南京航空航天大學

Nanjing University of Aeronautics and Astronautics

IMAGE AS SET OF POINTS

Xu Ma[1*], Yuqian Zhou[2*], Huan Wang[1], Can Qin[1], Bin Sun[1], Chang Liu[1], Yun Fu[1]

[1]Northeastern University [2]Adobe Inc.

{ma.xu1,wang.huan,qin.ca,sun.bi,liu.chang6}@northeastern.edu

yuqzhou@adobe.com yunfu@ece.neu.edu

ICLR 2023

1 Preface

The way we extract features depends a lot on how we interpret an image.

- **Convolutional Networks (ConvNets)** consider an image as organized pixels in a rectangular shape and extract features via convolutional operation in local region;
- **Vision Transformers (ViTs)** treat an image as a sequence of patches and extract features via attention mechanism in a global range;
- **Context clusters (CoCs)** view an image as a set of unorganized points and extract features via simplified clustering algorithm. In detail, each point includes the raw feature (e.g., color) and positional information (e.g., coordinates), and a simplified clustering algorithm is employed to group and extract deep features hierarchically.

2 Example

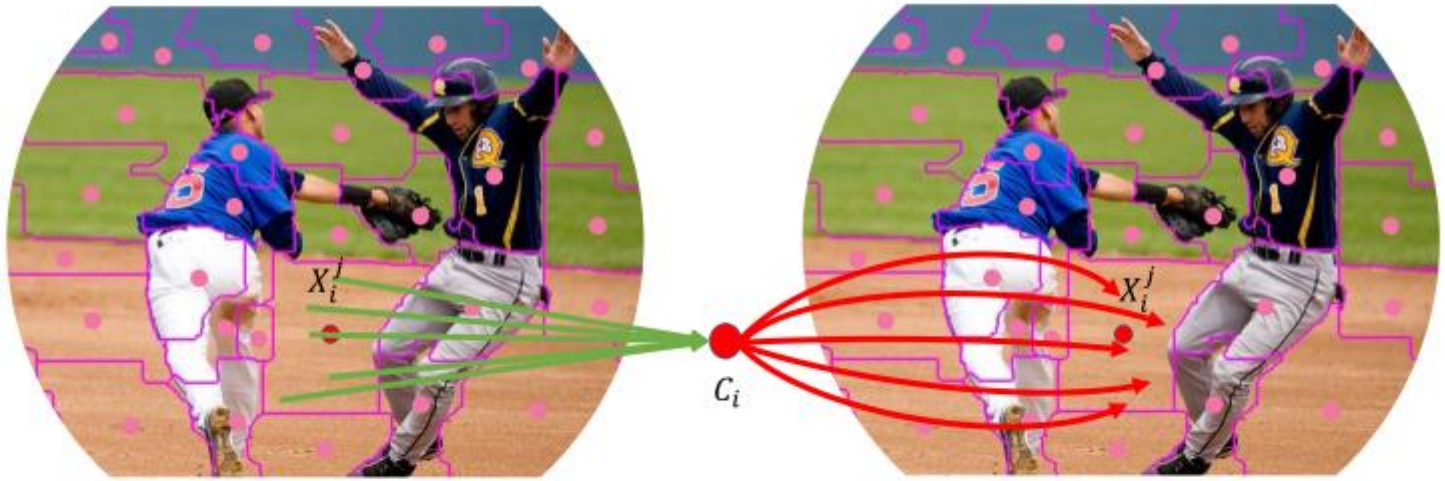


Figure 1: A context cluster in our network trained for image classification. We view *an image as a set of points* and sample c centers for points clustering. Point features are aggregated and then dispatched within a cluster. For cluster center C_i, we first aggregated all points $\{x_i^0, x_i^1, \cdots, x_i^n\}$ in ith cluster, then the aggregated result is distributed to all points in the clusters dynamically. See § 3 for details.

