RTAF: RESPONSE AWARE TENSOR ADAPTIVE FUNCTION BLOCK FOR EFFICIENT 3D MEDICAL IMAGE ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Volumetric medical features contain complex anatomical structures and simple repetitive responses, while dense 3D convolutions apply the same transformation to all feature responses. This uniform processing introduces unnecessary computation for simple responses and insufficient modeling capability for complex structures. Existing efficient operators mainly reduce redundancy or enhance feature interactions, but they lack a unified design that balances nonlinear representation and efficient local processing. We propose the Response Aware Tensor Adaptive Function (RTAF) block, a plug-and-play feature transformation module that dynamically combines nonlinear modeling and lightweight local processing according to feature characteristics. RTAF consists of three components: Router, Nonlinear Tensor Branch, and Local Convolution Branch. The Router generates an input-dependent modulation map that adjusts the inputs to the nonlinear and local processing branches.The Nonlinear Tensor Branch enables lightweight nonlinear feature modeling by combining Gaussian basis expansion with learnable centers and grouped CP-factorized tensor weights, while the Local Convolution Branch efficiently captures local spatial structures.Extensive experiments on six MedMNIST3D classification datasets and five Medical Segmentation Decathlon benchmarks demonstrate that RTAF achieves competitive performance across 3D ResNet-18/50 and nnU-Netv2, while introducing substantially lower parameter and computational costs than dense 3D convolution blocks.

1 INTRODUCTION

Deep neural networks learn hierarchical representations through successive feature transformations, where convolutional operators play a fundamental role in extracting spatial patterns and modeling channel interactions Wang et al. (2021a). However, conventional convolutional operators usually adopt a uniform transformation strategy, applying the same operation to all feature responses. This assumption ignores the fact that feature responses often have different levels of complexity. Some responses mainly contain repetitive patterns and can be processed with lightweight operations, whereas others encode complex structures and channel dependencies that require stronger nonlinear modeling. Applying identical transformations to all responses therefore results in inefficient computation for simple patterns and insufficient representation for complex structures. This problem is particularly evident in volumetric medical image analysis, where 3D features simultaneously contain large homogeneous regions and fine-grained anatomical details, such as boundaries and small lesions. Although recent 3D architectures have improved feature extraction and contextual modeling, their underlying transformation operators still rely on fixed processing mechanisms and cannot adapt transformation complexity to heterogeneous feature responses.

Recent studies have explored improving feature transformation efficiency from different perspectives. Redundancy-aware operators, such as Octave Convolution and SCConv, reduce unnecessary computation by exploiting spatial or channel redundancy in feature representations Chen et al. (2019); Li et al. (2023). However, they mainly focus on removing redundant information and do not provide stronger transformations for complex responses. Adaptive convolution methods, including Dynamic Convolution and CondConv, introduce input-dependent transformations to improve flexibility Chen et al. (2020); Yang et al. (2019). Nevertheless, these approaches mainly adjust trans-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

formation weights within predefined operators, without explicitly considering the combination of expressive nonlinear modeling and efficient local processing. Therefore, an effective feature transformation mechanism that dynamically balances transformation complexity and computational cost remains desirable for heterogeneous volumetric features.

To address this issue, we propose the Response Aware Tensor Adaptive Function (RTAF) block, a plug-and-play module for adaptive volumetric feature transformation. RTAF is designed to assign different transformation strengths to feature responses with different characteristics. It contains three key components: a Router, a Nonlinear Tensor Branch, and a Local Convolution Branch. The Router generates a soft modulation map to regulate the information flow between the two branches. The Nonlinear Tensor Branch provides efficient nonlinear feature modeling for complex responses, while the Local Convolution Branch preserves local spatial information with lightweight operations. By integrating these complementary transformations, RTAF enables flexible feature processing without modifying existing network architectures.

Our contributions are summarized as follows:

- We propose the Response Aware Tensor Adaptive Function (RTAF) block, a plug-and-play feature transformation module that can directly replace conventional convolutional blocks without modifying the original network architectures or training procedures.
- We develop a unified adaptive transformation framework, which dynamically balances nonlinear representation capability and efficient local processing.
- We conduct extensive experiments on six MedMNIST3D classification datasets and five Medical Segmentation Decathlon benchmarks, demonstrating RTAF achieves competitive performance and favorable accuracy-efficiency trade-offs across multiple 3D architectures.

2 Related Work

2.1 Volumetric Medical Image Representation

Deep learning-based volumetric medical image analysis has been extensively developed with convolutional and transformer-based architectures. 3D U-Net and V-Net extend encoder-decoder architectures into three-dimensional space and establish fundamental frameworks for volumetric segmentation Çiçek et al. (2016); Milletari et al. (2016). nnU-Net further introduces a self-configuring paradigm that automatically adapts network configurations, preprocessing strategies, and training procedures for different datasets Isensee et al. (2021). Beyond convolutional architectures, UNETR incorporates transformer encoders to capture long-range dependencies, while SwinUNETR introduces hierarchical representation learning through shifted-window attention Hatamizadeh et al. (2022); Tang et al. (2022). 3D UX-Net explores large-kernel convolution to enlarge receptive fields while preserving convolutional inductive biases Lee et al. (2022). These approaches improve volumetric representation through enhanced contextual modeling and feature interaction. However, their underlying feature transformation operators generally adopt fixed processing mechanisms, where heterogeneous feature responses are treated with similar transformations.

