Visual Dependency Transformers: Dependency Tree Emerges from Reversed Attention

Mingyu Ding[13*] Yikang Shen[2] Lijie Fan[3] Zhenfang Chen[2]
Zitian Chen[4] Ping Luo[1] Josh Tenenbaum[3] Chuang Gan[24]
[1]The University of Hong Kong [2]MIT-IBM Watson AI Lab [3]MIT [4]UMass Amherst

Abstract

Humans possess a versatile mechanism for extracting structured representations of our visual world. When looking at an image, we can decompose the scene into entities and their parts as well as obtain the dependencies between them. To mimic such capability, we propose Visual Dependency Transformers *(DependencyViT)* [1] *that can induce visual dependencies without any labels. We achieve that with a novel neural operator called* reversed attention *that can naturally capture long-range visual dependencies between image patches. Specifically, we formulate it as a dependency graph where a child token in reversed attention is trained to attend to its parent tokens and send information following a normalized probability distribution rather than gathering information in conventional self-attention. With such a design, hierarchies naturally emerge from reversed attention layers, and a dependency tree is progressively induced from leaf nodes to the root node unsupervisedly.*

DependencyViT offers several appealing benefits. (i) Entities and their parts in an image are represented by different subtrees, enabling part partitioning from dependencies; (ii) Dynamic visual pooling is made possible. The leaf nodes which rarely send messages can be pruned without hindering the model performance, based on which we propose the lightweight DependencyViT-Lite to reduce the computational and memory footprints; (iii) DependencyViT works well on both self- and weakly-supervised pretraining paradigms on ImageNet, and demonstrates its effectiveness on 8 datasets and 5 tasks, such as unsupervised part and saliency segmentation, recognition, and detection.

1. Introduction

Humans have a rich mental representation of our surrounding environments. When looking at an image (see Figure 1(a)), we can recognize the scene and also can quickly decompose it into hierarchical elements with dependencies, *e.g.*, a laptop consisting of a screen and a keyboard is placed on the table. This ability to construct dependencies between objects (and/or their parts) serves as the cornerstone of human intelligence, enabling us to perceive, interact, and reason about the world.

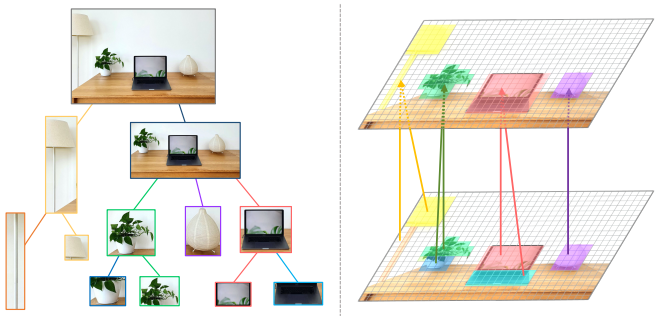

Figure 1. (a) is an example of hierarchical dependency structure. (b) illustrates the dynamic pooling and information aggregation process of DependencyViT.

From the pre-deeplearning era, many classical image dependency parsing algorithms [25, 27, 66, 70, 81, 98] have been proposed. For example, Bayesian framework [70], And-Or graph [27], and hierarchical probabilistic models [25, 66] for parsing images into their constituent visual patterns. Apart from that, Capsule Network [40, 61] shows the potential to learn geometrically organized parts from images. After that, visual grounding methods [10, 18, 21, 23, 85] try to align the semantic meaning between visual objects and words to distill effective structures for the vision branch from language. Similarly, human-object interaction approaches [39] learn the relationships between two objects, *e.g.*, a boy "holds" an ice cream, from manually annotated labels. Such methods struggle to learn hierarchical visual structures, such as different parts of an object, unless exhaustive and time-consuming manual annotations are provided. Recently, vision-language (VL) grammar induction [72] proposes to extract shared hierarchical object

*This work was done when Mingyu was visiting MIT.
[1] https://github.com/dingmyu/DependencyViT

dependencies for both vision and language unsupervisedly from image-caption pairs. However, the above works suffer two key issues: 1) the parsing relies heavily on supervision from natural language or human annotations rather than the image itself, and 2) their parsed structures are object-level based on a pre-trained object detection model, like Faster/Mask-RCNN [30, 58], hindering their generalizability in part-level and non-detector scenarios.

This paper answers a question naturally raised from the above issues: can we efficiently induce visual dependencies and build hierarchies from images without human annotations? Currently, visual parsing works mainly lie in semantic and instance segmentation. Unlike detector-based works that rely on pre-trained detectors, they parse the image at the pixel level, which is resource-intensive and costly. Inspired by vision transformers [22] that take image patches as input and leverage self-attention to perform interactions between patches, we propose to build a dependency tree at the patch level. Taking patches as basic elements and building a tree structure based on them has two benefits: 1) it unifies part-level and object-level dependencies, all of which are formulated into subtrees; 2) in the dependency structure, information can be aggregated from leaves to the parent (as shown in Figure 1(b)) to produce a hierarchy of representations for different parts and object along the path.

