Core-KAN: Continuous Vision Kernels with Kolmogorov-Arnold Networks

Lan Guo[1], Mengling Li[1], Haoran Li[2], Jun Shen[3], Yuanbo Jiang[1], Qingguo Zhou[1], Binbin Yong[1]*

[1] School of Information Science and Engineering, Lanzhou University
[2] Department of Data Science and AI, Monash University
[3] School of Computing and Information Technology, University of Wollongong

Abstract

Conventional convolutional kernels are typically defined on fixed discrete grids, limiting their ability to accommodate heterogeneous local structures. Existing adaptive operators improve flexibility, but often couple geometric scale variation with content-dependent filtering, while incurring high computational cost from per-location kernel generation. To decouple geometric scale adaptation from content-dependent filtering while avoiding expensive per-location kernel generation, we propose Continuous Relative-scale KAN (Core-KAN), a relative-scale-conditioned continuous convolution operator. In detail, Core-KAN maps input features into a compact latent basis space and uses a lightweight scale controller to predict local scales relative to an exponential moving average reference. A KAN-based generator represents depth-wise kernel bases as continuous coordinate functions, allowing the operator to synthesize spatial filters at arbitrary resolutions rather than being confined to a fixed lattice. Instead of synthesizing independent kernels at every location, it constructs a compact bank of scale-conditioned kernel responses and interpolates them according to the predicted local scale map. An independent mixing controller further combines the interpolated basis responses based on local content, explicitly decoupling geometric scale adaptation from content-dependent filtering. Together with lightweight pointwise projections, this design forms a low-rank dynamic convolution that scales efficiently with kernel size and can be readily integrated into hierarchical vision backbones. Extensive experiments across three representative computer vision tasks demonstrate that Core-KAN consistently outperforms strong convolutional and dynamic-kernel baselines while introducing only marginal parameter and computational overhead. Core-KAN provides an efficient and general framework for continuous, scale-adaptive convolution across diverse vision tasks.

Introduction

Local aggregation is a fundamental mechanism for building visual representations. Convolution remains one of the most reliable choices because a small spatially shared kernel provides translation equivariance, computational efficiency, and a strong locality bias (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2015; He et al. 2016; Liu et al. 2022; Woo et al. 2023). Yet the same property that makes convolution efficient also limits its adaptivity: a fixed discrete kernel profile is applied to textures, object boundaries, thin structures, and homogeneous regions alike. Modern vision backbones therefore face a recurring tension between the efficiency of shared convolutional operators and the need for location-specific spatial reasoning.

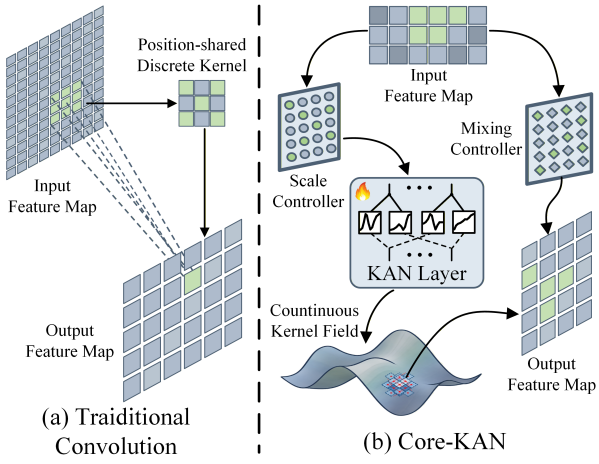

Figure 1: Conceptual comparison between conventional convolution and Core-KAN. (a) Conventional convolution shares a fixed discrete kernel across locations. (b) Core-KAN conditions a shared KAN-parameterized continuous kernel field on local relative scales and independently mixes basis responses according to content, enabling location-specific aggregation on a fixed sampling grid. Implementation details are omitted for clarity.

A direct way to increase adaptivity is to generate or select a different kernel at each location. However, fully position-specific kernels weaken the parameter sharing that makes convolution transferable, and dense kernel synthesis is expensive in both computation and memory. Existing approaches address parts of this tension. Large-kernel and multi-branch networks enlarge or diversify receptive fields, but still operate over finitely parameterized filters (Szegedy et al. 2015; Li et al. 2019; Ding et al. 2022; Liu et al. 2023; Ding et al. 2024). Conditional and dynamic convolutions improve content adaptivity by routing among learned components or modulating kernel weights (Yang et al. 2019; Chen et al.

*Corresponding author: Binbin Yong(yongbb@lzu.edu.cn)

2020; Ma et al. 2020; Li, Zhou, and Yao 2022; Jia et al. 2016; Zhou et al. 2021). Deformable convolutions adapt sampling locations, but still attach weights to a finite set of displaced points (Dai et al. 2017; Zhu et al. 2019; Wang et al. 2023; Xiong et al. 2024). Continuous convolutional representations learn coordinate-to-weight functions, but their kernel fields are typically shared at the layer level rather than conditioned densely on local structure (Wang et al. 2018; Wu, Qi, and Li 2019; Romero et al. 2022b,a). Thus, an efficient operator that provides dense spatial adaptation through a shared continuous kernel family remains underexplored.

Our key observation is that spatial adaptation in convolution involves two distinct decisions. The first is geometric: local scale and structure may require a sharper, broader, or differently shaped kernel profile. The second is content-dependent: local visual evidence may require a different mixture of latent filtering patterns. Existing dynamic-kernel mechanisms often bind these roles into a single routing signal, reducing interpretability by obscuring whether the operator changes kernel geometry, feature-dependent composition, or merely selects among discrete alternatives. Moreover, absolute scale predictions can drift across feature stages and optimization dynamics, making dense scale conditioning unstable without calibration to a feature-level reference.