2.2 Efficient Feature Transformation

The substantial computational cost of volumetric models has motivated studies on efficient feature transformation. Octave Convolution and SCConv exploit spatial and channel redundancy to reduce unnecessary computation in feature representations Chen et al. (2019); Li et al. (2023). ACS-Conv further improves the efficiency of 3D convolution through operator redesign for volumetric feature extraction Yang et al. (2021). From an architectural perspective, EffiDec3D reduces computational overhead by optimizing decoder structures for efficient volumetric reconstruction Rahman & Marculescu (2025). Although these methods effectively improve computational efficiency, they mainly focus on reducing redundant computation or simplifying network structures. They do not explicitly consider that feature responses with different characteristics may require different transformation strategies.

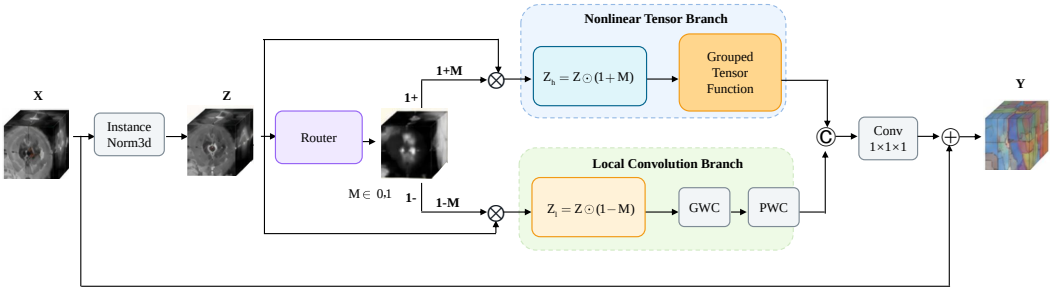

Figure 1: The architecture of RTAF.

2.3 Adaptive and Nonlinear Feature Transformation

Recent studies have explored adaptive and nonlinear transformations to improve the flexibility of feature learning. Dynamic Convolution and CondConv introduce input-dependent kernel aggregation to generate more flexible convolutional operators Chen et al. (2020); Yang et al. (2019). KAC-Net further investigates learnable nonlinear function transformations for medical image analysis, demonstrating the potential of nonlinear mappings beyond standard convolution Li et al. (2026). However, these approaches generally enhance transformation flexibility or nonlinear representation within a single transformation framework. They do not provide a mechanism to jointly exploit expressive nonlinear modeling for complex responses and efficient local processing for simple responses. RTAF addresses this limitation by adaptively combining complementary transformation branches according to feature characteristics.

3 Method

3.1 Overall Framework

As illustrated in Fig. 1, the proposed Response Aware Tensor Adaptive Function (RTAF) is a plug-and-play feature transformation module designed to adapt transformation strength for heterogeneous volumetric features. Unlike conventional convolutional operators that apply a uniform transformation to all responses, RTAF allows different features to receive different levels of nonlinear modeling and local processing. Given an intermediate feature representation $X \in \mathbb{R}^{C\times D\times H\times W}$, RTAF first applies Instance Normalization to obtain the normalized feature representation Z. A lightweight Router then extracts channel-wise and spatial response cues and generates a soft transformation map $M \in [0,1]^{C\times D\times H\times W}$. Instead of performing hard feature selection, the generated map softly regulates the transformation strength of different feature responses. Specifically, RTAF modulates the normalized feature as:

$$Z_h = (1 + M) \odot Z, \tag{1}$$

$$Z_l = (1 - M) \odot Z. \tag{2}$$

Larger values of M amplify the input to the nonlinear tensor branch and attenuate the input to the local convolution branch. Smaller values preserve more of the local branch input while keeping the nonlinear branch active. This modulation enables learned transformation preferences across heterogeneous feature responses.

The modulated features are then processed by two complementary branches:

$$Y_h = \mathcal{F}_{NTB}(Z_h), \tag{3}$$

$$Y_l = \mathcal{F}_{LC}(Z_l). \tag{4}$$

The Nonlinear Tensor Branch provides expressive nonlinear feature modeling, while the Local Convolution Branch efficiently captures local spatial structures. Their complementary representations

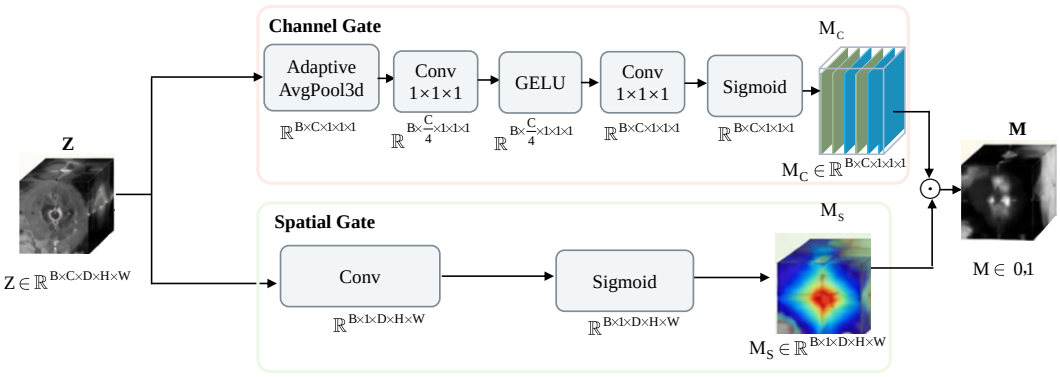

Figure 2: The architecture of Router.

are concatenated and projected through a $1 \times 1 \times 1$ convolution, followed by a residual connection with the original input. Through this adaptive cooperation, RTAF achieves a balance between nonlinear representation capability and efficient local processing without modifying existing network architectures.

