CONVOLUTIONAL NEURAL NETWORK CAN SATISFY REPRESENTATION AND INTERPRETABILITY WITHOUT NONLINEAR ACTIVATION FUNCTIONS SIMULTANEOUSLY

Anonymous authors
Paper under double-blind review

ABSTRACT

Nonlinear activation functions enable the selection of features and achieve strong representation when combined with various feature mappings. However, in this paper, we find that for general visual recognition tasks, such as classification, nonlinear activation functions may actually constrain the interpretability of the model's intermediate and final outputs. We observe that nonlinear activations, particularly ReLU, discard negative responses, thereby disrupting the model's semantic attribution and potentially leading to unreasonable output attributions. Moreover, models whose feature mappings can be represented by univariate polynomials, such as conventional convolutional neural networks, inherently possess limited feature discriminability and thus lower representational capacity. This observation motivates a rethinking of effective feature representation and selection methods without relying on nonlinear activation functions. We demonstrate that simple element-wise multiplication of feature responses can achieve complete semantic attribution and enhance feature discriminability. Building on this, we further optimize the approach by extending element-wise multiplication from single-head homogeneous inputs to a multi-head heterogeneous inputs scheme, which improves the model's representational capacity and reduces inference latency. Experiments on ImageNet and visualizations of feature outputs show that NANet is the first to surpass the representational capability of activation-based models while also improving the interpretability of model.

1 INTRODUCTION

Nonlinear activation functions (Agarap, 2018; Hendrycks & Gimpel, 2016) play a critical role in visual recognition models. However, models (Ni et al., 2024b; Ma et al., 2024) without such functions have historically struggled to achieve competitive performance. Meanwhile, nonlinear activations inevitably introduce information loss (Sandler et al., 2018), making it challenging to interpret output features directly from the model's intrinsic mappings.

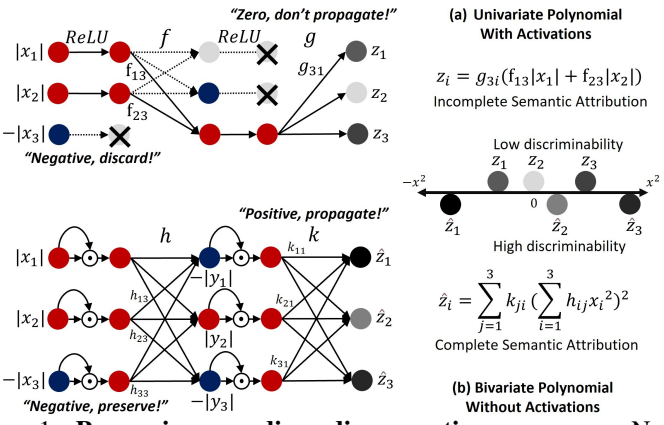

Figure 1: **Preserving vs. discarding negative responses.** Nonlinear activations discard negative responses and break semantic attribution, whereas element-wise multiplication preserves complete attribution pathways with improved feature discriminability.

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.

As shown in Figure 1 (top), nonlinear activation functions (e.g., ReLU (Agarap, 2018)) suppress negative responses generated during feature mapping by mapping them to zero. This operation disrupts the propagation pathways of shallow features associated with these negative responses, resulting in incomplete semantic attribution in deeper representations. Consequently, final predictions may be incorrectly attributed to incomplete or irrelevant shallow features, complicating model interpretation. Furthermore, the feature transformations in most existing convolutional neural networks (ConvNets) can be formulated as univariate polynomial mappings, inherently limiting their representation capacity. Therefore, simply removing activation functions from conventional ConvNets for interpretability substantially reduces their ability to learn effective feature representations.

This paper aims to investigate how a model can *preserve complete semantic attribution while maintaining effective feature representation and selection.* We find that a simple element-wise multiplication operation can achieve this objective. As illustrated in Figure 1 (bottom), element-wise multiplication transforms negative responses into positive values, allowing them to propagate forward, which maintains the complete semantic attribution. Meanwhile, the resulting squaring effect amplifies responses with large magnitudes while suppressing weak responses. This improves feature separability and enables effective representation and selection without relying on nonlinear activation functions.

At the model level, we propose directly replacing nonlinear activation functions in ConvNets with element-wise multiplication, as shown in Figure 2 (left), allowing the model to achieve feature representation and selection through bivariate polynomial mappings only. Specifically, we replace the GELU (Hendrycks & Gimpel, 2016) activation functions in ConvNeXt (Liu et al., 2022) with element-wise multiplication, achieving complete semantic attribution and enhanced feature discriminability. Furthermore, we extend element-wise multiplication from a single-head homogeneous input scheme to a multi-head heterogeneous input scheme, resulting in the final **N**on-**A**ctivation Conv**Net** (**NANet**). This design significantly reduces inference latency, improves model performance, and overcomes the limitations of nonlinear activation functions on interpretability.

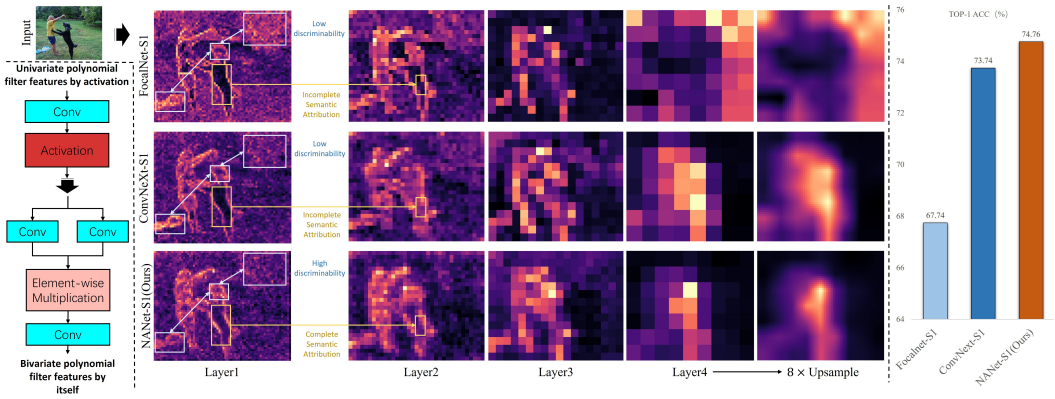

Figure 2: **Element-wise multiplication unlocks complete semantic attribution and enhanced feature discriminability.** *Left:* Replacing nonlinear activation functions with element-wise multiplication allows the model to perform feature representation and selection through bivariate polynomial mappings. *Middle:* Visualization of layer-wise output features shows that NANet preserves complete semantic attribution and better distinguishes target-related features from irrelevant ones, resulting in more interpretable representations. *Right:* Classification results demonstrate that NANet achieves effective feature representation without nonlinear activation functions, surpassing activated models.

We visualize the final output feature maps from each layer of NANet and two activation-based models (FocalNet (Yang et al., 2022) and ConvNeXt (Liu et al., 2022)), as shown in the middle of Figure 2. FocalNet employs GELU in every gating operation, resulting in frequent feature filtering in shallow layers. Consequently, torso features highly relevant to the "dog" class may be suppressed due to negative responses *not* associated with shallow edge features. Such information cannot propagate to deeper layers, causing incomplete semantic attribution and potentially leading to erroneous interpretations of the final feature representations. Similarly, although ConvNeXt employs fewer

activation functions, important torso information is still suppressed, limiting the interpretability of the final feature maps.

In contrast, NANet demonstrates stronger discrimination between dog-related and irrelevant features, whereas FocalNet and ConvNeXt fail to establish such clear distinctions. Moreover, NANet effectively preserves complete semantic information related to the "dog" class while distinguishing primary and secondary discriminative features (e.g., NANet has a complete outline of the "dog" category and head regions contribute more significantly). Classification accuracy further confirms that NANet achieves effective feature representation without nonlinear activation functions, outperforming activation-based models, as shown in Figure 2 (right).

We conduct extensive experiments on ImageNet. The results demonstrate that NANet substantially outperforms existing activation-free models, surpasses efficient activation-based models, and achieves comparable performance to large-scale models with nonlinear activations. Visualization of the intermidiate and final feature maps further demonstrates that NANet naturally focuses on target-related contour regions with clear semantic emphasis, without relying on post-hoc explanation methods.