3 Method

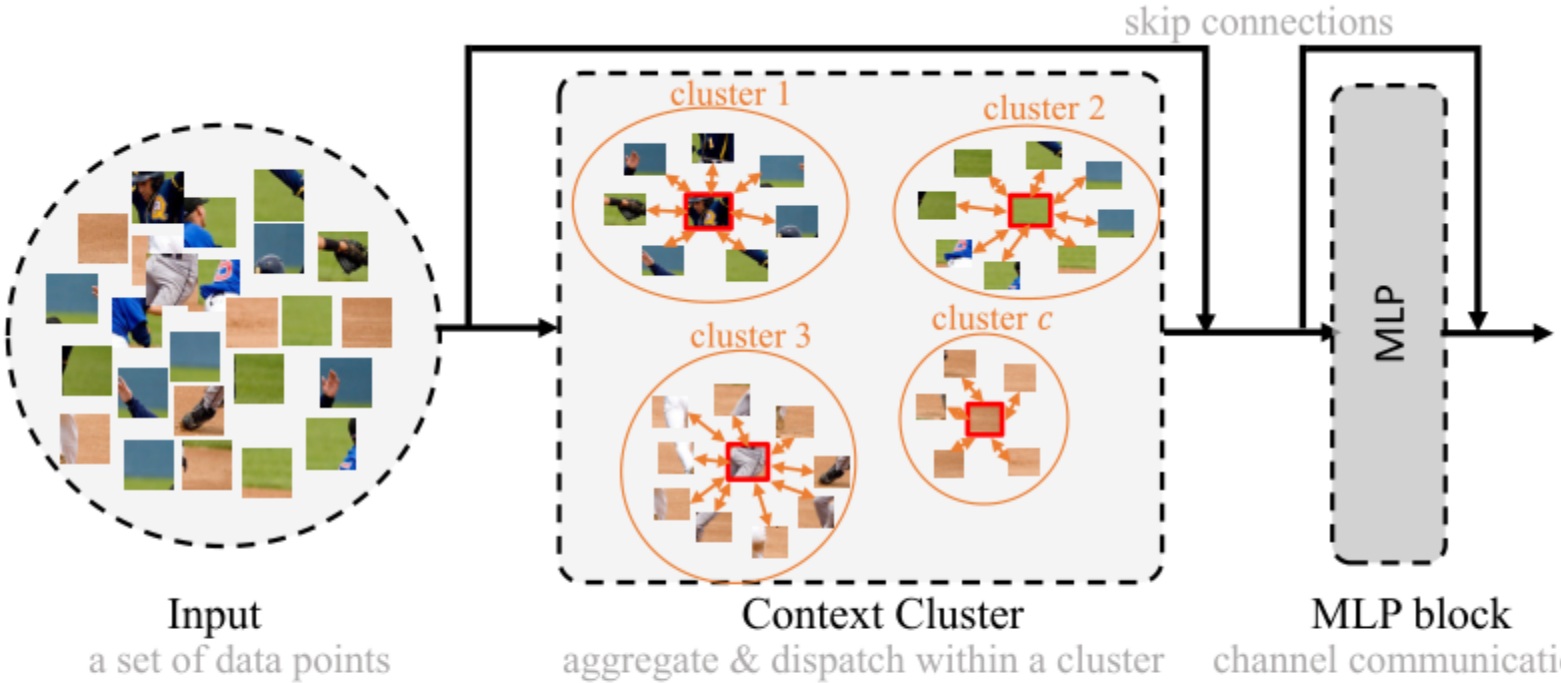


Figure 2: A Context Cluster block. We use a context cluster operation to group a set of data points, and then communicate the points within clusters. An MLP block is applied later.

