Revisiting the Integration of Convolution and Attention for Vision Backbone

Lei Zhu
City University of Hong Kong
ray.leizhu@outlook.com

Xinjiang Wang
Sensetime Research
wangxinjiang@sensetime.com

Wayne Zhang
Sensetime Research
wayne.zhang@sensetime.com

Rynson Lau[†]
City University of Hong Kong
Rynson.Lau@cityu.edu.hk

Abstract

Convolutions (Convs) and multi-head self-attentions (MHSAs) are typically considered alternatives to each other for building vision backbones. Although some works try to integrate both, they apply the two operators simultaneously at the finest pixel granularity. With Convs responsible for per-pixel feature extraction already, the question is whether we still need to include the heavy MHSAs at such a fine-grained level. In fact, this is the root cause of the scalability issue w.r.t. the input resolution for vision transformers. To address this important problem, we propose in this work to use MSHAs and Convs in parallel **at different granularity levels** instead. Specifically, in each layer, we use two different ways to represent an image: a fine-grained regular grid and a coarse-grained set of semantic slots. We apply different operations to these two representations: Convs to the grid for local features, and MHSAs to the slots for global features. A pair of fully differentiable soft clustering and dispatching modules is introduced to bridge the grid and set representations, thus enabling local-global fusion. Through extensive experiments on various vision tasks, we empirically verify the potential of the proposed integration scheme, named *GLMix*: by offloading the burden of fine-grained features to light-weight Convs, it is sufficient to use MHSAs in a few (*e.g.*, 64) semantic slots to match the performance of recent state-of-the-art backbones, while being more efficient. Our visualization results also demonstrate that the soft clustering module produces a meaningful semantic grouping effect with only IN1k classification supervision, which may induce better interpretability and inspire new weakly-supervised semantic segmentation approaches. Code will be available at https://github.com/rayleizhu/GLMix.

1 Introduction

Since the renaissance of deep learning over a decade ago, CNNs had dominated image analysis, until recently when transformers become popular in vision tasks. CNNs and transformers differ in how they model spatial feature interactions: CNNs use convolutions (Convs), while transformers use multi-head self-attentions (MHSAs). Both have their own advantages and limitations. For example, Convs have an inductive bias of translation equivariance, which matches the image property and enables decent performances with less data. They also have a linear complexity w.r.t. pixel number, making them scalable to high-resolution input. However, they have a limited receptive field, which cannot be remedied simply by stacking more layers together [39]. In contrast, MHSAs can model

[†] Rynson Lau is the corresponding author.

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

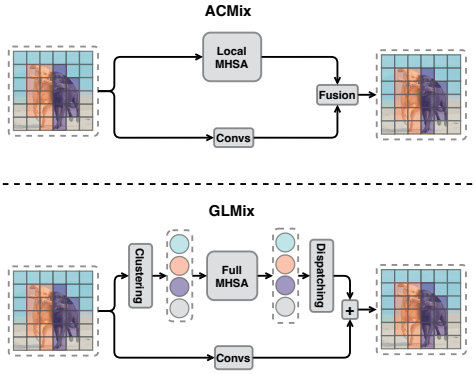

Figure 1: Existing integration schemes, *e.g.*, ACMix [40], apply MHSAs and Convs at the same granularity (top). In contrast, we affirm that by offloading the burden of extracting fine-grained features to lightweight Convs, MHSAs can be aggressively applied to coarse semantic slots to make spatial mixing more efficient (bottom).

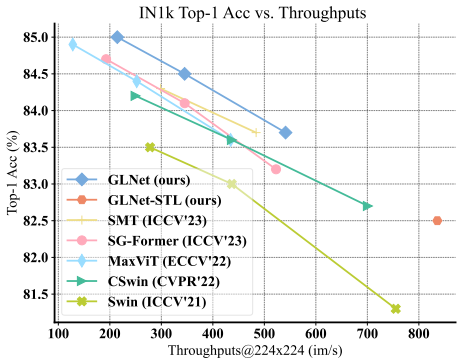

Figure 2: IN1k top-1 accuracy vs. FLOPs. While several recent state-of-the-art models (*i.e.*, MaxViT [47], CSWin [14] and SG-Former [43], SMT [34]) lie in almost the same Pareto frontier, our GLNet models move the frontier further to the upper-right with a clear margin.

long-range dependency flexibly, but suffer from a quadratic complexity w.r.t. input resolution and require more data to compensate for the lack of inductive bias. Besides, some discussions [41] also point out that MHSAs play the role of low-pass filters, while Convs play the role high-pass ones. Hence, they are complementary to each other.

There are indeed some works that use both Convs and MHSAs to build vision backbones. Some of them alternate Convs and MHSAs across different stages/blocks [29, 47], forming a loose collaboration. Others [40, 14, 5] integrate Convs and MHSAs tightly in each block. Specifically, they apply Convs and MHSAs in parallel at the same granularity level and fuse their outputs for further processing, as shown in Figure 1(top). With Convs responsible for fine-grained feature extraction, we ask if we still need to apply the heavy MHSAs at the pixel level. Meanwhile, recent vision-language models [1, 63] have shown that an image can be described as a fixed number of visual tokens regardless of its resolution, possibly stemming from the low-rank property of natural signals. Inspired by these works, we propose a global-local mixing (GLMix) block, which uses Convs and MHSAs *at different granularities* for different roles: while Convs focus on extracting local features, MHSAs focus on learning global inter-object relations. Specifically, in each block, we represent an image as both a fine-grained regular grid and a coarse-grained set of semantic slots, and then apply Convs to the grid and MHSAs to the slots in parallel. To enable local-global feature fusion, we introduce a pair of conjugated soft clustering and dispatching modules to bridge the grid and set representations. In this way, we achieve highly efficient local-global modeling by using lightweight Convs to extract high-resolution features and heavy MSHAs to process a fixed number of semantic slots.

To verify the performance of the proposed integration scheme for Convs and MHSAs, referred to as *GLMix*, we start by building a **S**win-**T**iny-**L**ayout model, referred to as GLNet-**STL**, based on the GLMix blocks. GLNet-STL achieves 82.5% top-1 accuracy on ImageNet-1k. It surpasses Swin-T (81.3% top-1 accuracy) significantly by 1.2%. Besides, we note that the macro architectural designs are also important factors for the performance of vision backbones. For example, PoolFormer [64] and ConvNext [36] reveal that with a deeper architecture, vision backbones can still achieve strong performances with simple token mixers such as average pooling and depth-wise convolution. Hence, we further adopt several macro designs from recent state-of-the-art vision backbones [34, 43] and scale the model up to derive a family of 3 models: GLNet-4G/9G/16G. As a result, the GLNet-4G/9G/16G models achieve 83.7%/84.5%/85.0% top-1 accuracy, while being more efficient than recent state-of-the-art works (as shown in Figure 2). Evaluations on downstream dense prediction tasks such as object detection, instance segmentation, and semantic segmentation demonstrate the strengths of GLNet consistently. We also observe that a meaningful semantic grouping effect has emerged in the soft clustering module, even with only image-level classification supervision.

Here, we refer to the pixels on the feature maps instead of the input image.

To summarize, our contributions are three-fold:

- We revisit existing integration approaches for Convs and MHSAs, and propose to integrate the two operations *at different granularities*. Such integration combines the strengths of Convs (*e.g.*, the inductive bias of translation equivariance) and MSHAs (*e.g.*, global interactions, data adaptivity) while avoiding the scalability issue w.r.t. the input resolution.
- We introduce a pair of conjugated, fully differentiable clustering and dispatching modules to bridge the set and grid representations of image features, hence enabling the fusion of the global features extracted by MHSAs and local features extracted by Convs. An advantage of the soft clustering module is that it produces meaningful semantic grouping effects without direct dense supervision.
- Through extensive experiments on various computer vision tasks, such as image classification, object detection, and instance/semantic segmentation, we empirically verify our proposed approach. Specifically, a new family of vision backbones, GLNet, demonstrates a favorable performance-computation trade-off to existing state-of-the-arts, under ImageNet-1k supervised training.