In practice, it is non-trivial to build the dependency tree with the standard transformer. Although the self-attention mechanism is designed to collect information dynamically from other patches, the number of attention heads constraints the number of tokens that a patch can attend to. 1) However, each parent could have an arbitrary number of children in a dependency tree, while each child only has one parent. Thus it's more straightforward for a node to select its parent instead of selecting the child. 2) Furthermore, the transformer treats each patch equally, it does not distinguish between root and leaf nodes. Contributions for different subtrees should be distinct.

Motivated by the above observations, in this work, we propose a dependency-inspired vision transformer, named Visual Dependency Transformers (DependencyViT). We propose three innovations to the standard self-attention, as shown in Figure 2. Firstly, to form a root-centric dependency parser, we introduce a reversed self-attention mechanism by transposing the adjacency matrix. In this way, leaf nodes can send information to their parents and form hierarchical subtrees. Secondly, we propose a message controller to determine how a node or subtree sends messages. Thirdly, a soft head selector is introduced to generate a unique dependency graph for each layer. As a result, self-attentions in DependencyViT naturally form a dependency tree parser. We did extensive studies in both supervised and self-supervised pretraining to show DependencyViT is capable of capturing either object- or part-level dependencies.

Intuitively, dependency parsing should ease scene understanding, as humans can understand complex scenes at a glance based on visual dependencies. Based on this, we further introduce a lightweight model DependencyViT-Lite by proposing a dynamic pooling scheme, reducing the computational cost largely. Within each subtree, we prune those leaf nodes with the least information received because they have sent information to their parent node. We show the pruned nodes can be retrieved by soft aggregations from their parents, preserving the model capability and dense representation capability.

We make three main contributions. (i) DependencyViT performs visual dependency parsing by reversed attention in self- or weakly-supervised manners. We demonstrate its effectiveness in both part-level and object-level parsing. (ii) We propose a visual dynamic pooling scheme for DependencyViT hence DependencyViT-Lite. The dependency tree can also be progressively built during the pruning process. (iii) Extensive experiments on both self- and weakly-supervised pretraining on ImageNet, as well as five downstream tasks, show the effectiveness of DependencyViT.

2. Related Work

Dependency Parsing in Vision. Unsupervised dependency parsing is a long-standing task in computer vision with many classical image dependency parsing algorithms that have been proposed in the pre-deeplearning era [25, 27, 66, 70, 81, 98]. Dating back to [70] proposed a Bayesian framework for parsing images into their constituent visual patterns. [27, 81, 98] surveyed on stochastic and context sensitive grammar of images with Bayesian framework, And-Or graph and probabilistic models. [25, 66] proposed to use hierarchical probabilistic models for detection and recognition of objects in cluttered environments.

In the deep learning era, a representative accomplishment is Capsule Network [61], where the activity vector of a capsule represents the instantiation parameters of an object part. After that, Stacked Capsule Autoencoders [40] leverages dynamic routing among capsules to automatically discover sub-patterns and recover the compositional relations on the MNIST dataset [17]. There are also works [24, 34, 37, 82, 83, 88] that further extend the composition relations in Capsule Networks and apply them to more tasks, *e.g.*, generative adversarial scenarios. However, it remains challenging to make them work on complex natural images. Most recently, supervised hierarchical semantic segmentation [44, 48, 49] became more popular. There are works to perform human parsing [76, 77] based on human part relations. Recently there are also attempts to perform part segmentation [4, 13, 36, 51] in an unsupervised manner. [64] explored spectral clustering on self-supervised features and pseudo labels on unsupervised saliency detection.

This work provides a *new perspective*, discovering visual

dependencies automatically from neural attention in vision transformers. We believe it is of great significance to both the traditional grammar induction field, and the recent vision transformer and multimodal learning research. We provide an initial study that enables a flexible model that can simultaneously work on hierarchical parsing, scene graph, and downstream tasks like detection and segmentation. Furthermore, our model can adaptively induce different kinds of structures conditions on the given task.

Vision Transformers. ViT [22] first applies self-attention directly to a sequence of image patches. Works [11, 26, 33, 59, 60, 65, 68, 75, 80, 92] follows the discipline to stack multiple self-attention layers to model the information across patch tokens. After that hierarchical designs are widely adopted to vision transformers [1, 12, 19, 20, 28, 35, 42, 43, 46, 47, 52, 55, 67, 71, 71, 75, 79, 84, 89, 89–91, 93, 95, 96] for better efficiency and lower memory cost. For example, Swin [52], ViL [93], and HaloNet [71] apply local windows attention to the patch tokens, which reduce the quadratic complexity to linear, but lose the ability of long-range dependency modeling. PVT [75] and CvT [79] perform attention on the squeezed tokens to reduce the computational cost. However, previous transformer models fail to discover object parts in images and resolve their dependencies.