Based on this observation, we propose Continuous Relative-scale KAN (Core-KAN), a Continuous Relative-scale KAN convolution operator for spatially adaptive visual aggregation. As illustrated in Figure 1, unlike conventional convolution with a fixed discrete kernel, Core-KAN represents depthwise kernel bases as a shared continuous coordinate field parameterized by a Kolmogorov-Arnold Network (KAN) (Liu et al. 2025). A lightweight scale controller predicts dense local scales and normalizes them with an exponential moving average reference. The resulting relative scales transform the queried coordinates, enabling location-specific kernel profiles along a continuous trajectory on a fixed regular sampling grid. An independent mixing controller composes basis responses according to local content, decoupling geometric scale adaptation from content-dependent composition. For efficiency, Core-KAN samples the continuous kernel field at a compact set of scale supports, constructs a response bank, and interpolates neighboring responses according to each local scale. This avoids position-wise kernel synthesis and confines adaptive computation to a compact low-rank basis space with lightweight pointwise projections.

Our contributions are threefold. First, we propose Core-KAN, a relative-scale-conditioned continuous convolution operator that parameterizes a shared coordinate-to-kernel field with a KAN, enabling smooth location-specific adaptation on a fixed sampling grid. Second, we develop an efficient support-and-interpolate scheme with a dual-control mechanism that decouples geometric scale adaptation from content-dependent basis composition. Finally, extensive evaluations on ImageNet-1K, COCO, and ADE20K demonstrate that Core-KAN is an effective plug-and-play operator with consistent performance gains and interpretable adaptive behavior.

Related Work

Adaptive Convolutional Operators

Adaptive ConvNets enlarge the receptive field, select among filters, or generate input-dependent weights. Selective Kernel Networks and large-kernel models provide multiple or wider spatial supports (Li et al. 2019; Ding et al. 2022; Liu et al. 2023; Ding et al. 2024). Recent efficient backbones further use selective large kernels, multi-scale kernels, anisotropic strip kernels, and partial-channel processing (Li et al. 2023; Cai et al. 2024; Yuan et al. 2026; Huang et al. 2026). RefConv reparameterizes filters from pretrained kernels, while SCConv reduces spatial and channel redundancy (Cai et al. 2025; Li, Wen, and He 2023). These designs improve context modeling or efficiency, but their filters remain finite tensors or branches.

Conditional and dynamic convolution instead adapts learned components from the input. CondConv and Dynamic Convolution combine experts (Yang et al. 2019; Chen et al. 2020); ODConv attends over spatial, channel, and kernel dimensions (Li, Zhou, and Yao 2022); KernelWarehouse assembles kernels from shared cells (Li and Yao 2024); and FDConv modulates frequency-grouped kernel parameters (Chen et al. 2025). Spatially varying filters are also produced by Dynamic Filter Networks, DDF, pixel-adaptive convolution, and involution (Jia et al. 2016; Zhou et al. 2021; Su et al. 2019; Li et al. 2021).

Sampling Adaptation and Continuous Kernels

Deformable convolution approaches spatial adaptivity from the sampling side. By predicting offsets and modulation weights, they allow the operator to collect evidence from irregular positions around each location (Dai et al. 2017; Zhu et al. 2019; Xiong et al. 2024). This mechanism is powerful when object geometry suggests that the sampling grid itself should move. Core-KAN follows a different principle. It keeps the regular convolutional grid intact and adapts the kernel function evaluated on that grid. Thus, the operator preserves the implementation structure and locality bias of convolution while allowing the filter profile to vary continuously.

Continuous kernel methods provide another route to parameter efficiency by representing weights as coordinate-conditioned functions rather than independent lattice parameters. Prior work has shown that coordinate-to-weight mappings can compactly describe kernels for point clouds, long sequences, or large spatial supports (Wang et al. 2018; Romero et al. 2022b,a; Kim and Park 2023). However, these continuous kernels are usually learned as layer-level or globally controlled functions. They are not primarily designed for dense, location-wise kernel adaptation inside standard 2D visual backbones.

KANs for Visual Modeling

Kolmogorov-Arnold Networks replace fixed scalar weights and activations with learnable univariate edge functions, giving them a different functional parameterization from conventional multilayer perceptrons (Liu et al. 2025). This makes the kernel transformation traceable as combinations

