西湖大學
WESTLAKE UNIVERSITY

Modern Convolutional Neural Networks

Siyuan Li

Westlake University, Zhejiang University
March, 2024

Timeline of Modern CNNs

InceptionNeXt
CVPR'2024 (2023)

DCN.V4
CVPR'2024

ConvNeXt
CVPR'2022

RepLKNet
CVPR'2022

SLaK
ICLR'2023

ConvNeXt.V2
CVPR'2023

DCN.V3
CVPR'2023

UniRepLKNet
CVPR'2024

Convolution Kernel Designs

Large-Kernel Conv + Gated Attentions

VAN (2022)
CVMJ'2023

HorNet
NeurIPS'2022

FocalNet
NeurIPS'2022

MogaNet (2022)
ICLR'2024

Mamba
arXiv'2023

VMamba
arXiv'2024

Content

1. **Modern CNNs: Macro Design and Pre-training**

   MetaFormer, ConvNeXt, ConvNeXt.V2 (SparK, A2MIM)

2. **Design of Convolution Kernels**

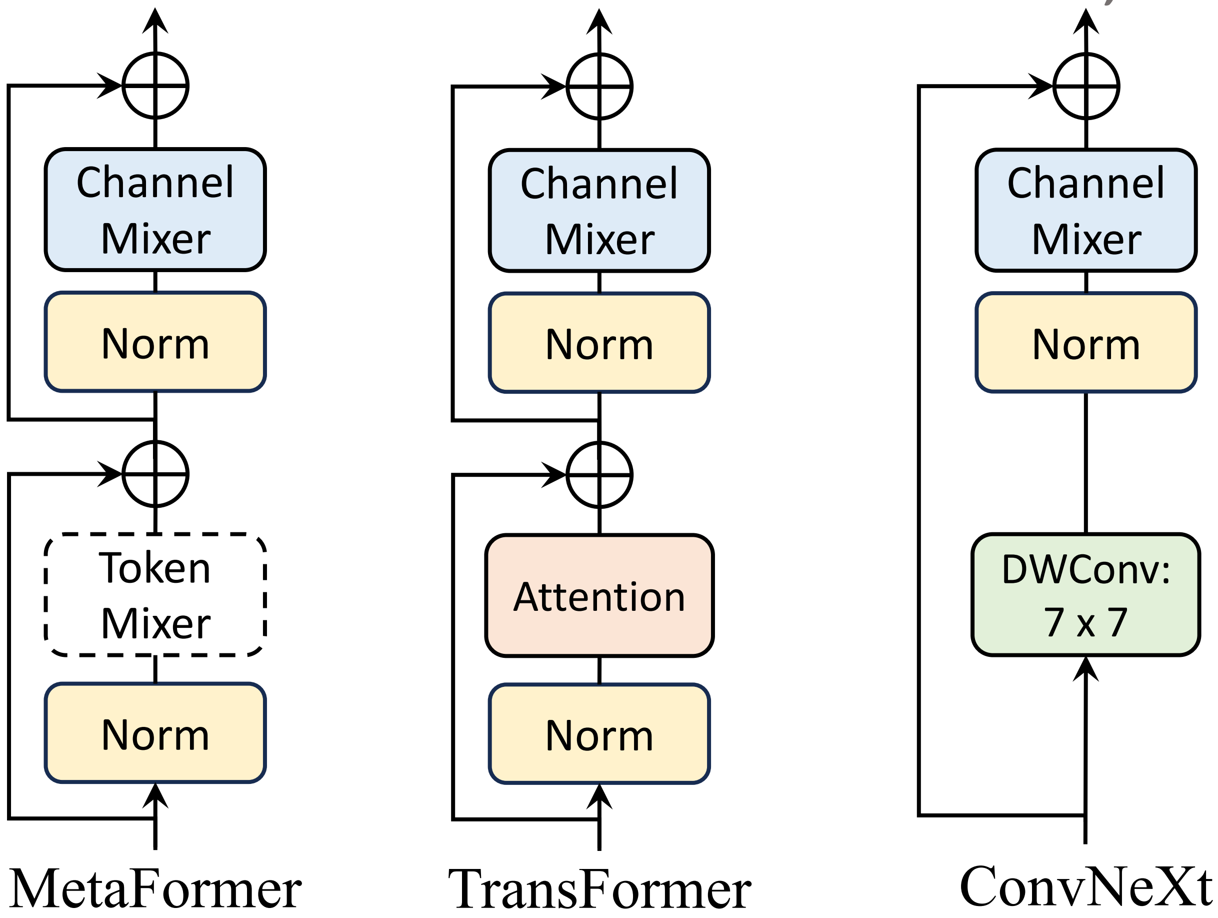
   RepLKNet, SLaK, InceptionNext, DCN.V3/V4, UniRepLKNet

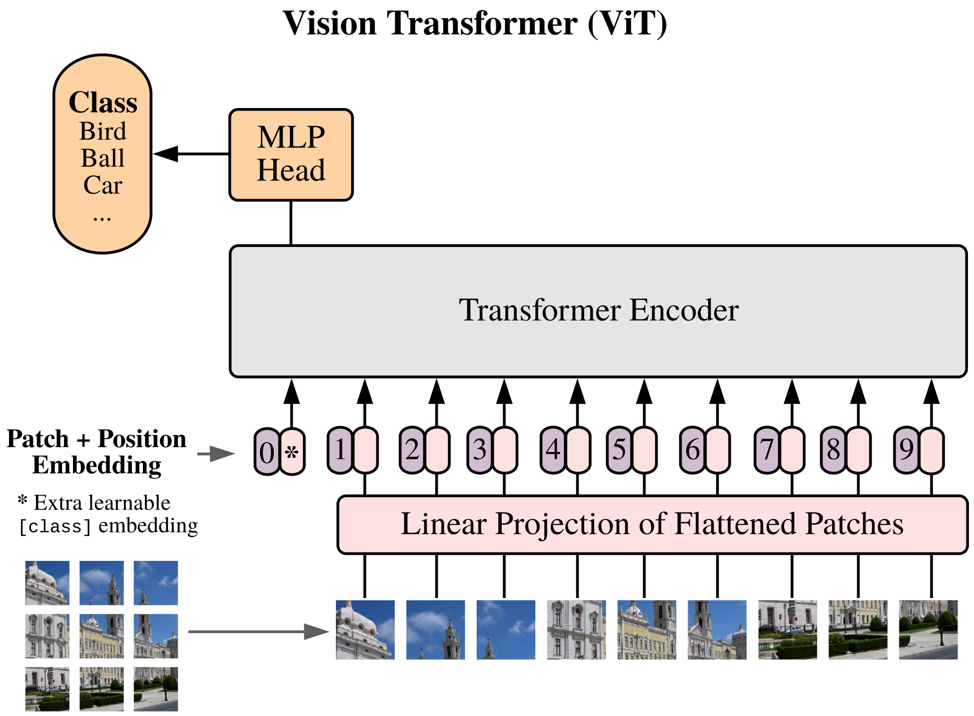
3. **Combining Large Kernel with Gated Attention**

   VAN, HorNet, FocalNet, MogaNet, Mamba, VMamba

Modern CNNs: Macro Design

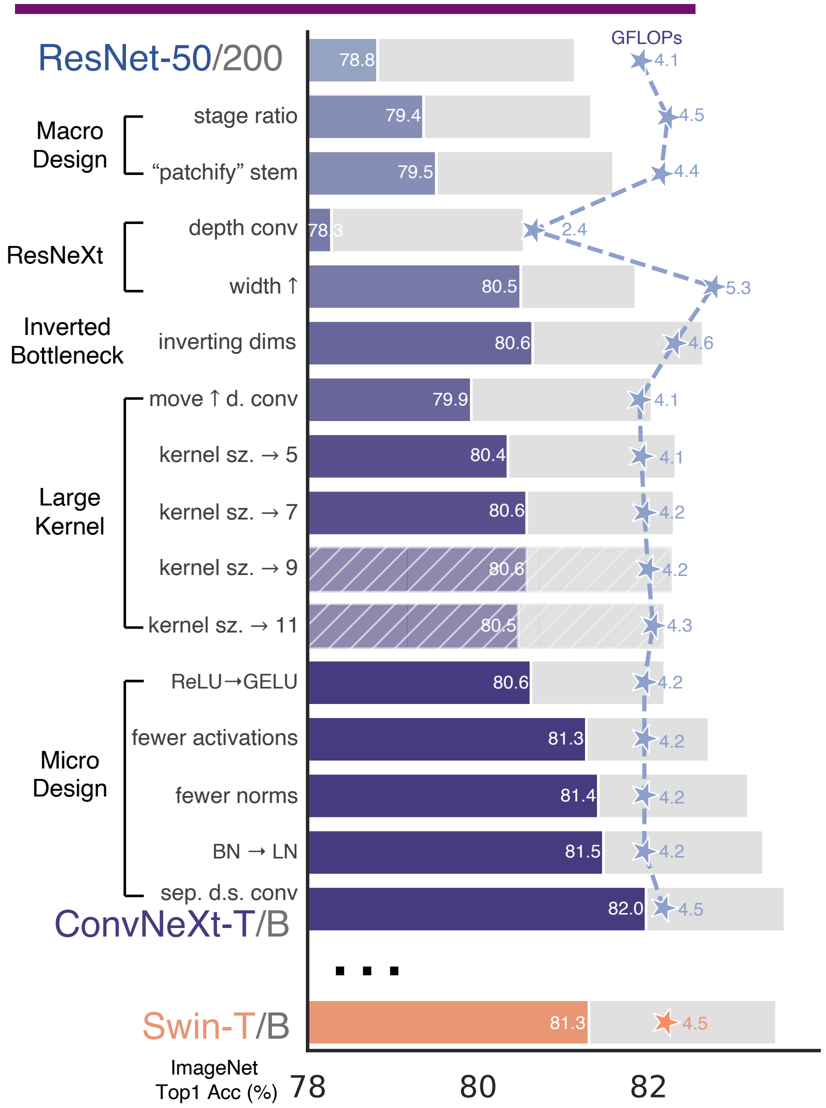
- Macro Design: Patch Embedding + Token Mixer + Channel Mixer + Pre-Norm & Short-cut).

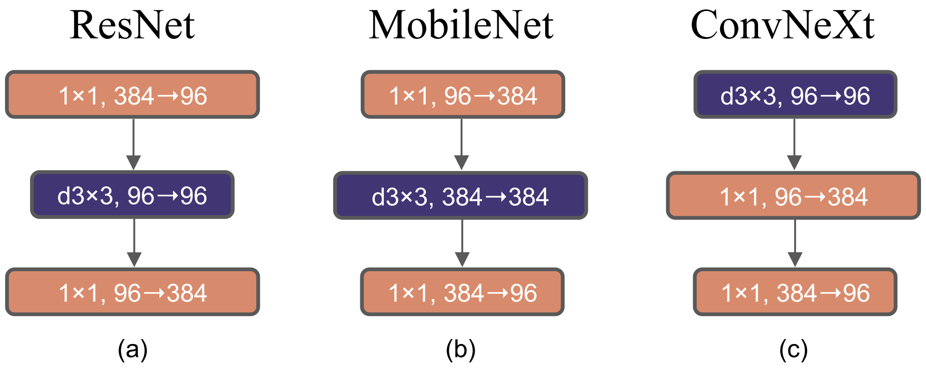


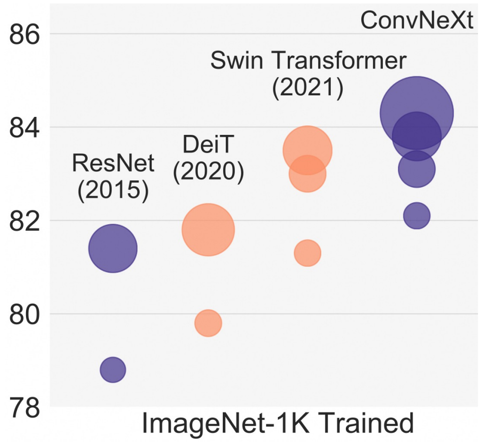


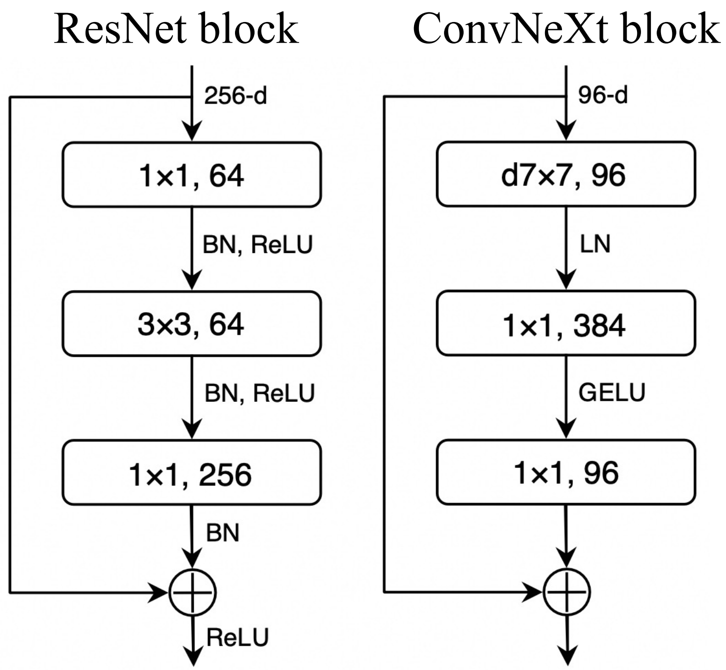
[1] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR, 2021
[2] PoolFormer: MetaFormer Is Actually What You Need for Vision. CVPR, 2022.
[3] A ConvNet for the 2020s. CVPR, 2022.

