Infinite-Dimensional Feature Interaction

Chenhui Xu[1,2] **Fuxun Yu**[1,3] **Maoliang Li**[4] **Zihao Zheng**[4]
Zirui Xu[1] **Jinjun Xiong**[2,*] **Xiang Chen**[1,4,*]
[1]George Mason University [2]University at Buffalo
[3]Microsoft [4]Peking University
{cxu26,jinjun}@buffalo.edu, xiang.chen@pku.edu.cn

Abstract

The past neural network design has largely focused on feature *representation space* dimension and its capacity scaling (e.g., width, depth), but overlooked the feature *interaction space* scaling. Recent advancements have shown shifted focus towards element-wise multiplication to facilitate higher-dimensional feature interaction space for better information transformation. Despite this progress, multiplications predominantly capture low-order interactions, thus remaining confined to a finite-dimensional interaction space. To transcend this limitation, classic kernel methods emerge as a promising solution to engage features in an infinite-dimensional space. We introduce InfiNet, a model architecture that enables feature interaction within an infinite-dimensional space created by RBF kernel. Our experiments reveal that InfiNet achieves new state-of-the-art, owing to its capability to leverage infinite-dimensional interactions, significantly enhancing model performance.

1 Introduction

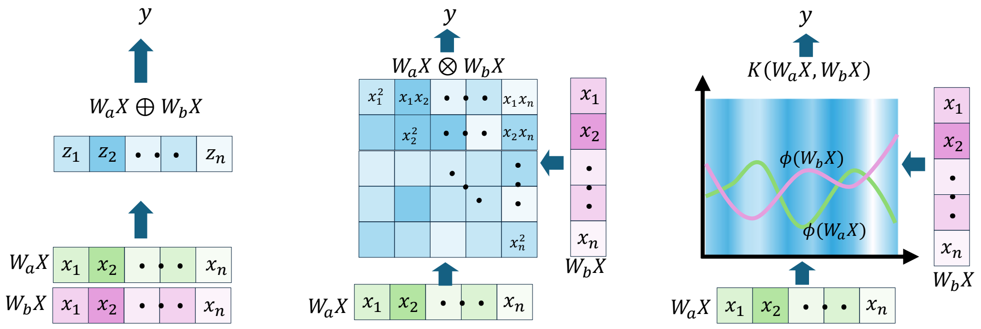
In the past decade, deep neural network architecture design has experienced several major paradigm shifts regarding the feature representation learning. As shown in Fig. 1(a), the early stage of neural network design is dominant by flat stream architectures in the form of *weight-feature interaction* (e.g., $W\mathbf{x}$), like multi-layer perceptron (MLP), convolution neural networks (CNN), ResNet, etc. These models usually adopt linear superposition (e.g., $W_i\mathbf{x} \oplus W_j\mathbf{x}$)[1] in the feature representation space. Therefore, the feature representation space scaling is limited to increase model channel width and depth [14, 19]. Nevertheless, this scaling approach has witnessed model development from the very small-scale MLPs or LeNet[20] to the recent huge ConvNext V2 [41]. With the ultra-scaled parameter amounts, computing complexity, and model size, the return of investment on model performance by further scaling feature dimensions has largely plateaued [10].

Despite the plateau in feature representation space, recent sporadic architecture design works [29] shed light on another potential dimension of scaling: feature interaction space. Specifically, as shown in Fig. 1(b), these neural network designs generally demonstrate *feature-feature interaction* (e.g., $W_i\mathbf{x} \otimes W_j\mathbf{x}$). As a mathematical example, the self-attention mechanism in Transformers [38] can be formulated as $\mathbf{x}^{L+1} = f_k(\mathbf{x}^L) \otimes f_q(\mathbf{x}^L) \otimes f_v(\mathbf{x}^L)$, which is also element-wise multiplication between processed input feature themselves. Characterized by element-wise interaction, these designs offer complementary feature correlation capabilities in addition to simple linear superposition. Such feature interactions have become the essential mechanisms of mainstream state-of-the-art neural architectures. For example, it's implemented in SENet with squeeze and excitation [17], non-local network with transposed multiplication [39], vision transformers with self-attention [4, 11, 38], gated aggregation [22, 35, 46], and quadratic neurons [12, 44, 45].

[*] Corresponding Author.

[1] The symbol $\oplus$ denotes the Direct Sum of vector spaces. This corresponds to structures such as channel expansion/bottleneck in commonly used neural networks. While the $\otimes$ is elementwise multiplication.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) Feature Representation Space (b) Feature Interaction Space (c) ∞-D Feature Interaction Space

Figure 1: (a) Traditional feature representation without interaction [14, 41]. (b) Recent work with finite feature interaction [35, 45]. (c) Our method: Kernel-enabled infinite feature interaction.

Although these models greatly improve the performance of state-of-the-arts, as mentioned above, these works provide diversified explanations that neglect the underlying shared design of element-wise feature multiplication operation [29], and thus may fail to reveal the fundamental source of improvement. To provide both explainability and quantifiability, in this paper, we propose a unified theoretical perspective to rethink the feature interaction scaling, i.e., *the dimensionality of feature interaction space*. For example, as shown in Fig 1(b), by employing the $\otimes$, element-wise multiplication, an implicit interactive quadratic space $\mathcal{Q} = \text{span}\left(x_1^2, x_1x_2, \cdots, x_{n-1}x_n, x_n^2\right)$ with degrees of freedom $n(n+1)/2$ is constructed from the original representative vector space $\mathcal{V} = \text{span}\left(x_1, x_2, \cdots, x_n\right)$ with degrees of freedom n [29]. Such space dimensionality scaling is the key to improving feature representation quality and end-task, as we will show later.

From the unified feature dimensionality perspective, a new opportunity emerges in neural architecture design, that is to scale to *infinite-dimensional feature interactions* than former methodologies (e.g., from $n(n+1)/2$ to $\lim_{k\to\infty} \frac{(n+k-1)!}{(n-1)!k!}$). However, scaling feature interaction space dimensionality from architectural enhancements (e.g., quadratic, self-attention, and recursive gates) comes with linear scaling cost w.r.t. interaction order k, which hinders the infinite dimensionality increase [11, 22]. Thus, there is an open question:

How can we efficiently extend interactions to an infinite-dimensional space?

Inspired by traditional machine learning, we propose an approach that introduces kernel methods for feature interaction in neural networks. We define a set of feature interactions between features $W_a\mathbf{x}$ and $W_b\mathbf{x}$ with a kernel function $K(W_a\mathbf{x}, W_b\mathbf{x})$ instead of the element-wise multiplication. As shown in Fig 1(c), the kernel method transforms the feature to an ultra-high dimensional space by an implicit mapping $\phi(\cdot)$. From there, the feature interaction space is defined by the inner product on the Reproducing Kernel Hilbert Space (RKHS) [1] $\mathcal{H}$ constituent with the kernel function $K(\cdot, \cdot)$. The RKHS can greatly expand the interaction space at very little cost, enabling infinite dimensions.