2 Related Works

Efficient Attention Mechanisms. Vanilla MHSAs [48] have a quadratic complexity w.r.t. the number of input tokens, causing huge computation burden and heavy memory footprints, especially in vision applications where the feature maps are in high-resolution. A large volume of works have been conducted to develop efficient variants to overcome such a limitation. These works can be roughly categorized as sparse approximations [6, 35, 70], low-rank approximations [50, 51], and kernel-based methods [27, 7, 19]. The global branch in the proposed GLMix block, which is a combination of soft clustering and dispatching modules and an MHSA, can be used independently as a low-rank attention approximation with not only key-value pairs but also queries being down-sampled. However, according to our experiments, such a usage produces poor performance (Table 6), possibly due to losing too many details and the lack of inductive bias. We find that using MSHAs and Convs in a complementary way is crucial to the success of our proposed GLNet family.

Hybrid Vision Backbones. Many works indicate that hybrid vision backbones, which use both Convs and MHSAs, can achieve better performances than pure transformers and CNNs. Among these works, some of them use Convs and MHSAs alternatively across different blocks or stages [34, 29, 47, 59, 12], forming a loose collaboration between the two operators. Another approach adopted by several recent state-of-the-art works [14, 43, 19, 18, 55] is to integrate Convs and MHSAs in each block tightly. Different from these works that apply Convs and MHSAs at the same granularity level, we find that by offloading the burden of extracting fine-grained and location-preserving features to lightweight depth-wise Convs, MHSAs can be applied aggressively on coarse semantic slots while achieving compelling performances with higher efficiency.

Clustering for Representation Learning. Clustering is a type of unsupervised learning method used to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples. Existing works, such as [58, 66, 30, 17, 49], have explored clustering for representation learning in deep neural networks. However, unlike ClusTR [58] and TCFormer [66], which use DPC-KNN [16] for the clustering, the soft clustering module in our work is fully learnable and does not rely on predefined rules. In comparison with ClusterFormer [30] and PaCaViT [17], which perform cross-attention between the feature grid and cluster representations/slots, our work performs self-attention over the slots (*i.e.*, queries and key-value pairs are both from the slots), making the attention even more lightweight. Besides, our soft clustering module is hardware-efficient because it is designed to be non-iterative and mainly involves a dense matrix multiplication.

3 Methodology

Modern vision backbones are usually built by alternating spatial modeling modules (*e.g.*, Convs, MHSAs, spatial MLPs) and per-location feed-forward networks (FFNs, *i.e.*, embedding MLPs). Much research has been dedicated to developing spatial modeling modules, which is also the primary focus of this work. Specifically, we seek an integration scheme for Convs and MHSAs, which can utilize the strengths of both and scale to high-resolution inputs. Without modifying the standard design of the two basic operators, our key idea is to represent input features twice as both a regular feature grid and a set of *semantic slots*, and then process the feature grid with Convs and the semantic slots with MHSAs (Sec. 3.1). A pair of fully differentiable soft clustering and dispatching modules is

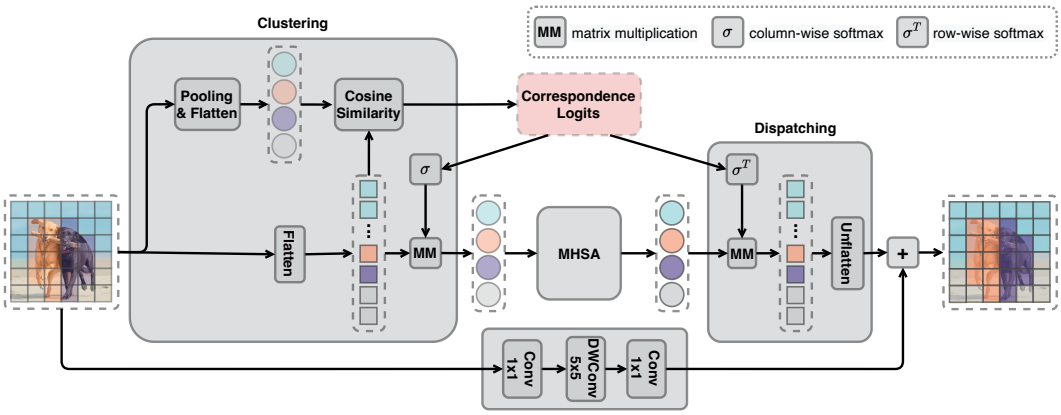

Figure 4: Structure of our GLMix block. At the core is a pair of conjugated soft clustering and dispatching modules to bridge the set and grid representations and enable local-global fusion.

introduced to bridge the two representations, enabling local-global fusion (Sec. 3.2). Based on such an integration scheme we propose a new family of vision backbones named GLNet (Sec. 3.3).

3.1 Convs and MHSAs at Different Granularities

Image features are usually organized as a regular grid in vision backbones. Such a representation preserves the spatial correspondence between features and the input image, which is necessary for downstream dense prediction tasks (*e.g.*, semantic segmentation). Besides, extracting local features with the grid representation is convenient and efficient.

In addition to the grid representation, we also create an intermediate set representation composed of a fixed number of *semantic slots* to enable efficient global context modeling. The reason is that although global interactions are usually expensive to compute, it is feasible to use a small amount (*e.g.*, 64 in our experiment) of semantic slots to summarize an image [1, 63], as images are natural signals with heavy spatial redundancy [22]. Notably, the set of semantic slots that we use here is different from the sequence of visual tokens in plain ViTs [15, 46]. While each visual token corresponds to a hard-divided regular patch (*e.g.*, 16×16 pixels), semantic slots are an abstraction of some "soft" irregular semantic regions, as shown in Figure 3.

We apply Convs to the grid representation to extract local features as they are lightweight and thus efficient in processing the fine-grained feature grid. To model global context, we apply MHSAs on semantic slots. This is a natural choice as MHSAs are permutation-equivariant operators, thus naturally suitable for the set representation. The scalability issue w.r.t. input resolution is avoided, as we have only a small number of semantic slots. Hence, the drawback of MHSAs is overcome.

Next, we illustrate how the set and grid representations are bridged by a pair of soft clustering and dispatching modules, so that local and global features can be fused.

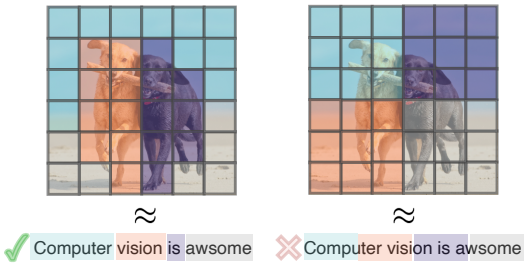

Figure 3: The semantic slots correspond to "soft" irregular semantic regions (left). Compared to using hard-divided regular patches (as adopted by plain ViTs [15, 46]) on the right, our formulation is closer to tokenization in NLP.

3.2 Bridging The Set and Grid Representations

To establish a connection between coarse-grained semantic slots and fine-grained feature grids, we need to create a correspondence between them. Although the classical k-means clustering is applicable for this purpose [49, 66], it is suboptimal for two reasons. First, it is an iterative algorithm, which is inefficient on GPUs. Second, it is a heuristic approach, which cannot be end-to-end optimized. Hence,

Table 1: System-to-system comparison with existing works under the **S**win-**T**iny-**L**ayout protocol [70]. †: implemented by us with modules provided by timm [54].

Models	Params (M)	FLOPs (G)	Throu. (im/s)	IN1K Top-1 (%)
Swin-T [35]	29	4.5	755.2	81.3
DAT-T [56]	29	4.6	—	82.0
Swin-ACMix-T [40]	30	4.6	—	81.9
Flatten-Swin-T [19]	29	4.5	—	82.1
NAT-STL [20]	29	4.5	—	81.4
MaxViT-STL† [47]	28	4.5	763.4	81.8
GLNet-STL (ours)	30	4.4	835.9	**82.5**

Table 2: Model configurations of the GLNet family. C: base channels (*i.e.*, feature channels of the first stage). e: FFN expansion ratio. #Blocks: the 4-stage block numbers. FLOPs are measured at 224×224 resolution.