Modern CNNs: ConvNeXt







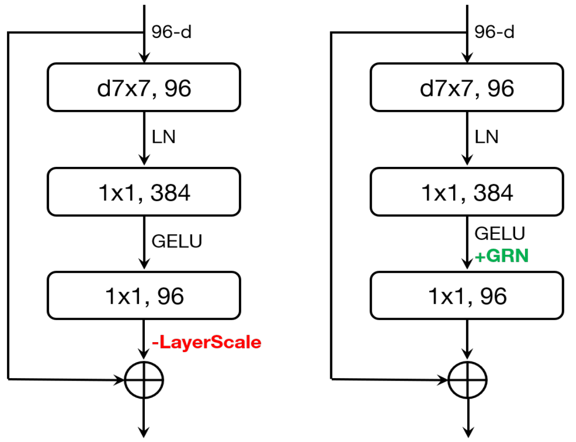


model	image size	#param.	FLOPs	throughput (image / s)	IN-1K top-1 acc.
ImageNet-1K trained models					
• RegNetY-16G [54]	224^2	84M	16.0G	334.7	82.9
• EffNet-B7 [71]	600^2	66M	37.0G	55.1	84.3
• EffNetV2-L [72]	480^2	120M	53.0G	83.7	85.7
○ DeiT-S [73]	224^2	22M	4.6G	978.5	79.8
○ DeiT-B [73]	224^2	87M	17.6G	302.1	81.8
○ Swin-T	224^2	28M	4.5G	757.9	81.3
• ConvNeXt-T	224^2	29M	4.5G	774.7	**82.1**
○ Swin-S	224^2	50M	8.7G	436.7	83.0
• ConvNeXt-S	224^2	50M	8.7G	447.1	**83.1**
○ Swin-B	224^2	88M	15.4G	286.6	83.5
• ConvNeXt-B	224^2	89M	15.4G	292.1	**83.8**
○ Swin-B	384^2	88M	47.1G	85.1	84.5
• ConvNeXt-B	384^2	89M	45.0G	95.7	**85.1**
• ConvNeXt-L	224^2	198M	34.4G	146.8	**84.3**
• ConvNeXt-L	384^2	198M	101.0G	50.4	**85.5**

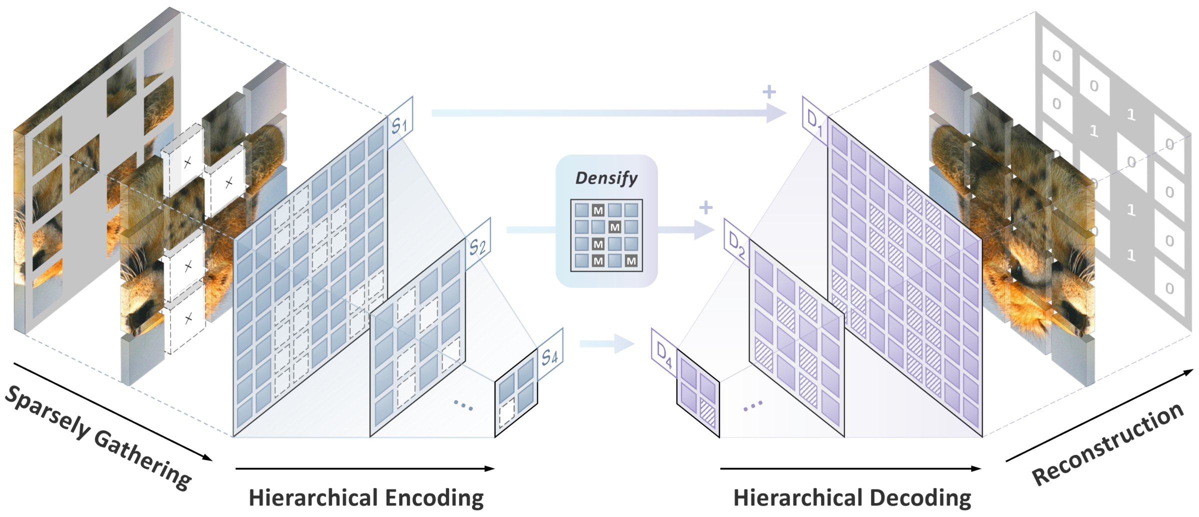
[1] A ConvNet for the 2020s. CVPR, 2022.

Modern CNNs: ConvNeXt.V2

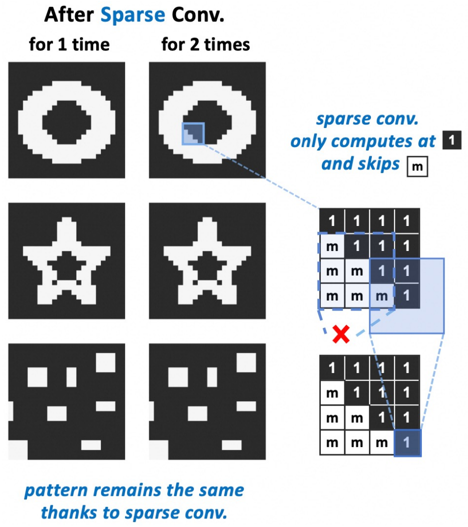
- CNNs benefit from Masked Image Modeling (MIM) Pre-training.



ConvNeXt.V1 ConvNeXt.V2



MIM pre-training with SparK (or FCMAE in ConvNeXt.V2)



Sparse Conv for Masking

Global Response Normalization (GRN)

```
# gamma, beta: learnable affine transform parameters
# X: input of shape (N,H,W,C)

gx = torch.norm(X, p=2, dim=(1,2), keepdim=True)
nx = gx / (gx.mean(dim=-1, keepdim=True)+1e-6)
return gamma * (X * nx) + beta + X
```

$$\mathcal{G}(X) := X \in \mathcal{R}^{H\times W\times C} \rightarrow gx \in \mathcal{R}^{C}$$

$$\mathcal{N}(||X_i||) := ||X_i|| \in \mathcal{R} \rightarrow \frac{||X_i||}{\sum_{j=1,\ldots,C} ||X_j||} \in \mathcal{R}$$

Backbone	Method	#param	FLOPs	Val acc.
ConvNeXt V1-B	Supervised	89M	15.4G	83.8
ConvNeXt V1-B	FCMAE	89M	15.4G	83.7
ConvNeXt V2-B	Supervised	89M	15.4G	84.3 (+0.5)
ConvNeXt V2-B	FCMAE	89M	15.4G	**84.6** (+0.8)
ConvNeXt V1-L	Supervised	198M	34.4G	84.3
ConvNeXt V1-L	FCMAE	198M	34.4G	84.4
ConvNeXt V2-L	Supervised	198M	34.4G	84.5 (+0.2)
ConvNeXt V2-L	FCMAE	198M	34.4G	**85.6** (+1.3)

Methods Target	#Para. (M)	Sup. Label	MoCoV3‡ CL	SimMIM‡ RGB	SparK RGB	A^2MIM RGB
ResNet-50	25.6	79.8	80.1	79.9	80.6	**80.4**
ResNet-101	44.5	81.3	81.6	81.3	82.2	**81.9**
ResNet-152	60.2	81.8	82.0	81.9	82.7	**82.5**
ResNet-200	64.7	82.1	82.5	82.2	83.1	**83.0**
ConvNeXt-T	28.6	82.1	82.3	82.1	82.7	**82.5**
ConvNeXt-S	50.2	83.1	83.3	83.2	84.1	**83.7**
ConvNeXt-B	88.6	83.5	83.7	83.6	84.8	**84.1**

[1] ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. CVPR, 2023. [2] Designing BERT for Convolutional Networks: Sparse and Hierarchical Masked Modeling. ICLR, 2023. [3] Architecture-Agnostic Masked Image Modeling - From ViT back to CNN. ICML, 2023.

Content

1. **Modern CNNs: Macro Design and Pre-training**

   MetaFormer, ConvNeXt, ConvNeXt.V2 (SparK, A2MIM)

2. **Design of Convolution Kernels**

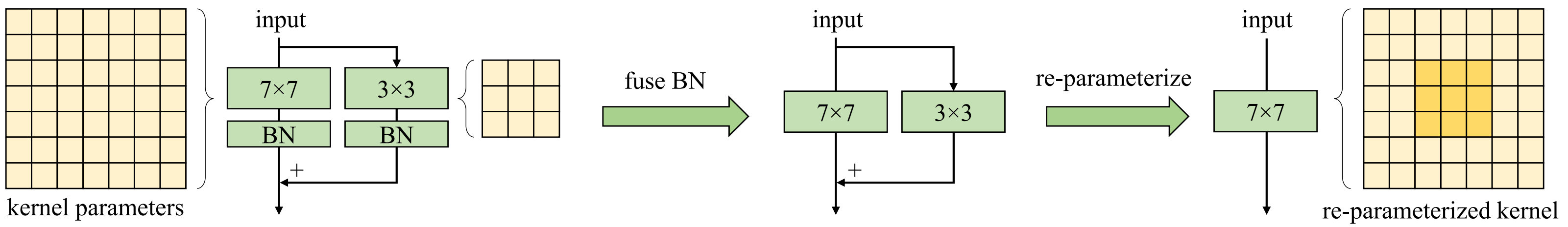
   RepLKNet, SLaK, InceptionNext, DCN.V3/V4, UniRepLKNet

3. **Combining Large Kernel with Gated Attention**

   VAN, HorNet, FocalNet, MogaNet, Mamba, VMamba

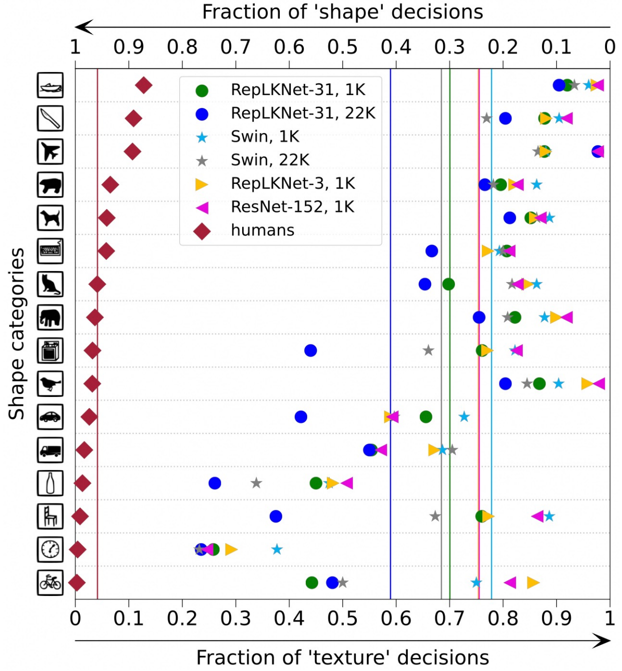
Large Kernels: RepLKNet

- Large-Kernel (LK) Convolutions are efficient and competitive as Self-attention.
- Training extremely large convolutions with Structural Re-parameterization.



DW7×7 = DW3×3 (BN) +DW7×7 (BN)+Short-cut.

Large kernels are **shape biased** as ViTs.

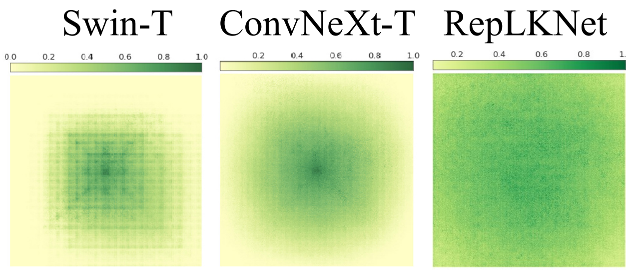


Resolution R	Impl	Latency (ms) @ Kernel size 3	5	7	9	13	17	21	27	29	31
16 × 16	Pytorch	5.6	11.0	14.4	17.6	36.0	57.2	83.4	133.5	150.7	171.4
	Ours	5.6	6.5	6.4	6.9	7.5	8.4	8.4	8.4	8.3	8.4
32 × 32	Pytorch	21.9	34.1	54.8	76.1	141.2	230.5	342.3	557.8	638.6	734.8
	Ours	21.9	28.7	34.6	40.6	52.5	64.5	73.9	87.9	92.7	96.7
64 × 64	Pytorch	69.6	141.2	228.6	319.8	600.0	977.7	1454.4	2371.1	2698.4	3090.4
	Ours	69.6	112.6	130.7	152.6	199.7	251.5	301.0	378.2	406.0	431.7

Large kernels are **memory bound** instead of compute bound.

Kernel size	Architecture	ImageNet Top-1	Params	FLOPs	ADE20K mIoU	Params	FLOPs
7-7-7-7	ConvNeXt-Tiny	81.0	29M	4.5G	44.6	60M	939G
7-7-7-7	ConvNeXt-Small	82.1	50M	8.7G	45.9	82M	1027G
7-7-7-7	ConvNeXt-Base	82.8	89M	15.4G	47.2	122M	1170G
31-29-27-13	ConvNeXt-Tiny	81.6	32M	6.1G	**46.2**	64M	973G
31-29-27-13	ConvNeXt-Small	82.5	58M	11.3G	**48.2**	90M	1081G

Extremely large kernels benefit both classification and downstream tasks and outperforms ViTs.

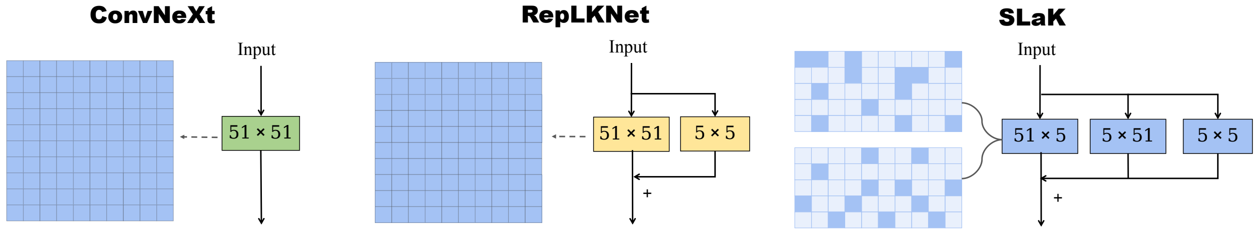


Effective receptive field

[1] Scaling Up Your Kernels to 31x31: Revisiting Large Kernel Design in CNNs. CVPR, 2022.

