Parameter-Efficient Cross-Layer Feature Fusion via Chebyshev Polynomial Unit

***Abstract*—Cross-layer feature fusion critically influences the performance of deep neural networks, where mainstream approaches like additive fusion of residual connections struggle to model high-order nonlinear interactions, limiting representational capacity, while concatenative fusion of dense connections incurs significant memory and computational overhead. Although recent attention-based feature fusion methods refine feature representations and strengthen hierarchical interactions, they often struggle to balance expressiveness and efficiency. To address these limitations, we propose the Chebyshev Fusion Unit (CFU), a lightweight yet effective cross-layer fusion mechanism. Specifically, CFU computes high-order Chebyshev polynomial terms between residual features and current-layer features, each of which is aggregated through learnable scalar weights, forming enhanced fusion features. This design enables explicit modeling of complex cross-layer dependencies with minimal additional parameters. Extensive experiments across various commonly used base models on image classification, medical image segmentation, and physical law learning tasks demonstrate its superior performance.**

***Index Terms*—feature fusion, Chebyshev polynomial, classification, segmentation, physical law learning.**

I. Introduction

Deep neural networks (DNNs) have achieved remarkable success across diverse domains such as computer vision [1]–[3], natural language processing [4], [5], and scientific computing [6], [7], yet their performance is fundamentally constrained by the design of cross-layer feature fusion mechanisms. Traditional approaches, including additive shortcuts in ResNets [2] and concatenated connections in DenseNets [3], exhibit complementary limitations, where the former struggles to model high-order nonlinear interactions due to its linear residual learning paradigm [8], while the latter introduces substantial memory and computational overhead through feature dimension expansion [9]. To address these limitations, attention-based feature fusion methods enable adaptive learning of hierarchical relationships by dynamically adjusting fusion weights through interdependencies between features. Self-attention and multi-head attention mechanisms [5] construct global feature interactions through inner products among features. Recent attentional feature fusion (AFF) [10] and multi-scale spatial attention modules [11] leverage contextual information to refine feature representations. However, these approaches often face a trade-off between expressiveness and efficiency: while they capture long-range dependencies, their quadratic complexity significantly limits scalability.

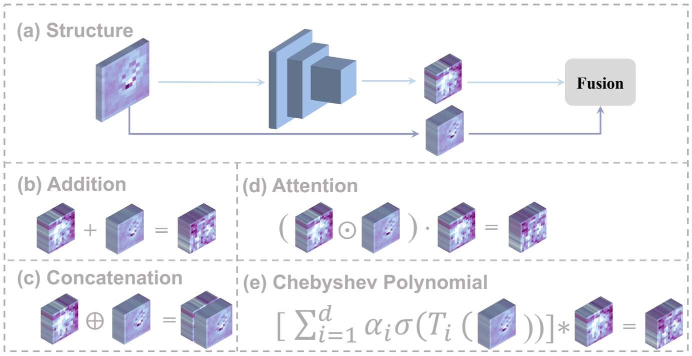


Fig. 1. (a) Overall structures with fusion, where the color mixing blocks indicate intermediate features. (b) The additive feature fusion. (c) The concatenation feature fusion, where $\oplus$ serves as the concatenation operator. (d) The attention-based feature fusion, where $\odot$ indicates inner product. (e) The proposed Chebyshev polynomial fusion, where $*$ indicates the Hadamard product, σ denotes the regularization transformation, α serves as a learnable scalar, and $T_i(\cdot)$ is the Chebyshev polynomial expansion.

To transcend the expressiveness-efficiency trade-off inherent in existing fusion paradigms, we propose the **Chebyshev Fusion Unit (CFU)**, a theoretically grounded cross-layer fusion mechanism that leverages the orthogonal and recursive properties of Chebyshev polynomials to model high-order nonlinear interactions with strict parameter efficiency, as shown in Fig. 1. Unlike heuristic attention modules or rigid additive shortcuts, CFU decomposes feature fusion into hierarchical orthogonal expansions, where dynamically weighted polynomial terms capture complex dependencies between residual and current-layer features through spectrally optimized aggregation. Theoretical derivations guarantee that this design enhances feature interactions, complexity analysis ensures its parameter and computational efficiency, and extensive experiments demonstrate its Pareto optimal balance between expressiveness and efficiency.