Model	C	e	#Blocks	Adv. designs	FLOPs
GLNet-STL	96	4	[2, 2, 6, 2]	No	4.4G
GLNet-4G	64	3	[4, 4, 18, 4]	Yes	4.5G
GLNet-9G	96	3	[4, 4, 18, 4]	Yes	9.7G
GLNet-16G	128	3	[4, 4, 18, 4]	Yes	16.7G

we introduce a simplified and fully differentiable clustering module and the conjugated dispatching module to address these two problems, as shown in Figure 4. We illustrate the process below.

Clustering (feature grid → semantic slots). Given an input feature grid, $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, where C is the number of channels, H is the height, and W is the width, we first initialize M semantic slots, $\mathbf{S}_{init} \in \mathbb{R}^{M\times C}$, via average pooling followed by shape flattening, as:

$$\mathbf{S}_{init} = \text{Flatten}(\text{AvgPool}(\mathbf{X})). \tag{1}$$

We then compute the correspondence logits, $\mathbf{A} \in \mathbb{R}^{M\times HW}$, as scaled cosine similarity between the initial semantic slots $\mathbf{S}_{init}$ and the flattened input features $\bar{\mathbf{X}} \in \mathbb{R}^{HW\times C}$, as:

$$\mathbf{A} = \text{CosineSimilarity}(\mathbf{S}_{init}, \bar{\mathbf{X}})/\sigma. \tag{2}$$

Here, the learnable scale factor σ smooths the distribution of $\mathbf{A}$, preventing dominance by salient entrances. With the correspondence logits, we perform a 1-step update to derive refined semantic slots, $\mathbf{S} \in \mathbb{R}^{M\times C}$, as the weighted sum of flattened features $\bar{\mathbf{X}}$, as:

$$\mathbf{S} = \text{Softmax}(\mathbf{A})\bar{\mathbf{X}}. \tag{3}$$

The refined semantic slots $\mathbf{S}$ are then fed to MHSAs as input.

Dispatching (semantic slots → feature grid). After transforming $\mathbf{S}$ to propagate global context with an MHSA module, the transformed semantic slots $\mathbf{S}'$ are dispatched to spatial locations for fusion with local features. Specifically, the dispatched features, $\mathbf{G} \in \mathbb{R}^{C\times H\times W}$, are computed as:

$$\mathbf{G} = \text{Unflatten}(\text{Softmax}(\mathbf{A}^T)\mathbf{S}'). \tag{4}$$

$\mathbf{G}$ can be readily fused with the feature grid processed by Convs due to shape compatibility. We follow the Feature Pyramid Network [32] to use additive fusion, as it is simple/lightweight and provides a regularization that aligns the global and local features in the same semantic space.

Discussion. The soft clustering and dispatching operations are highly efficient as the main computations lie in hardware-friendly dense matrix multiplications, and we perform only 1-step instead of iterative updates of the semantic slots. They are fully differentiable as we do not use hard assignments like k-means. The combination is similar to soft-routing in SoftMoE [42], which aims to build large mixture-of-expert models. However, as we target a better balance between cost/performance, our design has several differences: (1) slots are initialized with a different strategy (*i.e.*, per-image average pooling instead of learned parameters shared by all images); (2) the clustering module is placed at a different position (*i.e.*, in pair with token mixers instead of FFNs); and (3) significantly fewer slots are used (*i.e.*, 64, instead of 4096 which is even more than the number of tokens for an image).

3.3 GLNet

To verify the performance of the GLMix block proposed above, we start by creating a **S**win-**T**iny-**L**ayout architecture, named GLNet-**STL**, which follows the macro architectural designs of the Swin-T [35] model but with the spatial mixing modules (window attention and shift-window attention) replaced. Specifically, we use the GLMix block in the first three stages of GLNet-STL, where the feature maps are in high-resolution ($\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ of the input resolution). At the 4^{th} stage, which is

Table 3: Comparison with different state-of-the-art backbones on ImageNet-1K classification. All models are trained and evaluated on 224×224 input resolution. Top-1 refers to top-1 accuracy (%). We compare models trained with a standard supervised recipe and those trained with an advanced distillation recipe [26].

Method		FLOPs	#Param.	Top-1
Standard supervised training				
Swin-T [35]	[ICCV'21]	4.5G	29M	81.3
PVT-S [51]	[ICCV'21]	3.8G	25M	79.8
CSWin-T [14]	[CVPR'22]	4.5G	23M	82.7
CMT-S [18]	[CVPR'22]	4.0G	25M	83.5
RegionViT-S [3]	[ICLR'22]	5.7G	31M	83.3
CrossFormer-S [53]	[ICLR'23]	5.3G	31M	82.5
MaxViT-T [47]	[ECCV'22]	5.6G	31M	83.6
MOAT-0 [59]	[ICLR'23]	5.7G	28M	83.3
NAT-T [20]	[CVPR'23]	4.3G	28M	83.2
InternImage-T [52]	[CVPR'23]	5G	30M	83.5
Flatten-T [19]	[ICCV'23]	4.3G	21M	83.1
SG-Former-S [43]	[ICCV'23]	4.8G	23M	83.2
GLNet-4G (ours)		4.5G	27M	**83.7**
Swin-S [35]	[ICCV'21]	8.7G	50M	83.0
CSwin-S [14]	[CVPR'22]	6.9G	35M	83.6
RegionViT-M [3]	[ICLR'22]	7.9G	42M	83.4
MaxViT-S [47]	[ECCV'23]	11.7G	69M	84.4
MOAT-1 [59]	[ICLR'23]	9.1G	42M	84.2
NAT-S [20]	[CVPR'23]	7.8G	51M	83.7
InternImage-S [52]	[CVPR'23]	8G	50M	84.2
BiFormer-B [70]	[CVPR'23]	9.8G	57M	84.3
Flatten-S [19]	[ICCV'23]	6.9G	35M	83.8
SG-Former-M [43]	[ICCV'23]	7.5G	39M	84.1
SMT-B [34]	[ICCV'23]	7.7G	32M	84.3
GLNet-9G (ours)		9.7G	61M	**84.5**

Method		FLOPs	#Param.	Top-1
Standard supervised training				
Swin-B [35]	[ICCV'21]	15.4G	88M	83.5
CrossFormer-L [53]	[ICLR'22]	16.1G	92M	84.0
CSWin-B [14]	[CVPR'22]	15.0G	78M	84.2
CMT-L [18]	[CVPR'22]	19.5G	75M	84.8
RegionViT-B [3]	[ICLR'22]	13.6G	74M	83.8
MaxViT-B [47]	[ECCV'23]	23.4G	120M	84.9
MOAT-2 [59]	[ICLR'23]	17.2G	73M	84.7
NAT-B [20]	[CVPR'23]	13.7G	90M	84.3
InternImage-B [52]	[CVPR'23]	16G	97M	84.9
Flatten-B [19]	[ICCV'23]	15.0G	75M	84.5
SG-Former-B [43]	[ICCV'23]	15.6G	78M	84.7
GLNet-16G (ours)		16.7G	106M	**85.0**
Trained with distillation supervision				
Uniformer-S* [29]	[ICLR'22]	4.2G	24M	83.4
Wave-ViT-S* [61]	[ECCV'22]	4.7G	23M	83.9
DualViT-S* [62]	[TPAMI'23]	5.4G	25M	84.1
BiFormer-S* [70]	[CVPR'23]	4.5G	26M	84.3
GLNet-4G* (ours)		4.5G	27M	**84.4**
Uniformer-B* [29]	[ICLR'22]	8.3G	50M	85.1
Wave-ViT-B* [61]	[ECCV'22]	7.2G	34M	84.8
DualViT-B* [62]	[TPAMI'23]	9.3G	43M	85.2
BiFormer-B* [70]	[CVPR'23]	9.8G	58M	**85.4**
GLNet-9G* (ours)		9.7G	61M	85.3

$\frac{1}{32}$ of the input resolution, we use full attention because this is affordable, and beneficial for the performance [29, 41]. As shown in Table 1, our GLNet-STL achieves competitive 82.5% Top-1 accuracy at the highest throughput of 835.9 im/s among several comparable architectures.