3 Method——Context Clusters Pipeline

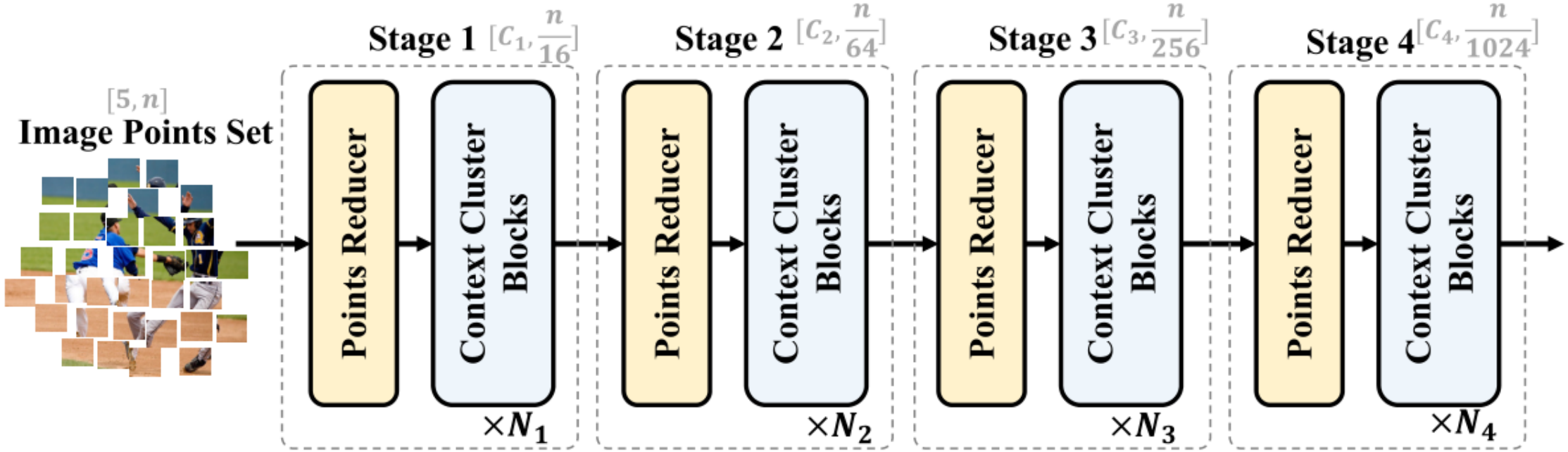
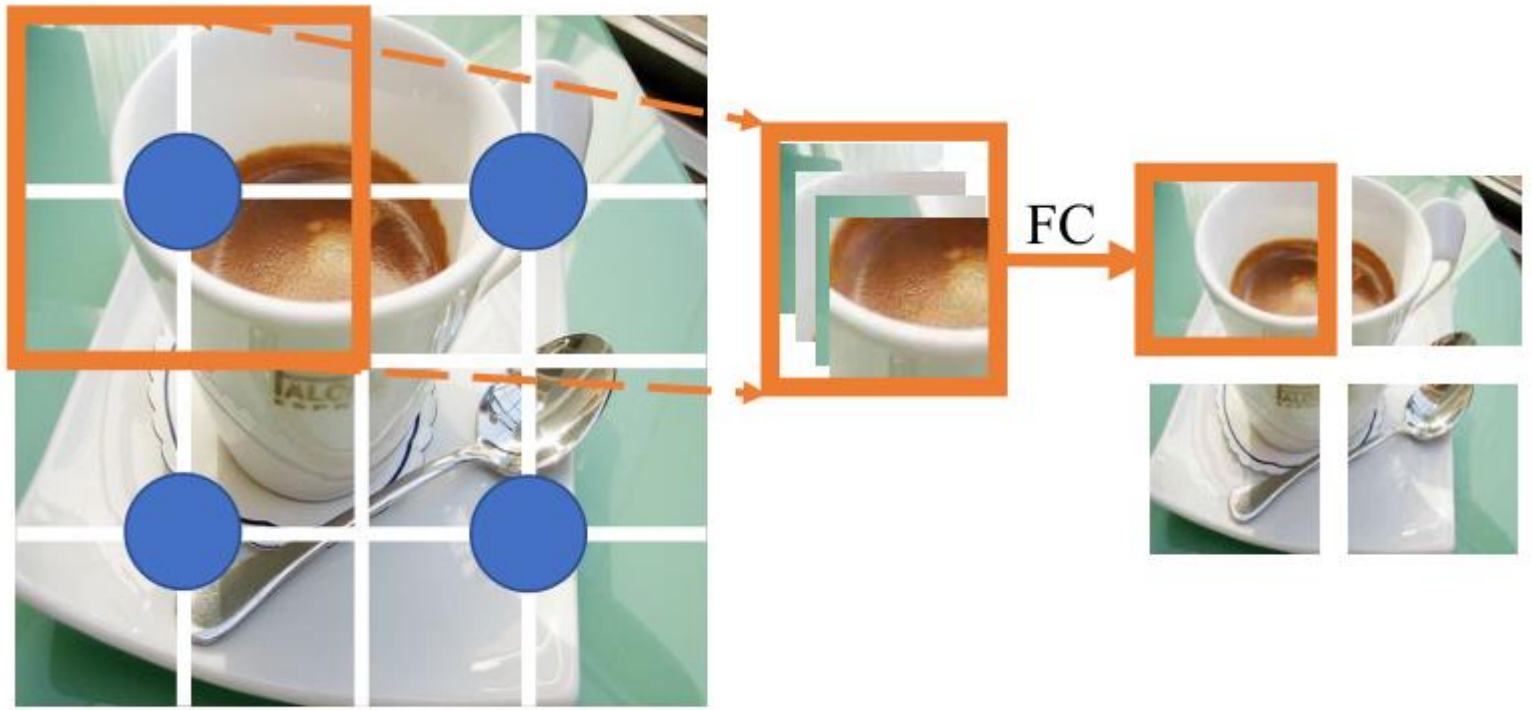