Our main contributions are summarized below:

- We introduce Chebyshev Fusion Unit (CFU), a novel feature fusion method that leverages Chebyshev polynomial connections to improve cross-layer interactions, thereby enhancing the network's representational capacity.
- We theoretically prove that CFU achieves stronger feature interactions than additive fusion from the perspective of the Hilbert-Schmidt Independence Criterion.
- We provide comprehensive empirical evidence across a range of tasks, demonstrating that CFU consistently surpasses commonly used feature fusion methods.

II. Related Work

Attention-based feature fusion has emerged as a critical paradigm to address the limitations of traditional additive and concatenative operations in hierarchical feature integration. This kind of feature fusion method starts from self-attention and multi-head-attention mechanisms [5], which construct global feature interactions through inner products between features. Latter attempts like SENet [12] introduced channel-wise gating mechanisms to recalibrate feature importance but ignored spatial and scale inconsistencies, leading to suboptimal fusion in multi-scale scenarios. Subsequent works such as SKNet [13] and ResNeSt [14] extended dynamic feature selection to intra-layer contexts by leveraging attention weights for multi-branch feature aggregation, though these methods remained confined to single-scale feature interactions. To unify cross-layer fusion, Attentional Feature Fusion (AFF) [10] and multi-scale channel attention module (MS-CAM) [11] are proposed to adaptively fuse features from short/long skip connections via iterative refinement (iAFF) [15], establishing a unified framework for hierarchical feature integration across layers and skip connections. Further innovations like POSTER [16] constructed a pyramid cross-fusion attention fusion method to maximize proper attention to salient regions. Despite these advances, existing attention-based fusion methods face critical computational and memory challenges.

III. Methodology

In this section, we introduce the design of Chebyshev Fusion Unit (CFU). We provide the fundamental mathematical properties of Chebyshev polynomials in Section III-A, and illustrate the specific structure of CFU in Section III-B, and analyze the interaction strength and computational complexity in Section III-C.

A. Preliminaries: Chebyshev polynomials

Chebyshev polynomials have great potential for application in DNNs due to the best uniform approximation for continuous functions [17] and the tightest upper and lower bounds compared to all other polynomials on the interval [-1, 1]. The former property enables them to effectively capture complex patterns in data, while the latter intrinsically ensures numerical stability. The most commonly used forms are the trigonometric definition and recursive definition, as shown in Eq. (1) and Eq. (2), respectively:

$$T_n(x) = \begin{cases} \cos(n \arccos x), & |x| \le 1 \\ \cosh(n\mathrm{arccosh} x), & x > 1 \\ (-1)^n \cosh(n\mathrm{arccosh}(-x)), & x < -1 \end{cases} \tag{1}$$

$$\begin{cases} T_0(x) = 1, \quad T_1(x) = x \\ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \ge 1 \end{cases} \tag{2}$$

B. Chebyshev Fusion Unit

We propose the Chebyshev Fusion Unit (CFU), a lightweight yet theoretically grounded approach that leverages the recursive and orthogonal properties of Chebyshev polynomials to explicitly model high-order nonlinear interactions while maintaining strict parameter efficiency. As shown in Fig. 1(e), CFU receives the shortcut feature and the current-layer feature as inputs, denoted as x and $f(x)$ for simplicity, where $f(\cdot)$ serves as the transformation of the neural network between them. Then the shortcut feature x recursively generates Chebyshev polynomial terms of various orders as intermediate features $T_0(x), \cdots T_d(x)$ according to Eq. (2), where the operation between feature vectors is replaced by the Hadamard product (element-wise multiplication), reformulated as Eq. (3):

$$\begin{cases} T_0(x) = 1, \qquad T_1(x) = x \\ T_{i+1} = 2x \circ T_i(x) - T_{i-1}(x), \quad n \geqslant 1 \end{cases} \tag{3}$$

where $\circ$ denotes the Hadamard product.

Then these intermediate features will pass through a regularization function σ to restrict their values. Then each regularized intermediate feature is multiplied by a scalar weight α and summed to obtain the high-order feature, which later aggregates with the current-layer feature through the Hadamard product to get the aggregated feature y, as denoted in Eq. (4):

$$\begin{aligned} y &= f(x) \circ [\alpha_0 \sigma(T_0(x)) + \alpha_1 \sigma(T_1(x)) + \cdots + \alpha_d \sigma(T_d(x))] \\ &= f(x) \circ \sum_{i=0}^{d} \alpha_i \sigma(T_i(x)) \end{aligned} \tag{4}$$

where $T_0(x)$ is an all-ones vector, meaning that $f(x) \circ \alpha_0 \sigma(T_0(x)) = C \cdot f(x)$ represents a preserved but controllable primary output of the current layer, leading to a stable training process.