We then propose InfiNet, a novel family of neural networks that generate high-quality feature representations. Specifically, we introduce the Radial Basis Function (RBF) kernel [32] as a replacement of the common $\oplus$ or $\otimes$ operations. With the infinite series expansion in RBF kernel, it enables a theoretical provable dimensionality approximation $\text{span}_{j\in\mathbb{N},\sum_k n_k=j}\{\frac{x_1^{n_1}\cdots x_k^{n_k}}{\sqrt{n_1!\cdots n_k!}}\}$ while with as low-overhead as evaluating an exponential function. In this way, InfiNet enables efficient infinite-dimensional feature interaction space scaling upon a finite set of branches in the model architecture, thus achieving better complexity performance tradeoffs than prior state-of-the-art.

Contributions. We make the following contributions:

- We unify the perspectives of recent feature interactive works and identify a novel direction of neural network performance scaling: the feature interaction space dimensionality.
- We propose a method to expand the feature interaction space to an infinite dimension with RBF kernel, that can effectively model the complex implicit correlations of features.
- We propose InfiNet, a novel series of neural networks that explore the neural interaction from infinite-dimensional space, and achieve state-of-the-art performance.

Extensive ablation studies verify that the shift from finite feature interaction space to infinite one is a key factor to learn better representations and therefore improving model performance. Meanwhile, large-scale experiments on ImageNet classification, MS COCO detection and ADE20K segmentation also demonstrate InfiNet design's effectiveness, which consistently outperforms the state-of-the-art flat stream networks [14, 26] and finite-order interaction networks [25, 35].

2 Related Work

Interactions in Neural Networks. Interaction in neural networks has undergone a long period of change. During the evolution from AlexNet [19] to ResNet [14], the model has for long followed the principle of summing the weighted pixels at each position in the layer-by-layer feature iteration. And then, as the potential of the attention mechanism was realized, model development is towards an element-wise multiplication way. This trend is punctuated by the emergence of models such as Non-Local [39], Transformer [38], and ViT [11]. Recent insights suggest that the crux of the attention mechanism lies in high-order information interactions, rather than the mechanism of "Attention" itself [4, 35, 43]. This revelation has spurred the development of innovative neural network architectures. For example, HorNet [35], QuadraNet [44] and MogaNet [22] examines high-order models from the perspective of spatial interactions through multiplication-integrated architecture design. However, limited by the support of existing deep learning platforms such as Pytorch [33], few attempts have been made to extend the model's feature interaction to an ultra-high dimensional situation through the kernel method for more potential.

Kernel Methods in Neural Networks. The fundamentals of kernel methods are well-studied in traditional ML domains like support vector machines [36] but they are less used in neural architectures. Early extensions to deep learning through the kernelized perceptron [7] have improved the performance but mainly in shallow neural networks. Recent advancements include the development of Convolutional Kernel Networks (CKN) [30], which merge CNNs' robust feature learning with kernel stability. This approach offers a theoretical foundation for deep learning's application in structured data. Additionally, the introduction of Kervolutional Neural Networks [6] replaces traditional convolution in CNNs with kernel-based operations to enhance feature extraction without excessive computational costs. The combination of Gaussian processes with neural networks to create Deep Kernel Learning [40] adjusts model complexity based on data while maintaining Bayesian inference. The neural tangent kernel (NTK) [18] framework establishes a direct connection between infinitely wide neural networks at initialization and kernel methods. Specifically, NTK shows that as the width grows, the network's training dynamics can be described by a kernel function, linking the neural network's behavior to that of kernel methods. Random features [34] provide an efficient approximation to the feature mappings used in kernel methods. By random projections, one can approximate the inner product defined by a kernel function, making it feasible to apply kernel methods [3, 13].

Despite these innovations, a significant limitation remains: Although these methods expand the feature representation space, they fail to scale up the feature interaction space, limiting the network to aggregate information in a superposition manner. As a result, these methods also fall short of potential feature interactions and thus are incapable of handling complex data correlations and functionalities.

3 From Feature Representation Space to Interaction Space

We start with considering the transformation of feature representation of a normal shape-preserving 2-layer perceptron block. Given an input $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ with n-dimension. We denote the first layer transformation as $\mathbf{z} = g(\mathbf{x}) = \sigma(W_1\mathbf{x})$ (we omit the bias term for simplicity), and the second layer transformation as $h(\mathbf{z}) = W_2\mathbf{z}$. Therefore the whole block is a mapping $h \circ g : \mathbb{R}^n \mapsto \mathbb{R}^n \mapsto \mathbb{R}^n$ from feature space $\mathbb{R}^n$ to a middle feature representation space $\mathbb{R}^n$ and eventually to a output space $\mathbb{R}^n$. Expanding the width of the neural networks, for example with a coefficient 2, is going to expand the feature representation space to $\mathbb{R}^{2n}$. The core idea of this dimensional expansion is that implicit associations in features in low dimensions will be expressed explicitly when projected into a high-dimensional space. Model architecture (i.e. convolution, multi-branch [37]) is an additive superposition of pixels, essentially no different from the perceptron in terms of representation space.

3.1 Feature Interaction Space

Although flat stream neural networks are defined by linear transformations and activations, modern network design also introduces multiplications in the network structure or neurons. They are called attentions from an interpretable point of view, that is the degree of interest of one position in relation to another; from a more abstract point of view, they are called interactions in spatial contexts. But invariably, these are realized with element-wise multiplication. We conduct the following definition.

Definition 1 (Feature Interaction) A Feature interaction refers to transformations between features with the same or different positions defined by element-wise multiplication.

For example, Star Operation [29] is a basic 2-order feature interaction, defined as $(W_a\mathbf{x}) * (W_b\mathbf{x})$, where $W_a, W_b \in \mathbb{R}^n$. This is a simple element-wise multiplication fusion of two linear transformations of the ordinary input $\mathbf{x}$. We write the expansion of such multiplication operation:

$$y = (W_a\mathbf{x}) * (W_b\mathbf{x}) = (\sum_{i=1}^{n} w_{ai}x_i)(\sum_{j=1}^{n} w_{bj}x_j) = \sum_{i\leq j} \alpha_{i,j}x_ix_j \tag{1}$$

where $\alpha_{i,j} = w_{ai}w_{bj} + w_{aj}w_{bi}$, if $i \neq j$, and $\alpha_{i,i} = w_{ai}w_{bi}$. Then we vectorize α and x_ix_j:

$$A = [\alpha_{1,1}, \alpha_{1,2}, \alpha_{2,2}, \cdots, \alpha_{n-1,n}, \alpha_{n,n}] \in \mathbb{R}^{n(n+1)/2} \tag{2}$$

$$\chi = [x_1x_1, x_1x_2, x_2x_2, \cdots, x_{n-1}x_n, x_nx_n] \tag{3}$$

χ can therefore define a basis of a space. Thus the output of the current layer can be rewritten as:

$$y = \sum_{i\leq j} \alpha_{i,j}x_ix_j = A\chi. \tag{4}$$

From the independence of the pixel level, we know that each term in χ is linearly independent, this indicates every dimension in χ is an individual dimension. Given a set of basis vectors χ, we define span(χ) as the feature interaction space. In this way, the generation of the next layer of features y is constituted by a linear superposition $A\chi$ on the feature interaction space span(χ) like in Eq.(4).