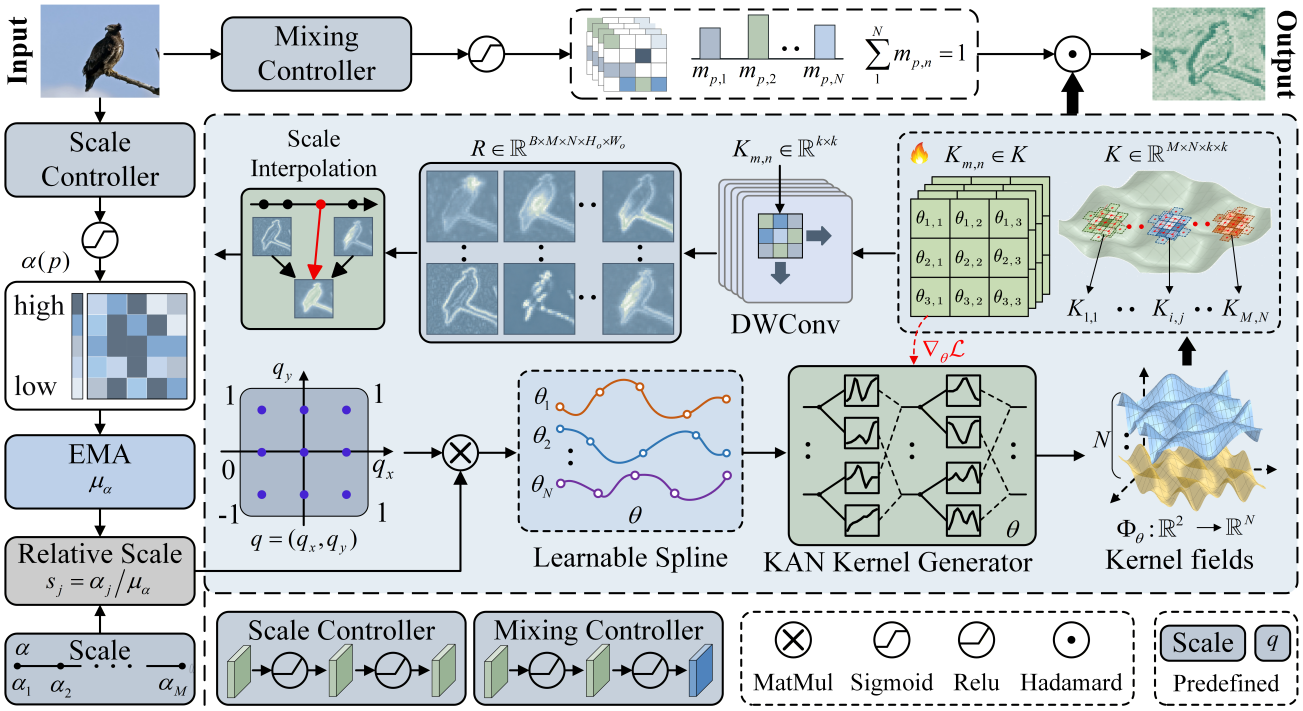

Figure 2: Overview of Core-KAN. The scale controller predicts a dense scale field $\alpha(p)$, and an operator-specific EMA reference μ_α converts absolute scale supports into relative supports. These supports transform a predefined coordinate grid q before it is evaluated by the KAN kernel generator. The learnable spline parameters θ define N shared continuous kernel fields, from which M sets of fixed-size kernels are sampled. Depthwise convolution produces a response bank $\mathcal{R}$, and each location interpolates its two neighboring support responses.

of one-dimensional functions. Unlike MLPs, which apply high-dimensional nonlinear changes, each input coordinate's effect on the output can be understood through the shape of its corresponding edge function. This gives the kernel evolution explicit geometric meaning. Existing visual uses of KANs mainly treat them as feature transformation modules, inserting KAN style nonlinear mappings into local or tokenized visual representations (Bodner et al. 2024; Li et al. 2025). In that setting, the KAN directly transforms image features.

Subsequent studies have broadened this paradigm to other building blocks of visual architectures. For instance, KAT replaces MLP blocks in Transformers with scalable rational-function KAN layers (Yang and Wang 2025), whereas KAC incorporates a KAN-based classifier for continual visual learning (Hu et al. 2025). Despite their emerging potential across vision tasks, leveraging the continuous functional parameterization of KANs to implement efficient, spatially adaptive convolution operators remains largely unexplored.

Method

We propose **Core-KAN**, a relative-scale-conditioned continuous convolution operator that decouples geometric scale adaptation from content-dependent basis composition. As illustrated in Figure 2, a scale controller predicts a dense local scale field, while an exponential moving average (EMA) provides an operator-specific reference for relative-scale normalization. A shared Kolmogorov-Arnold Network (KAN) maps scale-transformed coordinates to continuous kernel weights. Instead of synthesizing a different kernel at every spatial position, Core-KAN samples this continuous field at a compact set of scale supports, constructs a response bank using shared depthwise convolutions, and interpolates neighboring responses according to the local relative scale. An independent mixing controller then performs content-dependent composition over the latent basis responses.

Operator Formulation

Let $X \in \mathbb{R}^{B\times C_{\text{in}}\times H\times W}$ and $Y \in \mathbb{R}^{B\times C_{\text{out}}\times H_o\times W_o}$ denote the input and output features. Core-KAN first projects the input into a compact N-dimensional basis space:

$$V = P_{\text{in}}(X), \qquad V \in \mathbb{R}^{B\times N\times H\times W}, \tag{1}$$

where P_{in} is a 1×1 convolution. The operator computes one scale-adapted response $\widetilde{R}_{b,n}(p)$ for each sample b, basis n, and output location p. It then modulates these responses with content-dependent weights $m_b(p,n)$ and applies a pointwise output projection:

$$Y_{b,c}(p) = \sum_{n=1}^{N} W^{\text{out}}_{c,n}\, m_b(p,n)\, \widetilde{R}_{b,n}(p), \tag{2}$$

where W^{out} is the weight matrix of P_{out}. Equation (2) exposes the two complementary controls in Core-KAN: the

scale branch changes the spatial profile of each basis response, whereas the mixing branch changes its content-dependent contribution.

For clarity, the spatial equations below use unit-stride indexing. General stride, padding, and output resolution follow the standard convolutional convention.

Decoupled Scale and Mixing Controllers

Both controllers operate on the original input X. In our implementation, each controller contains two lightweight 3×3 convolution-GroupNorm-ReLU blocks followed by a 1×1 prediction layer.

Dense scale prediction. The scale controller C_α predicts a bounded scalar at every spatial location:

$$\begin{aligned} \widehat{\alpha}_b(p) &= \sigma(C_\alpha(X_b)(p)), \\ \alpha_b(p) &= \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\,\widehat{\alpha}_b(p), \end{aligned} \tag{3}$$

where σ is the sigmoid function and $\alpha_b(p) \in [\alpha_{\min}, \alpha_{\max}]$.

