

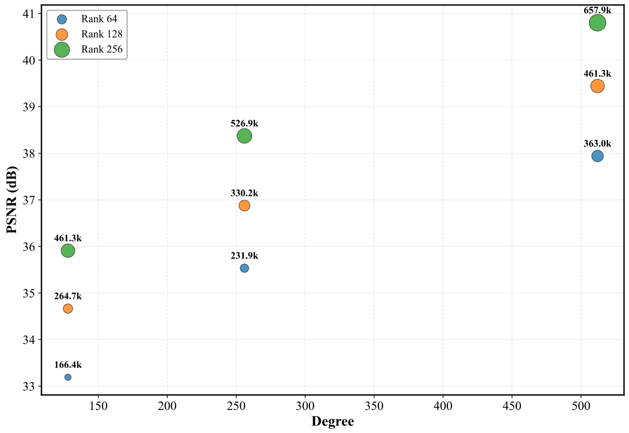

Figure 3. PSNR comparison across different rank-degree configurations and model sizes. Higher polynomial degrees lead to better performance, particularly when using lower-rank linear layers in deeper stages, supporting our architectural design choice.

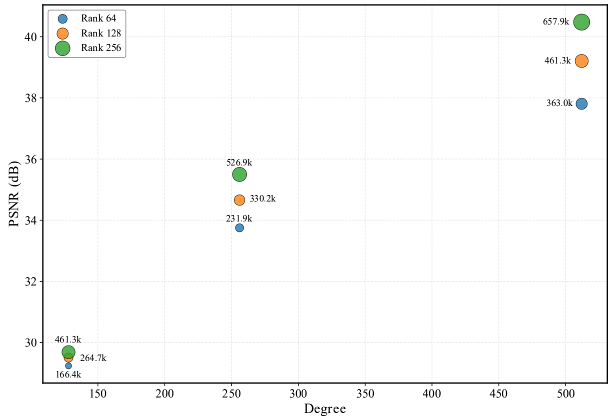

Figure 4. PSNR comparison without ReLU activation. The trend remains consistent, showing that modifying rank alone has limited effect on performance, further highlighting the importance of structured polynomial expansions.

4. Hyperparameter Settings

In accordance with the configurations used in prior studies, we employed the hyperparameter settings summarized in Tab. 2 for the image fitting task. Furthermore, we adopted the same initialization scheme as in these works, utilizing the publicly available implementation provided by the authors.

Table 2. Configuration details of different models used in the experiments.

Model	K	ω_0	s_0	Layers	Hidden Features
Finer	-	30	-	5	256
Wire	-	20	30	4	300
Gauss	-	-	30	5	256
Siren	-	30	-	5	256
ReLU+P.E.	-	-	-	5	256
SL^{2}A	256	-	-	5	256

5. Computational Complexity

We analyze the computational complexity in terms of the number of parameters, and training time (per image/SDF), across different tasks, as summarized in Tabs. 3 and 4. Although our method, SL^{2}A, utilizes more parameters compared to FINER (the best-performing method after ours), it does not require extensive training time. It is important to note that the reported training times correspond to a fixed number of iterations using optimal hyperparameters that maximize performance (PSNR) rather than configurations optimized for speed. Time differences between methods arise from their respective optimal hyperparameter settings. Our primary contribution is the elimination of the need for manual activation function design, in contrast to prior approaches. Moreover, as illustrated in Tab. 1, existing methods utilizing naive learnable activations, such as KAN [3], suffer from limited scalability and demand substantial training resources. We effectively address these limitations by introducing a scalable learnable activation paradigm that requires only marginally increased parameter counts while maintaining computational efficiency through the use of a single learnable activation layer and low-rank MLPs, which minimize overhead without sacrificing performance. Future research will explore further enhancements aimed at improving network compactness. In the image representation task (Tab. 3), SL^{2}A achieves competitive training time, considerably outperforming many existing methods, and remains efficient despite its marginally higher parameter count. For the occupancy representation task (Tab. 4), SL^{2}A requires slightly longer training times compared to FINER but still maintains efficiency within acceptable bounds. Overall, the complexity analysis indicates that SL^{2}A remains computationally efficient across tasks, with slight trade-offs in training time balanced by its superior performance in representation quality.

6. Single Image Super Resolution

To demonstrate the generalization capability of our approach, we evaluated it on the task of single image super-resolution using an image from the DIV2K dataset [9]. Specifically, we trained our architecture for image representation and downsampled the original image (with dimensions $1356 \times 2040 \times 3$) by factors of 2, 4, and 6 and evaluated it on the original resolution. The corresponding results, illustrated in Fig. 5, compare super-resolution reconstructions of a parrot image across multiple ratios. Our findings indicate that SL^{2}A-INR consistently surpasses FINER, achieving superior PSNR [2] metrics. Furthermore, SL^{2}A-INR distinctly preserves sharp details and produces less noisy result images, specifically in 6× settings, whereas the recent SOTA method FINER produces comparatively noisier results. These visual and quantitative improvements highlight our method's effectiveness in achieving high-quality super-resolution reconstructions and underscore its robustness and adaptability to inverse problems. We hypothesize that this trend could similarly hold for other inverse problem tasks involving complex reconstructions, a direction we plan to explore in future work.