Figure 3: Context Cluster architecture with four stages. Given a set of image points, Context Cluster gradually reduces the point number and extracts deep features. Each stage begins with a points reducer, after which a succession of context cluster blocks is used to extract features.

3.1 Points Reducer

To reduce the points number, we evenly select some anchors in space, and the nearest k points are concatenated and fused by a linear projection.



(a) Illustration of anchors for points reduction.

3.2 Context Cluster Blocks

1. Context Clustering

feature points $\mathbf{P} \in \mathbb{R}^{n \times d}$ linearly project $\mathbf{P}$ to $\mathbf{P}_s$ for similarity computation

evenly propose **c** centers in space, and the center feature is computed by averaging its **k** nearest points——the conventional SuperPixel method SLIC (Achanta et al., 2012)

similarity matrix $\mathbf{S} \in \mathbb{R}^{c \times n}$

2. Feature Aggregating

map the points to a value space $P_v \in \mathbb{R}^{m \times d'}$

$$g = \frac{1}{C}\left(v_c + \sum_{i=1}^{m} \mathrm{sig}\left(\alpha s_i + \beta\right) * v_i\right), \qquad \text{s.t.,} \quad C = 1 + \sum_{i=1}^{m} \mathrm{sig}\left(\alpha s_i + \beta\right). \tag{1}$$

3. Feature Dispatching

$$p_i' = p_i + \mathrm{FC}\left(\mathrm{sig}\left(\alpha s_i + \beta\right) * g\right). \tag{2}$$

4. Multi-Head Computing

4 Experiments—— Image Classification

Table 1: Comparison with representative backbones on ImageNet-1k benchmark. Throughput (images / s) is measured on a single V100 GPU with a batch size of 128, and is averaged by the last 500 iterations. All models are trained and tested at 224×224 resolution, except ViT-B and ViT-L.

	Method	Param.	GFLOPs	Top-1	Throughputs (images/s)
MLP	♣ ResMLP-12 (Touvron et al., 2022)	15.0	3.0	76.6	511.4
	♣ ResMLP-24 (Touvron et al., 2022)	30.0	6.0	79.4	509.7
	♣ ResMLP-36 (Touvron et al., 2022)	45.0	8.9	79.7	452.9
	♣ MLP-Mixer-B/16 (Tolstikhin et al., 2021)	59.0	12.7	76.4	400.8
	♣ MLP-Mixer-L/16 (Tolstikhin et al., 2021)	207.0	44.8	71.8	125.2
	♣ gMLP-Ti (Liu et al., 2021a)	6.0	1.4	72.3	511.6
	♣ gMLP-S (Liu et al., 2021a)	20.0	4.5	79.6	509.4
Attention	♦ ViT-B/16 (Dosovitskiy et al., 2020)	86.0	55.5	77.9	292.0
	♦ ViT-L/16 (Dosovitskiy et al., 2020)	307	190.7	76.5	92.8
	♦ PVT-Tiny (Wang et al., 2021)	13.2	1.9	75.1	-
	♦ PVT-Small (Wang et al., 2021)	24.5	3.8	79.8	-
	♦ T2T-ViT-7 (Yuan et al., 2021a)	4.3	1.1	71.7	-
	♦ DeiT-Tiny/16 (Touvron et al., 2021)	5.7	1.3	72.2	523.8
	♦ DeiT-Small/16 (Touvron et al., 2021)	22.1	4.6	79.8	521.3
	♦ Swin-T (Liu et al., 2021b)	29	4.5	81.3	-
Convolution	♠ ResNet18 (He et al., 2016)	12	1.8	69.8	584.9
	♠ ResNet50 (He et al., 2016)	26	4.1	79.8	524.8
	♠ ConvMixer-512/16 (Trockman et al., 2022)	5.4	-	73.8	-
	♠ ConvMixer-1024/12 (Trockman et al., 2022)	14.6	-	77.8	-
	♠ ConvMixer-768/32 (Trockman et al., 2022)	21.1	-	80.16	142.9
Cluster	♥ Context-Cluster-Ti (ours)	5.3	1.0	71.8	518.4
	♥ Context-Cluster-Ti‡ (ours)	5.3	1.0	71.7	510.8
	♥ Context-Cluster-Small (ours)	14.0	2.6	77.5	513.0
	♥ Context-Cluster-Medium (ours)	27.9	5.5	81.0	325.2