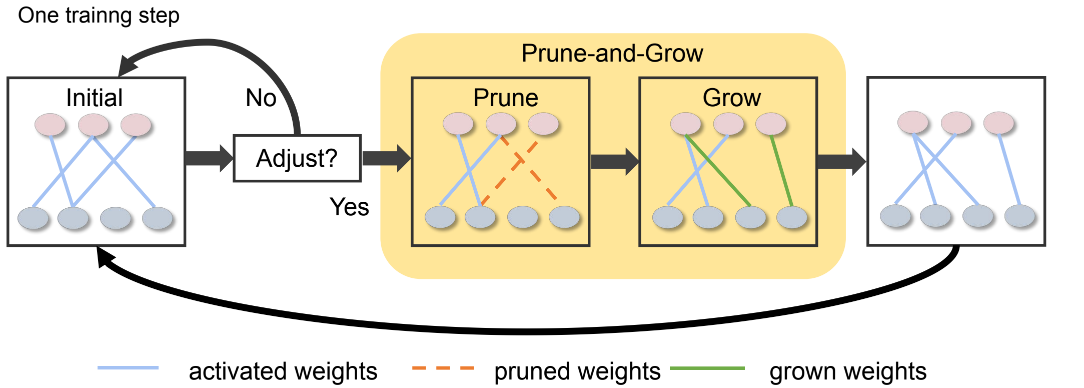
Large Kernels: SLaK

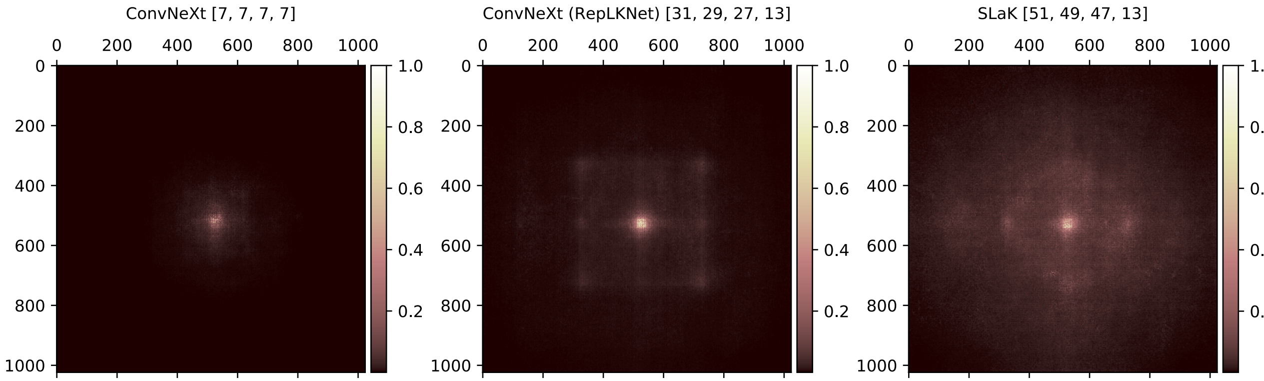
- Step 1: Decomposing a large kernel (61x61) into two rectangular, parallel kernels.
- Step 2: Using sparse groups training (speedup), expanding more width.



Kernel Size	Top-1 Acc	#Params	FLOPs	Top-1 Acc	#Params	FLOPs	Top-1 Acc	#Params	FLOPs
	Decomposed			Sparse groups			Sparse groups, expand more width		
7-7-7-7	81.0	29M	4.5G	80.0	17M	2.6G	81.1	29M	4.5G
31-29-37-13	81.3	30M	5.0G	80.4	18M	2.9G	81.5	30M	4.8G
51-49-47-13	81.5	31M	5.4G	80.5	18M	3.1G	81.6	30M	5.0G
61-59-57-13	81.4	31M	5.6G	80.4	19M	3.2G	81.5	31M	5.2G

Model	Kernel Size	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
pre-trained for 120 epochs, finetuned for 1× (12 epochs)							
ConvNeXt-T (Liu et al., 2022b)	7-7-7-7	47.3	65.9	51.5	41.1	63.2	44.4
ConvNeXt-T (RepLKNET)* (Ding et al., 2022)	31-29-27-13	47.8	66.7	52.0	41.4	63.9	44.7
SLaK-T	51-49-47-13	**48.4**	**67.2**	**52.5**	**41.8**	**64.4**	**45.2**
pre-trained for 300 epochs, finetuned for 3× (36 epochs)							
ConvNeXt-T (Liu et al., 2022b)	7-7-7-7	50.4	69.1	54.8	43.7	66.5	47.3
SLaK-T	51-49-47-13	**51.3**	**70.0**	**55.7**	**44.3**	**67.2**	**48.1**





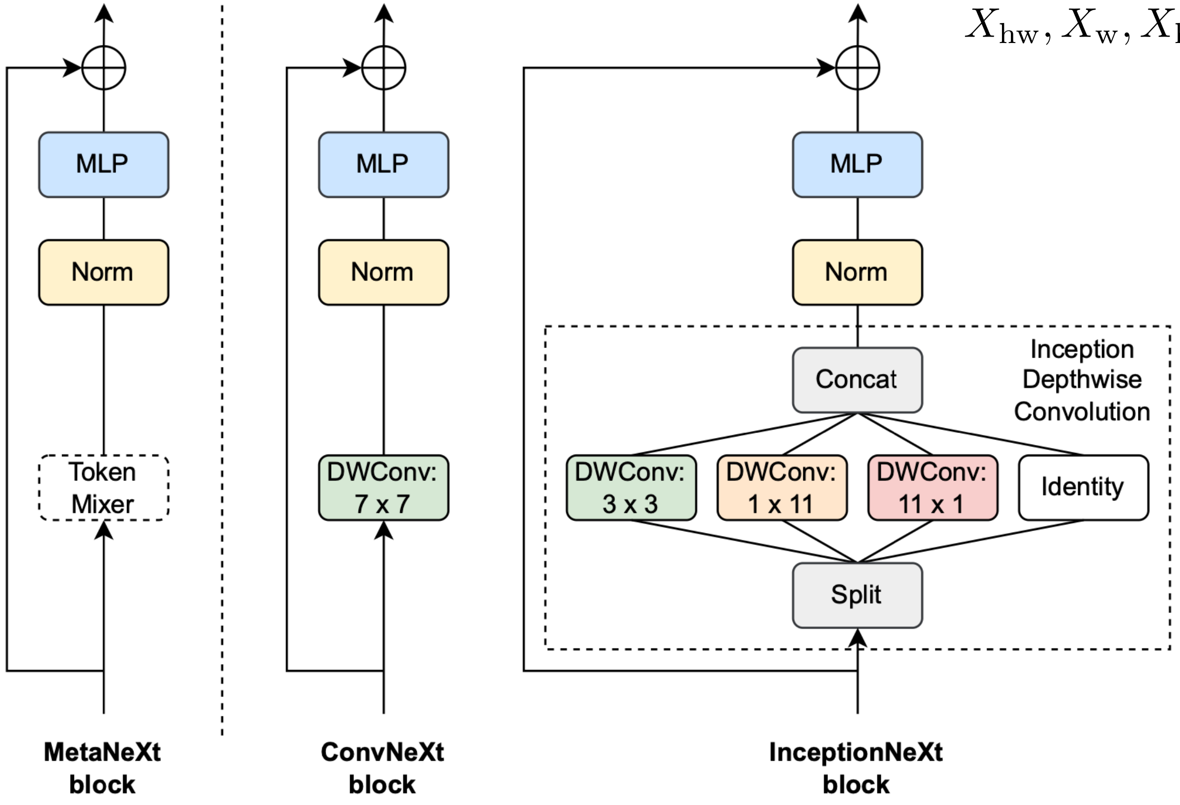
Effective receptive field (ERF)

(1) Initialization: Constructing Sparce Convolution based on SNIP[2]
(2) Dynamic sparsity: Pruning (the lowest magnitude) and growing

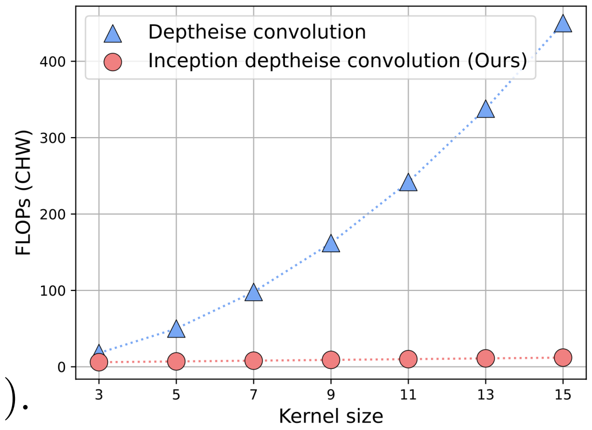
[1] More ConvNets in the 2020s: Scaling up Kernels Beyond 51x51 using Sparsity. ICLR, 2023.
[2] SNIP: Single-shot Network Pruning based on Connection Sensitivity. ICLR, 2019.

Large Kernels: InceptionNeXt

- **MetaNeXt: Fusing Token Mixer with Channel Mixer + PreNorm + ShortCut.**
- **Inception Kernels: Better performance and throughputs than Depth-wise Conv 7x7.**



$$
\begin{aligned}
X_{\mathrm{hw}}, X_{\mathrm{w}}, X_{\mathrm{h}}, X_{\mathrm{id}} &= \mathrm{Split}(X) \\
&= X_{:,:g}, X_{:g:2g}, X_{:2g:3g}, X_{:3g:} \\
X'_{\mathrm{hw}} &= \mathrm{DWConv}^{g \to g}_{k_s \times k_s} g(X_{\mathrm{hw}}), \\
X'_{\mathrm{w}} &= \mathrm{DWConv}^{g \to g}_{1 \times k_b} g(X_{\mathrm{w}}), \\
X'_{\mathrm{h}} &= \mathrm{DWConv}^{g \to g}_{k_b \times 1} g(X_{\mathrm{h}}), \\
X'_{\mathrm{id}} &= X_{\mathrm{id}}. \\
X' &= \mathrm{Concat}(X'_{\mathrm{hw}}, X'_{\mathrm{w}}, X'_{\mathrm{h}}, X'_{\mathrm{id}}).
\end{aligned}
$$

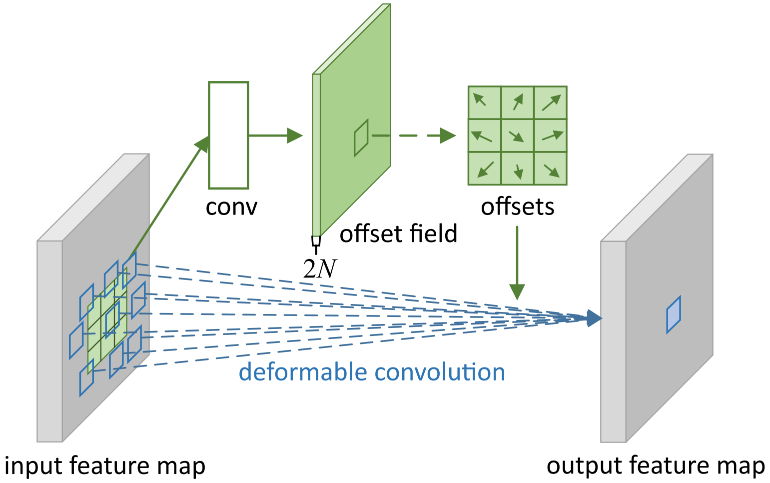


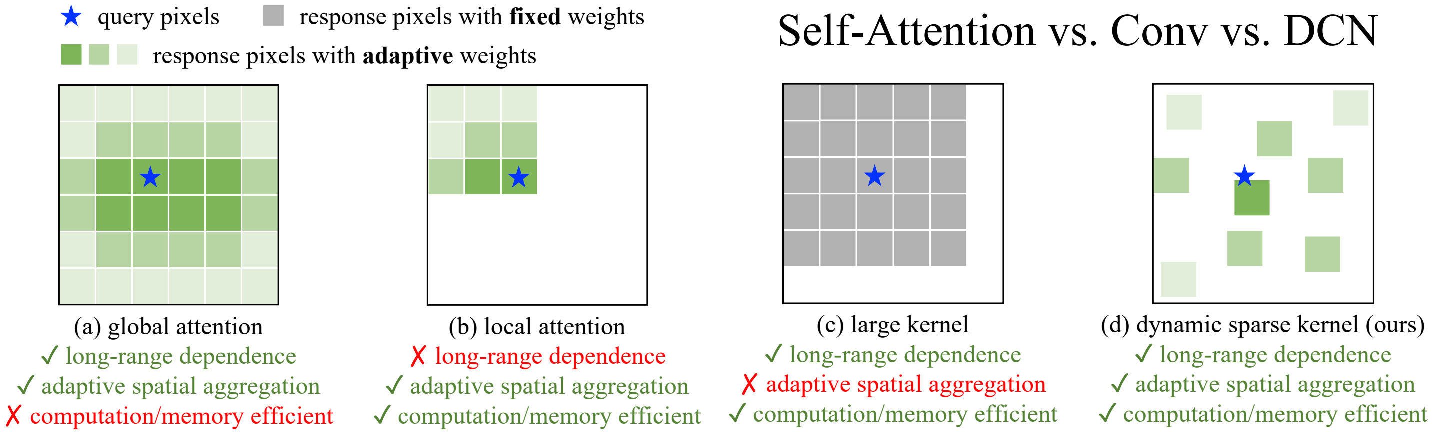
Model	Mixing Type	Image (size)	Params (M)	MACs (G)	Throughput (img/second) Train	Inference	Top-1 (%)
DeiT-S [61]	Attn	224^2	22	4.6	1227	3781	79.8
T2T-ViT-14 [76]	Attn	224^2	22	4.8	–	–	81.5
TNT-S [18]	Attn	224^2	24	5.2	–	–	81.5
Swin-T [37]	Attn	224^2	29	4.5	564	1768	81.3
Focal-T [73]	Attn	224^2	29	4.9	–	–	82.2
ResNet-50 [20, 69]	Conv	224^2	26	4.1	969	3149	78.4
RSB-ResNet-50 [20, 69]	Conv	224^2	26	4.1	969	3149	79.8
RegNetY-4G [46, 69]	Conv	224^2	21	4.0	670	2694	81.3
FocalNet-T [72]	Conv	224^2	29	4.5	–	–	82.3
ConvNeXt-T [38]	Conv	224^2	29	4.5	575	2413 (1943)	82.1
InceptionNeXt-T (Ours)	Conv	224^2	28	4.2	901 (+57%)	2900 (+20%)	82.3 (+0.2)

[1] InceptionNeXt: When Inception Meets ConvNeXt. CVPR, 2024.

Kernel Designs: DCN.V3 (InternImage)

- DCN.V3: Learnable offsets (V1) + Softmax-normalized modulation (V2) + Grouping.