The compelling performance of GLNet-STL encourages us to build stronger vision backbones based on it. We therefore investigate several recent state-of-the-arts [14, 47, 70, 43, 34], and incorporate the following advanced architectural designs adopted by them: **(1) Overlapped patch embedding**: use overlapped convolutions (3×3 Conv with stride 2) for image/feature down-sampling, instead of non-overlapped ones (2×2 Conv with stride 2) as in Swin-Transformer; **(2) Hybrid stage 3**: alternate full MHSAs and GLMix in consecutive blocks of stage 3; **(3) Convolutional position encoding**: add a 3×3 residual depth-wise convolution prior to each spatial mixing block; **(4) Deeper layout**: increase the depth ([2, 2, 6, 2] $\rightarrow$ [4, 4, 18, 4]) while reducing the width (base channel 96 $\rightarrow$ 64 and FFN expansion ratio 4 $\rightarrow$ 3); and **(5) Convlutional FFN**: add a 3×3 residual depth-wise convolution between the two linear projections of FFN.

Note that all designs above are widely used in the vision transformers. In addition, as this work is mainly to propose an effective as well as *efficient* integration scheme for MHSAs and Convs, we do not further incorporate some possibly useful designs, such as the squeeze-and-excitation (SE) block [23] used by MaxViT [47] and the gated linear unit (GLU) [44] used by SMT [34]. An ablation study for the incorporated architecture designs can be found in the Supplemental. After applying these modifications sequentially to GLNet-STL, we derive GLNet-4G, a model with 4.5G FLOPs. We scale it up to GLNet-9G and GLNet-16G FLOPs by increasing the number of base channels (96 for GLNet-9G and 128 for GLNet-16G). The model specifications are summarized in Table 2.

4 Experiments

We first empirically evaluate our proposed GLNet on a series of computer vision tasks, including ImageNet-1k [13] image classification (Sec. 4.1), COCO [31] object detection and instance segmentation (Sec. 4.2), and ADE20k [69] semantic segmentation (Sec. 4.3). Following existing works, we first train the models for image classification from scratch and then use the trained weights

For 224×224 input image used in IN1k classification, full attention at stage 4 is equivalent to 7×7 window attention used by Swin-Transformer.

Table 4: Results on the COCO 2017 dataset using the RetinaNet [33] framework for object detection, and Mask R-CNN [21] framework for instance segmentation. $1\times$ refers to 12 epochs, and $3\times$ refers to 36 epochs. MS means multi-scale training. mAP^b and mAP^m denote box mAP and mask mAP, respectively. FLOPs are measured at 800×1280 resolution.

Backbone	#Param.	FLOPs	RetinaNet 1×						RetinaNet 3× + MS					
	(M)	(G)	mAP^b	AP$^b_{50}$	AP$^b_{75}$	AP^b_S	AP^b_M	AP^b_L	mAP^b	AP$^b_{50}$	AP$^b_{75}$	AP^b_S	AP^b_M	AP^b_L
PVT-Small [51]	34	226	40.4	61.3	43.0	25.0	42.9	55.7	42.2	62.7	45.0	26.2	45.2	57.2
Swin-T [35]	39	245	41.5	62.1	44.2	25.1	44.9	55.5	43.9	64.8	47.1	28.4	47.2	57.8
Twins-SVT-S [8]	34	210	43.0	64.2	46.3	28.0	46.4	57.5	45.6	67.1	48.6	29.8	49.3	60.0
CrossFormer-S [53]	41	272	44.4	65.8	47.4	28.2	48.4	59.4	—	—	—	—	—	—
MaxViT-T [47]	46	263	44.7	66.3	47.7	28.0	48.3	58.9	—	—	—	—	—	—
BiFormer-S [70]	35	243	45.9	66.9	49.4	30.2	49.6	61.7	—	—	—	—	—	—
SMT-S [34]	30	247	—	—	—	—	—	—	47.3	67.8	50.5	32.5	51.1	62.3
GLNet-4G (ours)	37	214	**47.1**	**68.6**	**50.5**	**30.8**	**51.1**	**62.9**	**47.9**	**68.8**	**50.8**	**32.7**	**51.6**	**63.5**
Swin-S [35]	60	335	44.5	65.7	47.5	27.4	48.0	59.9	46.3	67.4	49.8	31.1	50.3	60.9
Twins-SVT-B [8]	67	326	45.3	66.7	48.1	28.5	48.9	60.6	46.9	68.0	50.2	31.7	50.3	61.8
CrossFormer-B [53]	62	389	46.2	67.8	49.5	30.1	49.9	61.8	—	—	—	—	—	—
ScalableViT-B [60]	85	330	45.8	67.3	49.2	29.9	49.5	61.0	48.0	69.3	51.4	32.8	51.6	62.4
MaxViT-S [47]	79	389	46.1	68.0	49.5	28.9	50.2	61.4	—	—	—	—	—	—
BiFormer-B [70]	67	356	47.1	68.5	50.4	31.3	50.8	62.6	—	—	—	—	—	—
GLNet-9G (ours)	70	292	**47.7**	**69.0**	**51.6**	**31.8**	**51.6**	**63.5**	**48.8**	**69.6**	**52.5**	**33.5**	**52.9**	**63.9**

Backbone	#Param.	FLOPs	Mask R-CNN 1×						Mask R-CNN 3× + MS					
	(M)	(G)	mAP^b	AP$^b_{50}$	AP$^b_{75}$	mAP^m	AP$^m_{50}$	AP$^m_{75}$	mAP^b	AP$^b_{50}$	AP$^b_{75}$	mAP^m	AP$^m_{50}$	AP$^m_{75}$
Swin-T [35]	47.8	264	42.2	64.6	46.2	39.1	61.6	42.0	46.0	68.2	50.2	41.6	65.1	44.8
Twins-SVT-S [8]	44.0	248	43.4	66.0	47.3	40.3	63.2	43.4	46.8	69.2	51.2	42.6	66.3	45.8
CSWin-T [14]	42	279	46.7	68.6	51.3	42.2	65.6	45.4	49.0	70.7	53.7	43.6	67.9	46.6
BiFormer-S [70]	45.2	262	47.8	69.8	52.3	43.2	66.8	46.5	—	—	—	—	—	—
SGFormer-S [43]	41	—	47.4	69.0	52.0	42.6	65.9	46.0	49.6	71.1	54.5	44.0	68.3	46.9
SMT-S [34]	40.0	265	47.8	69.5	52.1	43.0	66.6	46.1	49.0	70.1	53.4	43.4	67.3	46.7
GLNet-4G (ours)	46.6	233	**48.3**	**70.3**	**53.3**	**43.6**	**67.3**	**46.9**	**49.9**	**71.6**	**54.7**	**44.5**	**68.8**	**48.1**
Swin-S [35]	69.1	354	44.8	66.6	48.9	40.9	63.4	44.2	48.5	70.2	53.5	43.3	67.3	46.6
CrossFormer-B [53]	72	408	47.2	69.9	51.8	42.7	66.6	46.2	—	—	—	—	—	—
CSWin-S [14]	54	342	47.9	70.1	52.6	43.2	67.1	46.2	50.0	71.3	54.7	44.5	68.4	47.7
BiFormer-B [70]	76.3	375	48.6	70.5	53.8	43.7	67.6	47.1	—	—	—	—	—	—
SGFormer-M [43]	51	—	48.2	70.3	53.1	43.6	66.9	47.0	50.5	71.5	54.9	45.4	68.8	48.2
SMT-B [34]	51.7	328	49.0	70.2	53.7	44.0	67.6	47.4	49.8	71.0	54.4	44.0	68.0	47.3
GLNet-9G (ours)	79.5	311	**49.5**	**71.4**	**54.5**	**44.5**	**68.5**	**48.0**	**51.0**	**72.0**	**56.1**	**46.2**	**69.5**	**48.7**

for model initialization when performing downstream dense prediction tasks. Note that for dense prediction tasks which take high-resolution inputs, we keep the number of semantic slots to 64, which is consistent with that of image classification. We have found that 64 slots are sufficient to achieve state-of-the-art performances while increasing the number does not help. We then visualize the semantic slots to demonstrate that a meaningful semantic grouping effect emerges in the proposed soft clustering module (Sec. 4.4). Finally, we conduct an ablation study on the design choices of the GLMix integration scheme, which is the core of GLNet (Sec. 4.5).