In this work, we focus on efficient transformers for dependency parsing, based on the standard ViT [22]. We propose DependencyViT, a dependency-inspired vision transformer built on reversed self-attention, which captures hierarchies and dependencies between patches automatically. DependencyViT is orthogonal and seamlessly compatible with the SoTA transformer training methods, makes it more attractive than traditional grammar models from a practical perspective. Moreover, we show that the standard ViT layout can be highly efficient with our DependencyViT-Lite and dynamic pooling technique.

3. Method

This work proposes Visual Dependency Transformers (DependencyViT), a dependency-inspired backbone model based on reversed self-attention, capturing dependencies between patches automatically from self- or weakly-supervised signals for vision tasks.

Preliminaries. Let us assume a $\mathbb{R}^{N\times C}$ dimensional visual feature $\mathbf{X}$, where N is the number of total image patches and C is the number of token dimensions. The number of heads is H. The standard (forward) multi-head self-attention is defined as:

$$\begin{aligned} \mathcal{A}_f(\mathbf{Q},\mathbf{K},\mathbf{V}) &= \text{Concat}(\text{head}_1,\dots,\text{head}_H)\mathbf{W}_o \\ \text{where } \text{head}_h &= \text{Attention}(\mathbf{Q}_h,\mathbf{K}_h,\mathbf{V}_h) \\ &= \text{softmax}\left[\frac{\mathbf{Q}_h(\mathbf{K}_h)^{\mathrm{T}}}{\sqrt{C_h}}\right]\mathbf{V}_i \end{aligned} \tag{1}$$

where $\mathbf{Q}_h = \mathbf{X}\mathbf{W}_h^Q$, $\mathbf{K}_h = \mathbf{X}\mathbf{W}_h^K$, and $\mathbf{V}_h = \mathbf{X}\mathbf{W}_h^V$ are $\mathbb{R}^{N\times C_h}$ dimensional visual features of H heads, $\mathbf{X}\in\mathbb{R}^{N\times C}$ denotes the input feature and $\mathbf{W}_h\in\mathbb{R}^{C\times C_h}$ denotes the projection weights of the h_{th} head for $\mathbf{Q},\mathbf{K},\mathbf{V}$, $C = C_h * H$, and $\mathbf{W}_o$ is the weight of the output projection. $\mathbf{A}_{\mathrm{F}} = \text{softmax}(\mathbf{Q}\mathbf{K}^{\mathrm{T}})\in\mathbb{R}^{H\times N\times N}$ is called the attention matrix of the layer. In subsequent sections, we will omit the head dimension and focus on analyzing the attention within a single head.

3.1. Reversed Attention

The standard self-attention mechanism learns the $N\times N$ attention adjacency matrix to exchange information between different image patches. It treats all patches equally and does not follow a tree or graph structure, *i.e.*, it does not distinguish root and leaf nodes. To generate an adjacency graph, let us assume that each node can find its parent node via the `argmax(·)` function since the second dimension of the matrix follows a normalized probability distribution. In this case, the forward self-attention works by gathering information from parent nodes following the soft probabilities. All the nodes receive information from others, and eventually, they are dominated by the root node and the structural information of the image is lost. This learning scheme may perform well on visual recognition tasks due to its powerful attentive fusion and interaction capabilities, but it is not based on explicit hierarchical structures and dependencies.

Ideally, to build a dependency tree, we need to identify which patches are child nodes or parent nodes, so that information can be progressively aggregated to the root node. In turn, the root node distributes messages to leaf nodes. We achieve this by proposing reversed self-attention, which simply transposes the adjacency probability matrix so that the child node sends messages to the parent node. Considering each element a_{ij} in the attention matrix $\mathbf{A}$, we have:

$$a_{ij} = \text{softmax}\left(\left\{\frac{q_i k_j}{\sqrt{C_h}}\right\}_{j\in[0,N)}\right)_j, \tag{2}$$

where q_i is the i_{th} element of $\mathbf{Q}$, and k_j is the j_{th} element of $\mathbf{K}$. Then, after transposing the matrix $\mathbf{A}$, the information flow also changes as follows:

$$o_i = \left(\sum_j a_{ij}v_j\right)\mathbf{W}_o \implies o_i = \left(\sum_j a_{ji}v_j\right)\mathbf{W}_o, \tag{3}$$

where o_i denotes the i_{th} output and $\mathbf{W}_o$ is the weight of the output projection. We can see the child node 'receive' messages in forward attention but 'send' messages in reversed attention following the `softmax` probability distribution. Each child node has only one parent, but each

parent node can have multiple children. In this way, information can be collected progressively from leaf nodes to the root node through multiple reversed attention layers. At the same time, the dependency tree is also built bottom-up, and different subtrees may represent part-level or object-level semantics.