Because feature statistics vary across layers and training iterations, the same absolute prediction can have different meanings in different operators. Each Core-KAN layer therefore maintains a non-learnable EMA reference. At training iteration t, it is updated by

$$\begin{aligned} \overline{\alpha}^{(t)} &= \frac{1}{BHW} \sum_{b=1}^{B} \sum_{p} \alpha_b^{(t)}(p), \\ \mu_\alpha^{(t)} &= \mathrm{clip}_{[\alpha_{\min}, \alpha_{\max}]}\left(\rho \mu_\alpha^{(t-1)} + (1-\rho)\,\mathrm{sg}\left(\overline{\alpha}^{(t)}\right)\right), \end{aligned} \tag{4}$$

where ρ is the EMA momentum and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The stored reference is fixed at inference time. The local relative scale is then

$$s_b(p) = \frac{\alpha_b(p)}{\mu_\alpha}. \tag{5}$$

This normalization expresses the predicted scale relative to the typical scale of the current operator.

Content-dependent mixing. The independent mixing controller C_m predicts a simplex distribution over the N latent bases:

$$\begin{aligned} m_b(p,:) &= \mathrm{softmax}(C_m(X_b)(p)), \\ m_b(p,n) &\geq 0, \qquad \sum_{n=1}^{N} m_b(p,n) = 1. \end{aligned} \tag{6}$$

The mixing weights depend on local content but do not determine the relative scale used to query the kernel field. This separation prevents the adaptation of the geometric scale and the selection of the bases from being represented by a single routing signal.

KAN-Parameterized Continuous Kernel Fields

Relative-scale supports. We place M uniformly spaced absolute supports in the bounded prediction interval:

$$\alpha_j = \alpha_{\min} + \frac{j-1}{M-1}(\alpha_{\max} - \alpha_{\min}), \qquad j = 1, \ldots, M. \tag{7}$$

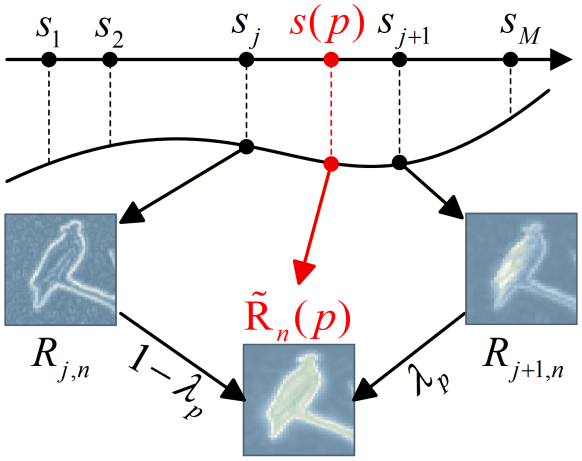

Figure 3: Pixel-wise scale interpolation in response space. For a location p with $s_j \leq s(p) \leq s_{j+1}$, Core-KAN retrieves the neighboring support responses $R_{j,n}(p)$ and $R_{j+1,n}(p)$ and linearly combines them using weights $1-\lambda_p$ and λ_p, respectively. Batch indices are omitted for clarity.

The supports are converted to relative coordinates using the same EMA reference as the dense scale map:

$$s_j = \frac{\alpha_j}{\mu_\alpha}, \qquad j = 1, \ldots, M. \tag{8}$$

Thus, $s_b(p)$ and $\{s_j\}_{j=1}^{M}$ lie in the same operator-calibrated scale coordinate system.

Continuous coordinate-to-kernel mapping. Let $\mathcal{G}_k = \{\delta_r\}_{r=1}^{k^2}$ denote the offsets of a regular $k \times k$ convolutional grid, and let $q_r = (q_{x,r}, q_{y,r}) \in [-1,1]^2$ be the normalized coordinate associated with δ_r. A shared KAN maps a continuous two-dimensional coordinate to N scalar kernel fields:

$$\Phi_\theta : \mathbb{R}^2 \to \mathbb{R}^N. \tag{9}$$

For a KAN with L layers, the ℓ-th layer can be written as

$$z_v^{(\ell+1)} = \sum_{u=1}^{d_\ell} \phi_{v,u}^{(\ell)}\left(z_u^{(\ell)}\right), \qquad z^{(0)} = x, \tag{10}$$

where each $\phi_{v,u}^{(\ell)}$ is a learnable univariate edge function parametrized by spline. The symbol θ collectively denotes all learnable spline parameters, and $\Phi_\theta(x) = z^{(L)}$.

For a relative scale s, Core-KAN evaluates the same KAN at the scale-transformed coordinates $\{sq_r\}_{r=1}^{k^2}$. The unnormalized n-th basis kernel is

$$G_n(s) = \mathrm{reshape}_{k\times k}\left(\{[\Phi_\theta(sq_r)]_n\}_{r=1}^{k^2}\right). \tag{11}$$

Each sampled kernel is standardized over its spatial entries and rescaled to a Kaiming-compatible magnitude:

$$K_n(s) = \sqrt{\frac{2}{k^2}}\,\frac{G_n(s) - \mathrm{mean}(G_n(s))}{\mathrm{std}(G_n(s)) + \varepsilon}. \tag{12}$$

Sampling the continuous fields at the M relative supports yields

$$\mathcal{K} = \{K_{j,n} = K_n(s_j)\}_{\substack{j=1,\dots,M \\ n=1,\dots,N}} \in \mathbb{R}^{M\times N\times k\times k}. \quad (13)$$

The kernels in $\mathcal{K}$ are therefore samples from the same shared continuous fields, rather than MN independently stored filters. Moreover, the discrete offsets δ_r remain fixed for all scales. Varying s changes the coordinates queried from the continuous kernel field, and hence the kernel weights, but does not resize or deform the $k \times k$ sampling grid.

Support-and-Interpolate Readout

Directly querying $K_n(s_b(p))$ at every spatial location would require synthesizing a distinct kernel for each position. Core-KAN instead evaluates the kernel field only at the M supports and realizes dense adaptation through response-space interpolation.

Support-sampled response bank. Each support kernel is applied depthwise to the projected basis feature V:

$$R_{b,j,n}(p) = \sum_{r=1}^{k^2} K_{j,n}(\delta_r)\, V_{b,n}(p+\delta_r). \quad (14)$$

Stacking the responses over all supports and bases gives

$$\mathcal{R} \in \mathbb{R}^{B\times M\times N\times H_o\times W_o}. \quad (15)$$

This response bank requires only M shared depthwise convolutions in the compact basis space.