3.2 Dimension of Feature Interaction Space

Now, we consider the number of dimensions of a feature interaction space. Given k-1 multiplication operations, we first define the k-order feature interaction space as follows:

Definition 2 (Feature Interaction Space) A k-order Feature Interaction Space $\mathcal{S}^k$ refers to the span of monomial basis $\{x_1^{d_1}x_2^{d_2}\cdots x_n^{d_n} | \sum d_i = k, d \in \mathbb{N}\}$ defined by a k-order Feature Interaction.

An element-wise multiplication generates a feature interaction space of $n(n+1)/2$ dimension, which is the number of elements of an upper triangular matrix. In general, considering the symmetry of the interactions and elements, for a k-order interaction on the n-dimensional feature, the dimension of the corresponding feature interaction space is:

$$dim(\mathcal{S}^k) = \frac{(n+k-1)!}{(n-1)!k!} \tag{5}$$

The next layer of feature generation based on the feature interaction space greatly expands the spatial dimensions to which the features are mapped compared to the original model based only on the feature representation space, i.e., it is possible to explore the feature's non-linearity in high-dimensional space. It is worth noting that this process introduces terms like x_1x_2, an interaction that cannot be captured by traditional plane networks in feature representation space.

For example, we consider the Self-Attention in the Transformers [38]. The Self-Attention contains two element-wise multiplications, so it explores a 3-order feature interaction space. This is due to the fact that: (1) in the first stage, in the query-key dot-product attention map computation $Att(\mathbf{x}) = Q \cdot K^T = W_Q\mathbf{x} \cdot \mathbf{x}^T W_K^T$ explores the feature interaction space $\mathcal{A} =$ span $\left(x_1^2, x_1x_2, \cdots, x_{n-1}x_n, x_n^2\right)$. (2) In the second stage, the multiplication between the attention map and the value $y = \sigma(Att(\mathbf{x})) \cdot V = \sigma(Att(\mathbf{x})) \cdot W_v\mathbf{x}$ explores the feature interaction space $\mathcal{S} = \mathcal{A} \otimes \mathbb{R}^n =$ span $\left(x_1^3, x_1^2x_2, x_1x_2x_3, \cdots, x_{n-1}x_n^2, x_n^3\right)$, which has $(n+2)(n+1)n/6$ dimensions in line with Eq.(5). Thereby we explain, from the perspective of feature interaction space, why the transformer family of models has, so far, generally outperformed recurrent neural networks and convolutional neural networks in various domains.

4 Expanding Interaction Space to Infinite Dimension

This element-wise multiplication-based interaction, since the construction of each order of the feature interaction space is based on a multiplication operator, leads to a problem in that feature interaction space expansion is still difficult. This is due to the fact that these interaction operators is explicit mapping, which tends to have quadratic or higher complexity w.r.t the input length (e.g. self-attention [38]) or lengthy recursive designs (e.g. HorNet [35]) and linear complexity w.r.t interaction order. But the problem with building this mapping explicitly is: (1) The complexity of mapping itself. The computational overhead associated with defining a set of explicit nonlinear mappings is non-negligible. A mapping from C channels to C' channels means a computation complexity of $O(CC')$. (2) The complexity of interaction. The complexity of the inner product used to interact with the two sets of features increases dramatically to $O(C')$ when the dimension is raised. Considering $C' >> C$, these two complexities will largely increase the computational overhead of the networks.

We would like to obtain a method that can expand the dimension of feature interaction space as much as possible in $O(1)$ time. Fortunately, the machine learning community has already given a method for increasing the dimension of a feature defined on the inner product: **kernel methods**.

4.1 Expanding Interaction Space with Reproducing Kernel

The nature of kernel methods is that they are substitutions for inner product operations. This requires combining element-wise multiplication and summation to define a set of inner products within the network. For this purpose, we rewrite the form of element-wise multiplication in Eq.(1) on two groups of features, which is a common design in literature architecture (e.g. multi-head self-attention [38]). It will therefore be a superposition of multiple interaction in the format:

$$y = \sum_{i=1}^{C} W_{ai}\mathbf{x} * W_{bi}\mathbf{x} = \langle \mathbf{W_a x}, \mathbf{W_b x} \rangle \tag{6}$$

where $\mathbf{W_a x} = [W_{a1}\mathbf{x}, W_{a2}\mathbf{x}, \cdots, W_{aC}\mathbf{x}]$, $C \in \mathbb{N}$ is the number of branches. Thereby we generalize the form of the feature interaction from element-wise multiplication to inner product which is a multi-branch paradigm. At this point, we have $\mathbf{W_a x} \in \mathbb{R}^C$, while it is generated from a feature representation space $\mathbb{R}^n$. In order to extend the interaction space, we need to further project $\mathbf{W_a x}$ and $\mathbf{W_b x}$ to a high-dimensional space. An obvious way to do this is to construct an implicit mapping $\Phi(\cdot)$ to a high-dimensional space, so we can compute $\langle \Phi(W_a\mathbf{x}), \Phi(W_b\mathbf{x}) \rangle$ for interaction.