C. Property Analysis

CFU enhances the complex feature interactions by weight-summing various orders of Chebyshev polynomial intermediate features and aggregating them with current-layer features. We propose and prove that it does enhance interaction gain from the perspective of Hilbert-Schmidt Independence Criterion (HSIC) [18], as illustrated in Proposition. 1:

Proposition 1: The interaction gain of polynomial fusion (PF) is larger compared to simple linear fusion (LF).

Proof

Consider the feature representations as elements in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with kernel K. The interaction degree is quantified by the Hilbert-Schmidt Independence Criterion (HSIC):

Let the input feature as x with its Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_x$ defined by the kernel function $k_x(\cdot,\cdot)$. Linear and polynomial fusion can be simplified as:

$$\begin{aligned} y_{\mathrm{LF}} &= g_{\mathrm{LF}}(x) = x + f(x) \\ y_{\mathrm{PF}} &= g_{\mathrm{PF}}(x) = x + \sum_{i=1}^{d} x^i f(x) \end{aligned} \tag{5}$$

The feature aggregation strength is measured by HSIC(x, y), defined as the squared Hilbert-Schmidt norm of the cross-covariance operator $\mathcal{C}_{x,y}$, as shown in Eq. (6)

$$\begin{aligned}\text{HSIC}(x, y) = \|\mathcal{C}_{x,y}\|^2_{\text{HS}} &= \mathbb{E}_{x,x',y,y'}[k_x(x, x')k_y(y, y')] \\ &+ \mathbb{E}_{x,x'}[k_x(x, x')]\mathbb{E}_{y,y'}[k_y(y, y')] \\ &- 2\mathbb{E}_{x,y}[\mathbb{E}_{x'}[k_x(x, x')]\mathbb{E}_{y'}[k_y(y, y')]]\end{aligned} \tag{6}$$

where k_x and k_y are feature kernels.

Intuitively, the higher-order terms $\sum_{i=1}^{d} x^i f(x)$ enable $g_{\text{PF}}(x)$ to span a higher-dimensional subspace in RKHS. Theoretically, since $g_{\text{PF}}(x) = g_{\text{LF}}(x) + \Delta g(x)$, where $\Delta g(x) = f(x) \circ \sum_{i=1}^{d} x^i$, and $\Delta g(x)$ is orthogonal to $g_{\text{LF}}(x)$ in $\mathcal{H}_y$ due to the polynomial basis orthogonality, thus:

$$\text{HSIC}(x, y_{\text{PF}}) = \|\mathcal{C}_{x,g_{\text{PF}}(x)}\|^2_{\text{HS}} = \|\mathcal{C}_{x,g_{\text{LF}}(x)} + \mathcal{C}_{x,\Delta g(x)}\|^2_{\text{HS}}.$$

By the triangle inequality and orthogonality of Hilbert-Schmidt norms, we get:

$$\|\mathcal{C}_{x,g_{\text{PF}}}\|_{\text{HS}} \geq \|\mathcal{C}_{x,g_{\text{LF}}}\|_{\text{HS}} + \|\mathcal{C}_{x,\Delta g}\|_{\text{HS}} - \epsilon > 0,$$

where the second inequality holds because ϵ serves as an upper bound on cross terms, which becomes negligible when k_x is universal and x is non-degenerate. Thus

$$\text{HSIC}(x, y_{\text{PF}}) > \text{HSIC}(x, y_{\text{LF}}).$$

□

Besides, CFU is parameter-efficient by expanding k-th order Chebyshev polynomials of the shortcut features, followed by a simple Sigmoid or Softmax operation and a Hadamard fusion with the current-layer feature, with only a polynomial order plus one $(d + 1)$ additional parameters per module. By exploiting the optimal approximation properties and numerical stability of Chebyshev polynomials, CFU ensures both theoretical rigor and practical scalability.

In this way, CFU achieves a Pareto optimal balance between constructing complex feature interactions and maintaining parameter efficiency.

IV. Experiments

To evaluate the effectiveness of CFU, we conduct comprehensive experiments across three major tasks: medical image segmentation, physical law learning, and image classification. Each experiment is designed to compare the performance of traditional neural network architectures with their CFU counterparts.