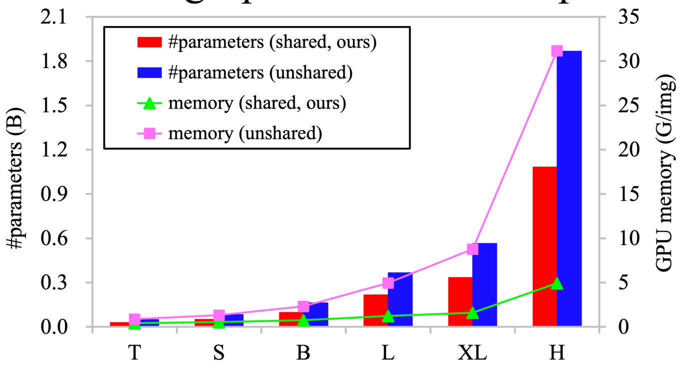
DCN.V1: $$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta \mathbf{p}_n)$$

DCN.V2: $$\mathbf{y}(p_0) = \sum_{k=1}^{K} \mathbf{w}_k \mathbf{m}_k \mathbf{x}(p_0 + p_k + \Delta p_k)$$

DCN.V3: $$\mathbf{y}(p_0) = \sum_{g=1}^{G} \sum_{k=1}^{K} \mathbf{w}_g \mathbf{m}_{gk} \mathbf{x}_g(p_0 + p_k + \Delta p_{gk})$$

Offsets Δp_n, Regular grids p_n, Modulation m_k, weights w

Scaling-up with efficient impl.

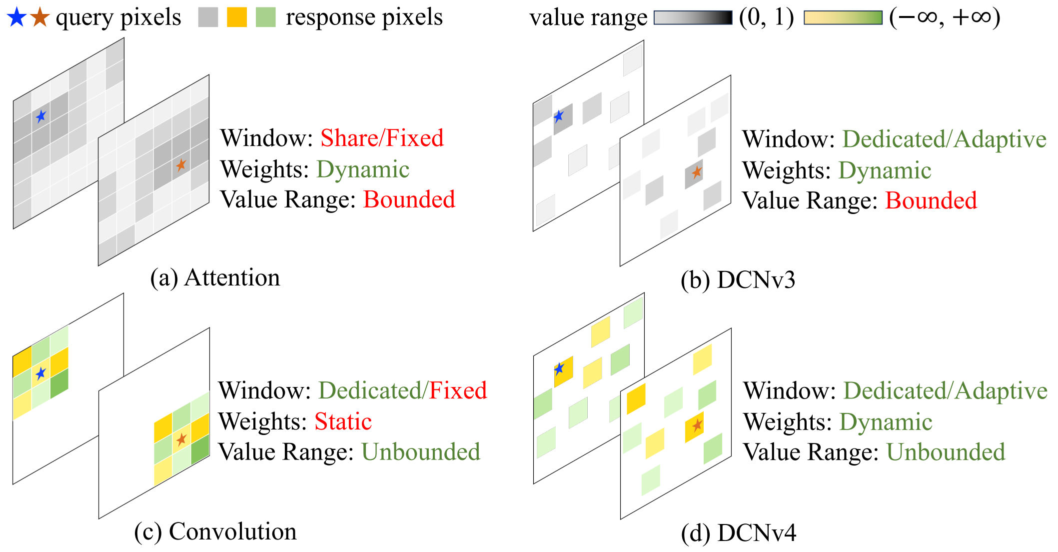


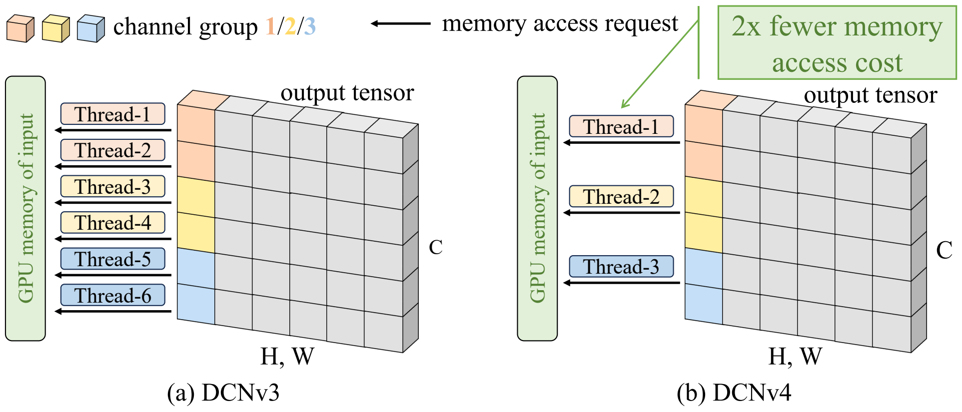
method	type	scale	#params	#FLOPs	acc (%)
SwinV2-L/24‡ [16]	T	384^2	197M	115G	87.6
RepLKNet-31L‡ [22]	C	384^2	172M	96G	86.6
HorNet-L‡ [43]	C	384^2	202M	102G	87.7
ConvNeXt-L‡ [21]	C	384^2	198M	101G	87.5
ConvNeXt-XL‡ [21]	C	384^2	350M	179G	87.8
InternImage-L‡ (ours)	C	384^2	223M	108G	87.7
InternImage-XL‡ (ours)	C	384^2	335M	163G	88.0
ViT-G/14# [30]	T	518^2	1.84B	5160G	90.5
CoAtNet-6# [20]	T	512^2	1.47B	1521G	90.5
CoAtNet-7# [20]	T	512^2	2.44B	2586G	90.9
Florence-CoSwin-H# [59]	T	–	893M	–	90.0
SwinV2-G# [16]	T	640^2	3.00B	–	90.2
RepLKNet-XL# [22]	C	384^2	335M	129G	87.8
BiT-L-ResNet152x4# [67]	C	480^2	928M	–	87.5
InternImage-H# (ours)	C	224^2	1.08B	188G	88.9
InternImage-H# (ours)	C	640^2	1.08B	1478G	89.6

[1] Deformable Convolutional Networks. ICCV, 2017. [2] Deformable ConvNets v2: More Deformable, Better Results. CVPR, 2018.
[3] InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. CVPR, 2023.

Kernel Designs: DCN.V4 (FlashInternImage)

- **DCN.V4: No Softmax normalization + Speed-up (reducing HRM as Flash-Attention).**





Model	5th EP	10th Ep	20th Ep	50th Ep	300th Ep
ConvNeXt	29.9	53.5	66.1	74.8	83.8
ConvNeXt + softmax	8.5 (-21.4)	25.3 (-28.2)	51.1 (-15.0)	69.1 (-5.7)	81.6 (-2.2)

Using Softmax in DWConv7×7 degenerating performance

Operator	Runtime (ms) 56 × 56 × 128	28 × 28 × 256	14 × 14 × 512	7 × 7 × 1024	14 × 14 × 768
Attention (torch)	30.8 / 19.3	3.35 / 2.12	0.539 / 0.448	0.446 / 0.121	0.779 / 0.654
FlashAttention-2	N/A / 2.46	N/A / 0.451	N/A / **0.123**	N/A / 0.0901	N/A / 0.163
Window Attn (7 × 7)	4.05 / 1.46	2.07 / 0.770	1.08 / 0.422	0.577 / 0.239	1.58 / 0.604
DWConv (7 × 7, torch)	2.02 / 1.98	1.03 / 1.00	0.515 / 0.523	0.269 / 0.261	0.779 / 0.773
DWConv (7 × 7, cuDNN)	0.981 / 0.438	0.522 / 0.267	0.287 / 0.153	0.199 / 0.102	0.413 / 0.210
DCNv3	1.45 / 1.52	0.688 / 0.711	0.294 / 0.298	0.125 / 0.126	0.528 / 0.548
DCNv4	**0.606 / 0.404**	**0.303 / 0.230**	**0.145 / 0.123**	**0.0730 / 0.0680**	**0.224 / 0.147**

ImageNet-1K Classification

Model	Size	Scale	Acc	Throughput
Swin-T	29M	224^2	81.3	1989 / 3619
ConvNeXt-T	29M	224^2	82.1	2485 / 4305
InternImage-T	30M	224^2	83.5	1409 / 1746
FlashInternImage-T	30M	224^2	**83.6**	2316 / 3154 (+64%/ + 80%)
Swin-S	50M	224^2	83.0	1167/2000
ConvNeXt-S	50M	224^2	83.1	1645/2538
InternImage-S	50M	224^2	84.2	1044/1321
FlashInternImage-S	50M	224^2	**84.4**	1625 / 2396

COCO2017 Det. and Seg.

Model	#param	FPS	Cascade Mask R-CNN 1× AP^b	1× AP^m	3×+MS AP^b	3×+MS AP^m
Swin-L	253M	20 / 26	51.8	44.9	53.9	46.7
ConvNeXt-L	255M	26 / 40	53.5	46.4	54.8	47.6
InternImage-L	277M	20 / 26	54.9	47.7	56.1	48.5
ConvNeXt-XL	407M	21 / 32	53.6	46.5	55.2	47.7
InternImage-XL	387M	16 / 23	55.3	48.1	56.2	48.8
FlashInternImage-L	277M	26 / 39	**55.6**	**48.2**	**56.7**	**48.9**

[1] DCNv4: Efficient Deformable ConvNets: Rethinking Dynamic and Sparse Operator for Vision Applications. CVPR, 2024.

Content

1. **Modern CNNs: Macro Design and Pre-training**

   MetaFormer, ConvNeXt, ConvNeXt.V2 (SparK, A2MIM)

2. **Design of Convolution Kernels**

   RepLKNet, SLaK, InceptionNext, DCN.V3/V4, UniRepLKNet

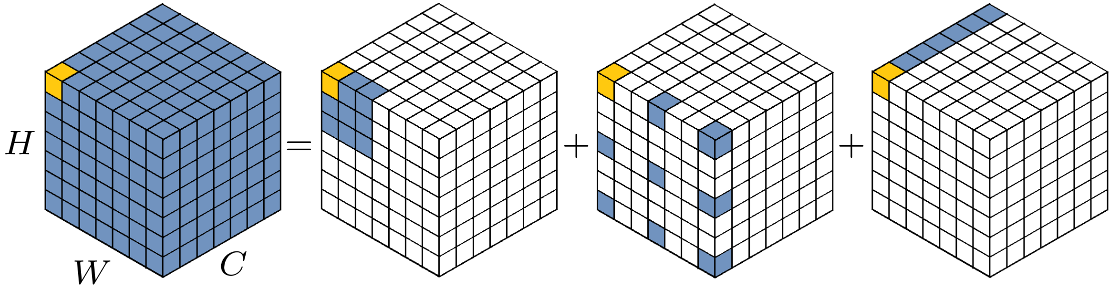
3. **Combining Large Kernel with Gated Attention**

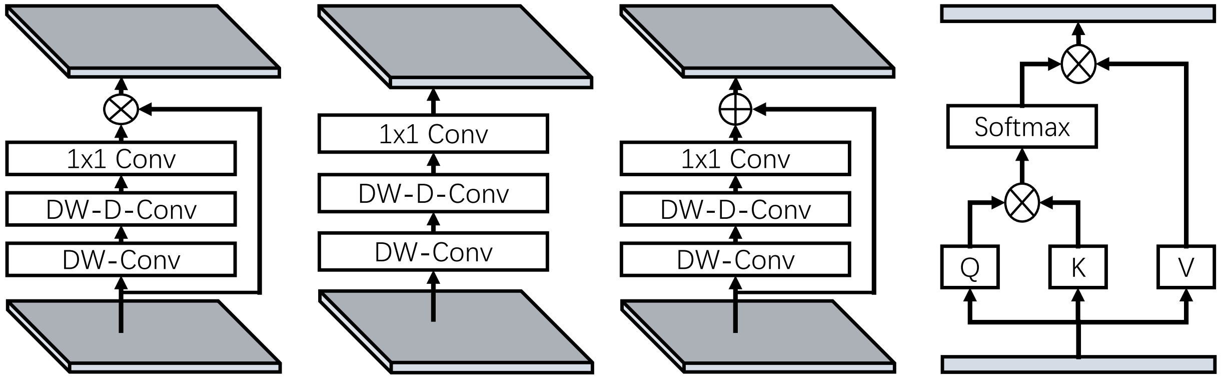
   VAN, HorNet, FocalNet, MogaNet, Mamba, VMamba

Gating & Large-kernel: VAN

- **Decomposed large kernel + Gating.**

Conv9×9 = DWConv3×3 + DWConv3×3 + PWConv1×1
(Dilation=3)





VAN (LKA) Non-attention Non-attention (add) Self-attention

[1] Visual Attention Network. CVMJ, 2023.