By Mercer's Theorm [31], taking a continuous symmetric positive semi-definite function $K(s,t)$, there is an orthonormal basis $\{\phi_i(\cdot)\}$, $i = 0, 1, \cdots, \infty$, consisting of eigenfunctions of function $K(\cdot,\cdot)$ such that the corresponding sequence of eigenvalues $\{\lambda_i\}$ is non-negative. These means:

$$K(s,t) = \sum_{i=1}^{\infty} \lambda_i \phi_i(s)\phi_i(t), \tag{7}$$

where $\forall i \neq j, \forall s$ and t, $\langle \phi_i(s), \phi_j(t) \rangle = 0$ since. Then we construct a Hilbert space $\mathcal{H}$ with the orthonarmal basis $\{\sqrt{\lambda_i}\phi_i(\cdot)\}$. Consider a vector $f = (f_1, f_2, \cdots)^T_{\mathcal{H}}$ on the space $\mathcal{H}$, then we have:

$$f = \sum_{i=1}^{\infty} f_i \lambda_i \phi_i(\cdot). \tag{8}$$

Thus for a vector $K(s,\cdot)$ in space $\mathcal{H}$:

$$K(s,\cdot) = \sum_{i=1}^{\infty} \lambda_i \phi_i(s)\phi_i(\cdot) = \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(s)\sqrt{\lambda_i}\phi_i(\cdot) = (\sqrt{\lambda_1}\phi_1(s), \sqrt{\lambda_2}\phi_2(s), \cdots)^T_{\mathcal{H}}. \tag{9}$$

Therefore, for the Hilbert space $\mathcal{H}$, we can define the reprodcuing kernel by:

$$\langle K(s,\cdot), K(t,\cdot) \rangle = \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(s)\sqrt{\lambda_i}\phi_i(t) = \sum_{i=1}^{\infty} \lambda_i \phi_i(s)\phi_i(t). \tag{10}$$

Implicit Mapping to High-Dimensional RKHS. Let $\Phi(s) = K(s,\cdot)$, then we have $\langle \Phi(s), \Phi(t) \rangle = K(s,t)$. The Hilbert Space $\mathcal{H}$ is known as the Reproducing Kernel Hilbert Space (RKHS) corresponding to kernel function $K(\cdot,\cdot)$. Note that the $\Phi(\cdot)$ is therefore defined on the RKHS $\mathcal{H}$, which

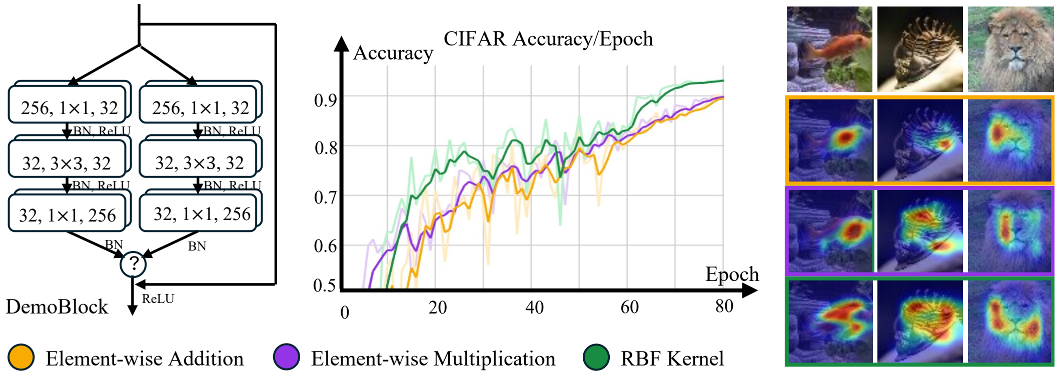


Figure 2: Comparison of simple representation, finite interaction, and infinite-dimensional interaction. The **?** circle in DemoBlock is chosen from element-wise Add, element-wise Mul. or RBF kernel.

can be infinite-dimensional given specific kernel $K(\cdot,\cdot)$. The mapping $\Phi(\cdot)$ does not have to have an explicit expression since we can get the result of $\langle\Phi(s),\Phi(t)\rangle$ by computing $K(s,t)$. This means that we can achieve an extension of the dimensions for the feature interaction space by simply replacing the inner product $\langle\mathbf{W_a x},\mathbf{W_b x}\rangle$ used in the interaction with a kernel $K(\mathbf{W_a x},\mathbf{W_b x})$.

4.2 Infinite-Dimensional Feature Interaction with RBF Kernel

To maximize the dimension of the feature interaction space, we consider Radial Basis Function (RBF) Kernel $K_{\text{rbf}}(\mathbf{s},\mathbf{t})=\exp\left(-\frac{1}{2}\|\mathbf{s}-\mathbf{t}\|_2^2\right)$with an infinite-dimensional RKHS, given the fact that:

$$\exp\left(-\frac{1}{2}\|\mathbf{s}-\mathbf{t}\|_2^2\right)=\sum_{j=0}^{\infty}\frac{\left(\mathbf{s}^\top\mathbf{t}\right)^j}{j!}\exp\left(-\frac{1}{2}\|\mathbf{s}\|_2^2+\|\mathbf{t}\|_2^2\right) \tag{11}$$

$$=\sum_{j=0}^{\infty}\sum_{n_1+n_2+\cdots+n_k=j}\exp\left(-\frac{1}{2}\|\mathbf{s}\|^2\right)\frac{s_1^{n_1}\cdots s_k^{n_k}}{\sqrt{n_1!\cdots n_k!}}\exp\left(-\frac{1}{2}\|\mathbf{t}\|^2\right)\frac{t_1^{n_1}\cdots t_k^{n_k}}{\sqrt{n_1!\cdots n_k!}} \tag{12}$$

$$=\langle\Phi_{\text{rbf}}(\mathbf{s}),\Phi_{\text{rbf}}(\mathbf{t})\rangle, \tag{13}$$

where $\Phi_{\text{rbf}}(\mathbf{x})=\sum_{j=0}^{\infty}\sum_{\sum_k n_k=j}\exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)\frac{s_1^{n_1}\cdots s_k^{n_k}}{\sqrt{n_1!\cdots n_k!}}\in\text{span}_{j\in\mathbb{N},\sum_k n_k=j}\{\frac{x_1^{n_1}\cdots x_k^{n_k}}{\sqrt{n_1!\cdots n_k!}}\}$.

Infinite-dimensional Feature Interaction Space Observing the RKHS of such a RBF kernel, $\text{span}_{j\in\mathbb{N},\sum_k n_k=j}\{\frac{x_1^{n_1}\cdots x_k^{n_k}}{\sqrt{n_1!\cdots n_k!}}\}$, We note that each of its dimensions is a j-order interaction within the feature $\mathbf{x}$, given the fact one of the bases of this RKHS is:

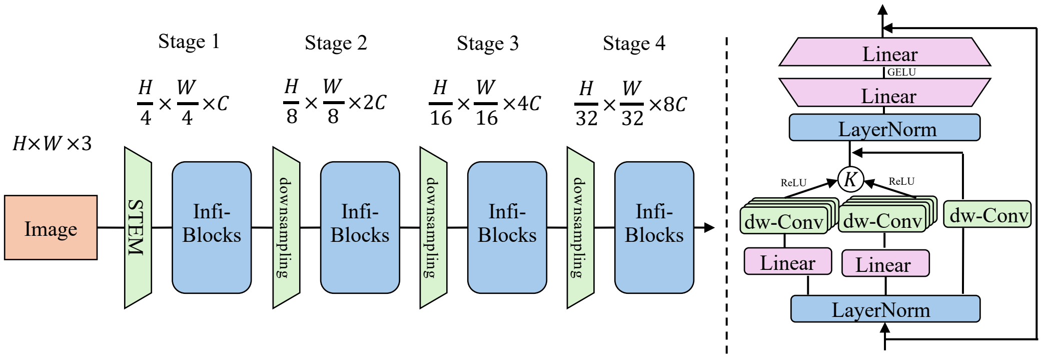
$$\{1,x_1,\cdots,x_n,x_1^2,\cdots,x_1x_n,\cdots,x_n^2,\cdots,x_1^j,x_1^{j-1}x_2,\cdots,x_n^j,\cdots\}, \tag{14}$$

which contains an all-order monomial among all elements of the feature $\mathbf{x}$. This means that we get an infinite-dimensional Hilbert space for the superposition of interaction information through such an RBF kernel, and most importantly, each dimension of this space is defined by a feature interaction. Therefore, we get an infinite-dimensional feature interaction space.