A. Medical Image segmentation

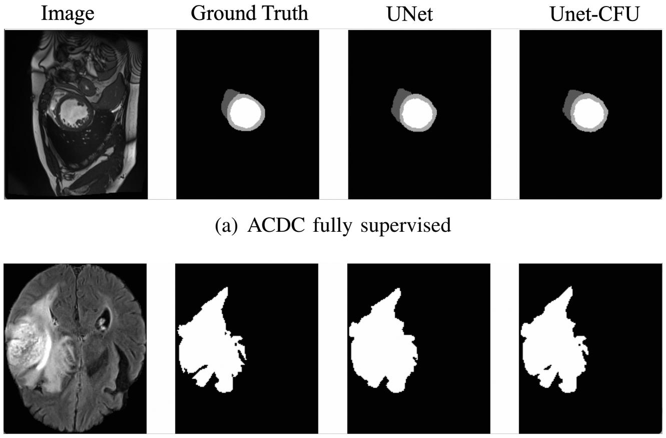
We conduct a series of experiments of medical image segmentation on two prominent datasets: ACDC (Automated Cardiac Diagnosis Challenge) [19] and BraTS19 (Brain Tumor Segmentation Challenge) [20]. To thoroughly assess performance across different settings, we implement a series of settings, including 2D fully supervised, 2D semi-supervised, and 3D fully supervised experiments. And we set UNet [21] as our baseline model, with concatenative feature fusion. We replace the concatenation fusion for CFU and evaluate the effectiveness among various polynomial orders.

TABLE I
The dice value of UNet and UNet-CFU on ACDC and BraTS19.

Order	1	2	3	4	5	6	7	8	9
ACDC 2D-fully-supervise with Baseline UNet: 0.7984									
UNet-CFU	**0.8122**	0.7900	**0.8070**	**0.8031**	0.7923	**0.8040**	**0.8077**	0.7896	**0.8013**
ACDC 2D-semi-supervise with Baseline UNet: 0.8225									
UNet-CFU	**0.8305**	**0.8270**	0.8205	**0.8320**	**0.8328**	**0.8243**	**0.8338**	**0.8328**	**0.8339**
BraTS19 3D-fully-supervise with Baseline UNet: 0.8291									
UNet-CFU	**0.8306**	**0.8389**	**0.8415**	**0.8448**	**0.8404**	0.8279	**0.8324**	**0.8417**	**0.8365**

The Dice scores of both UNet and UNet-CFU of various polynomial orders are presented in Tab. I. The results indicate that UNet-CFU consistently outperforms the traditional UNet architecture across all experimental configurations, as measured by the Dice metric.

We randomly select an image from the test set for visualization, with the results depicted in Fig. 2, exhibiting the power of UNet-CFU to identify and extract tiny objects and marginals, further highlighting the effectiveness of CFU in enhancing the model's representational power, enabling better feature extraction and segmentation accuracy.



(a) ACDC fully supervised

(b) BraTS19 fully supervised

Fig. 2. The segmentation results of ACDC and BraTS19. (a) ACDC for fully supervised task. (b) BraTS19 for fully supervised task.

B. Physical law learning

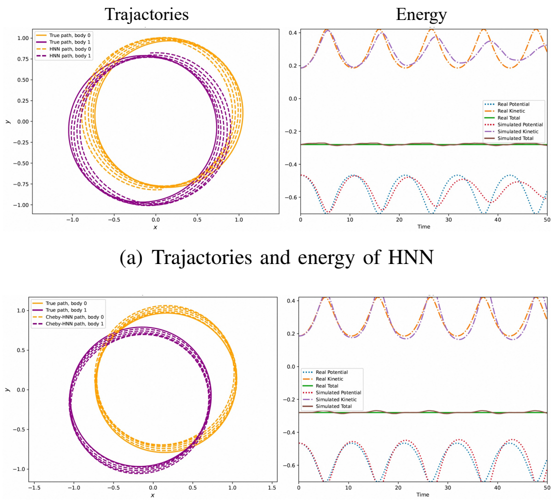
We conduct experiments on modeling object trajectories in real-world physical scenarios. To be concrete, we employ Neural ODEs (NODE) [22] to model the trajectory of a bouncing ball, and Hamiltonian Neural Networks (HNN) [23] to simulate two-body and three-body problems, as well as the motion of a real pendulum. Besides, we compare the learned kinetic energy, potential energy, and total mechanical energy of the objects between the baselines and the CFU variants to evaluate the generalization capacity because overfitting would manifest as small trajectory errors coupled with large

TABLE II
THE MSE-LOSS BETWEEN SIMULATED TRAJECTORY AND GROUND TRUTH PREDICTED BY NODE, HNN AND THEIR CFU VARIANTS.