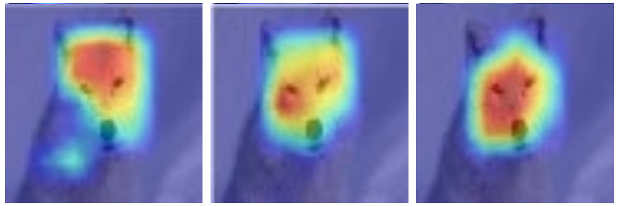
Properties	Convolution	Self-Attention	LKA
Local Receptive Field	✓	✗	✓
Long-range Dependence	✗	✓	✓
Spatial Adaptability	✗	✓	✓
Channel Adaptability	✗	✗	✓
Computational complexity	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$

Properties of DWConv vs. MHSA vs. Large-kernel Attention

Method	K	Dilation	Params. (M)	GFLOPs	Acc(%)
VAN-B0	7	2	4.03	0.85	74.8
VAN-B0	14	3	4.07	0.87	75.3
VAN-B0	21	3	4.11	0.88	75.4
VAN-B0	28	4	4.14	0.90	75.4

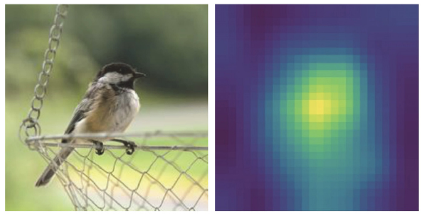
Kernel size vs. Dilation vs. ImageNet Acc (%)

Conv21×21 = DWConv5×5 +DWConv7×7 +PWConv1×1
(Dilation=3)



Swin-T ConvNeXt-T VAN-B2

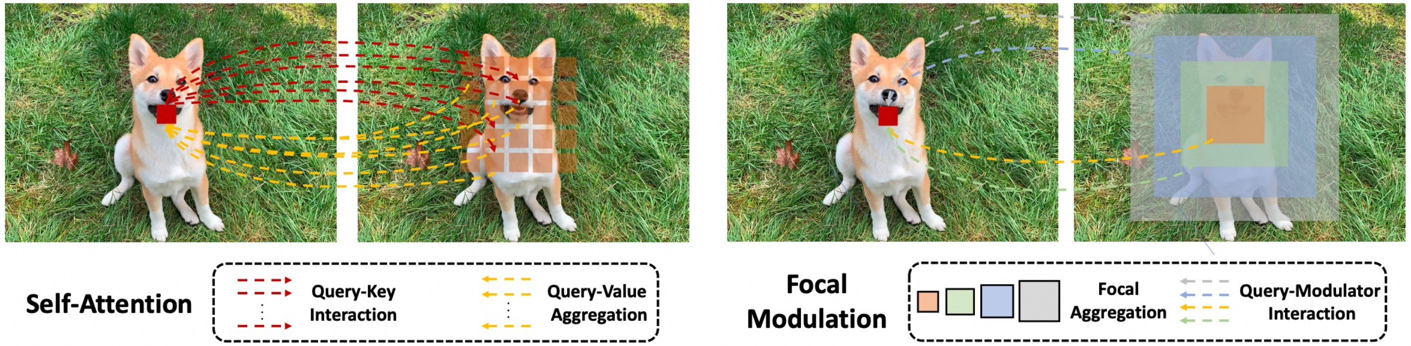
Grad-CAM visualization



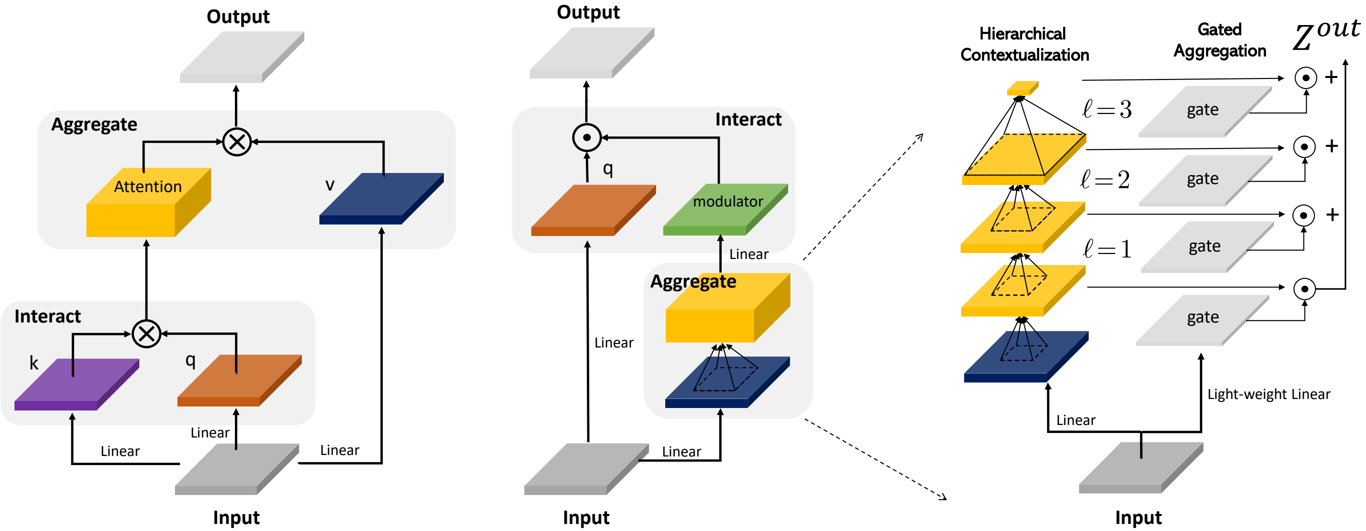
Attention map visualization

Gating & Hierarchical Kernel: FocalNet

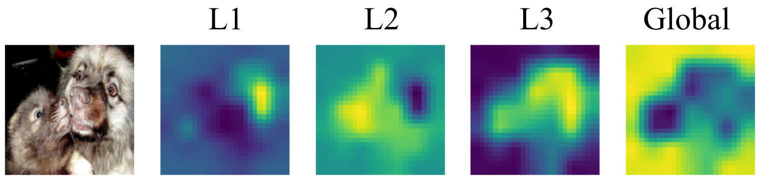
- Hierarchical Contextualization + Gated Aggregation.




```
def forward(x, m=0):
    x = pj_in(x).permute(0, 3, 1, 2)
    q, z, gate = split(x, (C, C, L+1), 1)
    for ℓ in range(L):
        z = hc_layers[ℓ](z)              # Eq.(4), hierarchical contextualization
        m = m + z * gate[:, ℓ:ℓ+1]      # Eq.(5), gated aggregation
    m = m + GeLU(z.mean(dim=(2,3))) * gate[:,L:]
    x = q * pj_cxt(m)                    # Eq.(6), Focal Modulation
    return pj_out( x.permute(0, 2, 3, 1) )
```



(a) Self-Attention (b) Focal-Modulation (c) Context Aggregation



$$\mathbf{Z}^{\ell} = f_a^{\ell}(\mathbf{Z}^{\ell-1}) \triangleq \text{GeLU}(\text{DWConv}(\mathbf{Z}^{\ell-1})) \in \mathbb{R}^{H\times W\times C} \quad \text{Eq. (4)}$$

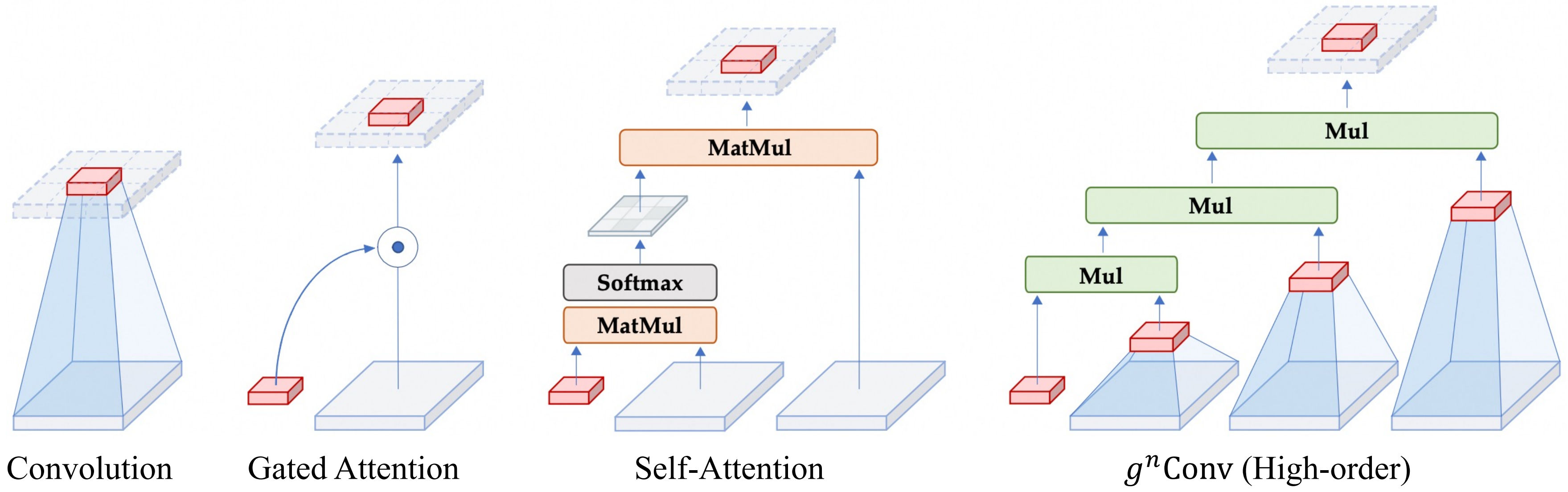
$$\mathbf{Z}^{out} = \sum_{\ell=1}^{L+1} \boxed{\mathbf{G}^{\ell}} \odot \mathbf{Z}^{\ell} \in \mathbb{R}^{H\times W\times C} \quad \text{Eq. (5)}$$

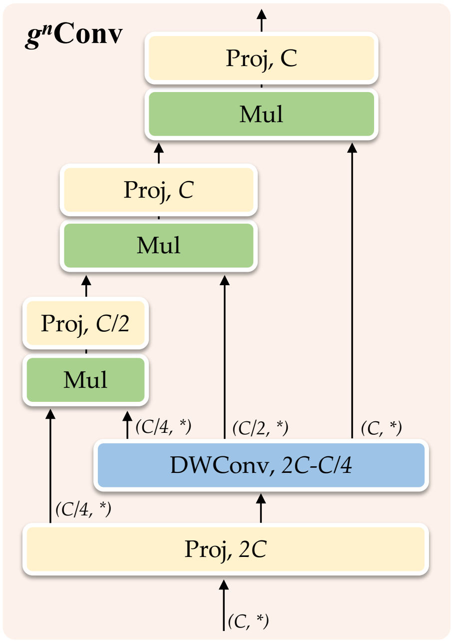
$$\boldsymbol{y}_i = q(\boldsymbol{x}_i) \odot \boxed{h\Big(\sum_{\ell=1}^{L+1} \boldsymbol{g}_i^{\ell} \cdot \boldsymbol{z}_i^{\ell}\Big)} \quad \text{Eq. (6)}$$

[1] Focal Modulation Networks. NeurIPS, 2022.

Gating & Hierarchical Kernel: HorNet

- High-order Interactions: Recursive DWConv + Gating.

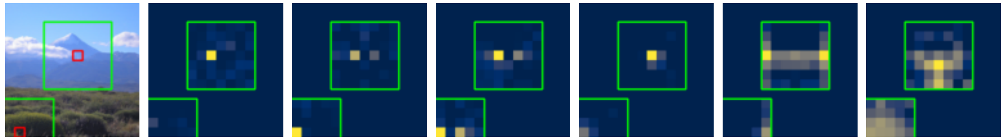




$$x_{g^n\text{Conv}}^{(i,c)} = p_n^{(i,c)} = \sum_{j\in\Omega_i}\sum_{c'=1}^{C} \underline{w_{n-1,i\to j}^{c}\mathbf{g}_{n-1}^{(i,c)}}\, w_{\phi_{\text{in}}}^{(c',c)} x^{(j,c')} \triangleq \sum_{j\in\Omega_i}\sum_{c'=1}^{C} \underline{h_{ij}^{c}}\, w_{\phi_{\text{in}}}^{(c',c)} x^{(j,c')} \quad \text{Eq. (3.8)}$$

```python
def forward(self, x):
    x = self.proj_in(x)
    y, x = torch.split(x, (self.dims[0], sum(self.dims)), dim=1)
    x = self.dwconv(x)
    x_list = torch.split(x, self.dims, dim=1)
    x = y * x_list[0]
    for i in range(self.order -1):
        x = self.projs[i](x) * x_list[i+1]
    return self.proj_out(x)
```

```python
self.projs = nn.ModuleList(
    [nn.Conv2d(self.dims[i], self.dims[i+1], 1)
    for i in range(order-1)])
self.proj_out = nn.Conv2d(dim, dim, 1)
```



Adaptive weights generated by g^nConv, i.e., $\frac{1}{C}\Sigma_{c=1}^{C} h_{ij}^{c}$ in Eq. (3.8)

[1] HorNet: Efficient High-Order Spatial Interactions with Recursive Gated Convolutions. NeurIPS, 2022.

Multi-order Interaction: MogaNet

- Representation Bottleneck[1]: Loss in the middle-order interactions.

Multi-order Interactions $I^{(m)}(i,j) = \mathbb{E}_{S \subseteq N \setminus \{i,j\}, |S|=m}[\Delta f(i,j,S)]$

$N = \{1, \dots, n\} \quad 0 \leq m \geq n-2$

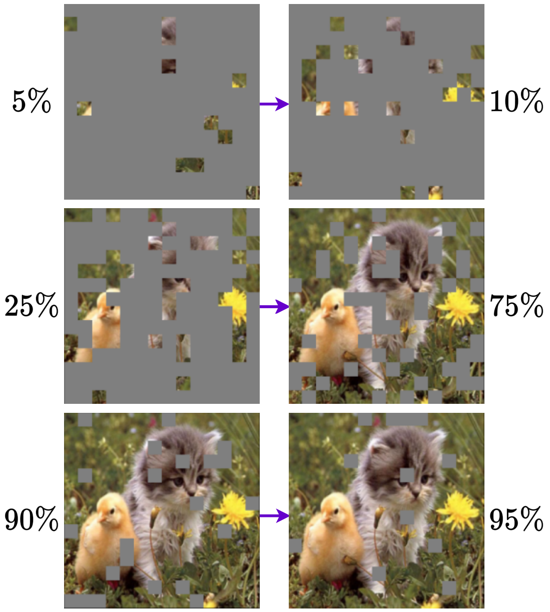
$\Delta f(i,j,S) = f(S \cup \{i,j\}) - f(S \cup \{i\}) - f(S \cup \{j\}) + f(S)$

Interaction Strengths $J^{(m)} = \dfrac{\mathbb{E}_{x \in \Omega} \mathbb{E}_{i,j} |I^{(m)}(i,j|x)|}{\mathbb{E}_{m'} \mathbb{E}_{x \in \Omega} \mathbb{E}_{i,j} |I^{(m')}(i,j|x)|}$

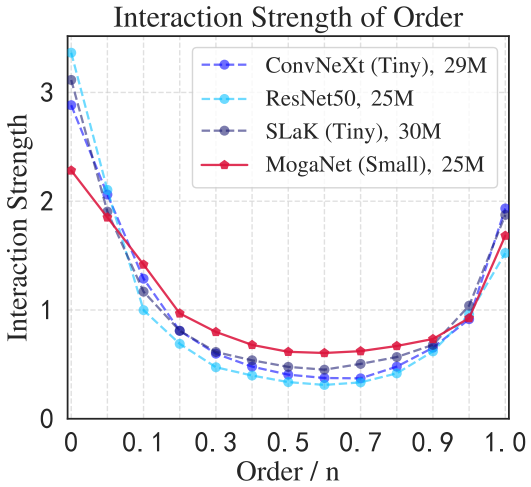
Much new information
Little new infomation

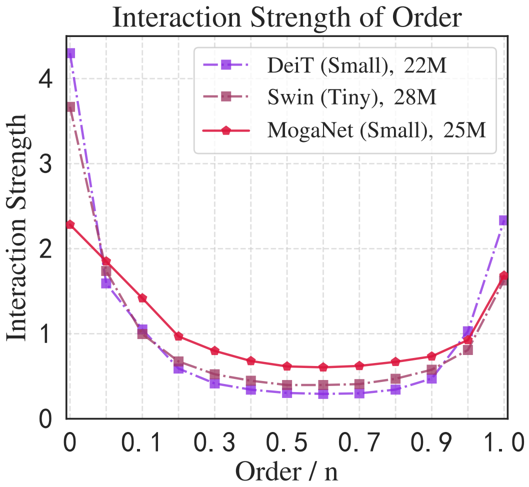
Little new information
Much new infomation

Much new information
Little new infomation



Both ViTs and modern CNN architectures fail to explore middle-order interactions, which are informative to humans.