2 ConvNet without Nonlinear Activation Functions

Existing visual recognition models, such as ConvNeXt (Liu et al., 2022) shown in Figure 3 (a), primarily rely on univariate polynomial representations (i.e., convolution-based univariate polynomial formulations in convolutional neural networks (ConvNets)) and nonlinear activation functions for feature representation and selection. The introduction of nonlinear activation functions in ConvNets is based on the implicit assumption that negative responses generated by convolutional aggregation are uninformative. During forward propagation, these activation functions (Agarap, 2018; Hendrycks & Gimpel, 2016) block or suppress negative responses. However, this design introduces two major challenges:

Incomplete semantic attribution. During forward propagation, nonlinear activation functions such as ReLU (Agarap, 2018) set negative responses generated by convolutional aggregation to zero and discard them, preventing the semantic information contained in these responses from being propagated to deeper layers. As nonlinear activation functions are repeatedly stacked, semantic attribution of deep ConvNet features become increasingly incomplete. Consequently, final ConvNet representations can only be attributed to hierarchical combinations of a subset of positive responses. The absence of negative responses introduces bias into the semantic evidence underlying model decisions, thereby limiting prediction interpretability.

Reduced feature discriminability. ConvNets distinguish feature responses through convolution operations, which compute weighted combinations of input features, resulting in limited discriminability. Furthermore, the large discrepancy originally existing between positive and negative responses is reduced when negative responses are clipped to zero by activation functions. This also decreases the discriminability of ConvNets.

2.1 ConvNet with Bivariate Polynomial

To address these issues, we propose replacing nonlinear activation functions with simple element-wise multiplication operations in ConvNets. This design is motivated by the observation that negative responses are not inherently meaningless. Activation-based networks cannot distinguish different magnitudes of negative responses generated by convolution kernels. In contrast, feature self-multiplication transforms negative responses into positive values, preserving all responses and maintaining complete semantic attribution. Meanwhile, the quadratic interaction between responses amplifies differences among positive and negative activations: strong responses are enhanced, whereas weak responses are suppressed, improving response-level discriminability.

2.1.1 Element-wise Multiplication with Homogeneous Input

As shown in Figure 3 (c), we modify ConvNeXt (Liu et al., 2022) by replacing the GELU (Hendrycks & Gimpel, 2016) activation function with an element-wise multiplication operation. Ac-

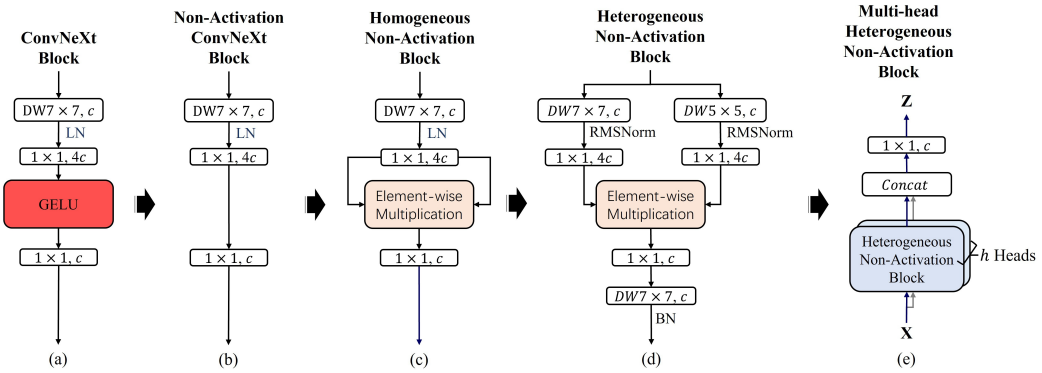

Figure 3: **Main block architectures of different model variants.** (a) ConvNeXt Block. (b) ConvNeXt Block without GELU. (c) Homogeneous-input Non-Activation Block, replacing GELU with element-wise multiplication. (d) Heterogeneous-input Non-Activation Block. (e) Multi-head architecture with heterogeneous-input Non-Activation Blocks. Residual connections from input to output are omitted for simplicity.

cordingly, the original univariate polynomial representation with activation-based feature selection is transformed into a bivariate polynomial representation. The output is obtained through element-wise multiplication between two homogeneous convolutional branches.

2.1.2 Element-wise Multiplication with Heterogeneous Input

Furthermore, we employ two heterogeneous convolution operations with contextual interactions as inputs to the element-wise multiplication module, as illustrated in Figure 3 (d). Compared with homogeneous convolution branches, this design introduces an additional response pattern where the two convolution outputs have opposite signs, besides both-positive and both-negative cases. This enhances response flexibility during propagation and enables the network to adaptively determine feature importance according to learned feature mappings.

2.1.3 Multi-head Mechanism with Heterogeneous Non-Activation Block

Finally, we incorporate a multi-head mechanism (Vaswani et al., 2017) into the NANet Block. Each head extracts relatively independent feature representations, thereby improving feature diversity. Let h denote the number of heads. Unlike conventional multi-head architectures, we do not apply separate projection mappings before splitting the input. Instead, the input is directly divided into h parts and fed into the corresponding heads. The outputs of all heads are then aggregated through a projection layer to integrate information across different heads, as illustrated in Figure 3 (e).

2.2 Towards Efficient ConvNet without Nonlinear Activation Functions: a Roadmap

In this section, we explore the trajectory of transforming ConvNeXt (Liu et al., 2022) into an efficient Non-Activation ConvNet by removing nonlinear activation functions (Hendrycks & Gimpel, 2016) and establishing bivariate polynomial representation, as illustrated in Figure 3, and analyze the performance of the main variants in terms of top-1 accuracy on ImageNet-1K (Deng et al., 2009) and inference latency on iPhone 13, as illustrated in Figure 4.

2.2.1 Training Strategy

All models in the trajectory, from ConvNeXt to NANet using Multi-head Heterogeneous Non-Activation Block, adopt a training scheme similar to DeiT (Touvron et al., 2021). We optimize the training recipe with a batch size of 4096, a learning rate of 4×10^{-3}, and a stochastic depth drop rate of 0.05 for ConvNeXt and NA-ConvNeXt, as ConvNeXt (Liu et al., 2022) did originally in their paper. We further optimize the training recipe with a batch size of 1024 and a learning

rate of 3×10^{-3}. We apply this training strategy to models with Homogeneous Non-Activation Block, Heterogeneous Non-Activation Block and Multi-head Heterogeneous Non-Activation Block in the process. The accuracy of each variant is averaged across training runs with three random seeds. To test inference latency on mobile devices, we deploy the models on an iPhone 13 using CoreML-Tools.

2.2.2 From ConvNeXt to Efficient Non-Activation ConvNet

Our exploration roadmap is summarized as follows. We first reduce the channel width of the ConvNeXt-T (Liu et al., 2022) block from 96 to 30, resulting in a compact ConvNeXt block. Based on this reduced block, we train a lightweight ConvNeXt model ($3M$ parameters) as our starting point. Experimental results show that ConvNeXt ($3M$) achieves a competitive Top-1 accuracy of 73.74%, which is comparable to StarNet (Ma et al., 2024) with a similar parameter budget (73.5% Top-1 accuracy).

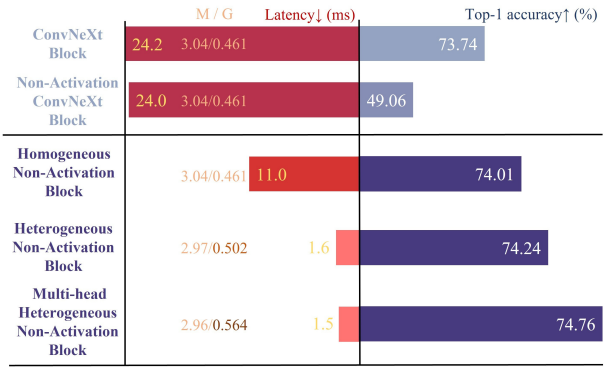

Figure 4: The evolution from ConvNeXt to an efficient Non-Activation ConvNet. Each variant is constructed using a different block design as the core building module.

We further evaluate the performance of the ConvNeXt block after removing all GELU (Hendrycks & Gimpel, 2016) nonlinear activation functions. The Top-1 accuracy drops significantly to 49.06%, indicating that prior to our study, the incomplete attribution pathways and reduced discriminability introduced by nonlinear activation functions were considered unavoidable trade-offs for achieving acceptable visual recognition accuracy in ConvNets.

Next, we replace the 4×4 convolution (stride=4) in the stem module with two stacked 3×3 convolutions (stride=2 and padding=1) and replace the 2×2 convolution (stride=2) in the downsample module with one 3×3 convolution (stride=2 and padding=1). After that, we investigate models equipped with the Homogeneous Non-Activation Block, where element-wise multiplication replaces the GELU activation function to preserve complete semantic attribution and enhance discriminability. The results demonstrate that replacing nonlinear activations with element-wise multiplication is not only effective but also efficient.

Furthermore, we evaluate the model with Heterogeneous Non-Activation Block, which introduces heterogeneous inputs with greater response flexibility. Experimental results show that this heterogeneous design improves classification accuracy while substantially reducing inference latency, making the model more suitable for efficient visual recognition.