3.2. Dependency Block

Simply applying transposed attention matrices does not guarantee a good dependency graph induced. This is because: (i) The amount of token that a patch can attend to is controlled by multiple attention heads, thus the dependency graph is not unique. (ii) Contributions for different subtrees are not well distinguished. In some downstream tasks like image classification, foreground and background trees should be distinct. To solve the above questions: we further introduce two modules: a head selector and a message controller. An overview of our dependency block is shown in Figure 2.

Head Selector. The head selector $\mathbf{P}$ is used to choose proper reversed attention heads for dependency induction. We obtain it by applying the `softmax(·)` function on the linear projections of the input tokens: $\mathbf{P} = \mathrm{softmax}(\mathbf{X}\mathbf{W}_p)$, where $\mathbf{W}_p \in \mathbb{R}^{C\times H}$ is the projection weight. By the head selector, we can build dependencies over all attention heads following the learnable soft probabilities and generate a unique dependency graph for each layer.

Message Controller. Similarly, the message controller $\mathbf{M}$ is learnable weights imposed on tokens during reversed self-attention. The goal of $\mathbf{M}$ is to determine the extent to which a node or a subtree sends messages. Specifically, we use two linear projection layers (who have the dimensions $\mathbb{R}^{C\times\frac{C}{2}}$ and $\mathbb{R}^{\frac{C}{2}\times 1}$) with a GELU activation [32] between them to learn it. After that, a `sigmoid(·)` function is used to get the probability in $[0, 1]$ of sending messages.

Note that the weights learned by the message controller are cumulative across all layers. The message controller $\mathbf{M}$ in the i_{th} layer is computed by $\mathbf{M}_1 \cdot \mathbf{M}_2 ... \cdot \mathbf{M}_i$, where the subscript represents the index of the layer. We also use $\mathbf{M}$ to weight the pooling to get the final representation over all patches. It has two benefits: (i) If a node does not send information in a layer, it keeps the status in subsequent layers, making the induced structure clearer. (ii) Different subtrees are weighted differently, which filters meaningless patches, benefiting downstream tasks like recognition and detection.

In summary, we have the reversed attention $\mathbf{A}_\mathrm{R} = \mathbf{A}_\mathrm{F} \cdot \mathbf{P} \cdot \mathbf{M}$ with dimensional permutations, where $\mathbf{A}_\mathrm{F}$ is the forward attention, as shown in Figure 2. We then compute the soft dependency mask by applying the `sum(·)` operator on $\mathbf{A}_\mathrm{R}$ over the head dimension. The dependency graph and tree structure are then obtained by `argmax(·)` and the chu-liu-edmonds algorithm [15], respectively.

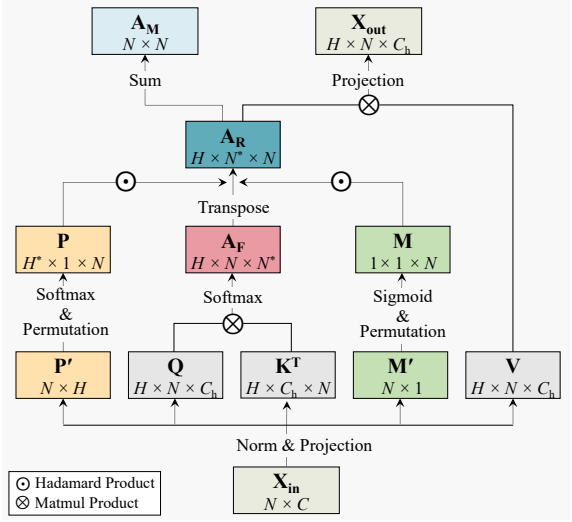

Figure 2. An architecture overview of our proposed reversed attention block in DependencyViT. FeedForward Networks (FFNs) and residual paths are omitted here. The input and output tokens are $\mathbf{X}_\mathrm{in}$ and $\mathbf{X}_\mathrm{out}$ with the number of tokens N and token dimensions C, respectively. The number of attention heads is H, and the per-head token dimension is C_h. We obtain the reversed attention matrices $\mathbf{A}_\mathrm{R}$ by transposing the forward attention weights $\mathbf{A}_\mathrm{F}$ with a head selector $\mathbf{P}$ and a message controller $\mathbf{M}$ imposing on it. After that, the soft dependency mask $\mathbf{A}_\mathrm{M}$ is induced by applying summation on $\mathbf{A}_\mathrm{R}$ over the head dimension. '*' indicates the dimension that is normalized by softmax probability distribution.

3.3. Dynamic Pooling based on Dependencies

Our dependency block is able to learn dynamic and comprehensive information flow between patches for dependency induction. Intuitively, with such visual dependencies, scene understanding can be simplified with less computational effort as most of the information can be represented by a few nodes. With this inspiration, we introduce a dynamic visual pooling scheme that reduces the computational cost largely (*i.e.*, FLOPs and GPU memory), and propose a lightweight model DependencyViT-Lite.