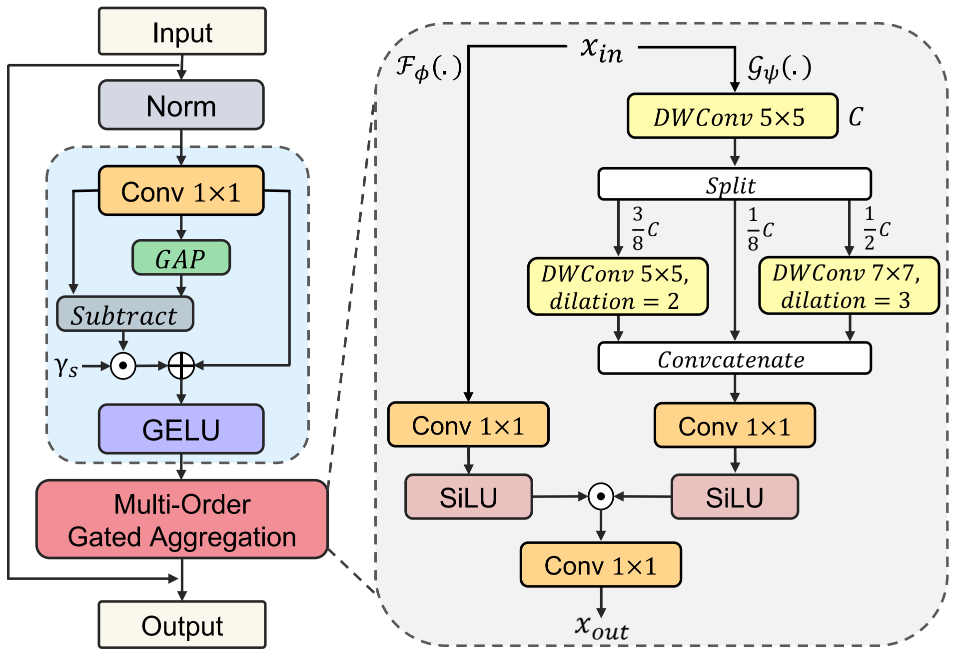




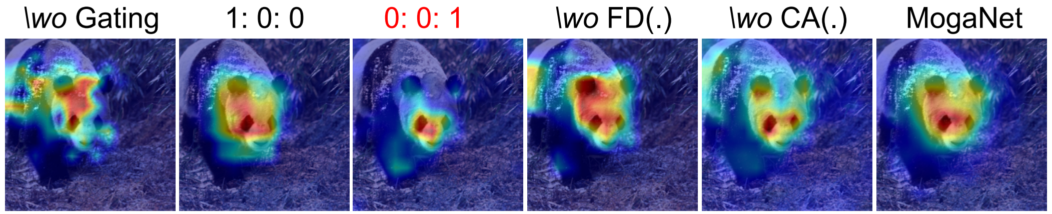
[1] Discovering and Explaining the Representation Bottleneck of DNNs. ICLR, 2022. [2] MogaNet: Multi-order Gated Aggregation Network. ICLR, 2024.

Multi-order Interaction: MogaNet

- Spatial Aggregation (SA): Multi-order context extraction + Gated aggregation.



$$Z = X + \mathrm{Moga}\Big(\mathrm{FD}\big(\mathrm{Norm}(X)\big)\Big)$$



Feature decomposition:

$$Y = \mathrm{Conv}_{1\times1}(X),$$
$$Z = \mathrm{GELU}\Big(Y + \gamma_s \odot \big(Y - \mathrm{GAP}(Y)\big)\Big)$$

Gated aggregation branch:

$$Z = \underbrace{\mathrm{SiLU}\big(\mathrm{Conv}_{1\times1}(X)\big)}_{\mathcal{F}_\phi} \odot \underbrace{\mathrm{SiLU}\big(\mathrm{Conv}_{1\times1}(Y_C)\big)}_{\mathcal{G}_\psi}$$

Multi-order DWConvs: DW5×5, DW5×5 (d=2), DW7×7 (d=3)

$$C_l + C_m + C_h = C,\ Y_C = \mathrm{Concat}(Y_{l,1:C_l}, Y_m, Y_h)$$

Modules	Top-1 Acc (%)	Params. (M)	FLOPs (G)
Baseline (+Gating branch)	77.2	5.09	1.070
$\mathrm{DW}_{7\times7}$	77.4	5.14	1.094
$\mathrm{DW}_{5\times5,d=1} + \mathrm{DW}_{7\times7,d=3}$	77.5	5.15	1.112
$\mathrm{DW}_{5\times5,d=1} + \mathrm{DW}_{5\times5,d=2} + \mathrm{DW}_{7\times7,d=3}$	77.5	5.17	1.185
+Multi-order, $C_l : C_m : C_h = 1:0:3$	77.5	5.17	1.099
+Multi-order, $C_l : C_m : C_h = 0:1:1$	77.6	5.17	1.103
+Multi-order, $C_l : C_m : C_h = 1:6:9$	77.7	5.17	1.104
+Multi-order, $C_l : C_m : C_h = 1:3:4$	**77.8**	5.17	1.102

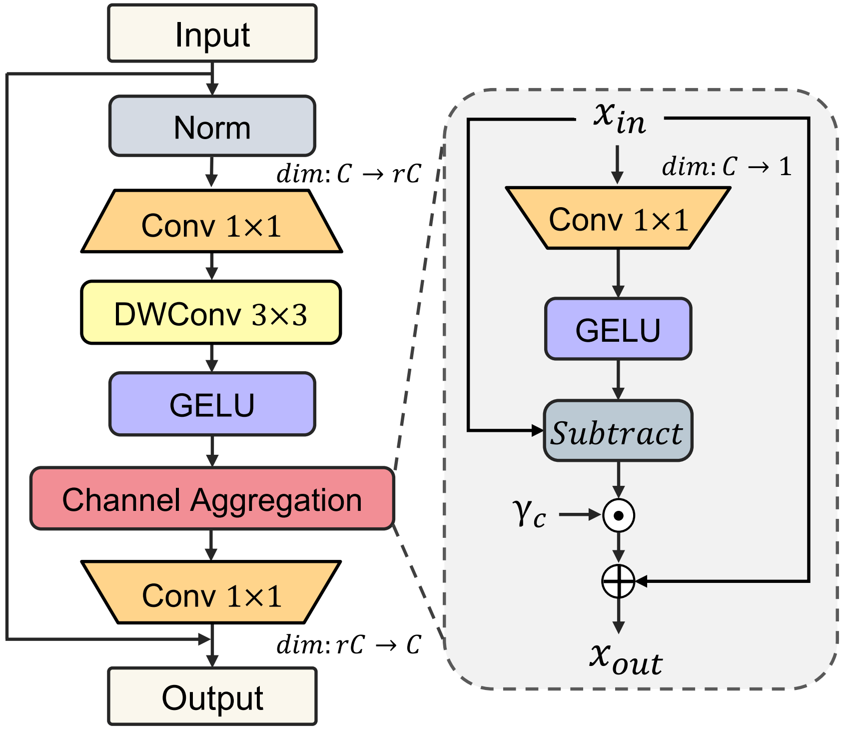
Top-1	Acc (%)	Context branch None	GELU	SiLU
Gating branch	None	76.3	76.7	76.7
	Sigmoid	76.8	77.0	76.9
	GELU	76.7	76.8	77.0
	SiLU	76.9	77.1	**77.2**

Ablation of SA module with MogaNet-T on ImageNet

[1] MogaNet: Multi-order Gated Aggregation Network. ICLR, 2024.

Multi-order Interaction: MogaNet

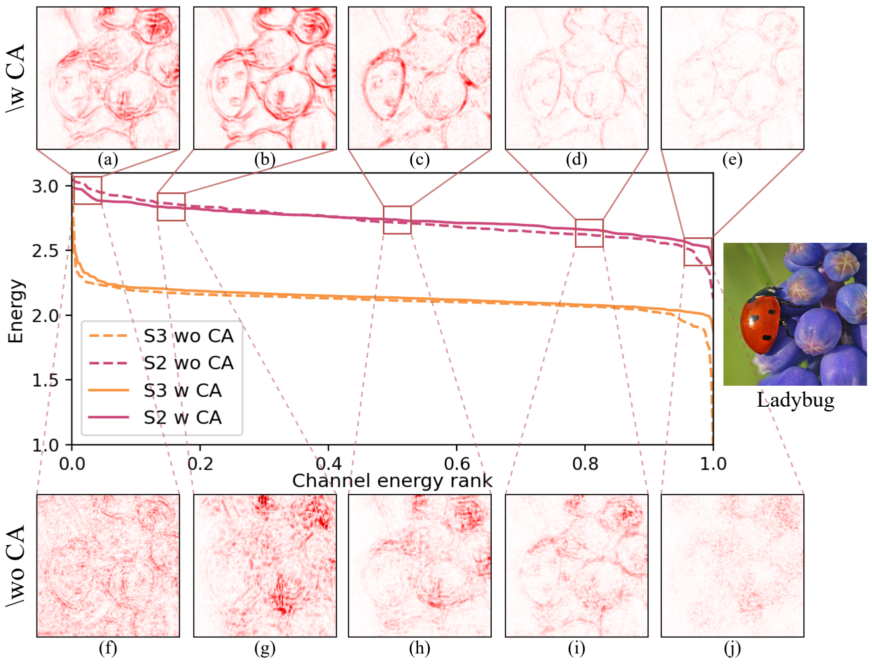
- **Channel Aggregation (CA): Multi-order Channel Reallocation.**



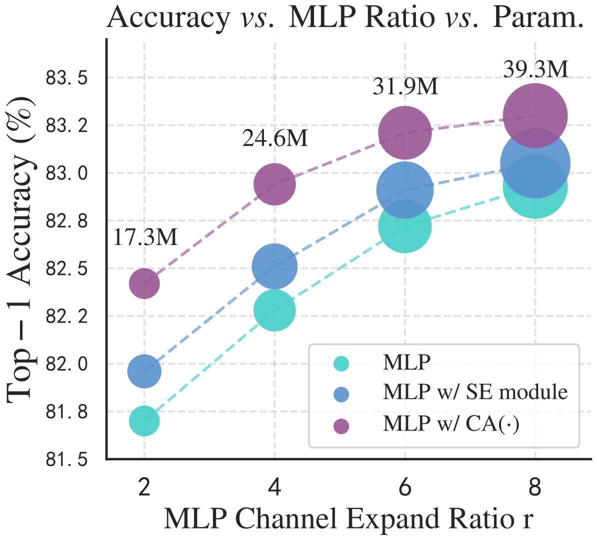
$$Y = \mathrm{GELU}\Big(\mathrm{DW}_{3\times3}\big(\mathrm{Conv}_{1\times1}(\mathrm{Norm}(X))\big)\Big),$$

$$Z = \mathrm{Conv}_{1\times1}\big(\mathrm{CA}(Y)\big) + X.$$

$$\mathrm{CA}(X) = X + \gamma_c \odot \big(X - \mathrm{GELU}(XW_r)\big)$$



Channel energy ranks and channel saliency maps (CSM)[1]



Modules		Top-1 Acc (%)	Params. (M)	FLOPs (G)
Baseline		76.6	4.75	1.01
SMixer	+Gating branch	77.3	5.09	1.07
	$+\mathrm{DW}_{7\times7}$	77.5	5.14	1.09
	+Multi-order DW(·)	78.0	5.17	1.10
	+FD(·)	78.3	5.18	1.10
CMixer	+SE module	78.6	5.29	1.14
	+CA(·)	**79.0**	5.20	1.10

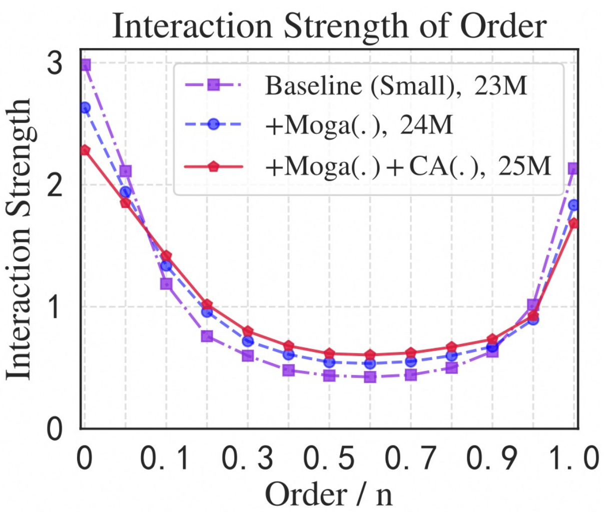
Ablation of MogaNet-S on ImageNet

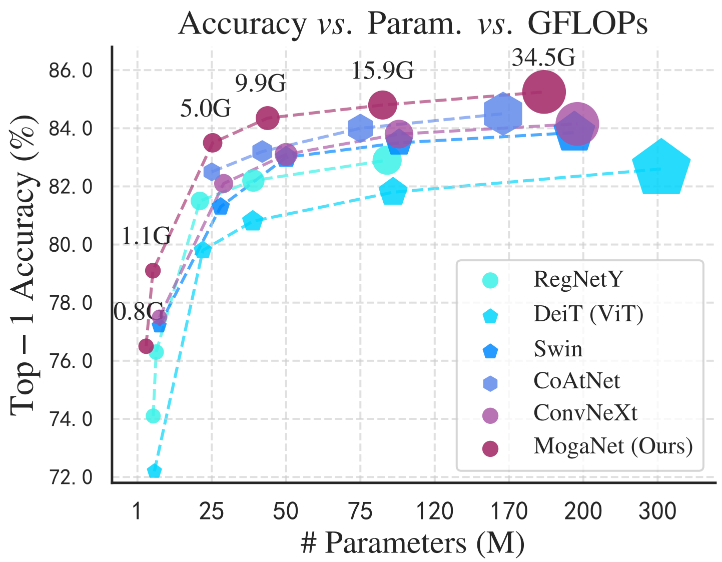
[1] Reflash dropout in image supe-resolution. CVPR, 2022.