Finally, incorporating a multi-head mechanism into the Heterogeneous Non-Activation Block further enhances the representation capability of the model, improving the Top-1 accuracy to 74.76%. Therefore, the proposed Non-Activation ConvNet (NANet) based on the Multi-head Heterogeneous Non-Activation Block achieves both effectiveness and efficiency for visual recognition while overcoming the inherent limitations of nonlinear activation functions. By preserving complete feature mappings, the network can adaptively determine feature importance based on feature interactions, thereby improving model interpretability.

3 Experiment: How effective is NANet?

Although nonlinear activation functions introduce certain limitations, they have played an essential role in improving the representation capability of deep neural networks. Therefore, before attributing improved interpretability to the removal of activation functions, it is necessary to verify that the proposed activation-free NANet can maintain competitive representation capability. We trained NANet using training strategies for NANet above, testing its top-1 classification accuracy on ImageNet-1K (Deng et al., 2009).

3.1 SIGNIFICANTLY LEADING ACTIVATION-FREE MODELS

To compare the performance of contemporary attempts on activation-free models, we design several NANet variants, with Multi-head Heterogeneous Non-Activation Block, of different scales.

The experimental results are shown in Table 1. NANet achieves significant improvements in classification accuracy. This demonstrates that, after removing all nonlinear activation functions, NANet can perform feature discrimination accurately. Compared with contemporary activation-free models, NANet enables more effective feature representation through learned feature mappings. This design preserves complete semantic attribution and provides a more faithful basis for model interpretation.

Table 1: Comparison of NANet and activation-free models on ImageNet-1K.

Model	**Params** (M)	**FLOPs** (G)	**TOP-1** ($\%$)
DTTN-T (2025)	7.1	2.4	77.9
NANet-S3	5.9	1.2	**78.3**
MONet-T (2024)	10.3	2.8	77.0
$\mathcal{D}$-PolyNets (2023)	11.4	-	70.0
DTTN-S (2025)	12.0	4.1	79.4
Π-Nets (2020)	12.3	1.9	65.2
$\mathcal{R}$-PolyNets (2023)	12.4	-	70.2
NANet-S4	7.6	1.5	**79.2**
MONet-S (2024)	32.9	6.8	81.3
DTTN-B (2025)	36.0	12.3	82.4
NANet-B	30.0	5.2	**82.5**

3.2 SLIGHTLY SURPASSING EFFICIENT ACTIVATED MODELS

Table 2: Comparison of NANet and efficient models with nonlinear activation functions on ImageNet-1K. Latency is evaluated on an iPhone 13 mobile device with the benchmark batch size set to 1, as in a real-world setting

Model	**Params**(M)	**FLOPs**(M)	**Latency**(ms)	**TOP-1**($\%$)
MobileOne-S0 (2023)	2.1	275	0.7	71.4
ShuffleV2-1.0 (2018)	2.3	146	4.1	69.4
MobileV3-S0.75 (2019)	2.4	44	5.5	65.4
GhostNet0.5 (2020)	2.6	42	10.0	66.2
MobileV3-S (2019)	2.9	66	6.5	67.4
StarNet-S1 (2024)	2.9	425	0.7	73.5
NANet-S1	3.0	564	1.5	**74.8**
MobileV2-1.0 (2018)	3.4	300	0.9	72.0
Shufflenetv2 1.5 (2018)	3.5	299	5.9	72.6
Mobileformer-52 (2022b)	3.6	52	6.6	68.7
FasterNet-T0 (2023)	3.9	338	0.7	71.9
StarNet-S2 (2024)	3.7	547	0.7	74.8
NANet-S2	3.9	754	1.8	**76.5**
MobileV3-L0.75 (2019)	4.0	155	10.9	73.3
EdgeViT-XXS (2022)	4.1	559	1.8	74.4
MobileOne-S1 (2023)	5.2	825	0.9	75.9
GhostNet1.0 (2020)	5.2	141	7.9	73.9
EfficientNet-B0 (2019)	5.3	390	1.6	77.1
MobileV3-L (2019)	5.4	219	11.4	75.2
StarNet-S3 (2024)	5.8	757	0.9	77.3
NANet-S3	5.9	1146	2.1	**78.3**
EdgeViT-XS (2022)	6.8	1166	3.5	77.5
MobileV2-1.4 (2018)	6.9	585	1.1	74.7
GhostNet1.3 (2020)	7.3	226	9.7	75.7
ShuffleV2-2.0 (2018)	7.4	591	19.9	74.9
FasterNet-T1 (2023)	7.6	855	0.9	76.2
MobileOne-S2 (2023)	7.8	1299	1.0	77.4
StarNet-S4 (2024)	7.5	1075	1.0	78.4
NANet-S4	7.6	1459	2.2	**79.2**

Furthermore, we test the top-1 classification accuracy of NANet and existing efficient models with activation functions on ImageNet-1K, as well as their inference latency on an iPhone 13. The experimental results are presented in Table 2. Compared to existing efficient models with activation functions, NANet achieves better classification accuracy while maintaining acceptable inference latency.

3.3 COMPARABLE TO LARGE-SCALE ACTIVATED MODELS

Finally, the scalability of activation-free models is another important aspect that requires investigation. We further evaluate the classification performance of NANet when scaling the model size to $15M$, $30M$, and $50M$ parameters. Experimental results show that NANet achieves representation capabilities comparable to a range of large-scale ConvNets and Transformer-based models. These results verify that removing nonlinear activation functions does not compromise the scalability of ConvNets, while providing a more interpretable feature interaction mechanism.

Table 3: Comparison of NANet and large-scale models with nonlinear activation functions on ImageNet-1K.

Model	**Params** (M)	**FLOPs** (G)	**TOP-1** (%)
CoAtNet-0 (2021)	25.0	4.0	81.6
DeiT-S (2021)	22.0	4.6	79.8
UniRepLKNet-N (2024)	18.3	2.8	81.6
RepViT-M1.5 (2024)	14.0	2.3	81.2
PVTv1-Tiny (2021)	13.2	1.9	75.1
NANet-T	15.0	2.5	**81.5**
DeiT-B (2021)	87.0	17.6	81.8
Swin-T (2021)	29.0	5.0	81.3
ConvNeXt-T (2022)	29.0	5.0	82.1
Focal-T (2023)	29.0	5.0	82.2
NANet-B	30.0	5.2	**82.5**
RegNet-Y-16G (2020a)	84.0	16.0	82.9
Swin-S (2021)	50.0	9.0	83.0
ConvNeXt-S (2022)	50.0	9.0	83.1
NANet-L	49.8	8.8	**83.1**

4 Quantitative Visualization: How interpretable is NANet?

4.1 Interpretability of different architecture

To investigate the interpretability of different architectures, we extract the output feature tensor from the last block of the final stage for each model. Each pixel in the feature tensor is a direct mapping result of the shallow image associated with it. These feature tensors are obtained directly from the final outputs of the feature aggregation process, without applying any post-hoc explanation methods such as CAM (Zhou et al., 2016) or Grad-CAM (Selvaraju et al., 2017). For visualization, each feature tensor is normalized to the range of [0,1] along the channel dimension, which is the basic operation for classification, and then upsampled by 8 times and rendered as the final feature map, as shown in Figure 5. Feature visualization remains faithful to the quantitative mapping relationship among features, without affecting the inherent interpretability of the model. A brighter red color means that this part of the features has a greater influence in the final feature recognition (e.g. classification), while darker areas have a smaller impact.

The evaluated models include FocalNet, which employs a nonlinear activation function for every gating operation (nonlinear modulation); ConvNeXt, a conventional ConvNet with a small number of activation functions; and NANet, which completely eliminates nonlinear activation functions. As illustrated in Figure 5, NANet consistently produces feature maps with clearer category-related object contours across diverse scenes. Furthermore, the aggregated regions are more concentrated on the vital part of the target objects.

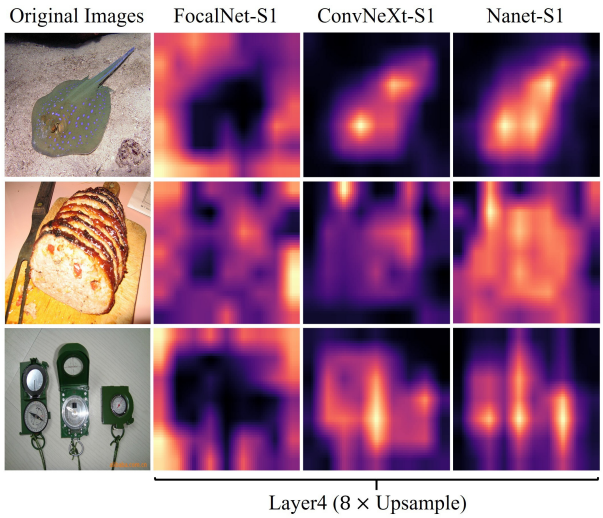

Figure 5: NANet exhibits inherent interpretability by progressively concentrating on category-relevant regions during inference, without requiring post-hoc visual explanation methods such as CAM or Grad-CAM.