7. Initialization Scheme

As previously mentioned, we employed the Xavier uniform initialization scheme for our method. Nevertheless, our approach (SL^{2}A) demonstrates robustness to various initialization schemes. To substantiate this claim, we evaluated our image representation (Image 00) using multiple initialization methods, as summarized in Tab. 5. The results indicate minimal variations in performance across different initialization schemes, confirming the robustness of our method. This characteristic contrasts markedly with typical

Table 3. Comparison of computational complexity across different methods for the Image Representation Task.

Method	#Params (M)↓	Training Time (min.) ↓
FINER	0.198	**0.595**
WIRE	**0.099**	1.713
Gauss	0.198	3.08
SIREN	0.198	0.643
ReLU+P.E.	0.204	3.425
SL^{2}A	0.330	0.77

Table 4. Computational complexity across different methods for the Occupancy Representation Task.

Method	#Params (M)↓	Training Time (min.) ↓
FINER	0.198	42.41
WIRE	**0.066**	63.03
Gauss	0.198	32.05
SIREN	0.198	**27.22**
ReLU+P.E.	0.214	39.32
SL^{2}A	0.248	45.59

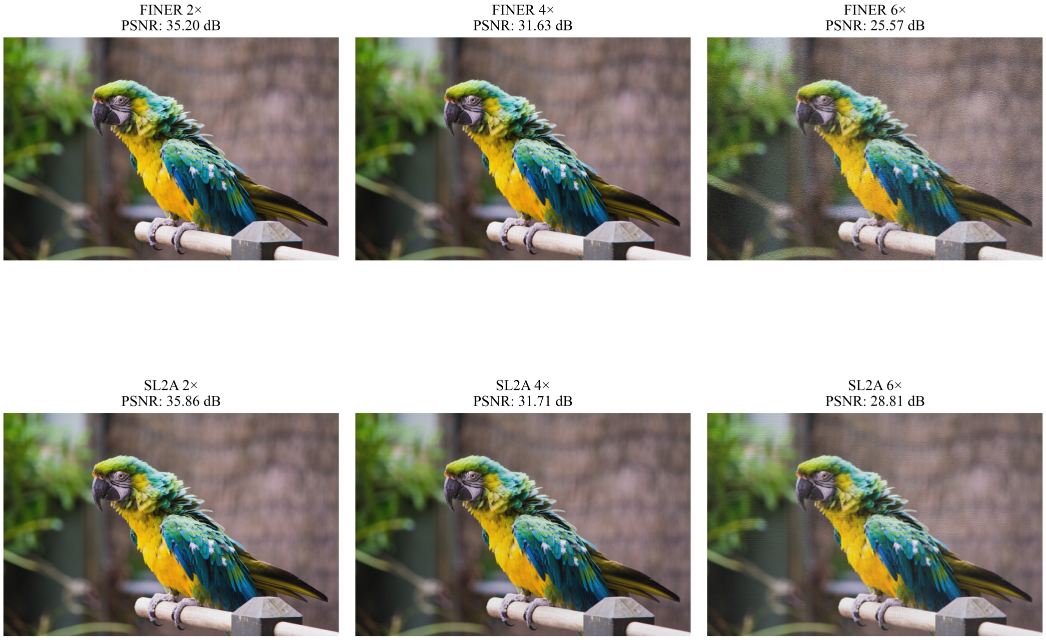

Figure 5. Results for a single image super resolution compared with FINER.

INR methods, such as SIREN [7], whose performance significantly depends on the chosen initialization scheme, often experiencing notable degradation under suboptimal initialization.

Table 5. Comparison of various initialization schemes evaluated on Image 00. Results demonstrate robustness of SL^{2}A, showing minimal variation across methods, except for the uniform initialization.

Initialization Scheme	PSNR
Xavier uniform	33.40
Kaiming uniform	33.22
Kaiming normal	33.12
Orthogonal	33.26
Uniform	31.86
Normal	33.28

References

[1] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[2] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.

[3] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

[4] John C Mason and David C Handscomb. *Chebyshev polynomials*. Chapman and Hall/CRC, 2002.

[5] Theodore J Rivlin et al. The chebyshev polynomials, pure and applied mathematics, 1974.

[6] Khemraj Shukla, Juan Diego Toscano, Zhicheng Wang, Zongren Zou, and George Em Karniadakis. A comprehensive and fair comparison between mlp and kan representations for differential equations and operator networks. *arXiv preprint arXiv:2406.02917*, 2024.

[7] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[8] Sidharth SS, Keerthana AR, Anas KP, et al. Chebyshev polynomial-based kolmogorov-arnold networks: An efficient architecture for nonlinear function approximation. *arXiv preprint arXiv:2405.07200*, 2024.

[9] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition workshops, pages 114–125, 2017.