3.2 Router

As illustrated in Fig. 2, the Router generates a soft transformation preference map by combining a learned channel prior with input-dependent spatial modulation. Given the normalized feature representation $Z \in \mathbb{R}^{C\times D\times H\times W}$, its two branches provide complementary channel-wise calibration and location-aware adjustment.

The channel branch applies global average pooling to Z, followed by two $1 \times 1 \times 1$ projections with an intermediate GELU activation and a final sigmoid function to produce $M_c \in (0, 1)^{C\times 1\times 1\times 1}$. Under the adopted non-affine instance normalization, the pooled channel means are zero, and this branch therefore learns input-independent calibration weights shared across samples. These weights provide a common channel-wise prior, encoding shared transformation preferences that complement the sample-specific adjustments of the spatial branch.

The spatial branch applies a lightweight $3 \times 3 \times 3$ convolution followed by a sigmoid function to obtain $M_s \in (0, 1)^{1\times D\times H\times W}$. By encoding local volumetric structures, it determines how the modulation varies across spatial locations for each input.

The two gates are combined through broadcast multiplication:

$$M = M_c \odot M_s, \tag{5}$$

where $\odot$ denotes element-wise multiplication with broadcasting. The resulting map $M \in [0, 1]^{C\times D\times H\times W}$ couples shared channel preferences with input-dependent spatial adjustments and regulates the features supplied to the two transformation branches. Through the modulation defined in Sec. 3.1, larger values of M amplify the input to the nonlinear tensor branch and attenuate the corresponding input to the local convolution branch, while smaller values retain more of the normalized feature in the local branch.

3.3 Nonlinear Tensor Branch

The Nonlinear Tensor Branch aims to enhance feature transformation by introducing a learnable tensorized nonlinear mapping. As illustrated in Fig. 3, the branch consists of two complementary components: the Adaptive Basis Function and the Factorized Tensor Weight. The Adaptive Basis Function constructs a learnable nonlinear representation space for feature responses, while the Factorized Tensor Weight models the interactions among input channels, output channels, and nonlinear basis components. By jointly learning nonlinear basis representations and tensorized feature interactions, the Nonlinear Tensor Branch provides stronger capability for modeling complex nonlinear relationships.

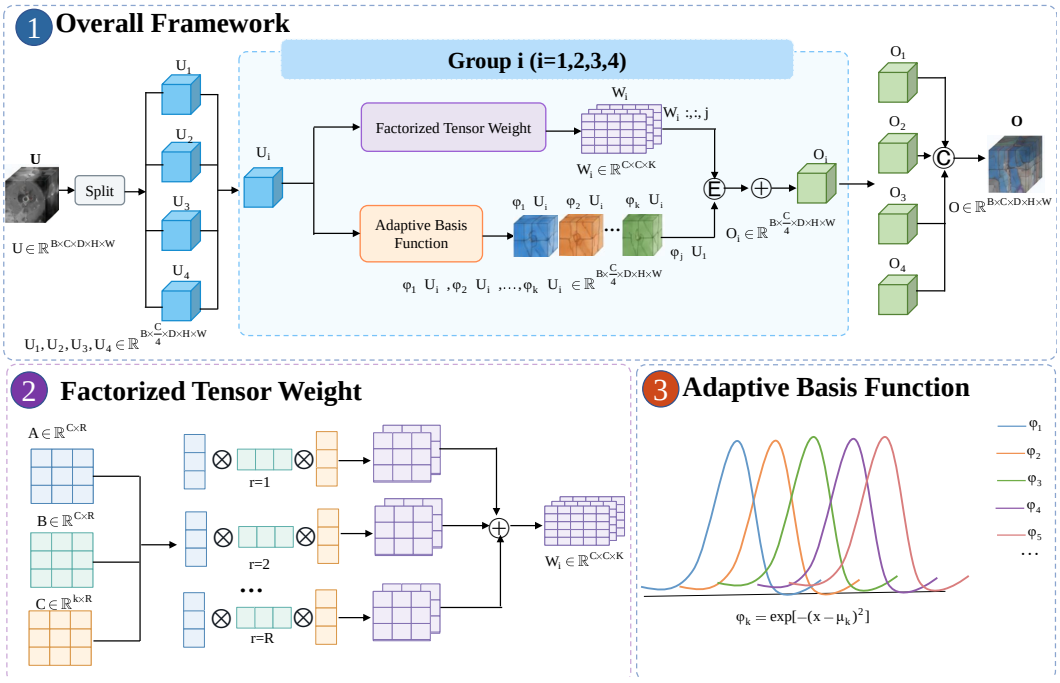

Figure 3: The architecture of Nonlinear Tensor Branch integrated with Factorized Tensor Weight and Adaptive Basis Function.