These observations suggest that the proposed feature interaction mechanism preserves more complete feature mappings throughout the forward propagation process. As a result, the final feature representations naturally highlight category-relevant regions without relying on post-hoc explanation methods, providing more interpretable evidence for the model's predictions.

4.2 INTERPRETABILITY OF DIFFERENT STAGES

We visualize the intermediate feature representations of the three models at different stages. Each feature tensor is normalized to the range of [0,1] along the channel dimension for layer 1 to 4 without upsample, which is more faithful to the quantitative mapping relationships among features. As shown in Figure 6, NANet consistently generates semantically aligned features throughout the entire feature aggregation process, concentrated on class-relevant object contours. In contrast, activation-based models may lose partial semantic structures at intermediate stages due to feature filtering. For example, FocalNet fails to preserve complete tortoise shell contours in its third-stage features. Furthermore, some intermediate representations of ConvNeXt are dominated by class-irrelevant regions, despite their seemingly interpretable final outputs, as observed in the second-stage features of ConvNeXt images. These observations demonstrate that NANet maintains more consistent and interpretable representations across all stages, benefiting from complete semantic attribution without nonlinear activation functions.

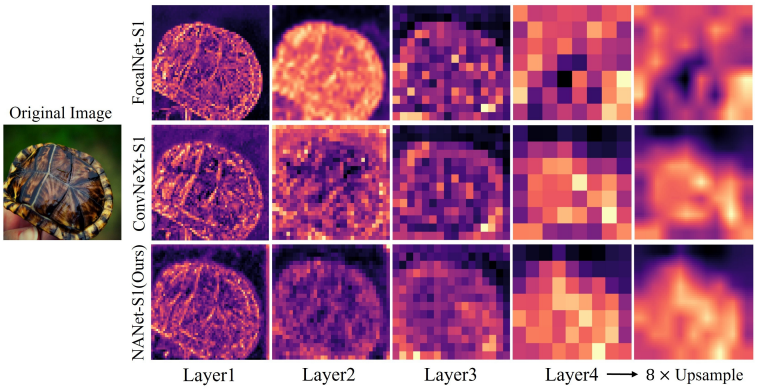

Figure 6: NANet achieves more interpretable quantitative mappings with complete semantic attribution across all network stages.

5 ABLATION STUDY: HOW WELL ELEMENT-WISE MULTIPLICATION WORKS?

We also conducted additional ablation experiments on considering element-wise multiplication.

5.0.1 THE PERFORMANCE OF NANET WITH NONLINEAR ACTIVATION FUNCTIONS

Throughout the construction of NANet, no nonlinear activation functions are employed. We further investigate the effect of reintroducing nonlinear activation functions after the element-wise multiplication operation. Experimental results show that adding ReLU after element-wise multiplication provides almost no performance improvement, whereas adding GELU leads to a slight improvement.

As illustrated in Figure 1, ReLU directly discards negative feature responses. The comparable performance after introducing ReLU indicates that element-wise multiplication itself provides sufficient nonlinear feature interaction and feature selection capability, without requiring ReLU-based feature selection.

Furthermore, negative responses generated after element-wise multiplication represent opposite responses between the convolutional features extracted from the two branches. When both branches produce strong but opposite responses, element-wise multiplication generates a large negative response. In this case, GELU improves feature discrimination by suppressing these strongly conflicting negative responses, which explains the performance improvement after introducing GELU. These results suggest that nonlinear activation functions are not fundamentally required for feature selection in NANet. Instead, the multiplicative interaction between feature branches naturally provides nonlinear discrimination capability while preserving complete response information for interpretability.

5.0.2 NANET VS. UNIVARIATE POLYNOMIAL WITH ACTIVATIONS

We next compare NANet, which performs both feature representation and selection solely through element-wise multiplication, with conventional models that employ univariate polynomial representations together with nonlinear activation functions for feature selection. Specifically, we replace the

Table 4: Ablation study of element-wise multiplication.

NANet variants	Multiplication / Addition	None / ReLU / GELU	RMSNorm / BN	TOP-1 Accuracy
NANet	Multiplication	None	RMSNorm	74.76%
+ ReLU	Multiplication	ReLU	RMSNorm	74.80%
+ GELU	Multiplication	GELU	RMSNorm	75.56%
Mul → Add & + ReLU	Addition	ReLU	RMSNorm	74.50%
Mul → Add & + GELU	Addition	GELU	RMSNorm	75.45%
Add + BN	Addition	None	BN	2.80%
Mul → Add	Addition	None	RMSNorm	61.09%
RMSNorm → BN	Multiplication	None	BN	69.64%

element-wise multiplication operation in NANet with element-wise addition, followed by a nonlinear activation function.

Experimental results show that, compared with the activation-free NANet, the linear variant equipped with ReLU exhibits inferior performance, while the counterpart using GELU achieves only a marginal improvement. These results suggest that element-wise multiplication naturally integrates feature representation and feature selection into a unified operation. Instead of relying on explicit nonlinear activation functions to suppress feature responses, NANet determines feature importance through multiplicative feature interactions, providing an alternative mechanism for nonlinear feature modeling.

5.0.3 Element-wise multiplication vs. RMSNorm

We first evaluate the model after removing all three sources of nonlinearity, namely nonlinear activation functions, element-wise multiplication, and RMSNorm. Under this setting, the Top-1 classification accuracy collapses to 2.8%, indicating that the nonlinear representation capability of the model is primarily derived from these three components.

Since layer normalization (Ni et al., 2024b) has been shown to introduce nonlinear representational capability, while element-wise multiplication is inherently nonlinear, we further investigate two variants of NANet: one retaining only RMSNorm (Zhang & Sennrich, 2019) (a variant of layer normalization) and the other retaining only element-wise multiplication as the source of nonlinearity. Experimental results show that retaining only element-wise multiplication results in only a slight performance degradation, whereas retaining only RMSNorm causes a substantial drop in accuracy compared with the full NANet. These results suggest that element-wise multiplication provides considerably stronger nonlinear representation capability than RMSNorm.

A possible explanation is that layer normalization applies the same nonlinear transformation to all features within a layer, regardless of their spatial locations or channel identities. Such a globally shared transformation limits the model's ability to discriminate among different feature responses, making layer normalization alone insufficient for effective nonlinear feature modeling. Unlike the globally shared nonlinearity introduced by layer normalization, element-wise multiplication generates input-dependent nonlinear transformations through feature interactions. This adaptive nonlinearity enables different spatial locations to undergo different transformations, thereby improving feature discriminability and representation capability.

6 Conclusion

We revisit nonlinear activation functions in visual recognition models and show that their information loss can disrupt semantic attribution and limit interpretability. We propose replacing nonlinear activations with element-wise multiplication, enabling complete semantic propagation and effective feature selection through polynomial mappings. Based on this idea, we develop Non-Activation ConvNet (NANet) with a multi-head heterogeneous multiplication design. Experiments on ImageNet demonstrate that NANet achieves competitive performance without nonlinear activations, outperforming existing activated and activation-free models while producing more interpretable representations. This work highlights the potential of polynomial transformations as an alternative for thorough interpretability.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have provided complete code and documentation in the supplementary materials.

REFERENCES

Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: chasing higher flops for faster neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12021–12031, 2023.

Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European conference on computer vision*, pp. 17–33. Springer, 2022a.

Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5270–5279, 2022b.

Yixin Cheng, Grigorios Chrysos, Markos Georgopoulos, and Volkan Cevher. Multilinear operator networks. In *International Conference on Learning Representations*, volume 2024, pp. 56698–56725, 2024.

Grigorios G Chrysos, Stylianos Moschoglou, Giorgos Bouritsas, Yannis Panagakis, Jiankang Deng, and Stefanos Zafeiriou. P–nets: Deep polynomial neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7323–7333. IEEE, 2020.

Grigorios G Chrysos, Bohan Wang, Jiankang Deng, and Volkan Cevher. Regularization of polynomial networks for image recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16123–16132. IEEE, 2023.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Xiaohan Ding, Yiyuan Zhang, Yixiao Ge, Sijie Zhao, Lin Song, Xiangyu Yue, and Ying Shan. Unireplknet: A universal perception large-kernel convnet for audio, video, point cloud, time-series and image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.

Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *Computational visual media*, 9(4):733–752, 2023.

Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1580–1589, 2020.

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

Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng. Conv2former: A simple transformer-style convnet for visual recognition. *arXiv preprint arXiv:2211.11943*, 2022.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. *Advances in neural information processing systems*, 30, 2017.

Siyuan Li, Zedong Wang, Zicheng Liu, Cheng Tan, Haitao Lin, Di Wu, Zhiyuan Chen, Jiangbin Zheng, and Stan Z Li. Moganet: Multi-order gated aggregation network. In *The Twelfth International Conference on Learning Representations*, 2023.

Weifeng Lin, Ziheng Wu, Jiayu Chen, Jun Huang, and Lianwen Jin. Scale-aware modulation meet transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6015–6026, 2023.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Xu Ma, Xiyang Dai, Yue Bai, Yizhou Wang, and Yun Fu. Rewrite the stars. *arXiv preprint arXiv:2403.19967*, 2024.

Yunhao Ni, Yuxin Guo, Junlong Jia, and Lei Huang. On the nonlinearity of layer normalization. *arXiv preprint arXiv:2406.01255*, 2024a.

Yunhao Ni, Yuxin Guo, Junlong Jia, and Lei Huang. On the nonlinearity of layer normalization. *arXiv preprint arXiv:2406.01255*, 2024b.

Chang Nie. Deep tree tensor networks. *arXiv preprint arXiv:2502.09928*, 2025.

Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais Martinez. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *European Conference on Computer Vision*, pp. 294–311. Springer, 2022.

Ilija Radosavovic, Justin Johnson, Saining Xie, Wan-Yen Lo, and Piotr Dollár. On network design spaces for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1882–1890, 2019.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020a.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020b.

Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser Nam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *Advances in Neural Information Processing Systems*, 35:10353–10366, 2022.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Mingxing Tan and QV Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 5, 2021.

Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2820–2828, 2019.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Mobileone: An improved one millisecond mobile backbone. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7907–7917, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ao Wang, Hui Chen, Zijia Lin, Hengjun Pu, and Guiguang Ding. Repvit: Revisiting mobile cnn from vit perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.

Jeffrey Wang, Jonathan Gregory, and Grigorios G Chrysos. Activation-free backbones for image recognition: Polynomial alternatives within metaformer-style vision models. *arXiv preprint arXiv:2605.20839*, 2026.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 568–578, 2021.

Yuhao Wang and Wei Xi. Uniconvnet: Expanding effective receptive field while maintaining asymptotically gaussian distribution for convnets of any scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20922–20933, October 2025.

Syed Talal Wasim, Muhammad Uzair Khattak, Muzammal Naseer, Salman Khan, Mubarak Shah, and Fahad Shahbaz Khan. Video-focalnets: Spatio-temporal focal modulation for video action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13778–13789, 2023.

Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. *Advances in Neural Information Processing Systems*, 35:4203–4217, 2022.

Guhnoo Yun, Juhan Yoo, Kijung Kim, Jeongho Lee, and Dong Hwan Kim. Spanet: Frequency-balancing token mixer using spectral pooling aggregation modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6113–6124, 2023.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *The Seventh International Conference on Learning Representations*, 2018a.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018b.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 13001–13008, 2020.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

A Related Work

Activation-free models NAFNet (Chen et al., 2022a) found that non-linear activation functions (such as Sigmoid, ReLU (Agarap, 2018), GELU (Hendrycks & Gimpel, 2016), Softmax, etc.) are not necessary: they can be replaced or removed by multiplication. PolyNets (Chrysos et al., 2023) has demonstrated that strongly regularized polynomial networks can achieve similar performance to other polynomial networks without activation functions. MONet (Cheng et al., 2024) proposed a model that relies solely on multilinear operators without activation functions, which captures the multiplication interaction of input token elements. DTTN (Nie, 2025) captures $2L$ order multiplication interactions between features through multilinear operations and essentially unfolds into a tree like TN topology with parameter sharing characteristics. PolyNeXt (Wang et al., 2026) has designed an inactive polynomial replacement scheme for three core primitives (MLP, convolution, and attention), where Hadamard product replaces standard nonlinearity to obtain the input polynomial function. But in its PolyNeXt cell, the sigmoid activation function is still used for the parameters, and it is not a completely activation-free model. Although these models have attempted various model architectures without activation functions, satisfactory performance has yet to be achieved.

Multiplication in neural networks. Models such as Conv2Former (Hou et al., 2022) employ modulation operations (element-wise multiplication) to achieve nonlinear representation. Architectures like VAN (Guo et al., 2023), FocalNet (Yang et al., 2022), Hornet (Rao et al., 2022), and some other models (Wang & Xi, 2025; Li et al., 2023; Lin et al., 2023; Yun et al., 2023; Wasim et al., 2023) have introduced innovations centered around modulation operations. StarNet (Ma et al., 2024) further analyzed the properties of element-wise multiplication from the perspective of feature dimensions. However, these models still rely on activation functions for acceptable performance. In this work, we explore a novel approach to effective representation that eliminates the activation functions, thereby maintaining performance and improving interpretability.

Efficient neural networks. Efficient networks aim to maximize performance while minimizing both the parameter scale and inference latency. Recent advancements in this field leverage techniques such as depthwise convolution (Howard et al., 2017; Sandler et al., 2018), reparameterization (Li et al., 2023; Wang et al., 2024), neural architecture (Tan & Le, 2019; 2021; Tan et al., 2019) search (NAS), and feature reorganization (Zhang et al., 2018b; Ma et al., 2018; Han et al., 2020; Chen et al., 2023). However, these approaches remain constrained by activation functions (Agarap, 2018; Hendrycks & Gimpel, 2016), which compromise interpretability of the model. We

demonstrate that effective nonlinear representation can ensure all of effectiveness, efficiency and interpretability.

B More visualizations of Interpretability

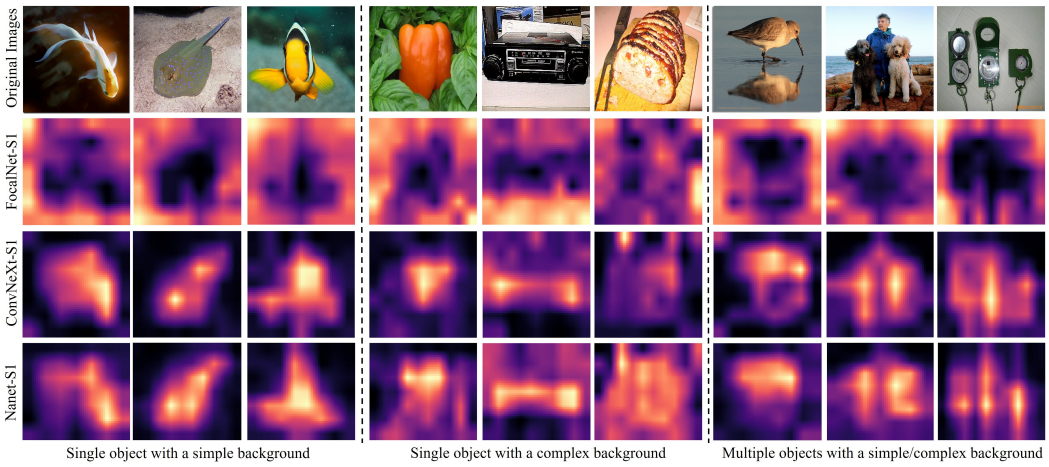

Figure 7: NANet exhibits inherent interpretability across various scenarios.

B.1 Interpretability of different scenario

We further classify the scene into (1) single object simple background, (2) single object complex background, and (3) multiple objects. On this basis, more final feature maps were visualized. The results show that the feature discrimination criteria of NANet in various scenarios fall in areas that humans can better understand.

B.2 Interpretability of different stages

We further visualize more intermediate feature representations of three models at different stages. As shown in Figure 8, NANet consistently generates semantically aligned features throughout the entire feature aggregation process, concentrated on class-relevant object contours. In contrast, activation-based models may lose partial semantic structures at intermediate stages due to feature filtering. For example, FocalNet and ConvNeXt fail to preserve complete bird contours in its third-stage features. Furthermore, some intermediate representations of ConvNeXt are dominated by class-irrelevant regions, despite their seemingly interpretable final outputs, as observed in the second-stage features of watch images.

C Contemporary Activated ConvNets without Nonlinear Activation Functions

To compare the performance of NANet variants with contemporary activated ConvNets without nonlinear activation functions, we select or modify ConvNeXt (Liu et al., 2022), StarNet (Ma et al., 2024), and Conv2former (Hou et al., 2022) to match the model scales of NANet S1-S4 and remove all activation functions from them. Subsequently, we train Non-Activation ConvNeXt, StarNet and Conv2former using their respective original training strategies for the S1-S4 models scales, testing their performance after removing activations under their optimal training conditions.