Pixel-wise scale interpolation. If the convolution changes spatial resolution, the scale map is first bilinearly resized to (H_o, W_o). We reuse $\alpha_b(p)$ and $s_b(p)$ for the resized values. For the neighboring supports satisfying $\alpha_j \le \alpha_b(p) \le \alpha_{j+1}$, equivalently $s_j \le s_b(p) \le s_{j+1}$, the interpolation coefficient is

$$\lambda_{b,p} = \frac{\alpha_b(p)-\alpha_j}{\alpha_{j+1}-\alpha_j} = \frac{s_b(p)-s_j}{s_{j+1}-s_j}, \qquad \lambda_{b,p}\in[0,1]. \quad (16)$$

The equality follows because the local prediction and all support values share the same positive reference μ_α. The scale-adapted response is then

$$\widetilde{R}_{b,n}(p) = (1-\lambda_{b,p})\, R_{b,j,n}(p) + \lambda_{b,p} R_{b,j+1,n}(p). \quad (17)$$

Values outside the supported interval are clamped to its nearest boundary; at an exact support location, the interpolation reduces to that support response.

To relate the efficient readout to direct continuous evaluation, define

$$R^{\star}_{b,n}(p) = \sum_{r=1}^{k^2} K_n(s_b(p))\,(\delta_r)\, V_{b,n}(p+\delta_r), \quad (18)$$

which evaluates the continuous kernel field separately at every position. Because convolution is linear in the kernel weights, Equation (17) is exactly equivalent to applying the interpolated local kernel

$$\widetilde{K}_{b,n,p} = (1-\lambda_{b,p})\, K_{j,n} + \lambda_{b,p} K_{j+1,n}, \quad (19)$$

Method	Params (M) ↓	Top-1 (%) ↑	Top-5 (%) ↑
ResNet-50 [CVPR'16]	25.56	78.44	94.24
DY-Conv [CVPR'20]	100.88	79.00	94.27
ODConv [ICLR'22]	90.67	78.52	94.01
SCConv [CVPR'23]	17.69	79.89	94.76
RefConv [TNNLS'25]	36.97	79.91	94.61
PartialNet [AAAI'26]	18.00	80.61	95.13
KernelWarehouse [ICML'24]	102.02	<u>81.05</u>	<u>95.21</u>
FDConv [CVPR'25]	29.20	80.36	95.02
Core-KAN (Ours)	26.61	**81.45**	**95.68**

Table 1: ImageNet-1K validation results. Reported or reproduced training recipes differ across methods, so the table is intended as a reference comparison. **Bold** and <u>underlined</u> values indicate the best and second-best accuracy, respectively.

namely,

$$\widetilde{R}_{b,n}(p) = \sum_{r=1}^{k^2} \widetilde{K}_{b,n,p}(\delta_r)\, V_{b,n}(p+\delta_r). \quad (20)$$

Thus, response interpolation exactly realizes the piecewise-linearly interpolated kernel in Equation (19), without materializing a location-specific kernel tensor. Relative to the direct query in Equation (18), it is a controllable piecewise-linear approximation along the learned scale-conditioned kernel trajectory. Increasing M improves the sampling density, while a compact M reduces response-bank computation.

Content Mixing and Low-Rank Interpretation

The interpolated basis responses encode geometric scale adaptation. Core-KAN subsequently applies the independent content weights from Equation (6):

$$Z_{b,n}(p) = m_b(p,n)\, \widetilde{R}_{b,n}(p). \quad (21)$$

When required, the mixing map is bilinearly resized to (H_o, W_o) and renormalized over n. The output is $Y = P_{\text{out}}(Z)$, yielding Equation (2).

The complete operator admits a low-rank interpretation. Let W^{in} and W^{out} denote the input and output projection matrices. Combining Equations (1), (20), and (2) gives the effective location-dependent kernel between input channel c' and output channel c:

$$W^{\text{eff}}_{b,c,c',p}(\delta) = \sum_{n=1}^{N} W^{\text{out}}_{c,n}\, m_b(p,n)\, \widetilde{K}_{b,n,p}(\delta)\, W^{\text{in}}_{n,c'}. \quad (22)$$

Here, $s_b(p)$ controls the spatial profile $\widetilde{K}_{b,n,p}$, whereas $m_b(p,n)$ controls the content-dependent composition of the latent bases. The surrounding pointwise projections share channel-mixing factors across locations, confining the spatially adaptive computation to the compact N-dimensional basis space.

Experiments

Experimental Setup

Core-KAN configuration. Unless otherwise specified, Core-KAN uses a fixed spatial support of $k = 3$, $N = 16$

Method	Object Detection						Instance Segmentation					
	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{box}_{S}	AP^{box}_{M}	AP^{box}_{L}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}_{S}	AP^{mask}_{M}	AP^{mask}_{L}
ResNet-50 [CVPR'16]	37.9	58.7	41.2	21.6	41.5	49.3	34.5	55.6	36.8	15.9	37.1	50.4
DY-Conv [CVPR'20]	39.2	60.3	42.5	23.0	42.9	51.4	34.7	56.0	37.1	16.4	36.9	51.1
ODConv [ICLR'22]	40.1	61.5	43.6	24.0	43.6	52.3	36.7	58.5	39.6	18.6	39.0	52.8
KernelWarehouse [ICML'24]	42.4	**65.4**	46.3	**27.2**	46.2	54.6	38.9	62.0	41.5	**22.7**	42.6	53.1
FDConv [CVPR'25]	42.5	64.8	46.2	26.4	47.0	54.9	38.3	61.8	41.0	19.6	42.4	54.3
Core-KAN (Ours)	**43.0**	65.1	**47.0**	27.1	**47.1**	**55.8**	**39.5**	**62.1**	**42.4**	20.5	**42.8**	**57.2**

Table 2: COCO val2017 results using Mask R-CNN under the standard $1\times$ schedule (12 epochs). **Bold** and underlined values denote the best and second-best results, respectively; ties are marked equally.