Order	1	2	3	4	5	6	7	8	9
bouncing ball with Baseline NODE: 0.231									
NODE-CFU	0.499	0.286	**0.225**	**0.020**	**0.071**	0.376	**0.024**	0.451	**0.084**
2-body problem with Baseline HNN: 6.404 (Unit: 1e-6)									
HNN-CFU	**4.474**	**5.994**	**2.437**	**2.220**	6.545	**5.727**	**4.245**	**4.454**	**3.087**
3-body problem with Baseline HNN: 4.437 (Unit: 1e-1)									
HNN-CFU	4.981	4.531	**4.242**	**4.373**	**4.121**	**4.428**	**4.029**	4.513	**4.219**
real pendulum problem with Baseline HNN: 5.982 (Unit: 1e-3)									
HNN-CFU	**5.807**	**5.794**	**5.805**	**5.808**	**5.803**	**5.802**	**5.793**	**5.806**	**5.807**

energy discrepancies. We train NODE and NODE-CFU for 1,000 iterations, while HNN and its CFU variant for 10,000 iterations.

The trajectories of the 2-body problem predicted by HNN and HNN-CFU are shown in Figure 3.



(a) Trajactories and energy of HNN

(b) Trajactories and energy of HNN-CFU

Fig. 3. The 2-body trajectories and corresponding energy predicted by HNN and HNN-CFU. (a) HNN. (b) HNN-CFU.

The mse-loss of energy of 2-body, 3-body, and real pendulum problem of HNN and HNN-CFU is shown in Table III:

TABLE III
THE MSE-LOSS OF ENERGY OF VARIOUS PHYSICAL SCENARIOS OF HNN AND HNN-CFU.

Scenario	2-body	3-body	real pendulum
HNN	2.903×10^{-5}	1.096×10^{-2}	7.500×10^{-3}
HNN-CFU	$\mathbf{1.085}\times 10^{-5}$	$\mathbf{6.093}\times 10^{-3}$	$\mathbf{7.494}\times 10^{-3}$

Through these comprehensive experiments, we have demonstrated that CFU exhibits superior approximation capabilities while show stronger generalization capacity compared to addition fusion. This robust performance suggests that CFU is well-suited for learning and modeling physical laws.

TABLE IV
THE TEST ACCURACY OF MODELS AND THEIR CFU VARIANTS ON CIFAR100.

Order	1	2	3	4	5	6	7	8	9
Using PCNN Architecture with Baseline: 59.8									
PCNN-FU	**59.8**	59.5	59.4	59.4	59.8	60.0	59.7	59.7	59.6
PCNN-CFU	59.7	**60.5**	**60.2**	**59.9**	**60.3**	**60.4**	**60.4**	**60.0**	**60.2**
Using MobileNet Architecture with Baseline: 60.0									
MobileNet-FU	59.7	59.8	59.3	60.1	60.2	**60.8**	59.6	**60.1**	O
MobileNet-CFU	**60.0**	**60.0**	**60.2**	**60.5**	**60.4**	60.4	**60.3**	60.0	**60.2**
Using ResNet18 Architecture with Baseline: 76.1									
ResNet18-FU	75.6	**76.7**	76.0	76.4	**76.4**	76.0	75.8	75.9	75.7
ResNet18-CFU	75.8	75.5	**76.3**	**76.6**	76.1	**76.6**	**76.4**	**76.3**	**76.1**

C. Image classification

We further evaluate the performance of CFU on the classification task using the CIFAR-10 [24] and CIFAR-100 [24] datasets. The baseline models include PCNN (a plain 5-layer CNN with 5 hidden states), MobileNetV2, ResNet18. We change the additive fusion to our CFU or plain polynomial fusion unit (PU, replacing the Chebyshev polynomials with plain polynomials) to construct the corresponding CFU and PU variants. We train these models from scratch for 120 epochs. The batch size for each model is set to 128, with an initial learning rate of 0.1, which is reduced by a factor of 10 at epochs 40, 60, 80, and 100. We use SGD with momentum of 0.9 and a weight decay of 5×10^{-4} as the optimizer. All other settings for both the baseline models and CFU variants are kept identical.