The experimental results are shown in Figure 9. NANet achieves significant improvements in classification accuracy. Compared with the univariate polynomial representation of ConvNeXt and the bivariate polynomial representations of StarNet and Conv2Former, NANet achieves significant performance improvements in the absence of nonlinear activation functions.

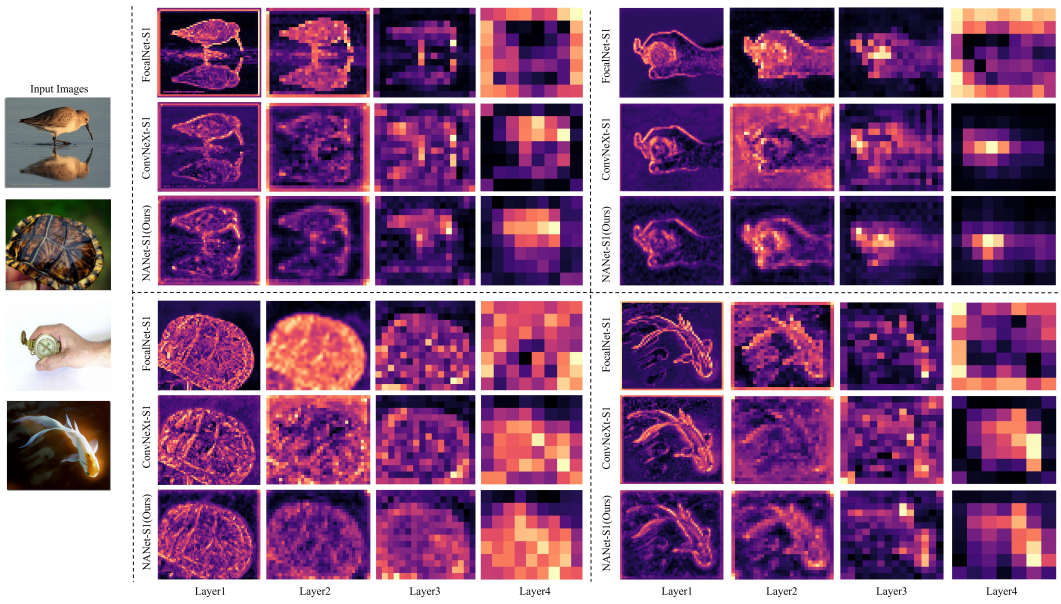

Figure 8: NANet achieves consistent interpretability across all network stages.

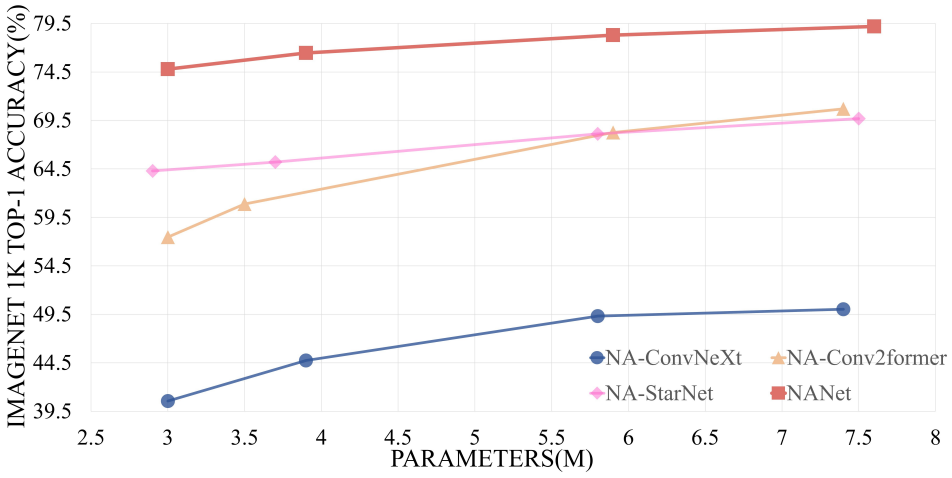

Figure 9: Comparison of NANet and ConvNets without nonlinear activation functions on ImageNet-1K.

D TOWARDS EFFICIENT NANET: A ROADMAP

In this section, we explore the trajectory of transforming ConvNeXt (Liu et al., 2022) into NANet by removing activation functions (Hendrycks & Gimpel, 2016) and establishing effective representation and selection, analyzing the performance of main steps in terms of top-1 accuracy on ImageNet-1K (Deng et al., 2009) and inference latency on iPhone 13. We deconstruct the process of transforming ConvNeXt (Liu et al., 2022) block into Heterogeneous Non-Activation Block into six steps: (1) removing activation functions, (2) performing element-wise multiplication of patterns with themselves, (3) separating the MLP module, (4) splitting the convolution and LN (Ba et al., 2016) modules, (5) reducing one kernel size from 7×7 to 5×5, and (6) adding a 7x7 convolution with Batch Normalization (BN) (Ioffe, 2017), as illustrated in Figure 10.

The roadmap of our exploration is as follows. We first compress the width (Zagoruyko & Komodakis, 2016) of ConvNeXt-T (Liu et al., 2022) (30M) from 96 to 30 and adopt a training recipe similar to DeiT (Touvron et al., 2021), resulting in the ConvNeXt-S1 model (3M) as our starting point. We remove all activation functions (Agarap, 2018) from the block and stem modules, then redesign the stem and down-sample modules by overlapping convolutions. Subsequently, we perform steps ② - ⑥ to nonlinearize the ConvNeXt (Liu et al., 2022) block. After separating the MLP module, the model parameters increase significantly, so we adopt a more suitable configuration for efficient models, compressing the width (Zagoruyko & Komodakis, 2016) and depth (Simonyan & Zisserman, 2014), to maintain a consistent model scale. In the final step, we find that using an IB block with $1.25\times$ channels for the stem module effectively enhances pattern diversity, further improving model performance. Finally, we apply the multi-head mechanism for the block acquired by the operations above and get the eventually blocks for NANet. Since the primary focus for deploying and inferring efficient models is parameter count and inference latency, we use parameter scale as a key metric to evaluate model scale during this trajectory. All models in this trajectory use the same training strategy as ConvNeXt-S1 to train and test on ImageNet-1K (Deng et al., 2009), with inference latency measured on an iPhone 13.

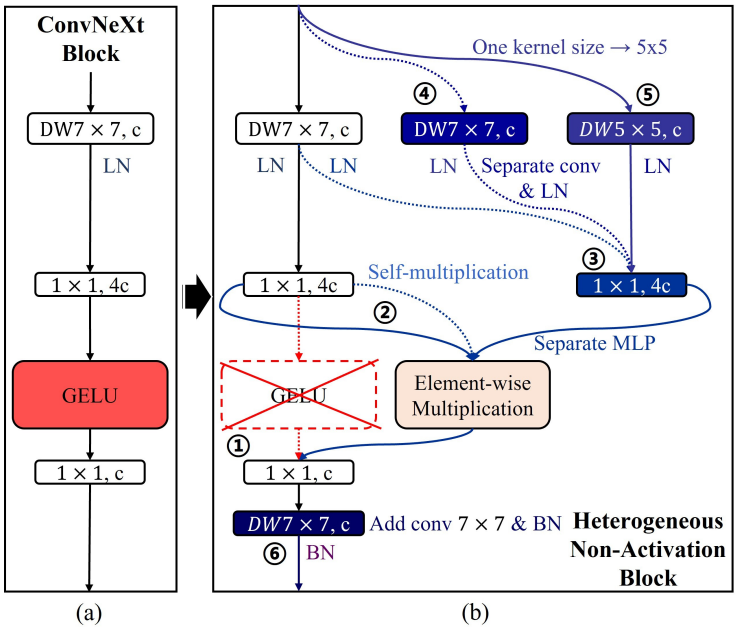

Figure 10: **Trajectory from ConvNeXt Block to Heterogeneous Non-Activation Block.** We systematically transform the ConvNeXt Block into the Heterogeneous Non-Activation Block through six distinct steps. At each stage, the corresponding model is trained independently to evaluate its performance and assess the contribution of each modification to the overall architecture.

D.1 Training Strategy

All models in the trajectory, from ConvNeXt-S1 to NANet-S1, adopt a training scheme similar to DeiT (Touvron et al., 2021). We optimize NANet with a batch size of 1024, a learning rate of 3×10^{-3}, and a stochastic depth drop rate of 0.05. We remove some data augmentation techniques, such as mixup (Zhang et al., 2018a), cutmix (Yun et al., 2019), and random erase (Zhong et al., 2020), as the weak data augmentation strategies benefit the performance of efficient models (Wang & Xi, 2025). This design allows us to systematically evaluate the effects of both data augmentation strategies and individual components introduced during the model evolution on overall performance. This adjusted training strategy improves the performance of ConvNeXt-S1 from 70.95% (obtained by the original training recipe in ConvNeXt) to 74.7%. We apply this training strategy to all models in the process. The accuracy at each step is averaged across training runs with three random seeds.