4 Experiments——Component Ablation

Table 2: Component ablation studies of Context-Cluster-Small on ImageNet-1k.

position info.	context cluster	multi head	top-1 acc.
✗	✗	✗	-
✓	✗	✗	74.2(↓3.3)
✓	✓	✗	76.6(↓0.9)
✓	✓	✓	**77.5**

4 Experiments——Visualization

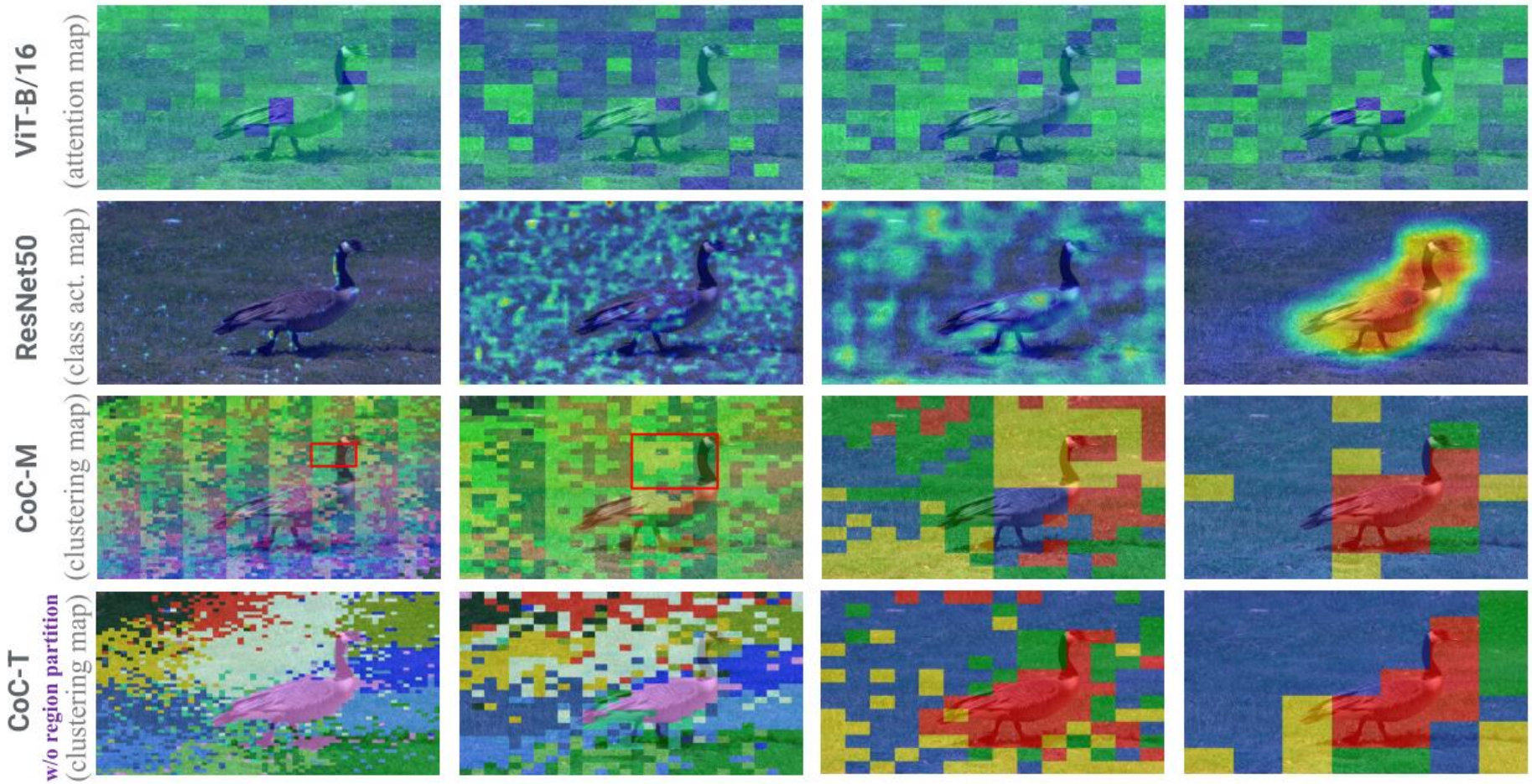


Figure 4: Visualization of activation map, class activation map, and clustering map for ViT-B/16, ResNet50, our CoC-M, and CoC-T without region partition, respectively. We plot the results of the last block in the four stages from left to right. For ViT-B/16, we select the [3rd, 6th, 9th, 12th] blocks, and show the cosine attention map for the `cls-token`. The clustering maps show that our Context Cluster is able to cluster similar contexts together, and tell what model learned visually.