To improve the efficiency of tensorized nonlinear modeling, the input feature representation Z_h is first divided into G channel groups:

$$Z_h = \{Z_h^1, Z_h^2, \cdots, Z_h^G\}, \tag{6}$$

where each group contains $C_g = C/G$ channels. Grouped transformation restricts tensor interactions within smaller channel subsets, reducing the interaction space while maintaining group-specific feature relationships.

For each feature group, the Adaptive Basis Function maps feature responses into a learnable nonlinear basis space. Specifically, the k-th Gaussian basis function for an input response x is defined as:

$$\phi_k(x) = \exp(-(x - \mu_k)^2), \tag{7}$$

where μ_k denotes the learnable center of the Gaussian basis. Multiple basis functions construct a richer nonlinear function space, where each feature response can be represented through adaptive combinations of different nonlinear bases. Compared with using a single nonlinear mapping form, this multi-basis formulation provides stronger capability to capture complex nonlinear variations by learning diverse response transformations within a unified representation space.

The generated nonlinear basis responses are subsequently aggregated through the Factorized Tensor Weight. For each feature group, the tensor weight is represented as:

$$W_g \in \mathbb{R}^{C_g^{out} \times C_g^{in} \times K}, \tag{8}$$

where K denotes the number of nonlinear basis functions. By introducing the basis dimension into feature transformation, the tensor formulation explicitly models the relationships between feature dimensions and nonlinear basis responses. Therefore, the Factorized Tensor Weight learns how different basis responses contribute to output features through joint channel-basis interactions.

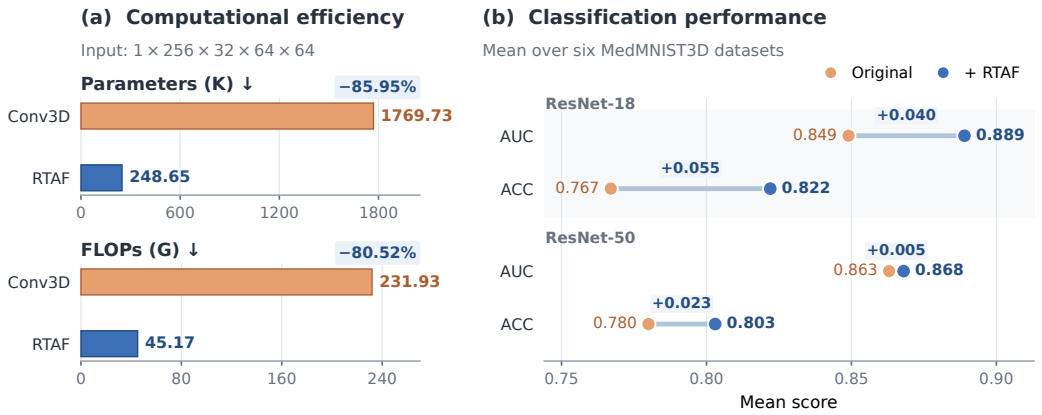

Figure 4: Efficiency and performance analysis of RTAF. (a) Comparison of parameters and FLOPs between Conv3D and RTAF. (b) Classification performance changes after integrating RTAF into ResNet-18 and ResNet-50.

However, directly parameterizing the complete tensor weight introduces a high-dimensional interaction space, making tensorized transformation difficult to scale. To preserve tensor interaction modeling while improving scalability, the Factorized Tensor Weight adopts CP decomposition:

$$W_g = \sum_{r=1}^{R} a_{g,r} \circ b_{g,r} \circ c_{g,r}, \tag{9}$$

where R denotes the tensor rank and $\circ$ represents the outer product. The factor vectors correspond to the output channel, input channel, and basis dimensions, respectively. This factorized formulation preserves the interaction structure among feature channels and nonlinear bases while avoiding dense tensor parameterization, reducing the number of learnable parameters from $C_g^{out} C_g^{in} K$ to $R(C_g^{out} + C_g^{in} + K)$.

The nonlinear basis responses are then aggregated through the Factorized Tensor Weight. For the g-th feature group, the output feature is computed as:

$$Y_h^g(b, i, d, h, w) = \sum_{k=1}^{K} \sum_{c=1}^{C_g^{in}} W_g(i, c, k) \phi_k(Z_h^g)(b, c, d, h, w), \tag{10}$$

where b, i, c, and k denote the batch index, output channel, input channel, and basis index, respectively. Through this formulation, the Nonlinear Tensor Branch jointly learns nonlinear response representations and tensorized feature interactions. The Adaptive Basis Function determines the nonlinear representation space, while the Factorized Tensor Weight controls the composition of nonlinear basis responses across feature dimensions. Finally, the outputs from different channel groups are finally concatenated to recover the complete feature representation:

$$Y_h = \text{Concat}(Y_h^1, Y_h^2, \cdots, Y_h^G). \tag{11}$$

The concatenation operation preserves the complementary nonlinear transformations learned from different channel groups and restores the full channel dimension for subsequent feature fusion. By integrating multi-basis nonlinear representation, factorized tensor interaction, and grouped transformation, the Nonlinear Tensor Branch enhances the capability of modeling complex nonlinear feature relationships while maintaining computational efficiency.