4.1 Image Classification on ImageNet-1k

Settings. For a fair comparison with existing works, we conduct image classification experiments on the ImageNet-1k dataset [13], using the standard training recipe provided by Swin-Transformer [35] and the advanced distillation recipe provided by LV-ViT [26]. The training details can be found in Appendix B. We then evaluate the models for classification accuracy and benchmark their throughputs with the script provided by the timm library [54], following the same hardware (a single Tesla V100 32G GPU) and batch size (128) configurations used in Swin-Transformer [35].

Results. In Table 3, we compare GLNet with several closely related methods and/or recent state-of-the-arts. Under the setting of standard supervised training, our GLNet-4G/9G/16G consistently show comparable or superior performances to existing best-performing models across different model scales. With dense distillation supervision, the potential of GLNets is further unleashed compared to standard supervised training. For example, the accuracy of the GLNet-4G model increases from 83.7% to 84.4%, a significant performance improvement of 0.7%. Both GLNet-4G and GLNet-9G provide a more competitive performance-FLOPs trade-off than other distilled models. As FLOPs is an indirect metric for practical inference speed and does not consider the memory access cost, we also plot the performance-throughput curve in Figure 2. The improvements become more pronounced when viewed w.r.t. throughputs. Interestingly, several of the latest vision backbones (*i.e.*, SG-Former [43],

MaxiViT [47], and CSwin [14]) lie in almost the same Pareto frontier, while our GLNet models further move the frontier to the upper-right corner with a clear margin. Such a result demonstrates the superiority of our integration philosophy: by applying the heavy MHSAs at a coarse granularity and light-weight Convs at a fine granularity, spatial modeling can be both effective and efficient.

4.2 Object Detection and Instance Segmentation

Settings. We evaluate the backbones for object detection and instance segmentation on COCO 2017 [31]. All experiments are conducted using the MMDetection [4] toolbox to ensure a fair comparison with existing works. The RetinaNet [33] framework is used for object detection, and the Mask R-CNN [21] framework is used for instance segmentation. During training, we initialize the backbone with weights trained on ImageNet-1K while leaving all other layers randomly initialized. Input images are resized by fixing the shorter side to 800 pixels while restricting the longer side to no more than 1,333 pixels. We train the RetinaNet and Mask R-CNN detectors with $1\times$ schedule (12 epochs) and $3\times$ schedule (36 epochs) provided by MMDetection. More training details are provided in Appendix B. We report the widely used average precision (AP) metric family, such as mean average precision (mAP), average precision at different thresholds (AP_{75} and AP_{50}), and average precision for objects of different sizes (AP_S, AP_M and AP_L). Details of these metrics can be found in MMDetection [4].

Results. We show the results for object detection and instance segmentation in Table 4. Our method achieves the best performances among the compared methods across all metrics and the two model sizes in both cases. These results indicate that local-global modeling with the GLMix block benefits object/instance-level tasks.

4.3 Semantic Segmentation on ADE20K

Settings. Our semantic segmentation experiments are conducted on the ADE20K dataset using the MMSegmentation [10] toolbox. We evaluate our approach using two frameworks - Semantic FPN [28] and UperNet [57]. In both cases, the backbone is initialized with ImageNet-1k weights, while the other layers are randomly initialized. For a fair comparison, we follow the same setting as PVT [51] to train the model 80k steps in our Semantic FPN experiments. On the other hand, for our UperNet experiments, we follow the settings used in Swin Transformer [35] and train the model for 160k iterations. More training details are provided in Appendix B. We report the mean intersection over union (mIoU) metric with no test-time augmentation.

Table 5: Performance comparison of different backbones on the ADE20K segmentation task. We report mIoU with no test-time augmentation. FLOPs are computed at 512 $\times$ 2048 resolution.

Backbone	Semantic FPN 80k			UperNet 160k		
	#Param. (M)	FLOPs (G)	mIoU (%)	#Param. (M)	FLOPs (G)	mIoU (%)
PVT-S [51]	28.2	161	39.8	—	—	—
Swin-T [35]	31.9	182	41.5	59.9	945	44.5
Twins-SVT-S [8]	28.3	144	43.2	54.4	901	46.2
CSWin-T [43]	26.1	202	48.2	59.9	959	49.3
BiFormer-S [70]	29.3	173	48.9	55.3	930	49.8
SG-Former-S [43]	25.4	205	49.0	52.5	989	49.9
SMT-S [34]	—	—	—	50.1	935	49.2
GLNet-4G (ours)	30.7	150	**49.6**	56.8	907	**50.6**
PVT-M [51]	48.0	219	41.6	—	—	—
Swin-S [35]	53.2	274	45.2	81.3	1038	47.6
Twins-SVT-B [8]	60.4	261	45.3	88.5	1020	47.7
CSWin-S [14]	38.5	271	49.2	64.6	1027	50.4
BiFormer-B [70]	60.4	282	49.9	88.5	1041	51.0
SG-Former-M [43]	38.2	273	50.1	68.3	1114	51.2
SMT-B [34]	—	—	—	61.8	1004	49.6
GLNet-9G (ours)	63.6	230	**51.3**	91.7	988	**51.4**

Results. Table 5 shows the results of the two different frameworks. Our GLNet-4G/16G achieve 49.6/51.3 mIoU with the Semantic FPN framework, improving the previous best SG-Former-S/M by 0.6/1.2 mIoU. A similar performance gain for the UperNet framework is also observed. The enhancements demonstrate the benefits of utilizing GLNet for high-resolution pixel-wise predictions.

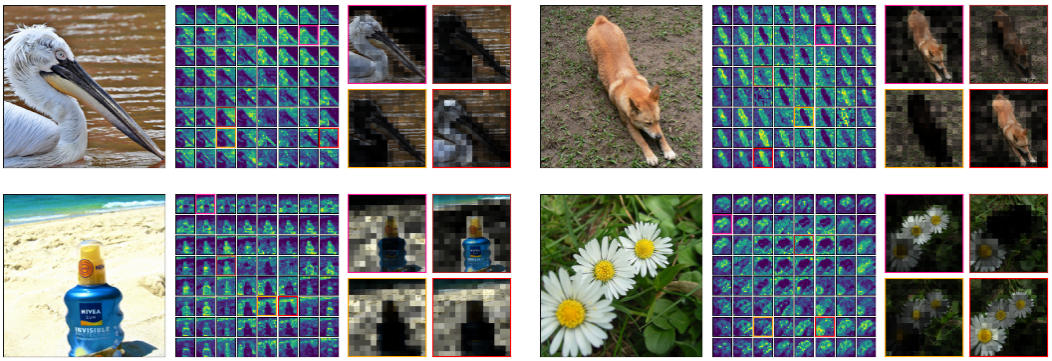

Figure 5: Visualization of semantic slots. For each sample, we show the input image (left), assignment maps of all semantic slots (middle), and four representative slots (right). We use the k-medoids algorithm to select the four representative slots automatically.

4.4 Visualization of Semantic Slots

As mentioned in Sec. 3.2, the conjugated clustering and dispatching modules construct a correspondence between the semantic slots and the feature grid. Such a formulation allows us to visualize which regions the semantic slots correspond to. Specifically, we extract the clustering weights in Eq. 3 and split them into M scalar maps of shape H $\times$ W. These scalar maps are then pseudo-colored for visualization. In addition, we use the k-medoids algorithm to select four representative slots for a closer look automatically. We find that a meaningful semantic grouping effect emerges in the first block of stage 3, as shown in Figure 5. Note that we use ImageNet-1k trained GLNet-STL for visualization. Hence, the model receives no dense supervision. Visualization for more samples, more blocks and at different epochs can be found in Appendix C.