Specifically, we rank and prune those leaf nodes which have the least information received, because 1) they are not the parent of any node and 2) they rarely transmit messages or they have sent enough information to their parent nodes. We progressively prune the leaf nodes with the least messages as the depth of the network increases. In this way, the memory and resource costs are largely reduced. Meanwhile, the tree architecture is still maintained by recording relationships between the pruned nodes and their parents to form a complete tree. Most importantly, DependencyViT-Lite is able to perform dense prediction tasks though some of its tokens are removed. According to the dependency graph, we retrieve those pruned nodes by a soft aggregation from their parents.

3.4. Model Analysis and Protocols

Model Instantiation. In this work, we follow the design strategy of the standard ViT (DeiT) [22, 68]. To show the efficiency and effectiveness of our model, we choose two different model sizes and build DependencyViT-T and DependencyViT-S based on tiny and small ViTs as backbones, respectively. We set the number of attention heads $H = 12$, the number of dependency blocks $L = 12$ with residual paths and FFNs of ratio 4 as in standard ViT. We set the token dimensions $C = \{192, 384\}$ for tiny and small models, respectively. Take an image with an arbitrary resolution, a C-dimensional $16\times$ down-sampling feature is obtained after the patch embedding layer. There are no overlaps between any of the two patches. Conditional positional encoding is used as in [14]. Based on our observation that the 'cls' token passes information between two visual patches and leads to confusion in dependencies, we remove it from our model. For DependencyViT-Lite models, we prune 16% number of nodes (*e.g.*, 32 of 196) at the $\{2, 5, 8, 11\}_{th}$ layers, respectively.

Complexity Analysis. Simply applying the standard global self-attention leads to a complexity of $O(2N^2C+12NC^2)$, which contains $O(2N^2C)$ for self-attentions, $O(4NC^2)$ for linear projections, and $O(8NC^2)$ for feedforward networks (FFNs). Our head selector and message controller lead to additional costs of $O(NCH)$ and $O(NC)$, respectively, which are much smaller than the costs of other components. In contrast, our DependencyViT-Lite reduces the number of tokens N to $0.3N$ and even smaller through dynamic pooling, which lowers the complexity exponentially (to 10% and even smaller). DependencyViT-Lite can run with batch sizes more than three times that of ViT on a same GPU.

Pretraining Protocols. We apply two different pretraining methods on DependencyViT: weakly-supervised and self-supervised. The first one is supervised pretraining on ImageNet by leveraging the information in class-level labels. The supervision encourages the model to learn high-level object-aware semantics, based on which DependencyViT learns to model object-aware dependencies by gathering information from subtrees to the root node (centered object).

For self-supervised pretraining, we take inspiration from recent contrastive learning and masked image modeling methods [2,5,7,9,29,97] as they can learn both object-level global representations and part-level local features. Specifically, we follow the same pretraining protocol as iBOT [97] (*e.g.*, employ self-distillation and masked image modeling on DependencyViT) and enjoy the benefit of its powerful pretraining capabilities. After pretraining, DependencyViT can establish a dependency tree for an unseen image, containing part-to-part, part-to-object, and object-to-object dependencies.

Figure 3 shows the dependency trees of an image from ImageNet parsed by weakly-supervised and self-supervised pretrained DependencyViT, respectively. It can be seen that weakly-supervised pretrained DependencyViT focuses more on the entire object, while the self-supervised pretrained DependencyViT can capture more fine-grained part-aware dependencies. The parsed dependency tree is expected to help many downstream tasks, such as saliency detection and part segmentation. For more analysis and detailed settings, please refer to Appendix.

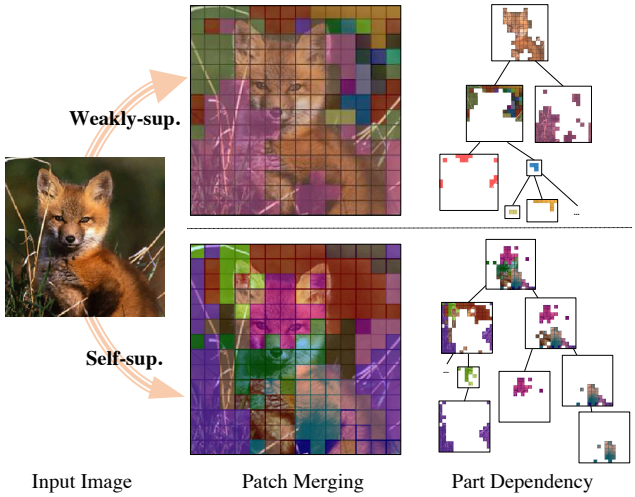

Figure 3. Visualizations of dependency trees parsed by self- and weakly-supervised pretrained DependencyViT, respectively. Patches are aggregated gradually until the root node is formed. To facilitate observation, the background area is not filled to the root node. It can be seen that weakly-supervised DependencyViT focuses more on the whole object, while the self-supervised DependencyViT captures more fine-grained part-aware dependencies.