4 Experiments——Visualization

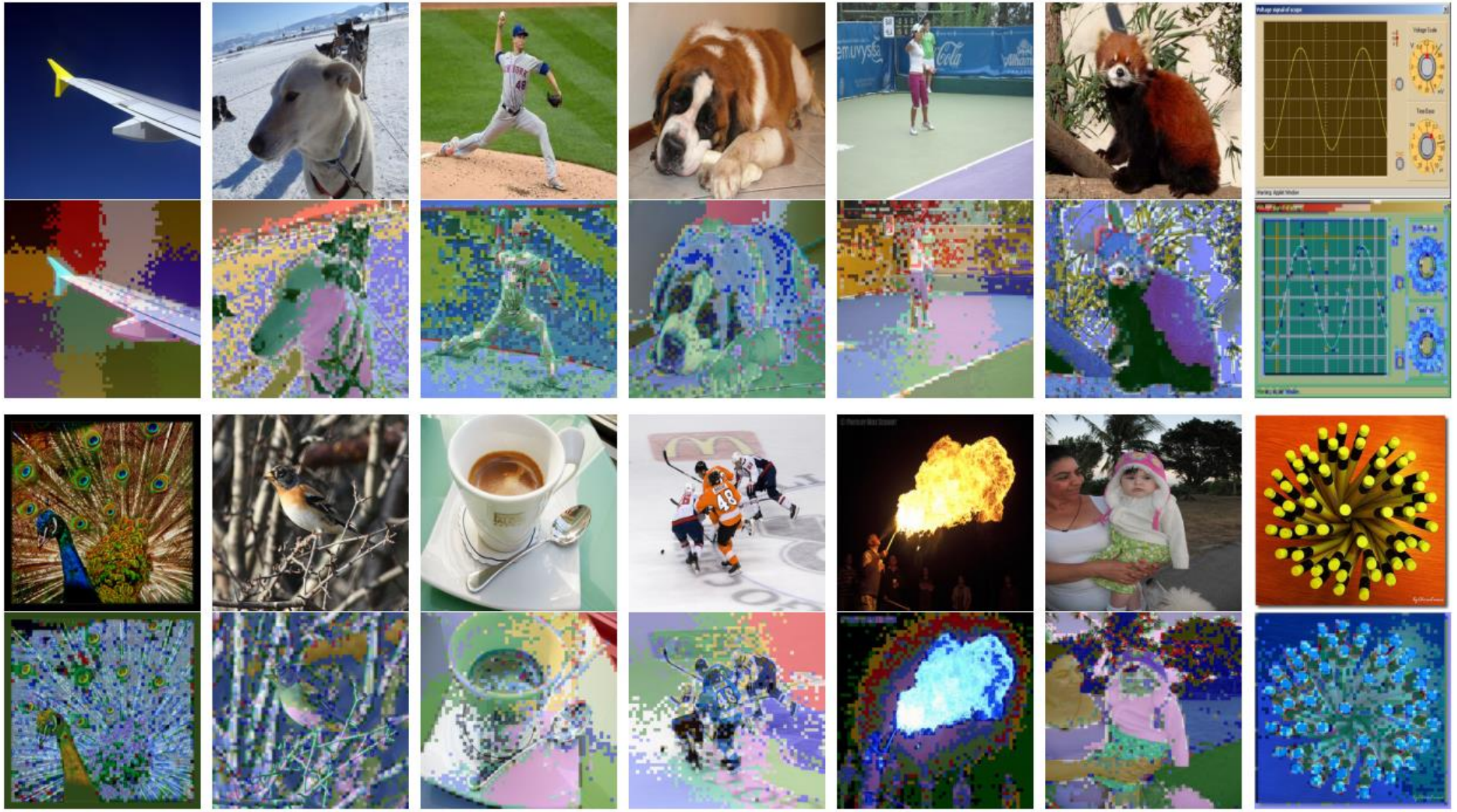


Figure 7: The clustering results of the last context cluster block in the first CoC-Tiny stage (without region partition). Without region partition, Our Context Cluster astonishingly displays "SuperPixel"-like clustering results, even in the early stage. we pick the most intriguing one out of the four heads.

4 Experiments——3D Point Cloud Classification

Table 3: Classification results on ScanObjectNN. All results are reported on the most challenging variant (PB_T50_RS).

Method	mAcc(%)	OA(%)
♠ SpiderCNN (Xu et al., 2018)	69.8	73.7
♠ DGCNN (Wang et al., 2019)	73.6	78.1
♠ PointCNN (Li et al., 2018)	75.1	78.5
♠ GBNet (Qiu et al., 2021)	77.8	80.5
♦ PointBert (Yu et al., 2022d)	-	83.1
♦ Point-MAE (Pang et al., 2022)	-	85.2
♦ Point-TnT (Berg et al., 2022)	81.0	83.5
♣ PointNet (Qi et al., 2017a)	63.4	68.2
♣ PointNet++ (Qi et al., 2017b)	75.4	77.9
♣ BGA-PN++ (Uy et al., 2019)	77.5	80.2
♣ PointMLP (Ma et al., 2022)	83.9	85.4
♣ PointMLP-elite (Ma et al., 2022)	81.8	83.8
♥ PointMLP-CoC (ours)	**84.4**$_{\uparrow 0.5}$	**86.2**$_{\uparrow 0.8}$