4.3 Demo Case Performance of Models on Different Feature Space

In order to compare networks that utilize the summing superposition of information mappings on the feature representation space, finite feature interaction space, and infinite-dimensional interaction space, we design a demo network with 8 DemoBlock, as shown in Fig. 2. DemoBlock has a two-group design, the only difference among models in different spaces is the interaction method at the end of the block (add for simple representation, multiplication for finite interaction, and RBF kernel for infinite dimensional interaction), the demo models are trained on CIFAR10 and Tiny-ImageNet.

As shown in Fig. 2, with the procedure of transferring from the simple feature representation space to a finite feature interaction and eventually to an infinite-dimensional interaction, the performance



(a) InfiNet Architecture (b) InfiBlock

Figure 3: **Overview of InfiNet.** (a) Four-stage hierarchical InfiNet design. (b) InfiBlock Design

on CIFAR10 of the networks is growing. This is done throughout the training process, implying the superiority of feature iteration in a high-dimensional interaction space. The right side of Fig. 2 shows the Class Activation Mapping [47] of three different demo nets on Tiny-ImageNet. From this, we can see that the network of feature interaction better reflects the pixel-level correlation within the image.

5 Method

5.1 InfiBlock: Infinite-Dimensional Spatial Feature Interaction

InfiBlock. In this section, we present InfiBlock, the basic block to build the high-performant InfiNet architecture to achieve infinite-dimensional spatial feature interaction. As presented in Fig. 3(b), InfiBlock starts with a LayerNorm layer and subsequently transforms the feature into separate representations of the two groups through two different linear layers. Then InfiBlock utilizes a depth-width convolution with an expansion coefficient of r and a ReLU activation to obtain a feature branch vector of length r on each group. The feature branch vectors on both groups are then fed into an RBF kernel for feature interaction. At the same time, we retain a pathway containing only one depth-wise convolution for summing superposition over the feature representation space to ensure that the linear connections between features are not neglected and overfitted. This is followed by a residual connection with the original input values after passing through a two-layer MLP. Starting with a input $X^l \in \mathbb{R}^{HWC}$, Infi-Block can be formulated as:

$$\hat{X}^l = \text{LayerNorm}(X^l), \tag{15}$$

$$[\vec{Z}_a^l, \vec{Z}_b^l, Z_c^l] = [\sigma(\text{Conv}_a(\hat{X}^l W_a)), \sigma(\text{Conv}_b(\hat{X}^l W_b)), \text{Conv}_c(\hat{X}^l)], \tag{16}$$

$$X^{l+1} = \text{MLP}(\text{LayerNorm}(K_{rbf}(\vec{Z}_a^l, \vec{Z}_b^l) + Z_c^l)) + X^l, \tag{17}$$

where $K_{rbf}(\vec{Z}_a^l, \vec{Z}_b^l) = exp(-\frac{\|\vec{Z}_a^l - \vec{Z}_b^l\|_2^2}{2}) \in \mathbb{R}^{HWC}$, is an RBF kernel with out any hyper-parameter, and $\| \cdot \|_2^2$ is squared L-2 norm. $\text{Conv}_{a,b}(\hat{X}^l W_{a,b}) \in \mathbb{R}^{HWC*r}$ expands the number of channels of feature to r times, where $W_{a,b}$ preserve the shape of the feature. The discretization of the feature is performed in a depth-width convolution. LayerNorm$(\cdot)$ is layer normalization [2] and MLP$(\cdot)$ is activated by GELU [15].

Specifically, following the ConvNeXt [26], we select a 7 × 7 depth-wise convolution to obtain a larger receptive field. We set the expansion coefficient r as 7 in practice. This means:

$$[\vec{Z}_a^l, \vec{Z}_b^l] = [[Z_{a1}^l, Z_{a2}^l, \cdots, Z_{a7}^l]^T, [Z_{b1}^l, Z_{b2}^l, \cdots, Z_{b7}^l]^T], \tag{18}$$

where $Z_{ai}^l = \sigma(\text{DW-Conv}_{ai}(X^l W_a)) \in \mathbb{R}^{HWC}$. Therefore, the RBF kernel performs feature interaction on multiple (7) branches of convolution filter and outputs a result still in $\mathbb{R}^{HWC}$.

5.2 Model Architectures

As shown in Fig. 3(a), InfiNet uses the widely adopted 4-stage hierarchical architecture as in ResNet [14], ConvNeXt [26] and Swin Transformer [25]. We stack InfiBlocks into each stage.

Table 1: **ImageNet classification results.** We compare our models with state-of-the-art models with comparable parameters, the Top-1 accuracy is reported on the ImageNet-1K validation set.

Model	Interact. Orders	Params (M)	FLOPs (G)	Top1 Acc.(%)
ConvNeXt-T[26]	no	29	4.5	82.1
SLaK-T[24]	no	30	5.0	82.5
Conv2Former-T[16]	2	27	4.4	83.1
UniFormer-S[21]	2	22	3.6	82.9
CoAtNet-0[8]	3	25	4.2	82.7
FocalNet-T[46]	3	28	4.4	82.1
Swin-T[25]	3	28	4.5	81.3
HorNet-T[35]	2-5	22	4	82.8
MogaNet-S[22]	4	25	5.0	83.5
InfiNet-T	∞	23	3.2	83.4
ConvNeXt-S[26]	no	50	8.7	83.1
SLaK-S[24]	no	55	9.8	83.8
Conv2Former-S[16]	2	50	8.7	84.1
UniFormer-B[21]	2	50	8.3	83.9
CoAtNet-1[8]	3	42	8.4	83.3
FocalNet-S[46]	3	50	8.7	83.5
Swin-S[25]	3	50	8.7	83.0
HorNet-S[35]	2-5	50	8.8	84.0
MogaNet-B[22]	4	44	9.9	84.3
InfiNet-S	∞	48	7.2	84.0