4. Experiments

In this section, we conduct extensive experiments to show the effectiveness of DependencyViT and DependencyViT-Lite on visual parsing and recognition. They are: unsupervised part segmentation on the Pascal-Part [8] and Car-Parts [56] datasets; unsupervised saliency detection on the ECSSD [63], DUTS [73] and DUT-OMRON [87] datasets; dependency parsing on the COCO dataset [50]; and image classification on ImageNet-1K [16].

4.1. Unsupervised Part Segmentation

To show the effectiveness of DependencyViT on visual dependency parsing, we apply it to the unsupervised part segmentation task without part labels, which is challenging and under-explored as it requires a comprehensive dependency understanding between parts. Considering available part parsing datasets, *e.g.*, Pascal-Part [8] and Car-Parts [56], are of small resolution and data scale, tiny ViT model is enough to work on this situation and

Table 1. Part segmentation results on the Pascal-Part [8] and Car-Parts [56] datasets. 'clustering' indicates applying k-means [53] on feature representations; 'maximum spanning' denotes the dependency tree is generated by Chu-Liu-Edmonds maximum spanning algorithm [15].

Method	Pretraining Type	Part Discovery by	Pascal-Part [8]		Car-Parts [56]	
			mIoU (%)	mAcc (%)	mIoU (%)	mAcc (%)
DeiT [68]	weakly-sup.	clustering	7.2	22.6	8.9	29.5
DeiT [68]	weakly-sup.	maximum spanning	18.9	35.5	17.8	37.7
DependencyViT	weakly-sup.	clustering	11.6	31.7	10.9	29.7
DependencyViT	weakly-sup.	maximum spanning	23.2	41.7	22.6	40.0
iBOT [97]	self-sup.	maximum spanning	25.1	44.8	25.7	46.1
DependencyViT	self-sup.	maximum spanning	**28.7**	**47.9**	**27.0**	**47.2**

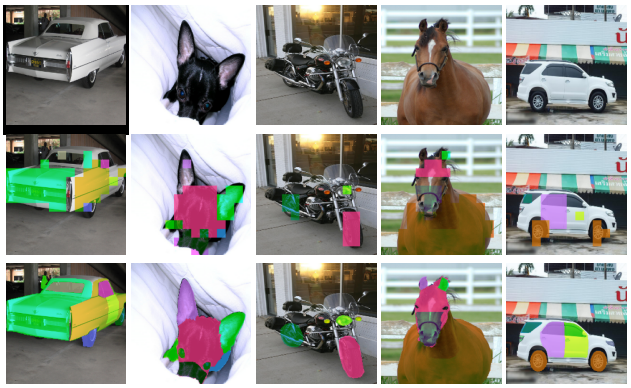

Figure 4. Visualization of part partitioning on the Pascal-Part [8] and Car-Parts [56] datasets. From top to bottom: 1) the original image; 2) our generated part mask of which each color represents a subtree in the hierarchy; 3) the ground truth part segments.

Figure 5. Visualization for unsupervised saliency detection on the ECSSD [63], DUTS [73] and DUT-OMRON [87] datasets. From top to bottom: 1) the original image; 2) our generated saliency mask; 3) our results post-processed by the bilateral solver [3]; 4) the ground truth part partitions.

further scaling model size up brings no gains. We take DependencyViT-T as our base model.

Both weakly- and self-supervised pretrained models are evaluated. For DependencyViT, we average the learned dependency masks of all layers, and then leverage the Chu-Liu-Edmonds maximum spanning algorithm [15] to generate the dependency tree. After that, we perform matching between all subtrees and part segments by the Hungarian maximum matching algorithm [41] and compute the mean intersection over union (mIoU) and mean accuracy (mAcc) metrics for evaluation. Note that we remove small part regions from evaluation for more reliable results. We take DeiT-Tiny [68] as the baseline. Since there are no explicit dependencies in it, the Naïve solution is to partition the patches in their latent representation space by k-means clustering [53] (k is set to 20 in this paper). To get a stronger baseline, we also build tree structures on DeiT by applying the maximum spanning algorithm on its mean pooled attention map. For self-supervised models, DependencyViT-T is evaluated with the maximum spanning algorithm for dependency tree generation. We take the self-supervised iBOT (tiny DeiT) as a strong baseline for fair comparison.

From the results shown in Table 1, we observe that DependencyViT consistently outperforms the baseline methods by a large margin on both weakly- and self-supervised settings and two datasets, demonstrating the effectiveness of our dependency parsing. Self-supervised DependencyViT shows better performance than the weakly-supervised one as it can learn more fine-grained dependencies. We visualize our patch-level part partitioning results in Figure 4.