4.5 Ablation Study

We ablate our GLMix integration scheme using the GLNet-STL model. By default, we use a global branch with 64 semantic slots and a local branch with 5×5 depth-wise conv in parallel in the GLMix blocks, as shown in Figure 4. With this default setting, we investigate the effect of **(a)** local-global collaboration, **(b)** the clustering strategy, **(c)** Conv kernel size in the local branch, and **(d)** number of slots in the global branch. Table 6 shows the experimental results. We summarize our findings below.

Local-global collaboration. First, using both local and global branches together is crucial. With the global/local branch removed, the model has a significantly degraded accuracy of 81.8%/78.0%, indicating that both coarse-grained inter-object relationship and fine-grained per-pixel local context are important. Second, using global and local branches in parallel instead of sequentially is important. A possible explanation is that the parallel layout provides a regularization for the global branch from the local branch. Otherwise, the global branch is difficult to optimize due to the lack of inductive bias. Finally, using Convs in the local branch is better than window MHSAs, as the latter are heavier and significantly decrease the throughput from 835.9 im/s to 660.9 im/s. This may be because Convs can implicitly bring position information via padding [25, 9] while window MHSAs cannot.

Clustering strategy. The soft clustering approach is an important component of the GLMix block. Using the k-means clustering results do not only produce a significantly lower throughput (835.9 im/s $\rightarrow$ 440.6 im/s) but also incurs unstable training. This can be attributed to the fact that k-means is an iterative, non-differentiable algorithm, as mentioned in Sec. 3.2. We also observe that initializing the semantic slots as learnable parameters decreases the accuracy from 82.5% to 82.1%. This implies that per-image adaptive initialization is better than static initialization. Possibly, there are difficulties to learn diverse contexts for each image with shared parameters as the slot initialization, according to visualizations in Appendix C.

Convolution kernel size in the local branch. The model is robust to the convolution kernel size in the local branch. Using a kernel size of 3 or 7 produces a similar accuracy (82.4%) to the kernel size of 5 (82.5%). This is because the global branch has provided a sufficient large receptive field.

Table 6: Ablation study on the GLMix design choices. We investigate the effect of **(a)** local-global collaboration, **(b)** the clustering strategy, **(c)** convolution kernel size of the local branch, and **(d)** number of slots in the global branch. †: W-MHSA stands for window MHSA; we use a window size of 7 because size divisibility is required. ‡: It is implemented with the official release of Clustered Attention [49], NaN loss occurred during training.

Model	Slot init.	Slot number	Conv k.s.	FLOPs (G)	Params (M)	Throu. (im/s)	IN1k Top-1 (%)
GLNet-STL	pooling	64	5	4.4	30.3	835.9	82.5
local branch only	pooling	-	5	3.8	26.4	999.7	81.8
global branch only	pooling	64	-	3.8	28.3	982.4	78.0
sequential (global $\rightarrow$ local)	pooling	64	5	4.4	30.3	860.1	80.6
sequential (local $\rightarrow$ global)	pooling	64	5	4.4	30.3	825.9	79.6
local branch w/ W-MHSA†	pooling	64	w7	5.0	32.2	660.9	81.1
k-means clustering‡	hashing	64	5	5.2	30.3	440.6	N/A
static slot initialization	param.	64	5	4.4	30.5	852.0	82.1
local w/ 7×7 DWConv	pooling	64	7	4.4	30.3	855.2	82.4
local w/ 3×3 DWConv	pooling	64	3	4.3	30.4	823.9	82.4
global w/ 9 slots	pooling	9	5	3.9	30.3	893.6	81.9
global w/ 25 slots	pooling	25	5	4.0	30.3	880.8	82.1
global w/ 36 slots	pooling	36	5	4.1	30.3	880.0	82.3
global w/ 49 slots	pooling	49	5	4.2	30.3	866.6	82.3
global w/ 81 slots	pooling	81	5	4.5	30.3	790.0	82.4

Number of semantic slots in the global branch. Using 64 semantic slots is sufficient to achieve a good performance. Although the accuracy decreases to 82.3% with fewer semantic slots (*e.g.*, 9, 25, 36 or 49), increasing the number to 81 also incurs a small performance drop to 82.4%. We hypothesize that this is due to the optimization difficulty caused by too many similar/near-duplicate slots [68].

5 Conclusion

In this paper, we have revisited the existing integration approaches for Convs and MHSAs, and proposed to apply the two operators at *different granularity levels*. We discover that by offloading the task of extracting fine-grained features to the lightweight Convs, the heavy MHSAs can be aggressively applied to a few semantic slots. Such an integration scheme, named GLMix, enables highly efficient local-global modeling to build high-performance vision backbones. A key component of GLMix is a pair of conjugated soft clustering and dispatching modules for bridging the feature grid and the set of semantic slots. Meaningful semantic grouping effects, which may induce better interpretability and inspire new weakly-supervised semantic segmentation approaches, are observed in the clustering process.

Currently, we only consider using a static number of semantic slots (*i.e.*, 64 in our experiments) for all images. This may cause many redundant slots representing the same content, as shown in Figure 5. It may be interesting to design a dynamic slot pruning mechanism for more efficient computation and end-to-end weakly-supervised segmentation. Another drawback of GLNet is that it still incorporates many hardware-inefficient depth-wise convolutions with low arithmetic intensity. Seeking more hardware-friendly alternatives will further improve its throughputs on modern hardware.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[2] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pages 1059–1071.

PMLR, 2021.

[3] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. In *International Conference on Learning Representations*. OpenReview.net, 2022. URL https://openreview.net/forum?id=T__V3uLix7V.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

[5] Qiang Chen, Qiman Wu, Jian Wang, Qinghao Hu, Tao Hu, Errui Ding, Jian Cheng, and Jingdong Wang. Mixformer: Mixing features across windows and dimensions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5249–5259, 2022.

[6] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv:1904.10509*, 2019.

[7] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[8] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021.

[9] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. In *International Conference on Learning Representations*, 2023.

[10] MMSegmentation Contributors. Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

[11] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops*, pages 702–703, 2020.

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

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[14] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[16] Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems*, 99:135–145, 2016.

[17] Ryan Grainger, Thomas Paniagua, Xi Song, Naresh Cuntoor, Mun Wai Lee, and Tianfu Wu. Paca-vit: learning patch-to-cluster attention in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18568–18578, 2023.

[18] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, 2022.

[19] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5961–5971, 2023.

[20] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6185–6194, 2023.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[23] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[24] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. In *Proceedings of the European conference on computer vision*, pages 646–661, 2016.

[25] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? *arXiv preprint arXiv:2001.08248*, 2020.

[26] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in Neural Information Processing Systems*, 34:18590–18602, 2021.

[27] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[28] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[29] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[30] James Liang, Yiming Cui, Qifan Wang, Tong Geng, Wenguan Wang, and Dongfang Liu. Clusterfomer: clustering as a universal visual learner. *Advances in neural information processing systems*, 36, 2024.

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European conference on computer vision*, pages 740–755. Springer, 2014.

[32] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[34] Weifeng Lin, Ziheng Wu, Jiayu Chen, Jun Huang, and Lianwen Jin. Scale-aware modulation meet transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6015–6026, 2023.

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

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

[37] I Loshchilov and F Hutter. Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, pages 1–16, 2017.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[39] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

[40] Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. On the integration of self-attention and convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–825, 2022.

[41] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=D78Go4hVcx0`.

[42] Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*, 2023.

[43] Sucheng Ren, Xingyi Yang, Songhua Liu, and Xinchao Wang. Sg-former: Self-guided transformer with evolving token reallocation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6003–6014, 2023.

[44] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[45] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[46] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[47] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Proceedings of the European conference on computer vision*, 2022.