Model	Interact. Orders	Params (M)	FLOPs (G)	Top1 Acc.(%)
ConvNeXt-B[26]	no	89	15.4	83.8
SLaK-B[24]	no	85	17.1	84.0
Conv2Former-B[16]	2	90	15.9	84.4
CoAtNet-2[8]	3	75	15.7	84.1
FocalNet-B[46]	3	89	15.4	83.9
Swin-B[25]	3	89	15.4	83.5
HorNet-B[35]	2-5	87	15.6	84.3
MogaNet-L[22]	4	83	15.9	83.5
InfiNet-B	∞	82	12.8	84.5
InfiNet-L	∞	116.8	19.1	84.8
ImageNet-21K Pretrained Models Fine-tuned @384^2				
ConvNeXt-L‡[26]	no	198	101	87.5
CoAtNet-3‡[8]	3	168	107	87.6
FocalNet-L‡[46]	3	197	101	87.3
Swin-L‡[25]	3	197	104	87.3
HorNet-L‡[35]	2-5	202	102	87.7
MogaNet-XL‡[22]	4	181	102	87.8
InfiNet-L‡	∞	116.8	60	87.8
ConvNeXt-XL‡[26]	no	350	179	87.8
InfiNet-XL‡	∞	255.8	126	88.2

We set the number of channels in each stage as [C,2C,4C,8C] following common practice. We build a family of InfiNets that InfiNet-T/S/B/L/XL with model variants hyper-parameters: $C = \{64, 96, 128, 128, 192\}$, number of blocks in each stages $= \{2, 2, 18, 2\}$ for InfiNet-T/S/B and $\{3, 3, 27, 3\}$ for InfiNet-L/XL. A down-sampling with 2×2 convolution with stride = 2 is used to connect each stage. STEM [26] is used to connect the input of InfiNet to Stage 1.

6 Experiments

We perform a series of experiments to validate the efficacy of InfiNets. The primary results on ImageNet [9] are showcased and contrasted with several architectures. Furthermore, our models were evaluated on downstream ADE20K[48] semantic segmentation and COCO [23] object detection. ImageNet-1K experiments are conducted on 4×Nvidia A100 GPUs and ImageNet-21K on 16×.

6.1 ImageNet Classification

Setups. We conduct image classification experiments on ImageNet-1K [9], which contains 1.28 million training samples belonging to 1000 classes and 50K samples for validation. We train the InfiNet-T/S/B/L models for 300 epochs with AdamW [28] optimizer. We use the cosine learning rate scheduler [27] with 20 warmup epochs and the basic learning rate is set as 4×10^{-3}. The training resolution is set as 224×224. To further evaluate the InfiNet's scalability, we train the InfiNet-L/XL models on 14M-sample ImageNet-22K dataset for 90 epochs and then fine-tune on ImageNet-1K at 384×384 resolution for 30 epochs following [26]. More details can be found in Appendix A.1.

Results. We present our ImageNet experiment results and comparison with baselines in Table 1. Our models achieve competitive performance with state-of-the-art. It is worth noting that InfiNet has about 20% fewer FLOPs[2] than the baseline models with similar parameter scales, but still achieves great performance. It demonstrates the effectiveness of our proposed generation of features in infinite-dimensional interaction spaces. Experiments on isotropic models can be found in Table 3(a).

[2] Difference between FLOPs and FLOPS: FLOPs (floating point of operations), the number of floating point operations, is used to measure the complexity of an algorithm/model. This number gets smaller the better. The FLOPs is different from FLOPS (floating point per second), which is a measure of hardware performance. Following other deep-learning architecture works, we use FLOPs to measure the computing demands. Since our model gets a smaller FLOPs, our method is more efficient.

Table 2: **Object detection and semantic segmentation results on MS COCO and ADE20K.**

	Object Detection with *Cascade Mask R-CNN 3×*				Semantic Segmentation with *UperNet 160K*			
Model	AP^{box}	AP^{mask}	Params	FLOPs	$mIoU^{ss}$	$mIoU^{ms}$	Params	FLOPs
ConvNeXt-T[26]	50.4	43.7	86M	741G	46.0	46.7	60M	939G
Swin-T[25]	50.4	43.7	86M	745G	44.5	45.8	60M	945G
HorNet-T[35]	51.7	44.8	80M	730G	48.1	48.9	52M	926G
InfiNet-T	51.9	46.2	77M	724G	46.7	47.4	50M	924G
ConvNeXt-S[26]	51.9	45.0	108M	827G	48.7	49.6	82M	1027G
Swin-S[25]	51.8	44.7	107M	838G	47.6	49.5	81M	1038G
HorNet-S[35]	52.7	45.6	107M	830G	49.2	49.8	81M	1030G
InfiNet-S	52.8	46.4	98M	802G	49.4	49.9	78M	1002G
ConvNeXt-B[26]	52.7	45.6	146M	964G	49.1	49.9	122M	1170G
Swin-B[25]	51.9	45.0	145M	982G	48.1	49.7	121M	1188G
HorNet-B[35]	53.3	46.1	144M	969G	50.0	50.5	121M	1174G
InfiNet-B	53.7	47.3	126M	906G	50.2	50.9	105M	1111G
ConvNeXt-L‡[26]	54.8	47.6	255M	1354G	53.2	53.7	235M	2458G
Swin-L‡[25]	53.9	46.7	253M	1382G	52.1	53.5	234M	2468G
HorNet-L‡[35]	55.4	48.0	251M	1363G	54.1	54.5	232M	2473G
InfiNet-XL‡	56.3	48.9	273M	1454G	54.6	55.2	253M	2544G

Table 3: More Results on isotropic models and different kind of Reproducing Kernel

(a) Isotropic Models

Model	Interact. Orders	Params (M)	FLOPs (G)	Top1 Acc.(%)
ConvNeXt-S(iso.)	no	22	4.3	79.7
Conv2Former(iso.)	2	23	4.3	81.2
DeiT-S[26]	3	22	4.6	79.8
HorNet-S(iso.)	2-5	22	4.5	80.6
InfiNet-S(iso.)	∞	22	4.3	81.4

(b) Ablation Study

Model	Interact. Orders	Params (M)	FLOPs (G)	Top1 Acc.(%)
InfiNet-$\oplus$	no	23	3.2	81.6
InfiNet-$\otimes$	2	23	3.2	82.1
InfiNet-2-polyno.	4	23	3.2	82.3
InfiNet-3-polyno.	6	23	3.2	82.5
InfiNet-T	∞	23	3.2	83.4

6.2 MS COCO Detection

Setups. We evaluate our models for object detection tasks on widely used MS COCO [23] benchmark. In the experiments, InfiNet-T/S/b/XL serves as the backbone network within Cascade Mask RCNN [5]. We use AdamW as the optimizer and a batch size of 16 and adhere to the 3× schedule, following ConvNeXt [26] and Swin [25]. We resize the input so that the longer side is at most 1333 and the short side is at most 800. We initialize the backbone model with ImageNet-1K pre-trained weights for T/S/B models and ImageNet-22K pre-trained weights for XL model.