4 Experiments——Object Detection and Instance Segmentation

Table 4: COCO object detection and instance segmentation results using Mask-RCNN (1×).

Family	Backbone	Params	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
Conv.	♠ ResNet-18	31.2M	34.0	54.0	36.7	31.2	51.0	32.7
Attention	♦ PVT-Tiny	32.9M	36.7	59.2	39.3	35.1	56.7	37.3
Cluster	♥ CoC-Small/4	33.6M	35.9	58.3	38.3	33.8	55.3	35.8
	♥ CoC-Small/25	33.6M	**37.5**	**60.1**	**40.0**	**35.4**	**57.1**	**37.9**
	♥ CoC-Small/49	33.6M	37.2	59.8	39.7	34.9	56.7	37.0

4 Experiments——Semantic Segmentation

Table 5: Semantic segmentation performance of different backbones with Semantic FPN on the ADE20K validation set.

Backbone	Params	mIoU(%)
♠ ResNet18	15.5M	32.9
♦ PVT-Tiny	17.0M	35.7
♥ CoC-Small/4	17.7M	**36.6**
♥ CoC-Small/25	17.7M	**36.4**
♥ CoC-Small/49	17.7M	**36.3**

5 Conclusion

关键：
提出基于上下文集群的深度网络模型CoCs，新设计本质上不同于ConvNets或ViTs，可作为许多任务的Backbone

优势：

◆通过将图像视为一组点，CoCs对不同的数据域表现出很强的泛化能力，如点云、RGBD图像等

◆上下文聚类处理提供CoCs令人满意的可解释性。通过可视化每一层中的聚类，我们可以明确地了解每一层中的学习

南京航空航天大學

Nanjing University of Aeronautics and Astronautics

Thanks for Listening