4.2. Unsupervised Saliency Detection

Besides part-level partitioning, DependencyViT can also work on object-level comprehensions, thanks to its built-in hierarchical dependencies. We evaluate the unsupervised saliency detection results of DependencyViT on three datasets. Except baseline methods [45, 54, 86, 99] that are specifically designed for the task based on pseudo-labels, we evaluate weakly-supervised DeiT for fair comparison. Following [78], we leverage normalized cut [62] on token representations to get the salient area of an image for DeiT. For DependencyViT, the soft dependency mask is added to the representation for better results. Bilateral solver [3] is used as post-processing for segment smoothing.

Figure 5 visualizes the saliency detection map of our method DependencyViT (weakly-sup.). From the figure and Table 2, we see that: 1) DependencyViT is superior to its counterparts, including the pseudo label-based saliency detection methods and DeiT. 2) Weakly-supervised DependencyViT outperforms the self-supervised one, indicating

Table 2. Unsupervised saliency detection on ECSSD [63], DUTS [73] and DUT-OMRON [87]. Tiny models, token normalized cut [62, 78] and bilateral solver [3] post-processing are used for DEiT and DependencyViT.

Method	ECSSD [63]			DUTS [73]			DUT-OMRON [87]		
	$maxF_\beta$ (%)	IoU (%)	Acc. (%)	$maxF_\beta$ (%)	IoU (%)	Acc. (%)	$maxF_\beta$ (%)	IoU (%)	Acc. (%)
DeepUSPS [54]	58.4	44.0	79.5	42.5	30.5	77.3	41.4	30.5	77.9
HS [86]	67.3	50.8	84.7	50.4	36.9	82.6	56.1	43.3	84.3
wCtr [99]	68.4	51.7	86.2	52.2	39.2	83.5	54.1	41.6	83.8
WSC [45]	68.3	49.8	85.2	52.8	38.4	86.2	52.3	38.7	86.5
DeiT [68]	49.3	40.5	72.7	34.2	26.8	72.7	33.2	27.2	71.1
DependencyViT (self-sup.)	62.1	55.0	78.4	43.0	35.9	73.2	32.5	28.0	67.2
DependencyViT (weakly-sup.)	62.0	48.4	83.6	53.8	37.0	87.5	52.0	39.7	**88.4**

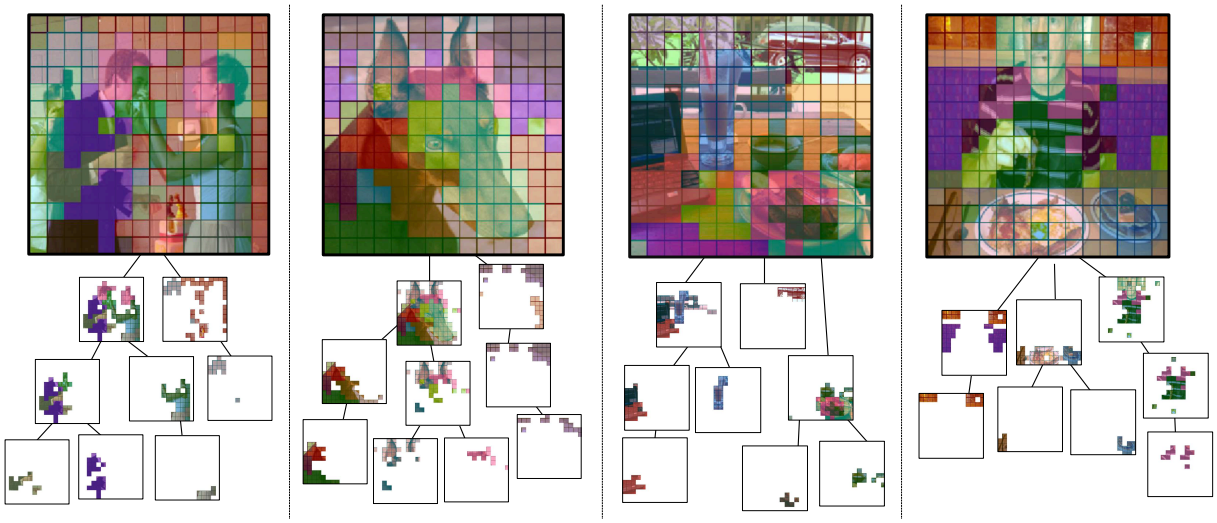

Figure 6. Visualizations of dependency trees parsed by self-supervised DependencyViT (small). Different colors represent different subtrees. Here we ignore the nodes in the small region (less important) and display the main subtrees.

weakly-supervised model is better at modeling object-level semantics. 3) The failure case in the last column of Figure 5 demonstrates how DependencyViT works. The two birds belong to the same semantic category but different objects, hence two subtrees.