Results. As shown in Table 2, InfiNets comprehensively beat the non-interactive model ConvNeXt [26], and space-limited interactive Swin [25] and HorNet [35] under the same cascade Mask RCNN framework in box AP and mask AP. This means that for such dense prediction tasks, spatial interaction of features in a high-dimension space is crucial. The InfiNet series model obtain$0.9 \sim 1.5$ box AP and $1.3 \sim 2.5$ mask AP gain compared with non-interactive ConNeXt.

6.3 ADE20 Segmentation

Setups. We evaluate our models for the semantic segmentation task on widely used ADE20K [48] benchmark covering 150 semantic categories on 25K images, in which 20K are used for training. We use UperNet [42] for as the basic framework and adopt InfiNets as the backbone model. Training details follow the Swin [25], we use AdamW optimizer with learning rate 1×10^{-4} and batch size 16.

Results. The right half of Table 2 lists the mIoU and corresponding model size and FLOPs for different configurations. Our models beat most of the baseline in the segmentation task. The results show that as the model size increases, the performance gap between InfiNet and other baselines is getting larger, illustrating the scalability of InfiNet on segmentation.

6.4 Ablation Study

We use the additive operator, Hadamard product, quadratic polynomial kernel and cubic polynomial kernel, and RBF kernel in the kernel methods section of InfiNet, respectively, on a tiny size model, to verify the effect of the gradual expansion of the order of the interaction space up to infinite dimensions on the performance of the model. As in Table 3(b), we can see that the performance of the model is gradually improving as the order of the model interaction space increases up to infinite dimensions.

7 Conclusion

As a conclusion, in this paper, we propose that one of the key points of success of today's element-wise multiplication-based models is that they explore a high-dimensional feature interaction space through feature interactions. And the RBF kernel can greatly expand this interaction space into an infinite dimensional feature interaction space. Based on this observation, we propose InfiNet, a high-performance neural network that explores infinite-dimensional feature interactions while using a modern model structure, which has achieved state-of-the-art results on several visual tasks.

Limitations

Although we propose the use of kernel methods for infinite-dimensional feature interaction, the only kernel methods we have tried so far are the RBF kernel function and some polynomial kernels of finite dimension. Substitutions utilizing a variety of kernel, including Laplace kernels, exponential kernels, a learnable kernel, etc., can be considered in subsequent studies. Our model has only been performed in some basic computer vision tasks, and validation in language and other modalities still requires some effort. The training of our model is only performed in the supervised learning paradigm, and more training and validation on self-supervised tasks still require effort. In addition, to avoid additional hyperparameter tuning, we fixed the σ parameter in the RBF kernel to 1. This may have deprived us of the possibility of exploring the optimal InfiNet, but due to the high cost of training the model, we will leave the impact of this hyperparameter on the InfiNet as a follow-up work.

Broader Social Impact

InfiNet is a state-of-the-art vision neural network architecture. The advancements in computer vision neural network architectures hold significant potential for positive societal impact, particularly in enhancing healthcare diagnostics, improving security systems, and advancing autonomous transportation. However, it is crucial to address potential negative implications such as privacy concerns, algorithmic biases, and job displacement. Ensuring ethical development and deployment involves implementing strict data protection measures, promoting fairness and inclusivity in algorithm design, and supporting workforce retraining programs. By proactively managing these challenges, we can maximize the benefits of computer vision technologies while minimizing their risks. Engaging with diverse stakeholders will be essential to ensure these technologies are used responsibly and equitably.

Acknowledgement

We sincerely thank the reviewers for their valuable suggestions during the review stage.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. *Advances in neural information processing systems*, 21, 2008.

[4] B. Bai, J. Liang, G. Zhang, H. Li, K. Bai, and F. Wang. Why attentions may not be interpretable? In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 25–34, 2021.

[5] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[6] C. Chen, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold. Kervolutional neural networks. *arXiv preprint arXiv:1904.03955*, 2019.

[7] Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350, 2009.

[8] Z. Dai, H. Liu, Q. V. Le, and M. Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] E. Dohmatob, Y. Feng, P. Yang, F. Charton, and J. Kempe. A tale of tails: Model collapse as a change of scaling laws. *International Conference on Machine Learning*, 2024.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[12] F.-L. Fan, H.-C. Dong, Z. Wu, L. Ruan, T. Zeng, Y. Cui, and J.-X. Liao. One neuron saved is one neuron earned: On parametric efficiency of quadratic networks. *arXiv preprint arXiv:2303.06316*, 2023.

[13] A. Hashemi, H. Schaeffer, R. Shi, U. Topcu, G. Tran, and R. Ward. Generalization bounds for sparse random feature expansions. *Applied and Computational Harmonic Analysis*, 62:310–330, 2023.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[16] Q. Hou, C.-Z. Lu, M.-M. Cheng, and J. Feng. Conv2former: A simple transformer-style convnet for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[17] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[18] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

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

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] K. Li, Y. Wang, G. Peng, G. Song, Y. Liu, H. Li, and Y. Qiao. Uniformer: Unified transformer for efficient spatial-temporal representation learning. In *International Conference on Learning Representations*, 2022.

[22] S. Li, Z. Wang, Z. Liu, C. Tan, H. Lin, D. Wu, Z. Chen, J. Zheng, and S. Z. Li. Moganet: Efficient multi-order gated aggregation network. *International Conference on Learning Representation*, 2024.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[24] S. Liu, T. Chen, X. Chen, X. Chen, Q. Xiao, B. Wu, T. Kärkkäinen, M. Pechenizkiy, D. C. Mocanu, and Z. Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. In *The Eleventh International Conference on Learning Representations*, 2023.

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[26] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[27] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[28] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[29] X. Ma, X. Dai, Y. Bai, Y. Wang, and Y. Fu. Rewrite the stars. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.

[30] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks. In *Advances in neural information processing systems*, pages 2627–2635, 2014.

[31] J. Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446, 1909.

[32] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris, and D. M. Hummels. On the training of radial basis function classifiers. *Neural networks*, 5(4):595–603, 1992.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[34] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[35] Y. Rao, W. Zhao, Y. Tang, J. Zhou, S. N. Lim, and J. Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *Advances in Neural Information Processing Systems*, 35:10353–10366, 2022.

[36] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[39] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[40] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378, 2016.

[41] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023.

[42] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

[43] C. Xu, X. Wang, F. Yu, J. Xiong, and X. Chen. Quadranet v2: Efficient and sustainable training of high-order neural networks with quadratic adaptation. *arXiv preprint arXiv:2405.03192*, 2024.