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

[49] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. *Advances in Neural Information Processing Systems*, 33:21665–21674, 2020.

[50] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv:2006.04768*, 2020.

[51] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[52] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023.

[53] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=_PHymLIxuI`.

[54] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[55] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021.

[56] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4794–4803, 2022.

[57] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision*, pages 418–434, 2018.

[58] Yutong Xie, Jianpeng Zhang, Yong Xia, Anton van den Hengel, and Qi Wu. Clustr: Exploring efficient self-attention via clustering for vision transformers. *arXiv preprint arXiv:2208.13138*, 2022.

[59] Chenglin Yang, Siyuan Qiao, Qihang Yu, Xiaoding Yuan, Yukun Zhu, Alan Yuille, Hartwig Adam, and Liang-Chieh Chen. Moat: Alternating mobile convolution and attention brings strong vision models. In *International Conference on Learning Representations*, 2023.

[60] Rui Yang, Hailong Ma, Jie Wu, Yansong Tang, Xuefeng Xiao, Min Zheng, and Xiu Li. Scalablevit: Rethinking the context-oriented generalization of vision transformer. In *European Conference on Computer Vision*, pages 480–496. Springer, 2022.

[61] Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. Wave-vit: Unifying wavelet and transformers for visual representation learning. In *Proceedings of the European conference on computer vision*, pages 328–345. Springer, 2022.

[62] Ting Yao, Yehao Li, Yingwei Pan, Yu Wang, Xiao-Ping Zhang, and Tao Mei. Dual vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 2023.

[63] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

[64] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.

[65] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[66] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11101–11111, 2022.

[67] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1Ddp1-Rb`.

[68] Shilong Zhang, Xinjiang Wang, Jiaqi Wang, Jiangmiao Pang, Chengqi Lyu, Wenwei Zhang, Ping Luo, and Kai Chen. Dense distinct query for end-to-end object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7329–7338, 2023.

[69] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

[70] Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson WH Lau. Biformer: Vision transformer with bi-level routing attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10323–10333, 2023.

A Effect of Advanced Architecture Designs

Architecture design	Params (M)	FLOPs (G)	Throu. (im/s)	IN1k Top-1 (%)
Swin-T layout (GLNet-STL)	30.3	4.4	835.9	82.5
+ overlapped patch emb.	32.3	4.7	782.4	82.7
+ hybrid stage 3	31.4	4.8	784.0	83.1
+ convolution pos. enc.	31.4	4.8	762.6	83.2
+ deeper layout	26.8	4.5	630.7	83.5
+ conv. FFN (GLNet-4G)	27.0	4.5	541.2	83.7

Table 7: The evolution path from GLNet-STL to GLNet-4G. Modifications are applied sequentially.

As mentioned in Sec. 3.3, we incorporate several advanced architecture designs adopted by recent vision backbones in our GLNet family to achieve state-of-the-art performance. Here, we list the effects of these designs in Table 7. All these designs improve the accuracy. However, they also decrease the throughput.

B Training Details

This section provides more training details for ImageNet-1k image classification, COCO object detection and instance segmentation, and ADE20K semantic segmentation.

Image classification. For the standard supervised training recipe, training details are in Table 8. When training with the advanced distillation recipe [26], we add an extra distillation head to the GLNet-4G/9G model and use the NFNet-F6 [2] to generate distillation targets; other training details are shown in Table 9. Experiments are run on 16 Tesla V100 SXM2 (32GB) GPUs. Each experiment takes 2-4 days, depending on model size.

Object detection and instance segmentation. For COCO experiments, all models are trained using the AdamW [38] optimizer with a batch size of 16. We use a linear schedule with 500 warm-up iterations and set the peak learning rate as $1e-4$. The weight decay is 0.05 for Mask R-CNN [21] and 0.001 for RetinaNet [33]. Experiments are run on 8 or 16 Tesla V100 SXM2 (32GB) GPUs. Each experiment takes 1-2 days, depending on model size.

Semantic segmentation. For the ADE20K semantic segmentation task, we apply the AdamW optimizer with a batch size of 32. In Semantic FPN [28] experiments, we use the cosine annealing learning rate schedule with 1000 warm-up iterations and a peak learning rate of 2e-4. The weight decay is $1e-4$. In UperNet [57] experiments, a polynomial learning rate schedule is employed with a linear warm-up phase of 1500 iterations. We set the learning rate as $6e-4$ and weight decay as $1e-2$. Experiments are run on 8 Tesla V100 SXM2 (32GB) GPUs. Each experiment takes 1-2 days, depending on model size.

C More Visualization Results

In this section, we provide more visualization results, including (1) visualization of semantic slots of blocks at different depths (Figure 6), (2) visualization of slot evolution over training epochs (Figure 7), and (3) visualization of slots using learned parameters as clustering initialization (Figure 8). We summarize the main observations as below:

- The semantic slots at the lower block (2^{nd} block) tends to group pixels according to color cues. At the middle block (5^{th} block), an object-level grouping effect has emerged. The upper block (10^{th} block) pays attention to discriminative local regions.
- During the training, we found that at the end of the 1^{st} epoch we can already distinguish the foreground objects and the backgrounds, although the grouping has not very concentrated patterns, this is possibly due to the fact that even a random projection can preserve distances/similarities well. At the end of the 5^{th} epoch, the semantic grouping becomes more concentrated and similar to that of the final stage.

Table 8: Training details of standard supervised training for ImageNet-1k classification.

config	value
optimizer	AdamW [38]
learning rate	2e-3
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	2048
learning rate schedule	cosine decay [37]
warmup epochs	5
training epochs	300
augmentation	RandAug(9, 0.5) [11]
label smoothing [45]	0.1
mixup [67]	0.8
cutmix [65]	1.0
gradient clip	5.0
drop path [24]	0.15/0.3/0.4

Table 9: Training details of the advanced distillation recipe [26] for ImageNet-1k classfication.

config	value
optimizer	AdamW
learning rate	2e-3
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	2048
learning rate schedule	cosine decay
warmup epochs	5
training epochs	310
augmentation	RandAug(9, 0.5)
label smoothing	0.1
mixup	0.8
cutmix	1.0
gradient clip	5.0
drop path	0.1

- When the semantic slots are initialized with learned parameters instead of adaptive average pooling as we proposed, the slots are quite more chaotic and tend to focus on foreground objects only, indicating that there are difficulties to learn diverse and global contexts. This possibly accounts for the degraded classification accuracy with such a design (82.5% $\rightarrow$ 82.1%, Table 6).