Method	Object Detection						Instance Segmentation					
	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{box}_{S}	AP^{box}_{M}	AP^{box}_{L}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}_{S}	AP^{mask}_{M}	AP^{mask}_{L}
ResNet-50 [CVPR'16]	40.9	61.3	44.8	24.4	44.6	52.3	37.1	58.3	39.9	18.4	39.8	52.9
DY-Conv [CVPR'20]	41.9	63.0	45.7	25.8	45.6	53.8	36.9	58.5	39.7	18.8	39.4	53.2
ODConv [ICLR'22]	42.6	64.0	46.6	27.0	46.3	54.8	38.7	60.6	42.0	21.4	41.2	54.7
KernelWarehouse [ICML'24]	45.6	67.5	49.8	29.8	49.4	**59.0**	41.5	63.0	44.8	24.2	43.9	58.5
FDConv [CVPR'25]	45.7	67.3	50.4	30.5	49.7	58.4	41.7	62.5	45.0	24.0	44.2	58.7
Core-KAN (Ours)	**46.2**	**67.8**	**50.5**	**31.3**	**49.8**	58.9	**42.2**	**63.8**	**45.1**	**24.8**	**44.5**	**59.2**

Table 3: COCO val2017 results using Mask R-CNN under the standard $3\times$ schedule (36 epochs). **Bold** and underlined values denote the best and second-best results, respectively; ties are marked equally.

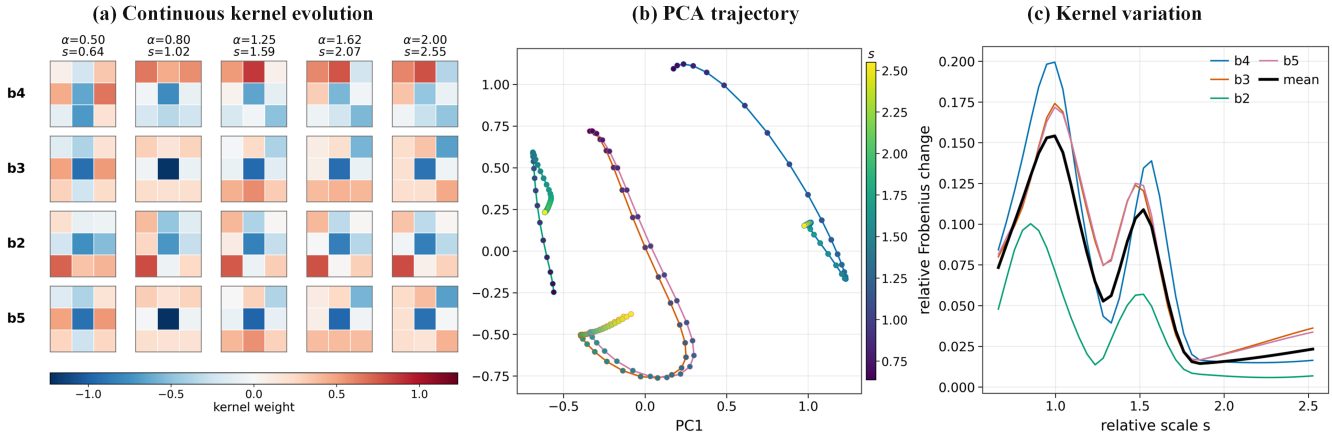

Figure 4: Evolution of learned 3×3 kernels in the last Core-KAN block. (a) Four basis fields queried at five relative scales. (b) PCA trajectories of densely sampled kernels. (c) Relative Frobenius change between adjacent queries. Smooth, basis-specific trajectories indicate a continuous kernel field rather than a discrete lookup table.

continuous kernel basis fields, and $M = 8$ uniformly spaced scale supports. The predicted local scale is bounded to $[\alpha_{\min}, \alpha_{\max}] = [0.5, 2.0]$, and the momentum of the exponential moving average scale reference is set to $\rho = 0.99$. Both the scale controller and the content-mixing controller use a hidden width of 32. The continuous kernel generator consists of two KAN layer with a grid size of 7 and a spline order of 3. For details, see Appendix A.2-A.6 and E.1-E.4.

Training and evaluation. Core-KAN is trained on ImageNet-1K (Russakovsky et al. 2015) for 300 epochs with AdamW, a batch size of 4,096, an initial learning rate of 4×10^{-3}, 20 warm-up epochs, cosine decay, and single-crop 224×224 validation. COCO 2017 (Lin et al. 2014) uses Mask R-CNN (He et al. 2017) with FPN (Lin et al. 2017) under the standard $1\times$ (12-epoch) and $3\times$ (36-epoch) schedules. For ADE20K (Zhou et al. 2017), the models are trained for 160K iterations. Within each downstream benchmark, all reproduced backbones share the same data pipeline, schedule, task head, and evaluation protocol.

Method	Params (M) ↓	mIoU (%) ↑	mAcc (%) ↑
ResNet-50 [CVPR'16]	66.00	40.00	49.61
DY-Conv [CVPR'20]	140.00	41.20	51.05
ODConv [ICLR'22]	131.00	42.36	52.51
KernelWarehouse [ICML'24]	141.00	43.20	53.30
FDConv [CVPR'25]	70.00	<u>43.50</u>	<u>53.63</u>
Core-KAN (Ours)	68.50	**44.19**	**54.38**

Table 4: ADE20K validation results with a ResNet-50 backbone. Params denotes the complete segmentation model. **Bold** and <u>underlined</u> values denote the best and second-best results, respectively.