The test accuracy on CIFAR100 of baselines and their CFU and FU variants are shown in Tab. IV, through which we can conclude that CFU-based models of most orders perform better than the baseline models. Besides, the recursive formulation of Chebyshev polynomials intrinsically avoids numerical overflow, representing a significant advantage over plain polynomials.

V. CONCLUSION

In this paper, we propose CFU, a lightweight fusion mechanism that utilizes orthogonal Chebyshev polynomial connections to enhance representational capacity while maintaining strict parameter efficiency. By introducing recursive spectral expansions of shortcut features and Hadamard product with current-layer features, CFU establishes a mathematically principled framework for high-order feature interactions, enabling superior approximation capabilities with minimal computational overhead. Our empirical results demonstrate that CFU consistently outperforms traditional fusion paradigms across medical image segmentation, physical law learning, and image classification tasks.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Advances in neural information processing systems*, vol. 27, 2014.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[6] S. Dong, P. Wang, and K. Abbas, “A survey on deep learning and its applications,” *Computer Science Review*, vol. 40, p. 100379, 2021.

[7] M. Wainberg, D. Merico, A. Delong, and B. J. Frey, “Deep learning in biomedicine,” *Nature biotechnology*, vol. 36, no. 9, pp. 829–838, 2018.

[8] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, p. 1798–1828, Aug. 2013.

[9] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” *arXiv preprint arXiv:1510.00149*, 2015.

[10] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, “Attentional Feature Fusion,” *arXiv e-prints*, p. arXiv:2009.14082, Sep. 2020.

[11] D. Ouyang, S. He, G. Zhang, M. Luo, H. Guo, J. Zhan, and Z. Huang, “Efficient Multi-Scale Attention Module with Cross-Spatial Learning,” *arXiv e-prints*, p. arXiv:2305.13563, May 2023.

[12] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[13] X. Li, W. Wang, X. Hu, and J. Yang, “Selective Kernel Networks,” *arXiv e-prints*, p. arXiv:1903.06586, Mar. 2019.

[14] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, “ResNeSt: Split-Attention Networks,” *arXiv e-prints*, p. arXiv:2004.08955, Apr. 2020.

[15] Q. Li and F. Chen, “A multiscale self-adaptive attentional feature fusion network for sound event localization and detection,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, C. Qin and Q. Cheng, Eds., vol. 13637, May 2025, p. 136370L.

[16] C. Zheng, M. Mendieta, and C. Chen, “POSTER: A Pyramid Cross-Fusion Transformer Network for Facial Expression Recognition,” *arXiv e-prints*, p. arXiv:2204.04083, Apr. 2022.

[17] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials*. CRC press, 2002.

[18] W.-D. K. Ma, J. P. Lewis, and W. Bastiaan Kleijn, “The HSIC Bottleneck: Deep Learning without Back-Propagation,” *arXiv e-prints*, p. arXiv:1908.01580, Aug. 2019.

[19] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, X. Yang, P.-A. Heng, I. Cetin, K. Lekadir, O. Camara, M. A. Gonzalez Ballester, G. Sanroma, S. Napel, S. Petersen, G. Tziritas, E. Grinias, M. Khened, V. A. Kollerathu, G. Krishnamurthi, M.-M. Rohé, X. Pennec, M. Sermesant, F. Isensee, P. Jäger, K. H. Maier-Hein, P. M. Full, I. Wolf, S. Engelhardt, C. F. Baumgartner, L. M. Koch, J. M. Wolterink, I. Išgum, Y. Jang, Y. Hong, J. Patravali, S. Jain, O. Humbert, and P.-M. Jodoin, “Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved?” *IEEE Transactions on Medical Imaging*, vol. 37, no. 11, pp. 2514–2525, 2018.

[20] S. S. Bakas, “Brats miccai brain tumor dataset,” 2020. [Online]. Available: https://dx.doi.org/10.21227/hdtd-5j88

[21] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[22] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” *Advances in neural information processing systems*, vol. 31, 2018.

[23] S. Greydanus, M. Dzamba, and J. Yosinski, “Hamiltonian neural networks,” *Advances in neural information processing systems*, vol. 32, 2019.

[24] A. Krizhevsky, G. Hinton *et al.*, “Learning multiple layers of features from tiny images,” 2009.