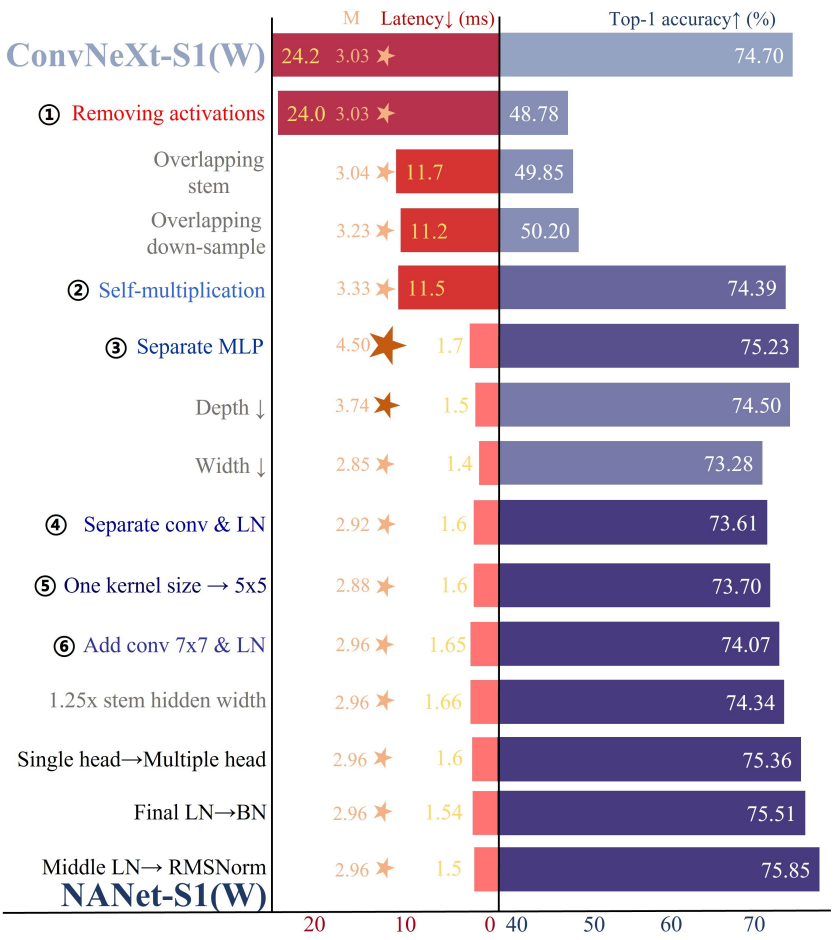

Figure 11: We remove the activation functions from the ConvNeXt Block and systematically evaluate the impact of each step in the process of transformation. This evolved ConvNeXt from a model combining linear representation and activation functions into model with Multi-head Heterogeneous Non-Activation Block. Each modification in the roadmap served a distinct purpose, collectively enhancing model performance.

To test inference latency on mobile devices, we deploy the models on an iPhone 13 using CoreML-Tools.

D.2 Evaluation and Analysis of Vital Steps

D.2.1 Removing Activation Functions

We first remove all activation functions (Agarap, 2018) from the ConvNeXt-S1 model, causing the top-1 accuracy to drop significantly from 74.70% to 48.78%. On one hand, this demonstrates the critical role of nonlinear representations provided by activation functions (Agarap, 2018; Hendrycks & Gimpel, 2016), highlighting the necessity of replacing them with effective nonlinear representations, making element-wise multiplication a key factor in model representation and selection. On the other hand, the model still maintains decent accuracy, indicating that the representations have not completely collapsed into linear ones, as LN (Ba et al., 2016) modules provide some proven nonlinear representation capabilities (Ni et al., 2024a) in deep networks, making LN (Ba et al., 2016) another key factor in model representation.

D.2.2 OVERLAPPING STEM AND DOWN-SAMPLE

We replace the 4×4 convolution (stride=4) in the stem module with two stacked 3×3 convolutions (stride=2 and padding=1), reducing inference latency by over 50%. Since the 4×4 convolution (stride=4) involves non-overlapping sliding blocks, it leads to (1) inefficient computation due to non-reusable data, (2) higher memory access overhead due to jumpy data access, and (3) difficulty in parallelizing due to sparse data distribution. Therefore, stacked convolutions are a better choice for efficient models prioritizing inference speed. The down-sample module, with a smaller input resolution and only one 3×3 convolution (stride=2 and padding=1), has a smaller impact in reducing inference latency.

D.2.3 SELF-MULTIPLICATION

As a key factor in model representation and selection, we directly multiply the expanded four-channel patterns with themselves, improving the performance from 50.20% to 74.39%. This shows that element-wise multiplication applied to the same pattern can already achieve results comparable to models (Ma et al., 2018; Han et al., 2020; Chen et al., 2022b; Pan et al., 2022; Vasu et al., 2023; Howard et al., 2019; Sandler et al., 2018; Chen et al., 2023) with activation functions. However, the main issue is that this simple application does not effectively reduce inference latency.

D.2.4 SEPARATING MLP

To address this issue, we split the 1×1 convolution and perform two distinct 1×1 convolution on patterns from element-wise multiplication. This increases the model scale (parameter count) by about 35% but reduces inference latency by an order of magnitude. Unlike self-multiplication, this approach avoids frequent data accesses during computation, significantly saving memory access time and greatly reducing inference latency.

D.2.5 COMPRESSING WIDTH AND DEPTH

Due to the increased model scale, we adjust the stage ratio from $[3, 3, 9, 3]$ to $[2, 2, 9, 2]$, as previous work (Radosavovic et al., 2019; 2020b) has shown that the third stage blocks achieve a better balance between accuracy and computational cost, which is more important for efficient models. Additionally, to maintain sufficient depth (Simonyan & Zisserman, 2014) for the extraction and nonlinear representation capabilities of LN (Ba et al., 2016), we further compress the model width (Zagoruyko & Komodakis, 2016) from 30 to 26, keeping the model scale comparable to ConvNeXt-S1.

D.2.6 SEPARATING CONVOLUTION & LN

After compressing the width (Zagoruyko & Komodakis, 2016) and depth (Simonyan & Zisserman, 2014) of the model, the top-1 accuracy drops from 75.23% to 73.28%. Building on the separated MLP, we further separate the convolution followed by LN (Ba et al., 2016) to enable the model to perform representations and selections on different spatial patterns, increasing the top-1 accuracy to 73.61%.

D.2.7 REDUCING ONE KERNEL SIZE TO 5X5

We reduce one convolution kernel size from 7×7 to 5×5, allowing the model to perform representations and selections on neighborhood patches. The nonlinear representations of two 7×7 kernels can be decomposed into (1) a nonlinear neighborhood representation of a 5×5 pixel block and a 7×7 pixel block, and (2) a 7×7 pixel block and a 5×5 pixel block's neighborhood. The second representation cannot extract meaningful features, so we reduce the kernel size to remove this part, enhancing the representation capability and saving parameters.

D.2.8 ADDING 7X7 CONVOLUTION & LN

At this stage, the model achieves nonlinear neighborhood representations of patterns. Adding a 7×7 convolution enables the model to aggregate neighborhood representations of different locations, improving the top-1 accuracy from 73.70% to 74.07%.

D.2.9 1.25× STEM HIDDEN WIDTH

Since the stem module processes input images at the highest resolution (224×224), a large number of channels can lead to a higher inference latency on mobile devices. Therefore, the channel number of the output from the stem module should remain as small as possible and match that of the first stage. To further enhance pattern diversity after the stem module, we set the output channels of the first convolution to $1.25\times$ the output channels of the stem module, making it similar to an IB block. This improves the top-1 accuracy from 74.07% to 74.34%.

D.2.10 ADDING A MULTI-HEAD MECHANISM

We further introduce a multi-head design to replace the original single-head structure. This design enables different heads to learn relatively independent feature representations and mitigates the redundancy caused by overly wide feature channels. Specifically, we divide the feature dimensions into five heads, resulting in model depths of [3,3,10,2] and model widths of [40,80,160,320], with per-head widths of [8,16,32,64]. This modification increases both the overall model width and depth, allowing each head to capture distinct feature representations while reducing feature redundancy. As a result, the classification accuracy improves from 74.34% to 75.36%.