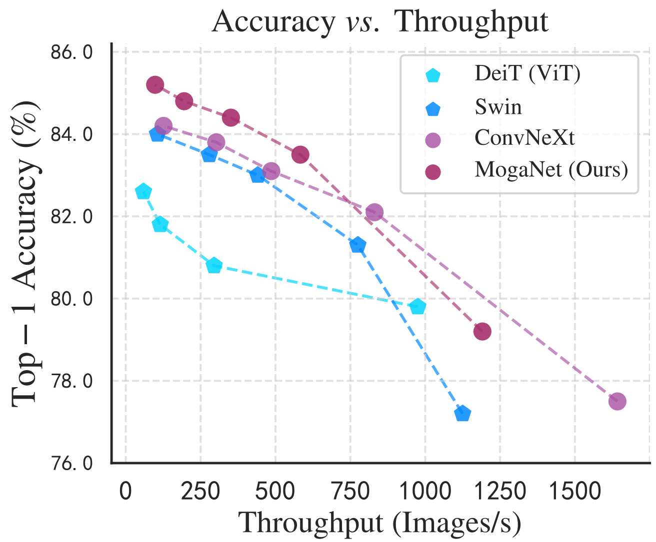
Multi-order Interaction: MogaNet

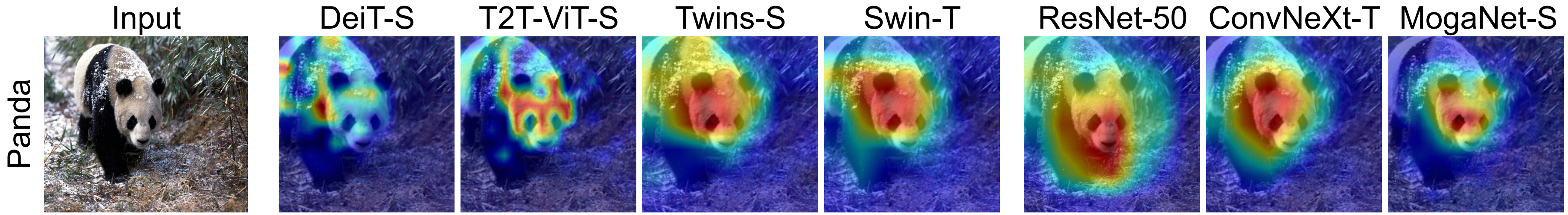
- **Great scalability and efficiency of parameters.**
- **Relieving representation bottleneck.**

Modules	Top-1 Acc (%)
ConvNeXt-T	82.1
Baseline	82.2
Moga Block	**83.4**
−FD(·)	83.2
−Multi-DW(·)	83.1
−Moga(·)	82.7
−CA(·)	82.9









西湖大學 WESTLAKE UNIVERSITY

Comparison Experiments of MogaNet

- ImageNet-1K Classification: 3M to 200M.

Architecture	Date	Type	Image Size	Param. (M)	FLOPs (G)	Top-1 Acc (%)
ResNet-18	CVPR'2016	C	224^2	11.7	1.80	71.5
ShuffleNetV2 2×	ECCV'2018	C	224^2	5.5	0.60	75.4
EfficientNet-B0	ICML'2019	C	224^2	5.3	0.39	77.1
RegNetY-800MF	CVPR'2020	C	224^2	6.3	0.80	76.3
DeiT-T†	ICML'2021	T	224^2	5.7	1.08	74.1
PVT-T	ICCV'2021	T	224^2	13.2	1.60	75.1
T2T-ViT-7	ICCV'2021	T	224^2	4.3	1.20	71.7
ViT-C	NIPS'2021	T	224^2	4.6	1.10	75.3
SReT-T$_{Distill}$	ECCV'2022	T	224^2	4.8	1.10	77.6
PiT-Ti	ICCV'2021	H	224^2	4.9	0.70	74.6
LeViT-S	ICCV'2021	H	224^2	7.8	0.31	76.6
CoaT-Lite-T	ICCV'2021	H	224^2	5.7	1.60	77.5
Swin-1G	ICCV'2021	H	224^2	7.3	1.00	77.3
MobileViT-S	ICLR'2022	H	256^2	5.6	4.02	78.4
MobileFormer-294M	CVPR'2022	H	224^2	11.4	0.59	77.9
ConvNext-XT	CVPR'2022	C	224^2	7.4	0.60	77.5
VAN-B0	CVMJ'2023	C	224^2	4.1	0.88	75.4
ParC-Net-S	ECCV'2022	C	256^2	5.0	3.48	78.6
MogaNet-XT	Ours	C	256^2	3.0	1.04	77.2
MogaNet-T	Ours	C	224^2	5.2	1.10	79.0
MogaNet-T§	Ours	C	256^2	5.2	1.44	**80.0**

Light-weight (3-10M)

Architecture	Date	Type	Image Size	Param. (M)	FLOPs (G)	Top-1 Acc (%)
Deit-S	ICML'2021	T	224^2	22	4.6	79.8
Swin-T	ICCV'2021	T	224^2	28	4.5	81.3
CSWin-T	CVPR'2022	T	224^2	23	4.3	82.8
LITV2-S	NIPS'2022	T	224^2	28	3.7	82.0
CoaT-S	ICCV'2021	H	224^2	22	12.6	82.1
CoAtNet-0	NIPS'2021	H	224^2	25	4.2	82.7
UniFormer-S	ICLR'2022	H	224^2	22	3.6	82.9
RegNetY-4GF†	CVPR'2020	C	224^2	21	4.0	81.5
ConvNeXt-T	CVPR'2022	C	224^2	29	4.5	82.1
SLaK-T	ICLR'2023	C	224^2	30	5.0	82.5
HorNet-T$_{7\times7}$	NIPS'2022	C	224^2	22	4.0	82.8
MogaNet-S	Ours	C	224^2	25	5.0	**83.4**
Swin-S	ICCV'2021	T	224^2	50	8.7	83.0
Focal-S	NIPS'2021	T	224^2	51	9.1	83.6
CSWin-S	CVPR'2022	T	224^2	35	6.9	83.6
LITV2-M	NIPS'2022	T	224^2	49	7.5	83.3
CoaT-M	ICCV'2021	H	224^2	45	9.8	83.6
CoAtNet-1	NIPS'2021	H	224^2	42	8.4	83.3
UniFormer-B	ICLR'2022	H	224^2	50	8.3	83.9
FAN-B-Hybrid	ICML'2022	H	224^2	50	11.3	83.9
EfficientNet-B6	ICML'2019	C	528^2	43	19.0	84.0
RegNetY-8GF†	CVPR'2020	C	224^2	39	8.1	82.2
ConvNeXt-S	CVPR'2022	C	224^2	50	8.7	83.1
FocalNet-S (LRF)	NIPS'2022	C	224^2	50	8.7	83.5
HorNet-S$_{7\times7}$	NIPS'2022	C	224^2	50	8.8	84.0
SLaK-S	ICLR'2023	C	224^2	55	9.8	83.8
MogaNet-B	Ours	C	224^2	44	9.9	**84.3**

Normal size (25-50M)

- COCO Det. and Ins. Seg.

Architecture	Type	#P. (M)	FLOPs (G)	Mask R-CNN 1× AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
RegNet-800M	C	27	187	37.5	57.9	41.1	34.3	56.0	36.8
MogaNet-XT	C	23	185	**40.7**	**62.3**	**44.4**	**37.6**	**59.6**	**40.2**
ResNet-18	C	31	207	34.0	54.0	36.7	31.2	51.0	32.7
RegNet-1.6G	C	29	204	38.9	60.5	43.1	35.7	57.4	38.9
PVT-T	T	33	208	36.7	59.2	39.3	35.1	56.7	37.3
PoolFormer-S12	T	32	207	37.3	59.0	40.1	34.6	55.8	36.9
MogaNet-T	C	25	192	**42.6**	**64.0**	**46.4**	**39.1**	**61.3**	**42.0**
ResNet-50	C	44	260	38.0	58.6	41.4	34.4	55.1	36.7
RegNet-6.4G	C	45	307	41.1	62.3	45.2	37.1	59.2	39.6
PVT-S	T	44	245	40.4	62.9	43.8	37.8	60.1	40.3
Swin-T	T	48	264	42.2	64.6	46.2	39.1	61.6	42.0
MViT-T	T	46	326	45.9	**68.7**	50.5	42.1	**66.0**	45.4
PoolFormer-S36	T	32	207	41.0	63.1	44.8	37.7	60.1	40.0
Focal-T	T	49	291	44.8	67.7	49.2	41.0	64.7	44.2
PVTV2-B2	T	45	309	45.3	67.1	49.6	41.2	64.2	44.4
LITV2-S	T	47	261	44.9	67.0	49.5	40.8	63.8	44.2
CMT-S	H	45	249	44.6	66.8	48.9	40.7	63.9	43.4
Conformer-S/16	H	58	341	43.6	65.6	47.7	39.7	62.6	42.5
Uniformer-S	H	41	269	45.6	68.1	49.7	41.6	64.8	45.0
ConvNeXt-T	C	48	262	44.2	66.6	48.3	40.1	63.3	42.8
FocalNet-T (SRF)	C	49	267	45.9	68.3	50.1	41.3	65.0	44.3
FocalNet-T (LRF)	C	49	268	46.1	68.2	50.6	41.5	65.1	44.5
MogaNet-S	C	45	272	**46.7**	68.0	**51.3**	**42.2**	65.4	**45.5**
ResNet-101	C	63	336	40.4	61.1	44.2	36.4	57.7	38.8
RegNet-12G	C	64	423	42.2	63.7	46.1	38.0	60.5	40.5
PVT-M	T	64	302	42.0	64.4	45.6	39.0	61.6	42.1
Swin-S	T	69	354	44.8	66.6	48.9	40.9	63.4	44.2
Focal-S	T	71	401	47.4	69.8	51.9	42.8	66.6	46.1
PVTV2-B3	T	65	397	47.0	68.1	51.7	42.5	65.7	45.7
LITV2-M	T	68	315	46.5	68.0	50.9	42.0	65.1	45.0
UniFormer-B	H	69	399	47.4	69.7	52.1	43.1	66.0	46.5
ConvNeXt-S	C	70	348	45.4	67.9	50.0	41.8	65.2	45.1
MogaNet-B	C	63	373	**47.9**	**70.0**	**52.7**	**43.2**	**67.0**	**46.6**
Swin-B	T	107	496	46.9	69.6	51.2	42.3	65.9	45.6
PVTV2-B5	T	102	557	47.4	68.6	51.9	42.5	65.7	46.0
ConvNeXt-B	C	108	486	47.0	69.4	51.7	42.7	66.3	46.0
FocalNet-B (SRF)	C	109	496	48.8	70.7	53.5	43.3	67.5	46.5
MogaNet-L	C	102	495	**49.4**	**70.7**	**54.1**	**44.1**	**68.1**	**47.6**

- ADE20K Sematic Seg.
- Video Prediction
- COCO 2D / 3D Pose Estimation

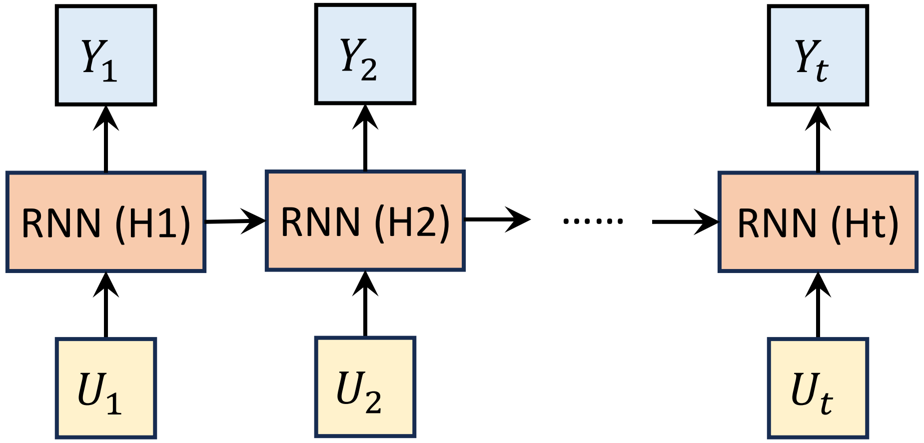
西湖大學 WESTLAKE UNIVERSITY

State-Space Models

- State-Space Model (SSM): "Parallel RNN"
- SSM vs. Convolution: "Long Convolution"

SSM: $$\begin{aligned} \hat{h}(t) &= \boldsymbol{A}h(t) + \boldsymbol{B}u(t), \\ y(t) &= \boldsymbol{C}h(t) + \boldsymbol{D}u(t). \end{aligned}$$

RNN: $$\begin{aligned} h_t &= \sigma(W_1 U_t + W_2 h_{t-1}), \\ o_t &= \sigma(W_3 h_t). \end{aligned}$$



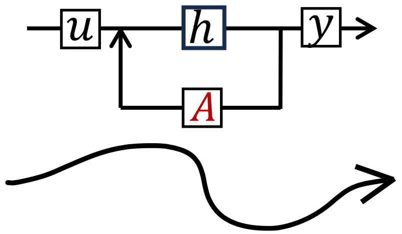
$$y_k = \overline{\boldsymbol{C}\boldsymbol{A}}^k \overline{\boldsymbol{B}} u_0 + \overline{\boldsymbol{C}\boldsymbol{A}}^{k-1} \overline{\boldsymbol{B}} u_1 + \cdots + \overline{\boldsymbol{C}\boldsymbol{A}\boldsymbol{B}} u_{k-1} + \overline{\boldsymbol{C}\boldsymbol{B}} u_k$$

$$y = \overline{\boldsymbol{K}} * u.$$

HiPPO Matrix

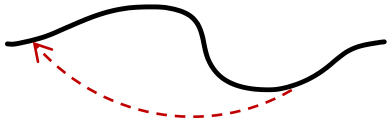
$$\boldsymbol{A}_{nk} = \begin{cases} (-1)^{n-k}(2k+1) & n > k \\ k+1 & n = k \\ 0 & n < k \end{cases}$$

$$\boldsymbol{A} = \begin{bmatrix} 1 \\ -1 & 2 \\ 1 & -3 & 3 \\ -1 & 3 & -5 & 4 \\ 1 & -3 & 5 & -7 & 5 \\ -1 & 3 & -5 & 7 & -9 & 6 \\ 1 & -3 & 5 & -7 & 9 & -11 & 7 \\ -1 & 3 & -5 & 7 & -9 & 11 & -13 & 8 \\ \vdots & & & & & & & & \ddots \end{bmatrix}$$



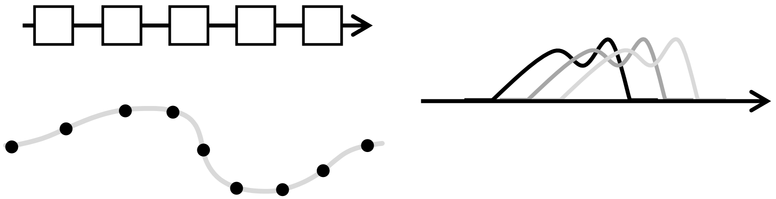
$$\begin{aligned} \hat{h} &= Ah + Bu \\ y &= Ch + Du \end{aligned}$$

Continuous State Space



$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 3 & 3 \end{bmatrix}$$

Long-Range Dependencies



$$\begin{aligned} h &= \bar{A}h + \bar{B}u \\ y &= \bar{C}h + \bar{D}u \end{aligned}$$

$$y = \overline{K} * u$$

Fast Discrete Representations

[1] Efficiently Modeling Long Sequences with Structured State Spaces. ICLR, 2022. [2] HiPPO: Recurrent Memory with Optimal Polynomial Projections. arXiv, 2020.