[44] C. Xu, F. Yu, Z. Xu, C. Liu, J. Xiong, and X. Chen. Quadranet: Improving high-order neural interaction efficiency with hardware-aware quadratic neural networks. *The 29th Asia and South Pacific Design Automation Conference*, 2024.

[45] Z. Xu, F. Yu, J. Xiong, and X. Chen. Quadralib: A performant quadratic neural network library for architecture optimization and design exploration. *Proceedings of Machine Learning and Systems*, 4:503–514, 2022.

[46] J. Yang, C. Li, X. Dai, and J. Gao. Focal modulation networks. *Advances in Neural Information Processing Systems*, 35:4203–4217, 2022.

[47] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[48] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

A Appendix

A.1 Training Details

The training details for ImageNet experiments are shown in Table 4 and Table 5.

Table 4: Training details for ImageNet-1K experiments

Configuration	InifNet-T/S/B/L
Input resolution	224^2
Epochs	300
Batch size	192/128/64/64
Optimizer	AdamW
AdamW (β_1, β_2)	$0.9, 0.999$
Learning rate	0.004
Learning rate decay	Cosine
Weight decay	0.05
Warmup epochs	20
Label smoothing ϵ	0.1
Stochastic Depth	Y
Rand Augment	9/0.5
Repeated Augment	Y
Erasing prob.	0.25
ColorJitter	N
Gradient Clipping	Y
EMA decay	Y

Table 5: Training details for ImageNet-21K experiments

Configuration	IN-21K PT		IN-1K FT	
	L	XL	L	XL
Input resolution	224^2		384^2	
Epochs	90		30	
Batch size	256		64	
Optimizer	AdamW		AdamW	
AdamW (β_1, β_2)	$0.9, 0.999$		$0.9, 0.999$	
Learning rate	4×10^{-3}		5×10^{-5}	
Learning rate decay	Cosine		Cosine	
Weight decay	0.1		0.00001	
Warmup epochs	5		0	
Label smoothing ϵ	0.2		0.1	0.2
Rand Augment	9/0.5		9/0.5	
Repeated Augment	N		N	
Erasing prob.	0.25		0.25	
Gradient Clipping	5		5	
EMA decay	N		Y	

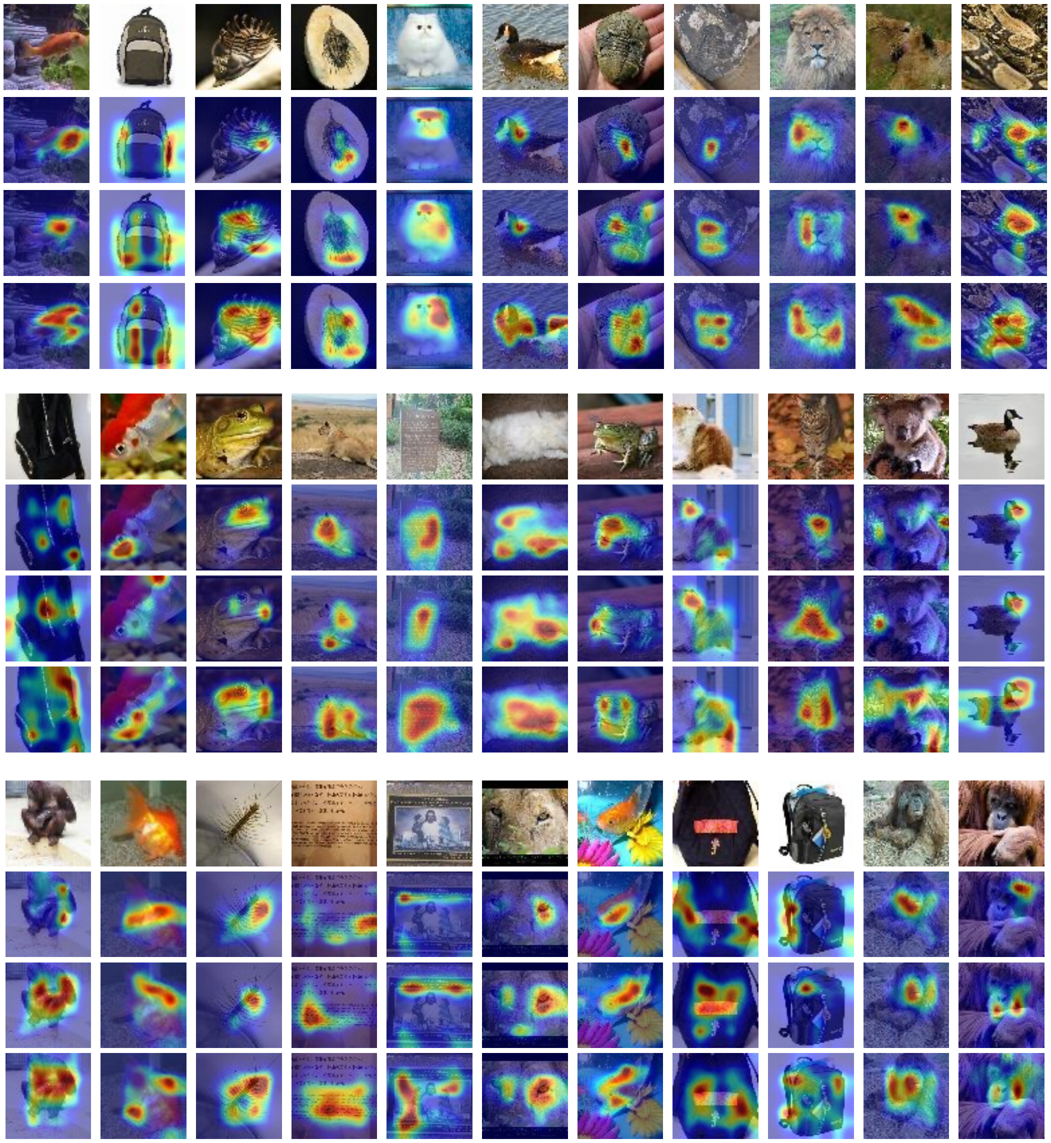


Figure 4: Visualization Comparison of (1) Feature Representation Space model, (2) Finite Feature Interaction Space model, (3) Infinite-Dimensional Feature Interaction model

A.2 More Visualization Comparison

In Fig. 4, we provide more Class Activation Mapping to illustrate the interpretability of interaction models. Infinite-dimensional Interaction models capture more totality of objects within a category, illustrates the importance of this infinite-dimensional interaction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We attribute the success of modern models to an feature interaction and claim the kernel methods can expand this feature interaction space. And propose the InfiNet. They are included in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: After the Conclusion

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: In Section 3 and 4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Section 6 and Appendix A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All datasets we use in this paper are third-party open source.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In Section 6 and Appendix A.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Training for multiple times is too expensive.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section 6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper is with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: After the Conclusion

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and baselines are cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.