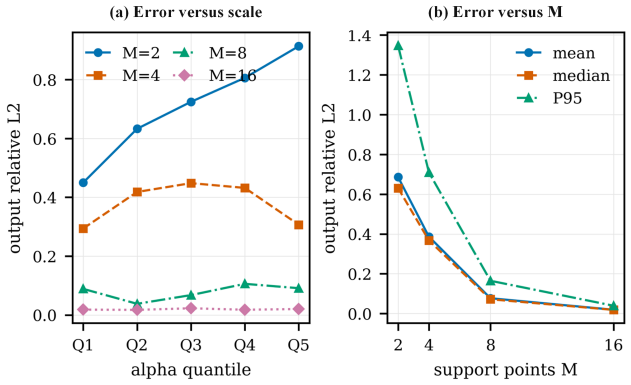

Figure 5: Interpolation fidelity to direct continuous-kernel evaluation. (a) Relative output error across five scale quantiles for different support counts M. (b) Mean, median, and 95th-percentile error. Denser supports consistently reduce the approximation error.

Main Results

Table 1 compares Core-KAN with dynamic convolution operators and recent efficient backbones on ImageNet-1K. Core-KAN achieves 81.45% top-1 and 95.68% top-5 accuracy with 26.61M parameters. Compared with ResNet-50, it delivers relative improvements of 3.84% and 1.53% in top-1 and top-5 accuracy, respectively, with a parameter overhead of only 4.11%. Core-KAN also provides relative gains of 0.49% in both accuracy metrics over KernelWarehouse while using 73.92% fewer parameters.

On COCO, Core-KAN consistently improves Mask R-CNN under both training schedules. As shown in Table 2, Core-KAN achieves 43.0 AP^{box} and 39.5 AP^{mask}, corresponding to relative improvements of 13.46% and 14.49% over ResNet-50, respectively. Compared with the strongest competing methods, Core-KAN further improves box AP over FDConv by 1.18% and mask AP over KernelWarehouse by 1.54%. The improvements are consistent across object scales: relative to ResNet-50, Core-KAN achieves gains of 25.46%, 13.49%, and 13.18% in AP^{box}_S, AP^{box}_M, and AP^{box}_L, respectively, together with corresponding gains of 28.93%, 15.36%, and 13.49% for mask prediction. Under the longer 3× schedule, Core-KAN obtains 46.2 AP^{box} and 42.2 AP^{mask} (Table 3), yielding relative improvements of 12.96% and 13.75% over ResNet-50, respectively. It also outperforms FDConv, the strongest competing method in terms of overall AP under this schedule, by 1.09% in box AP and 1.20% in mask AP. Core-KAN ranks first on five of the six box metrics and all six mask metrics. The only exception is AP^{box}_L, where its result is only 0.17% lower than that of KernelWarehouse (58.9 versus 59.0). These results demonstrate that the advantages of Core-KAN remain consistent under longer training and across detection and instance-segmentation tasks.

On ADE20K, Core-KAN achieves 44.19% mIoU and 54.38% mAcc with 68.50M parameters (Table 4). These results represent relative improvements of 10.47% and 9.61% over ResNet-50, respectively, with a parameter overhead of only 3.79% in the complete segmentation model. Core-KAN further improves mIoU and mAcc over FDConv by 1.59% and 1.40%, respectively, while using 2.14% fewer parameters.

Continuous Kernels and Interpolation

Figure 4 examines whether the KAN represents a continuous kernel family and, critically, whether this representation provides interpretable kernel evolution. Queries at increasing relative scales change both the signs and spatial arrangements of the normalized weights while retaining a fixed 3 × 3 sampling grid. The ordered, basis-specific PCA trajectories reveal that each basis function captures a distinct and interpretable pattern of scale-dependent kernel transformations. Dense queries form ordered, basis-specific PCA trajectories, demonstrating that the KAN learns semantically meaningful continuous trajectories rather than arbitrary discrete lookup tables.

Figure 5 compares support-based interpolation with direct per-location continuous-kernel evaluation. The approximation error decreases consistently as the number of supports increases across all scale quantiles and summary statistics. The default setting of $M = 8$ substantially reduces the error relative to smaller support sets, while $M = 16$ provides further improvement at additional response-bank cost. Thus, eight supports offer a practical trade-off between approximation fidelity and computational efficiency. For details, see Appendix B.1-B.4.

Conclusion

Core-KAN introduces dense spatial adaptation through a shared continuous kernel field. Relative-scale queries change kernel geometry, a separate controller mixes latent bases from image content, and support-based interpolation avoids explicit kernel synthesis at every position. Across ImageNet-1K, both COCO schedules, and ADE20K, Core-KAN improves on ResNet-50 with modest parameter overhead. Kernel trajectories and interpolation errors support the intended continuous formulation, while the controller maps show complementary spatial behavior. A remaining limitation is that the response-bank cost grows with the number of scale supports; reducing this cost without weakening interpolation fidelity is a useful direction for further work.

References

Bodner, A. D.; Tepsich, A. S.; Spolski, J. N.; and Pourteau, S. 2024. Convolutional Kolmogorov–Arnold Networks. *arXiv preprint arXiv:2406.13155*.

Cai, X.; Lai, Q.; Wang, Y.; Wang, W.; Sun, Z.; and Yao, Y. 2024. Poly Kernel Inception Network for Remote Sensing Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 27706–27716.

Cai, Z.; Ding, X.; Shen, Q.; and Cao, X. 2025. RefConv: Reparameterized Refocusing Convolution for Powerful ConvNets. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6): 11617–11631.

Chen, L.; Gu, L.; Li, L.; Yan, C.; and Fu, Y. 2025. Frequency Dynamic Convolution for Dense Image Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 30178–30188.

Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; and Liu, Z. 2020. Dynamic Convolution: Attention over Convolution Kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11030–11039.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 764–773.

Ding, X.; Zhang, X.; Han, J.; and Ding, G. 2022. Scaling Up Your Kernels to 31x31: Revisiting Large Kernel Design in CNNs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11963–11975.

Ding, X.; Zhang, Y.; Ge, Y.; Zhao, S.; Song, L.; Yue, X.; and Shan, Y. 2024. 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*, 5513–5524.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hu, Y.; Liang, Z.; Yang, F.; Hou, Q.; Liu, X.; and Cheng, M.-M. 2025. KAC: Kolmogorov–Arnold Classifier for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15297–15307.

Huang, H.; Xia, T.; Zhao, W.; and Ren, P. 2026. PartialNet: Compute Less, Perform Better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 21930–21938.

Jia, X.; De Brabandere, B.; Tuytelaars, T.; and Van Gool, L. 2016. Dynamic Filter Networks. In *Advances in Neural Information Processing Systems*, volume 29.

Kim, S.; and Park, E. 2023. SMPConv: Self-Moving Point Representations for Continuous Convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10289–10299.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25.

Li, C.; Liu, X.; Li, W.; Wang, C.; Liu, H.; Liu, Y.; Chen, Z.; and Yuan, Y. 2025. U-KAN Makes Strong Backbone for Medical Image Segmentation and Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(5): 4652–4660.

Li, C.; and Yao, A. 2024. KernelWarehouse: Rethinking the Design of Dynamic Convolution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 29201–29221. PMLR.

Li, C.; Zhou, A.; and Yao, A. 2022. Omni-Dimensional Dynamic Convolution. In *International Conference on Learning Representations*.

Li, D.; Hu, J.; Wang, C.; Li, X.; She, Q.; Zhu, L.; Zhang, T.; and Chen, Q. 2021. Involution: Inverting the Inherence of Convolution for Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12321–12330.

Li, J.; Wen, Y.; and He, L. 2023. SCConv: Spatial and Channel Reconstruction Convolution for Feature Redundancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6153–6162.

Li, X.; Wang, W.; Hu, X.; and Yang, J. 2019. Selective Kernel Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 510–519.

Li, Y.; Hou, Q.; Zheng, Z.; Cheng, M.-M.; Yang, J.; and Li, X. 2023. Large Selective Kernel Network for Remote Sensing Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16794–16805.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2117–2125.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, 740–755. Springer.

Liu, S.; Chen, T.; Chen, X.; Chen, X.; Xiao, Q.; Wu, B.; Kärkkäinen, T.; Pechenizkiy, M.; Mocanu, D. C.; and Wang, Z. 2023. More ConvNets in the 2020s: Scaling Up Kernels Beyond 51x51 Using Sparsity. In *International Conference on Learning Representations*.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11976–11986.

Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2025. KAN: Kolmogorov–Arnold Networks. In *The Thirteenth International Conference on Learning Representations*.

Ma, N.; Zhang, X.; Huang, J.; and Sun, J. 2020. WeightNet: Revisiting the Design Space of Weight Networks. In *Computer Vision – ECCV 2020*, volume 12360 of *Lecture Notes in Computer Science*, 776–792. Springer.

Romero, D. W.; Bruintjes, R.-J.; Bekkers, E. J.; Tomczak, J. M.; Hoogendoorn, M.; and van Gemert, J. C. 2022a. FlexConv: Continuous Kernel Convolutions with Differentiable Kernel Sizes. In *International Conference on Learning Representations*.

Romero, D. W.; Kuzina, A.; Bekkers, E. J.; Tomczak, J. M.; and Hoogendoorn, M. 2022b. CKConv: Continuous Kernel Convolution for Sequential Data. In *International Conference on Learning Representations*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

Su, H.; Jampani, V.; Sun, D.; Gallo, O.; Learned-Miller, E.; and Kautz, J. 2019. Pixel-Adaptive Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11166–11175.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going Deeper with Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Wang, S.; Suo, S.; Ma, W.-C.; Pokrovsky, A.; and Urtasun, R. 2018. Deep Parametric Continuous Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2589–2597.

Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; Wang, X.; and Qiao, Y. 2023. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14408–14419.

Woo, S.; Debnath, S.; Hu, R.; Chen, X.; Liu, Z.; Kweon, I. S.; and Xie, S. 2023. ConvNeXt V2: Co-Designing and Scaling ConvNets with Masked Autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16133–16142.

Wu, W.; Qi, Z.; and Li, F. 2019. PointConv: Deep Convolutional Networks on 3D Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9621–9630.

Xiong, Y.; Li, Z.; Chen, Y.; Wang, F.; Zhu, X.; Luo, J.; Wang, W.; Lu, T.; Li, H.; Qiao, Y.; Lu, L.; Zhou, J.; and Dai, J. 2024. Efficient Deformable ConvNets: Rethinking Dynamic and Sparse Operator for Vision Applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5652–5661.

Yang, B.; Bender, G.; Le, Q. V.; and Ngiam, J. 2019. CondConv: Conditionally Parameterized Convolutions for Efficient Inference. In *Advances in Neural Information Processing Systems*, volume 32.

Yang, X.; and Wang, X. 2025. Kolmogorov–Arnold Transformer. In *The Thirteenth International Conference on Learning Representations*.

Yuan, X.; Zheng, Z.; Li, Y.; Liu, X.; Liu, L.; Li, X.; Hou, Q.; and Cheng, M.-M. 2026. Strip R-CNN: Large Strip Convolution for Remote Sensing Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 12259–12267.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene Parsing Through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 633–641.

Zhou, J.; Jampani, V.; Pi, Z.; Liu, Q.; and Yang, M.-H. 2021. Decoupled Dynamic Filter Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6647–6656.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable ConvNets V2: More Deformable, Better Results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9308–9316.