Methods	OrganMNIST3D		NoduleMNIST3D		FractureMNIST3D	
	AUC	ACC	AUC	ACC	AUC	ACC
ResNet-18+2.5D	0.977	0.788	0.838	0.835	0.587	0.451
ResNet-18+3D	<u>0.996</u>	0.907	0.863	0.844	0.712	0.508
ResNet-18+ACS	0.994	0.900	0.873	0.847	0.714	0.497
ResNet-50+2.5D	0.974	0.769	0.835	0.848	0.552	0.397
ResNet-50+3D	0.994	0.883	0.875	0.847	0.725	0.494
ResNet-50+ACS	0.994	0.889	0.886	0.841	**0.750**	0.517
ACS-Conv	0.992	0.897	0.790	0.819	0.581	0.438
auto-sklearn	0.977	0.814	**0.914**	0.874	0.628	0.453
AutoKeras	0.979	0.804	0.844	0.834	0.642	0.458
FPVT	0.923	0.800	0.814	0.822	0.640	0.438
TTM	0.995	0.915	0.873	0.870	0.662	0.546
LMTTM	**0.997**	**0.948**	0.891	**0.883**	0.691	0.562
KACNet	0.993	0.916	0.894	0.865	0.714	0.542
ResNet-18 3D/w RTAF	**0.997**	0.925	0.889	0.855	0.729	**0.579**
ResNet-50 3D/w RTAF	0.996	0.910	0.899	0.868	0.744	0.542
Methods	AdernalMNIST3D		VesselMNIST3D		SynapseMNIST3D	
	AUC	ACC	AUC	ACC	AUC	ACC
ResNet-18+2.5D	0.718	0.772	0.748	0.846	0.634	0.696
ResNet-18+3D	0.827	0.721	0.874	0.877	0.820	0.745
ResNet-18+ACS	0.839	0.754	0.930	0.928	0.705	0.722
ResNet-50+2.5D	0.732	0.763	0.751	0.877	0.669	0.735
ResNet-50+3D	0.828	0.745	0.907	0.918	0.851	0.795
ResNet-50+ACS	0.828	0.758	0.912	0.858	0.719	0.709
ACS-Conv	0.791	0.801	0.843	0.910	0.615	0.711
auto-sklearn	0.828	0.802	0.910	0.915	0.631	0.730
AutoKeras	0.804	0.705	0.773	0.894	0.538	0.724
FPVT	0.801	0.704	0.770	0.888	0.530	0.712
TTM	0.816	0.808	0.855	0.922	0.709	0.753
LMTTM	0.847	**0.842**	0.917	0.931	0.737	0.777
KACNet	0.832	0.658	0.942	0.942	**0.939**	**0.886**
ResNet-18 3D/w RTAF	**0.886**	0.812	**0.964**	**0.948**	0.867	0.810
ResNet-50 3D/w RTAF	0.876	0.826	0.951	0.929	0.742	0.741

Table 1: Comparison with existing methods on MedMNIST3D classification benchmarks. Best results are highlighted in bold.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate RTAF as a feature transformation in volumetric classification and segmentation by replacing convolutional blocks in existing 3D networks. Classification uses six MedMNIST3D datasets, OrganMNIST3D, NoduleMNIST3D, FractureMNIST3D, AdrenalMNIST3D, VesselMNIST3D, and SynapseMNIST3D Yang et al. (2023), which span diverse volumetric patterns and levels of structural complexity. Segmentation uses five Medical Segmentation Decathlon (MSD) tasks, including Task02 Heart, Task04 Hippocampus, Task09 Spleen, Task05 Prostate, and Task10 Colon Antonelli et al. (2022)—covering different anatomical structures and spatial distributions in dense prediction.

We insert RTAF into 3D ResNet-18, 3D ResNet-50, and nnU-Netv2. It replaces the second convolutional block in the last two residual stages of each ResNet and in the last three encoder stages of nnU-Netv2; the remaining architecture and training procedures are unchanged. Classification follows the standard MedMNIST3D settings, and segmentation follows the official nnU-Netv2 pipeline. We report the mean over five cross-validation folds for nnU-Netv2 and use results from the original papers for comparison methods when available.

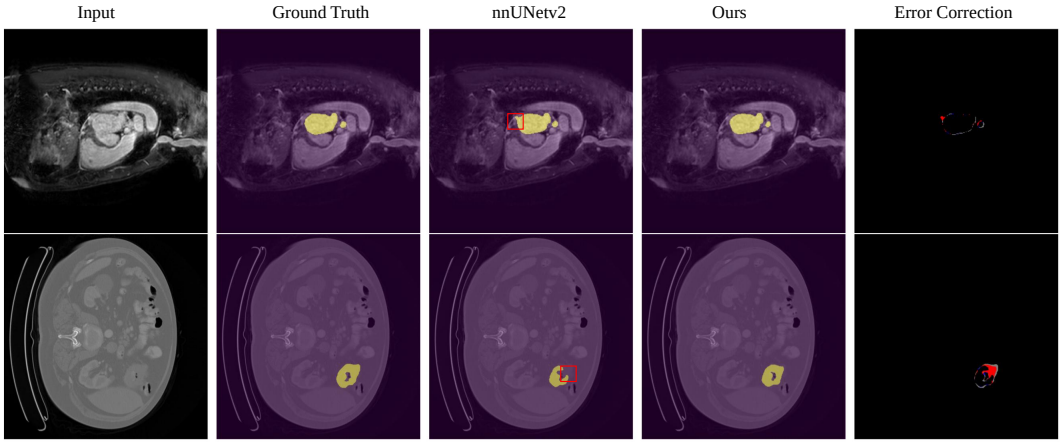

Figure 5: Qualitative comparison between nnU-Netv2 and nnU-Netv2 with RTAF.