D.2.11 SUBSTITUTING FINAL LN WITH BN

Layer Normalization (LN) only normalizes features within the same sample and layer. Although it provides nonlinear representation capability, it has two limitations: (1) LN lacks interactions across features from different batches; and (2) LN introduces additional computational overhead, making it an important factor affecting inference efficiency. Therefore, we replace the LN operation after the final convolution with Batch Normalization (BN), enabling normalization across different batches at the final stage while preserving the feature extraction capability of different heads in previous layers.

This modification is based on the observation that feature distributions from different batches may vary significantly during feature extraction. Applying BN to intermediate features forces them into a unified distribution, which may introduce feature distortion and weaken the representation of original features. By applying BN only after the final convolution, we leverage cross-batch normalization without interfering with the hierarchical feature extraction process. This change improves the classification accuracy to 75.51% while further reducing inference latency.

D.2.12 SUBSTITUTING MIDDLE LN WITH RMSNORM

Furthermore, to improve inference efficiency, we replace the two intermediate LN operations with RMSNorm, a lightweight normalization scheme without mean subtraction. This modification reduces inference latency while unexpectedly improving classification accuracy to 75.85%. We hypothesize that LN may reduce the effectiveness of element-wise multiplication by aligning the distributions of its two input features through mean subtraction. In contrast, RMSNorm preserves richer distributional differences between the inputs, allowing element-wise multiplication to better discriminate and select informative features.

D.3 EXPLORATIONS OF ARCHITECTURE

In this section, we investigate the impact of different configurations (convolution kernels) of key spatial operations on model performance during the evolution from ConvNeXt Block to Heterogeneous Non-Activation Block. As illustrated in Figure 12, after removing the activation function, we initially experiment with varying kernel sizes for self-multiplication. The findings reveal that a 9×9 kernel achieves the highest accuracy and the lowest inference latency. Meanwhile, models employing 5×5 and 7×7 kernels also demonstrate competitive performance, prompting us to retain these three configurations for further exploration. Similarly, when separating the convolutions, we test the performances on three different kernel sizes, with the 9×9 kernel outperforming the 5×5 and 7×7 variants. Building on this, we further examine the performance of models with one normal kernel and another reduced kernel. Based on the kernels 5×5, 7×7, and 9×9, we implement the ⑤ step, resulting in two distinct kernel configurations: 5×5 & 7×7 and 7×7 & 9×9. We then add these configurations with the corresponding 5×5, 7×7, or 9×9 convolutions, creating four distinct configurations for nonlinear context representation. The final results indicate that while the 9×9

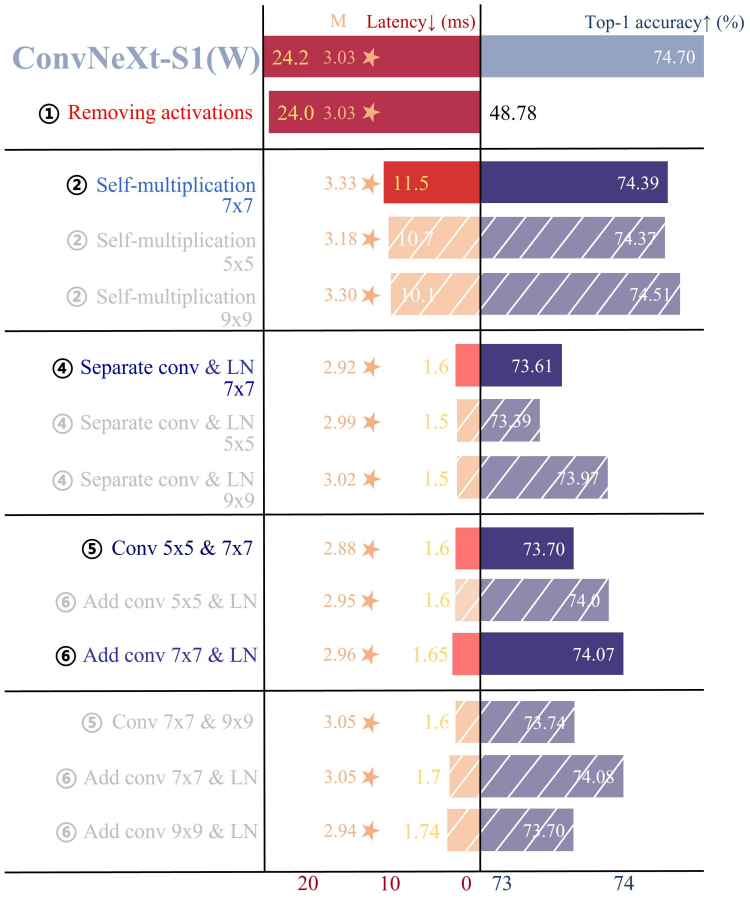

Figure 12: **Explorations of Architecture.**

kernel achieves the best performance in the ④ step, its performance declines after the ⑤ and ⑥ steps, ultimately aligning with that of the model based on the 7×7 kernel of the ④ step. This study adopts smaller kernel configurations, which offer certain advantages in terms of inference latency.

E Hyper-parameters & Training Settings

E.1 Training Settings on ImageNet-1K

We provide detailed settings on the weak training recipes used by ConvNeXt S1 (W) $\rightarrow$ NANet S1 (W), the standard training recipes used by ConvNeXt S1/S2/S3/S4/T/B/L, and the optimized training recipes used by ConvNeXt B*/L*, as shown in Table. 5. In this paper, we attempt to explore the representation capability of a native nonlinear representation strategy, without using techniques such as EMA and layer scale techniques to enhance model performance.

E.2 Hyper-parameters for models of different scales

As shown in Table 6, we provide the configurations of different NANet variants, including their widths, depths, numbers of heads, and parameter counts. The architectural hyper-parameters of NANet are defined as follows:

C_i: channel dimension of the i-th stage;

L_i: number of blocks in the i-th stage;

Table 5: **Training settings for NANet variants on ImageNet-1K.** The training recipes are similar to standard practices Touvron et al. (2021), with some optimizations. "W" indicates that the NANet variants are trained using the weak training recipe.

Super Parameters	**ConvNeXt-S1(W) → NANet-S1(W)** ImageNet-1K	**NANet-S1/S2/S3/S4/T/B/L** ImageNet-1K
Input Scale	224×224	224×224
Training Epochs	300	300
Batch Size	1024	1024
Optimizer	AdamW	AdamW
Optimizer Momentum	$\beta_1, \beta_2 = 0.9, 0.999$	$\beta_1, \beta_2 = 0.9, 0.999$
Base Learning Rate	3×10^{-3}	3×10^{-3}
Learning Rate Schedule	cosine decay	cosine decay
Learning Rate Decay	0.05	0.05
Warmup Epochs	20	20
Warmup Schedule	linear	linear
Label Smoothing ε	0.1	0.1
Dropout Rate	0.	0.
Drop Path Rate	0.05	0.05/0.05/0.05/0.05/0.1/0.2/0.4
AutoAugment	rand-m9-mstd0.5-inc1	rand-m9-mstd0.5-inc1
Color Jitter	0.4	0.
Mixup Alpha	0.	0.8
Cutmix Alpha	0.	0.2
Erasing Probability	0.	0.25
Loss	Cross Entropy	Cross Entropy
EMA	None	None
AMP	None	None
Layer Scale	None	None

Table 6: **Hyper-parameters for models of different scales.** '#Head' denotes the Head of NANet variants. '#parameters' denotes the number of parameters.

Model	$[C_1, C_2, C_3, C_4]$	$[L_1, L_2, L_3, L_4]$	#Head	#parameters
NANet-S1/S1(W)	[40,80,160,320]	[3,3,10,2]	5	3.0M
NANet-S2	[48,96,192,384]	[3,3,11,2]	6	3.9M
NANet-S3	[64,128,256,512]	[3,3,11,2]	8	5.9M
NANet-S4	[72,144,288,576]	[3,3,13,2]	9	7.6M
NANet-T	[80,160,320,640]	[3,3,18,3]	5	15.0M
NANet-B	[120,240,480,960]	[3,3,15,3]	5	30.0M
NANet-L	[144,288,576,1152]	[4,4,21,4]	6	50.0M

#Head: number of heads for feature channel partitioning.

E.3 Latency Benchmark Settings

We utilize the Apple CoreML tool to convert the model to a format compatible with the iPhone and perform benchmark tests for inference latency with a batch size of 1, aligning with real-world scenarios. To account for variations caused by device state and other factors during multiple runs, we calculate the average inference latency across ten tests. Despite this, the observed variations in the inference latency on the iPhone were generally within a range of $0.1ms$, which is negligible compared to the general latency. This does not affect the conclusion of the models in terms of inference latency.