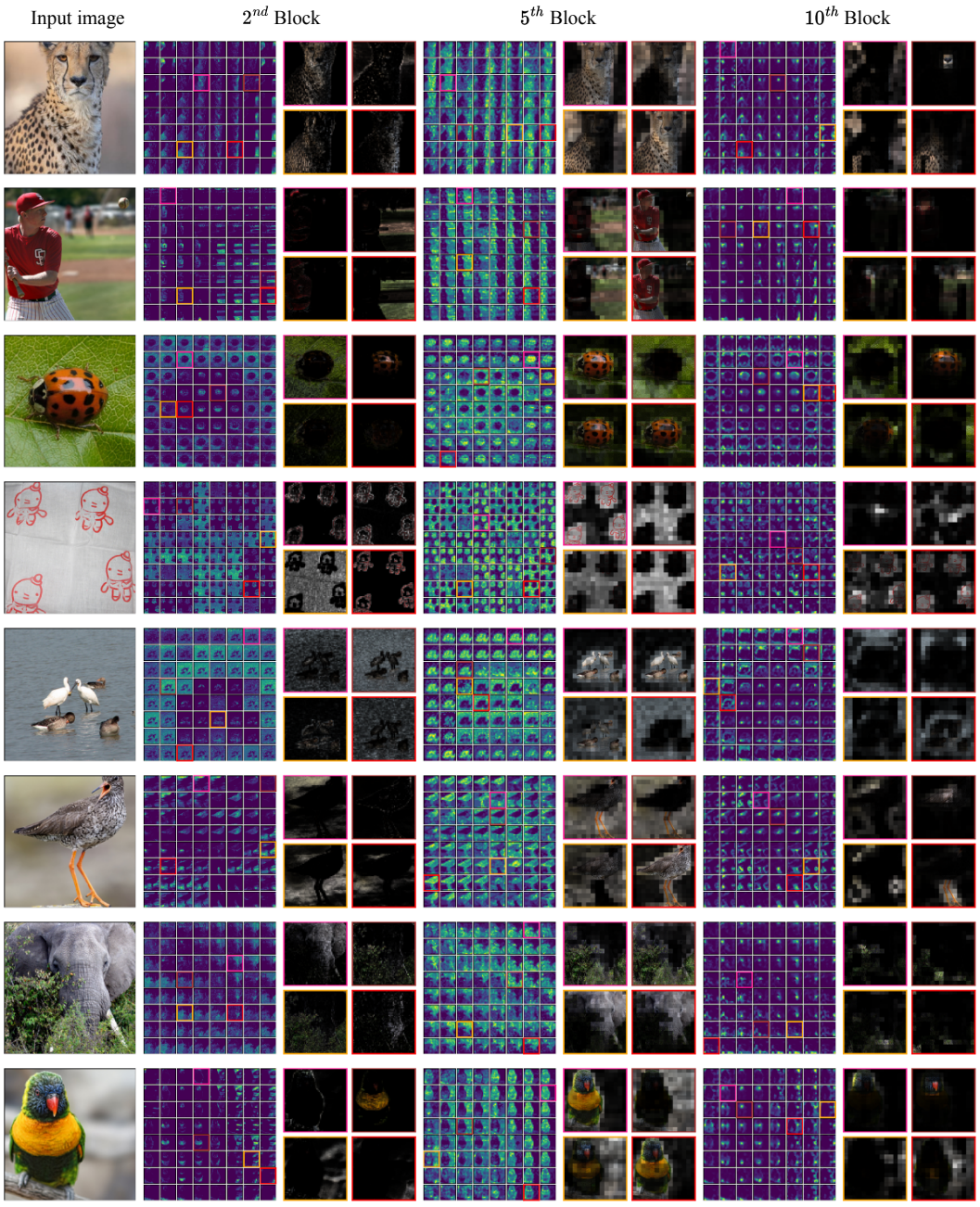

Figure 6: Visualization of semantic slots of blocks at different depths. The setting is the same as in Figure 5, except that we add the visualizations for a shallower block (columns 2-3) and a deeper block (columns 6-7).

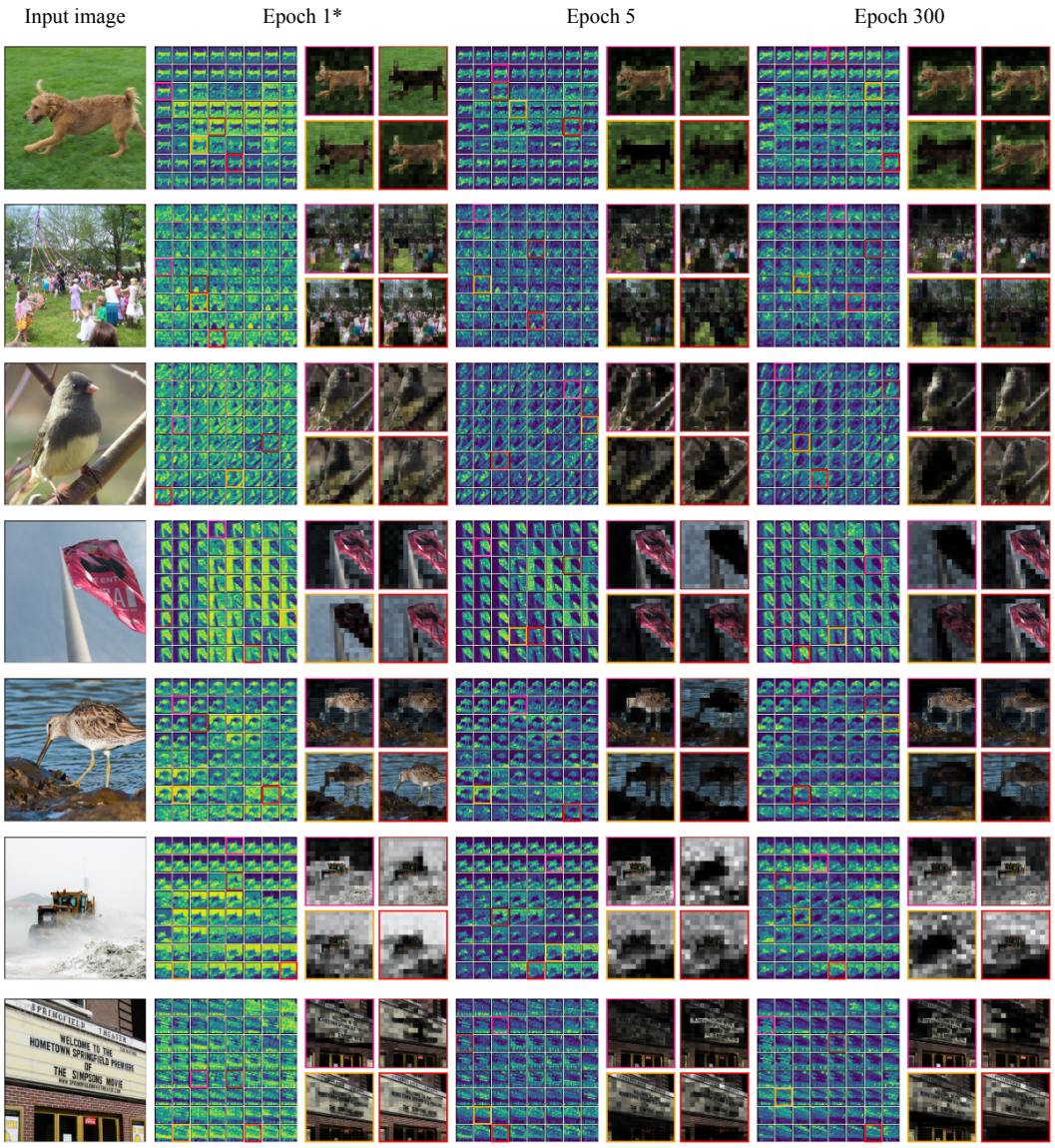

Figure 7: Slot evolution over training epochs. The setting is the same as in Figure 5, except that the checkpoint is replaced. *: slot assignments of epoch 1 are normalized to range $[0, 1]$ for better visibility, otherwise most of them will look like either empty or randomly scattered patterns.

Figure 8: Visualization of slots using learned parameters as clustering initialization. The setting is the same as in Figure 5, except that in the soft clustering initialization is modified.

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: The scope is stated at the beginning of the abstract. The contributions are summarized at the end of the Introduction (Sec. 1).
   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: We mentioned the limitations at the end of the conclusion (Sec. 5).
   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: [NA]
   Justification: The paper does not include theoretical results.
   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: Experimental settings are provided in Sec. 4. More details are given in Appendix (Sec. B).
   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: [No]
   Justification: All datasets used in this paper are existing public ones. We will release the code upon the acceptance of this paper.
   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: Experimental settings are provided in Sec. 4. More details are given in Appendix (Sec. B).
   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: Datasets such as ImageNet-1K, COCO, and ADE20K are usually considered large-scale. The experimental results are not sensitive to random initialization. And running these experiments multiple times is too costly. Following existing works, this paper does not report error bars.
   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: The information on the computer resources is included in the appendix (Sec. B).
   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 does not involve human subjects or participants. We only use existing and publicly available datasets for evaluations.

   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: [NA]

    Justification: This paper focuses on designing general vision backbones. It is too broad to discuss the societal impacts of such a general topic.

    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: This paper focuses on designing general vision backbones. It poses 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: We only use existing and publicly available datasets or tools for evaluations. These works are properly 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: [Yes]
Justification: Our code and documents will be released at `https://github.com/rayleizhu/GLMix`.
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.