Method	Task02_Heart	Task04_Hippocampus			Task09_Spleen
	Avg.	Ant.	Post.	Avg.	Avg.
SlimUNETR	90.42	87.19	85.38	86.29	95.42
SegFormer3D	91.64	87.44	85.48	86.46	95.68
UNETR	91.42	88.01	86.34	87.18	94.60
nnFormer3D	92.21	87.58	85.84	86.71	91.40
TransBTS	90.12	88.39	86.68	87.54	94.54
UNETR++	92.39	88.51	87.01	87.76	96.49
SwinUNETR	91.92	88.61	87.12	87.87	96.13
SwinUNETRv2	91.96	88.48	86.86	87.67	96.77
3DUX-Net	92.03	89.33	87.64	88.49	96.68
EffiDec3D	92.63	88.89	87.20	88.05	96.56
nnUNetv2	92.71	89.86	88.24	89.05	97.35
nnUNetv2 w/RTAF	**94.16**	**90.26**	**88.47**	**89.37**	**97.47**

Method	Task05_Prostate			Task10_Colon	5 Tasks
	PZ	TZ	Avg.	Avg.	Avg.
SlimUNETR	28.07	71.22	49.65	25.13	69.38
SegFormer3D	51.06	80.43	65.75	33.09	74.52
UNETR	35.70	76.96	56.33	28.03	71.51
nnFormer	45.87	80.90	63.39	24.74	71.69
TransBTS	43.83	77.06	60.45	29.53	72.44
UNETR++	32.56	74.91	53.74	40.82	74.24
SwinUNETR	49.21	76.64	62.93	42.94	76.36
SwinUNETRv2	52.03	81.31	66.67	48.87	78.39
3DUX-Net	48.99	82.67	65.83	48.77	78.36
EffiDec3D	56.97	82.87	69.92	48.42	79.12
nnUNetv2	**73.31**	85.68	79.50	50.55	81.83
nnUNetv2 w/RTAF	70.60	**89.30**	**79.95**	**52.25**	**82.64**

Table 2: Comparison with existing methods on MSD. Best results are highlighted in bold.

4.2 Classification Results

Table 1 compares RTAF-equipped 3D ResNets with ResNet-18/50 in 2.5D, 3D, and ACS formsYang et al. (2021), ACS-Conv Yang et al. (2021), auto-sklearn Feurer et al. (2015), AutoKeras Jin et al. (2019), FPVT Liu et al. (2022), TTM Ryoo et al. (2023), LMTTM Wei et al. (2025), and KACNet Li et al. (2026) on six MedMNIST3D datasets.

With ResNet-18, RTAF improves both AUC and accuracy over the matched 3D backbone on every dataset. The gains are most pronounced on VesselMNIST3D and AdrenalMNIST3D, while those

Variant	Dice ↑	HD95 ↓	Params (K) ↓	FLOPs (G) ↓
w/o Adaptive Basis Function	96.71	37.18	260.7	30.13
w/o Factorized Tensor Weight	97.40	24.90	375.4	45.17
w/o Router	97.00	35.47	208.6	44.26
w/o local_path	97.15	34.06	175.7	35.54
RTAF	97.47	25.11	248.6	45.17

Table 3: Ablation study of RTAF components on Task09 Spleen segmentation.

on OrganMNIST3D and NoduleMNIST3D are smaller. Averaging the six entries in Table 1 places this variant first in mean AUC and second in mean accuracy. Figure 4b summarizes its improvement over the original backbone. The aggregate ranking does not extend to every task, such as KACNet achieves a higher SynapseMNIST3D AUC, and LMTTM has the highest mean accuracy. Table 1 shows a less consistent effect with ResNet-50. Although five datasets improve, SynapseMNIST3D AUC drops from 0.851 to 0.742, leaving only a modest increase in the six-dataset mean shown in Figure 4b. Figure 4a compares the blocks at the profiled feature size, RTAF uses 248.65K parameters and 45.17G FLOPs, versus 1769.73K and 231.93G for a dense Conv3D block.

4.3 Segmentation Results

Table 2 compares RTAF-equipped nnU-Netv2 with SlimUNETR Pang et al. (2024), SegFormer3D Perera et al. (2024), UNETR Hatamizadeh et al. (2022), nnFormer Zhou et al. (2023), TransBTS Wang et al. (2021b), UNETR++ Shaker et al. (2024), SwinUNETR Tang et al. (2022), SwinUNETRv2 He et al. (2023), 3D UX-Net Lee et al. (2022), EffiDec3D Rahman & Marculescu (2025), and the original nnU-Netv2 on five Medical Segmentation Decathlon tasks.