State-Space Models: Mamba

Structured state space sequence models (S4)

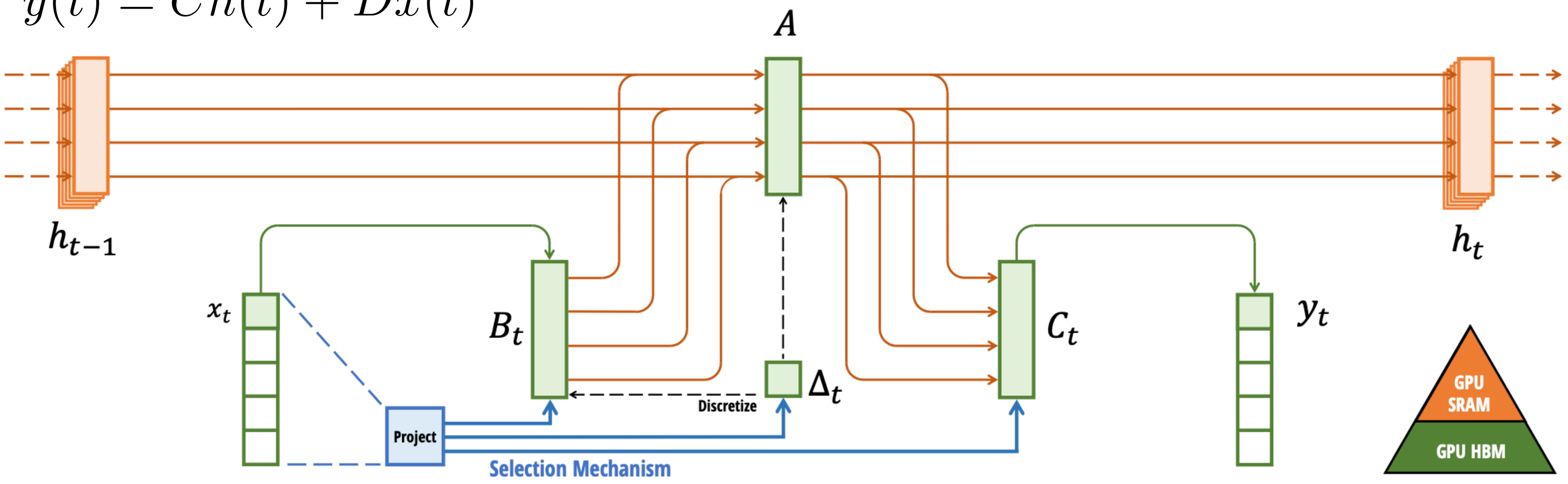
$$h'(t) = \boldsymbol{A}h(t) + \boldsymbol{B}x(t) \quad \text{(1a)}$$
$$y(t) = \boldsymbol{C}h(t) \quad \text{(1b)}$$

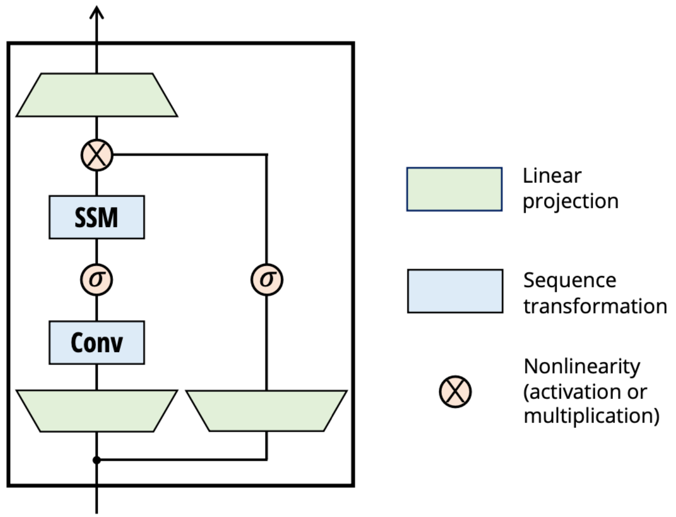
$$h_t = \overline{\boldsymbol{A}}h_{t-1} + \overline{\boldsymbol{B}}x_t \quad \text{(2a)}$$
$$y_t = \boldsymbol{C}h_t \quad \text{(2b)}$$

$$\overline{\boldsymbol{K}} = (\boldsymbol{C}\overline{\boldsymbol{B}}, \boldsymbol{C}\overline{\boldsymbol{A}\boldsymbol{B}}, \dots, \boldsymbol{C}\overline{\boldsymbol{A}}^{k}\overline{\boldsymbol{B}}, \dots) \quad \text{(3a)}$$
$$y = x * \overline{\boldsymbol{K}} \quad \text{(3b)}$$

$$x(t) \in \mathbb{R}^L \to y(t) \in \mathbb{R}^L, \mathrm{A} \in \mathbb{C}^{N\times N}, B, C \in \mathbb{C}^N, D \in \mathbb{C}^1$$

$$h'(t) = Ah(t) + Bx(t)$$
$$y(t) = Ch(t) + Dx(t)$$





→ **Mamba**

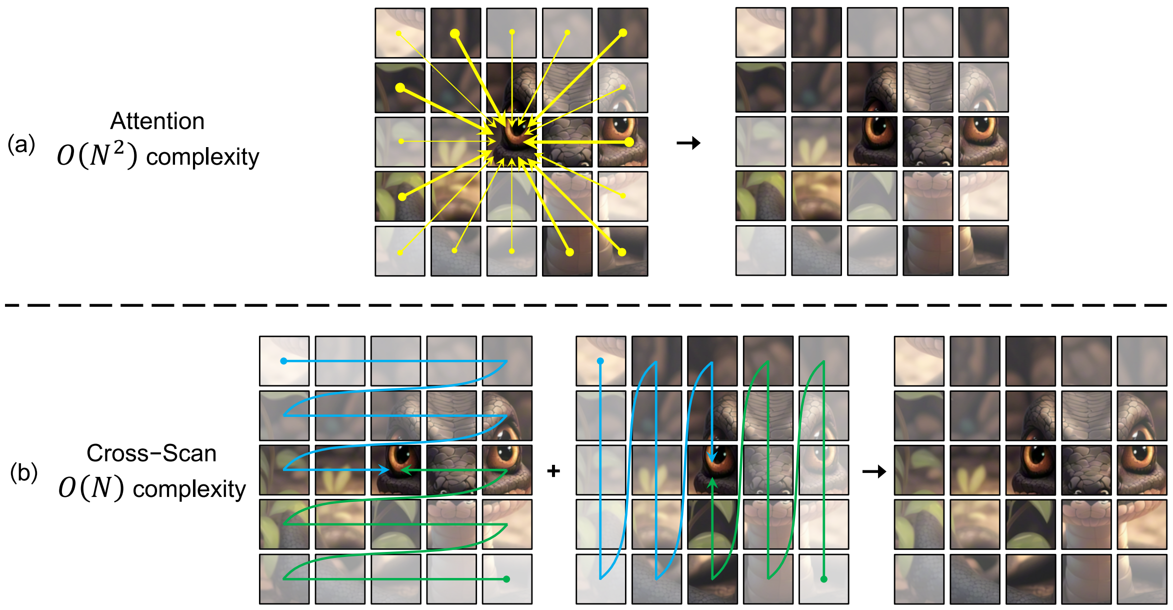
$$g_t = \sigma(\mathrm{Linear}(x_t))$$
$$h_t = (1 - g_t)h_{t-1} + g_t x_t$$

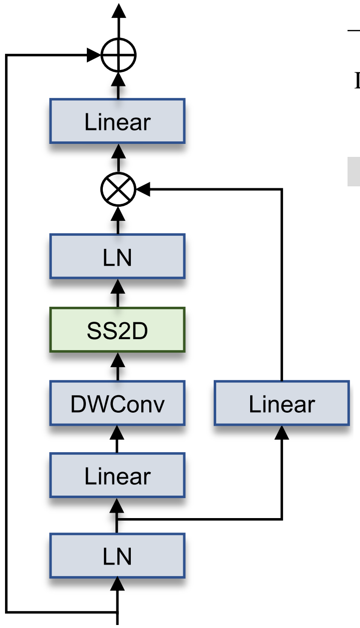
Model	Params	Accuracy (%) at Sequence Length					
		2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}
HyenaDNA	1.4M	28.04	28.43	41.17	42.22	31.10	54.87
Mamba	1.4M	31.47	27.50	27.66	40.72	42.41	**71.67**
Mamba	7M	30.00	29.01	31.48	43.73	56.60	**81.31**

Great Apes DNA Classification

[1] Linear-Time Sequence Modeling with Selective State Spaces. arXiv, 2023.

State-Space Models: VMamba



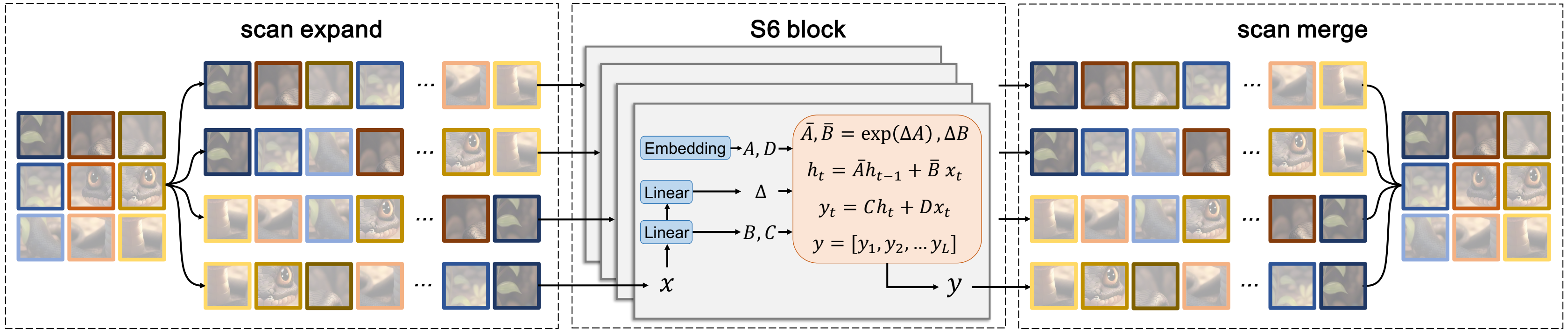


ADE20K Segmentation

method	crop size	mIoU (SS)	mIoU (MS)	#param.	FLOPs
ResNet-50	512^2	42.1	42.8	67M	953G
DeiT-S + MLN	512^2	43.8	45.1	58M	1217G
Swin-T	512^2	44.4	45.8	60M	945G
ConvNeXt-T	512^2	46.0	46.7	60M	939G
VMamba-T	512^2	47.3	48.3	55M	939G

ImageNet-1K Classification

method	image size	#param.	FLOPs	ImageNet top-1 acc.
RegNetY-4G [36]	224^2	21M	4.0G	80.0
RegNetY-8G [36]	224^2	39M	8.0G	81.7
RegNetY-16G [36]	224^2	84M	16.0G	82.9
EffNet-B3 [42]	300^2	12M	1.8G	81.6
EffNet-B4 [42]	380^2	19M	4.2G	82.9
EffNet-B5 [42]	456^2	30M	9.9G	83.6
EffNet-B6 [42]	528^2	43M	19.0G	84.0
ViT-B/16 [10]	384^2	86M	55.4G	77.9
ViT-L/16 [10]	384^2	307M	190.7G	76.5
DeiT-S [45]	224^2	22M	4.6G	79.8
DeiT-B [45]	224^2	86M	17.5G	81.8
DeiT-B [45]	384^2	86M	55.4G	83.1
Swin-T [28]	224^2	29M	4.5G	81.3
Swin-S [28]	224^2	50M	8.7G	83.0
Swin-B [28]	224^2	88M	15.4G	83.5
S4ND-ViT-B [35]	224^2	89M	-	80.4
VMamba-T	224^2	22M	4.5G	82.2
VMamba-S	224^2	44M	9.1G	83.5



[1] VMamba: Visual State Space Model. arXiv, 2024.

西湖大學
WESTLAKE UNIVERSITY

Thank you!

Paper: MogaNet

Code: MogaNet

Homepage

lisiyuan@westlake.edu.cn