In Table 2, RTAF-equipped nnU-Netv2 leads every task mean and the five-task mean, exceeding the result from other models. Relative to the matched nnU-Netv2 baseline, the overall mean rises from 81.83 to 82.64; Heart and Colon contribute the clearest gains, whereas Hippocampus and Spleen change little. The Prostate columns in Table 2 reveal a finer-grained trade-off: transition-zone Dice increases, but peripheral-zone Dice decreases from 73.31 to 70.60.Figure 5 shows a Heart slice and a Colon slice. In Heart, RTAF reduces the baseline's contour mismatch; in Colon, it better preserves the foreground boundary and the small internal void. Most of the surrounding prediction remains unchanged. The red regions outweigh the blue in these examples, consistent with the improved overlap reported in Table 2.

4.4 Ablation Studies

Table 3 ablates RTAF on MSD Task09 Spleen. The largest decrease in mean Dice appears in the *w/o Adaptive Basis Function* row, which replaces the Gaussian basis with a grouped linear map. The *w/o Router* row fixes the router mask to zero, whereas *w/o local path* supplies zeros to the fusion layer; both yield smaller Dice decreases relative to the complete *RTAF* row. On this task, the basis has the clearest observed effect, with more modest contributions from routing and local processing. The *w/o Factorized Tensor Weight* row in Table 3 replaces the CP factors with a dense weight tensor. It yields nearly the same Dice as *RTAF* and a marginally lower HD95, but requires about 51% more parameters at the same reported FLOPs. CP factorization therefore retains performance with fewer learned weights in this setting.

5 Conclusion

We introduced RTAF, a feature transformation block that combines a Gaussian-basis tensor function and local 3D convolution through a spatial–channel router. The router softly modulates the inputs to the two pathways, allowing nonlinear channel interactions and local spatial modeling to complement one another within existing networks. Across six MedMNIST3D classification datasets and five Medical Segmentation Decathlon tasks, RTAF improves the aggregate scores of the matched 3D ResNets and nnU-Netv2.

AI USE STATEMENT

In this work, generative AI tools were used to edit and restructure author-written text for grammar, clarity, concision, and academic style, and to assist with drafting the presentation and interpretation of author-produced experimental results. Generative AI tools were not used to formulate the research questions, develop the methodology, generate or analyze data, implement code, conduct experiments, create figures or tables, or select references. All AI-assisted text was checked against the authors' data and revised by the authors to ensure technical accuracy and consistency. The final scientific judgments were made by the authors. We take responsibility for the final content of this work, including text, claims, results, and conclusions.

REFERENCES

Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022.

Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11027–11036, 2020. doi: 10.1109/CVPR42600.2020.01104.

Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pp. 424–432, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46723-8.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R. Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 574–584, January 2022.

Yufan He, Vishwesh Nath, Dong Yang, Yucheng Tang, Andriy Myronenko, and Daguang Xu. Swinunetr-v2: Stronger swin transformers with stagewise convolutions for 3d medical image segmentation. In Hayit Greenspan, Anant Madabhushi, Parvin Mousavi, Septimiu Salcudean, James Duncan, Tanveer Syeda-Mahmood, and Russell Taylor (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, pp. 416–426, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-43901-8.

Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1946–1956, 2019.

Ho Hin Lee, Shunxing Bao, Yuankai Huo, and Bennett A. Landman. 3D UX-Net: A Large Kernel Volumetric ConvNet Modernizing Hierarchical Transformer for Medical Image Segmentation. *arXiv e-prints*, art. arXiv:2209.15076, September 2022. doi: 10.48550/arXiv.2209.15076.

Deguang Li, Zeyan Jin, Chengyue Guan, Liubing Ji, Yudong Zhang, Zhaozhao Xu, and Jiyong Zhang. KACNet: Enhancing CNN feature representation with Kolmogorov–Arnold networks for medical image segmentation and classification. *Information Sciences*, 726:122760, 2026. doi: 10.1016/j.ins.2025.122760.

Jiafeng Li, Ying Wen, and Lianghua He. Scconv: Spatial and channel reconstruction convolution for feature redundancy. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6153–6162, 2023. doi: 10.1109/CVPR52729.2023.00596.

Jinwei Liu, Yan Li, Guitao Cao, Yong Liu, and Wenming Cao. Feature pyramid vision transformer for medmnist classification decathlon. In *2022 International joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2022.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation . In *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, Los Alamitos, CA, USA, October 2016. IEEE Computer Society. doi: 10.1109/3DV.2016.79. URL https://doi.ieeecomputersociety.org/10.1109/3DV.2016.79.

Yan Pang, Jiaming Liang, Teng Huang, Hao Chen, Yunhao Li, Dan Li, Lin Huang, and Qiong Wang. Slim unetr: Scale hybrid transformers to efficient 3d medical image segmentation under limited computational resources. *IEEE Transactions on Medical Imaging*, 43(3):994–1005, 2024. doi: 10.1109/TMI.2023.3326188.

Shehan Perera, Pouyan Navard, and Alper Yilmaz. Segformer3d: An efficient transformer for 3d medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4981–4988, June 2024.

Md Mostafijur Rahman and Radu Marculescu. Effidec3d: An optimized decoder for high-performance and efficient 3d medical image segmentation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10435–10444, 2025. doi: 10.1109/CVPR52734.2025.00976.

Michael S Ryoo, Keerthana Gopalakrishnan, Kumara Kahatapitiya, Ted Xiao, Kanishka Rao, Austin Stone, Yao Lu, Julian Ibarz, and Anurag Arnab. Token turing machines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19070–19081, 2023.

Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Unetr++: Delving into efficient and accurate 3d medical image segmentation. *IEEE Transactions on Medical Imaging*, 43(9):3377–3390, 2024. doi: 10.1109/TMI.2024.3398728.

Yucheng Tang, Dong Yang, Wenqi Li, Holger R. Roth, Bennett Landman, Daguang Xu, Vishwesh Nath, and Ali Hatamizadeh. Self-supervised pre-training of swin transformers for 3d medical image analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20730–20740, June 2022.

Jian Wang, Hengde Zhu, Shui-Hua Wang, and Yu-Dong Zhang. A review of deep learning on medical image analysis. *Mobile Networks and Applications*, 26(1):351–380, 2021a.

Wenxuan Wang, Chen Chen, Meng Ding, Hong Yu, Sen Zha, and Jiangyun Li. Transbts: Multimodal brain tumor segmentation using transformer. In Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pp. 109–119, Cham, 2021b. Springer International Publishing. ISBN 978-3-030-87193-2.

Hongkai Wei, Yang Yang, Shijie Sun, Mingtao Feng, Rong Wang, and Xianfeng Han. Lmttm-vmi: Linked memory token turing machine for 3d volumetric medical image classification. *Computer Methods and Programs in Biomedicine*, 262:108640, 2025.

Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f2201f5191c4e92cc5af043eebfd0946-Paper.pdf.

Jiancheng Yang, Xiaoyang Huang, Yi He, Jingwei Xu, Canqian Yang, Guozheng Xu, and Bingbing Ni. Reinventing 2d convolutions for 3d images. *IEEE Journal of Biomedical and Health Informatics*, 25(8):3009–3018, 2021.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Hong-Yu Zhou, Jiansen Guo, Yinghao Zhang, Xiaoguang Han, Lequan Yu, Liansheng Wang, and Yizhou Yu. nnformer: Volumetric medical image segmentation via a 3d transformer. *IEEE Transactions on Image Processing*, 32:4036–4045, 2023. doi: 10.1109/TIP.2023.3293771.

A Additional Router Ablation

We further examine the channel and spatial components of the router on MSD Task09 Spleen. Table 4 shows that retaining either component alone reduces Dice and increases HD95 relative to the full router. The results support the use of their combination on this task; they do not establish the same effect on every dataset.

Variant	Dice ↑	HD95 ↓	Params (K) ↓	FLOPs (G) ↓
Channel only	96.45	36.07	241.7	44.26
Spatial only	96.35	36.60	215.6	45.17
RTAF	**97.47**	**25.11**	248.6	45.17

Table 4: Additional router ablation on Task09 Spleen.

B Training Protocols

MedMNIST3D. The public MedMNIST3D reference protocol uses the official training, validation, and test splits, 28^3 inputs, a batch size of 32, and 100 training epochs. Its PyTorch baseline uses Adam with an initial learning rate of 10^{-3} and cross-entropy loss. The learning rate is multiplied by 0.1 at epochs 50 and 75, and the checkpoint is selected using validation AUC. The same official data partitions and evaluation metric are used for the matched backbone comparison.

MSD with nnU-Netv2. As stated in the main paper, segmentation is evaluated over five cross-validation folds using the nnU-Netv2 pipeline. The public default trainer uses 1000 epochs with 250 iterations per epoch, SGD with momentum 0.99 and Nesterov acceleration, an initial learning rate of 10^{-2}, weight decay of 3×10^{-5}, and a polynomial learning-rate schedule. Patch size, batch size, target spacing, and normalization are planned separately for each task. The task-specific plans are held fixed for the matched nnU-Netv2 and RTAF comparison.

C Implementation of RTAF

Our implementation uses four channel groups, eight Gaussian basis functions, and CP rank eight. The Gaussian centers are shared across groups, initialized at eight equally spaced values in $[-1, 1]$, and optimized during training. For each group, the CP factors are stored as parameters and reconstructed into a group-level tensor weight at every forward pass. Gaussian responses are then generated and accumulated one basis at a time; the code does not stack all eight basis responses into a single tensor.

The channel gate uses global average pooling and two pointwise convolutions with hidden width $\max(\lfloor C/4 \rfloor, 8)$. The spatial gate uses a $3 \times 3 \times 3$ convolution followed by sigmoid. The local path consists of a depthwise $3 \times 3 \times 3$ convolution and a pointwise convolution. After concatenation, a pointwise convolution and instance normalization precede the residual addition.

D TASK-SPECIFIC TRADE-OFFS

The improvement in mean Prostate Dice does not extend to both regions. Relative to nnU-Netv2, RTAF increases transition-zone Dice from 85.68 to 89.30 but decreases peripheral-zone Dice from 73.31 to 70.60. The reported region average consequently changes only from 79.50 to 79.95. Aggregate scores alone do not show whether the peripheral-zone errors are concentrated near boundaries or in particular cases.

On SynapseMNIST3D, ResNet-50 with RTAF decreases AUC from 0.851 to 0.742 and accuracy from 0.795 to 0.741, whereas the ResNet-18 counterpart improves on both metrics. This indicates that the effect of adding RTAF is backbone-dependent on this task. The current aggregate results cannot determine whether the ResNet-50 decrease comes from run-to-run variation, optimization, or the interaction between RTAF and the deeper backbone.