To verify the effectiveness of dependency for object-level understanding, we make ablative comparisons by whether adding the soft dependency (+dependency) to the feature representation, see Figure 7). We see that 1) the dependency mask improves the performance significantly, showing the effectiveness of DependencyViT in learning object-level dependencies; and 2) the bilateral solver brings considerable improvement over all models.

4.3. Visualization

As shown in Figure 6, we visualize our visual dependency parsing on images from the COCO dataset, which does not overlap with the pretraining ImageNet dataset. We can see that the foreground and the background areas are represented by different subtrees, which further construct the overall scene dependency tree. The root subtree is generally an important part of the foreground object.

More downstream experiments, *e.g.*, semantic segmentation on ADE20K [94], object detection on the COCO dataset [50], and video recognition on Kinetics-400 [38], can be found in Appendix.

4.4. Visual Recognition

We show DependencyViT can work as a visual backbone for recognition and its downstream tasks. Two different model configurations, *i.e.*, DependencyViT and DependencyViT-Lite, are evaluated and compared with many counterparts. We make the following summaries from Table 4. 1) DependencyViT outperforms all counterparts, *e.g.*, 3.2% and 2.3% improvements over DeiT-Tiny and DeiT-small, respectively, indicating dependency parsing is likely to contribute to visual recognition tasks. 2) DependencyViT-Lite is the most efficient one (0.8 GFLOPs only) of all models and shows good performance, demonstrating the effectiveness of our progressively dynamic

Table 3. Comparisons of image classification on ImageNet-1K. All models (tiny) are trained and evaluated with 224×224 resolution.

Model	Direction	Head Selector	Message Controller	#Params (M)	FLOPs (G)	Top-1 (%)
Baseline (DeiT) [68]	forward	×	×	5.7	1.3	73.3
Forward + P	forward	√	×	5.7	1.3	73.4
Forward + M	forward	×	√	6.1	1.3	74.8
Forward + P + M	forward	√	√	6.2	1.3	74.8
Reverse + P	reverse	√	×	5.7	1.3	73.6
Reverse + M	reverse	×	√	6.1	1.3	74.9
Reverse + P + M (DependencyViT)	reverse	√	√	6.2	1.3	**75.4**
DependencyViT-Lite (forward)	forward	√	√	6.2	0.8	71.1
DependencyViT-Lite (reverse)	reverse	√	√	6.2	0.8	73.7

Figure 7. Ablative comparsions (tiny) of saliency detection on ECSSD dataset.

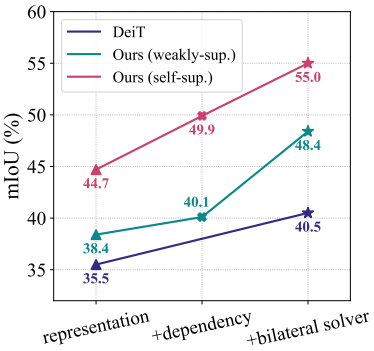

Table 4. Comparison of image classification on ImageNet-1K. All models are trained and evaluated with 224×224 resolution. * denotes the method can not be used for dense predictions.

Model	Hierarchical	Cost	#Params (M)	FLOPs (G)	Top-1 (%)
ResNet-18 [31]	√	low	11.7	1.8	69.9
ConvMixer-512/16 [69]	×	high	5.4	–	73.7
DeiT-Tiny/16 [68]	×	high	5.7	1.3	72.2
CrossViT-Tiny [6]	×	high	6.9	1.6	73.4
PVT-Tiny [75]	√	low	13.2	1.9	75.1
DependencyViT-Lite-T	×	low	6.2	0.8	73.7
DependencyViT-T	×	high	6.2	1.3	**75.4**
ResNet-50 [31]	√	low	25.0	4.1	76.2
ConvMixer-768/32 [69]	×	high	21.1	–	80.2
DeiT-Small/16 [68]	×	high	22.1	4.5	79.8
CrossViT-Small [6]	×	high	26.7	5.6	81.0
PVT-Small [75]	√	low	24.5	3.8	79.8
Swin-Tiny [52]	√	low	28.3	4.5	81.2
CvT-13 [79]	√	high	20.0	4.5	81.6
DynamicViT-LV-S/0.5 [57]*	×	–	26.9	3.7	82.0
PVTv2-B2 [74]	√	low	25.4	4.0	82.0
DependencyViT-Lite-S	×	low	24.0	3.0	80.6
DependencyViT-S	×	high	24.0	5.0	**82.1**

pooling. Typically, hierarchical transformers are more efficient and save computations for downstream tasks. Our DependencyViT-Lite reduces costs through induced dependencies even using a standard ViT layout. 3) DynamicViT [57] is a pruning-based transformer for the classification task. However, it can not perform dense predictions because the information of its pruned patches is lost. On the contrary, the pruned nodes in our DependencyViT-Lite can be retrieved from their parents for dense predictions, showing the importance of dependency induction.

4.5. Ablation Study

We perform ablation studies on tiny models in Table 3. P denotes the head selector, and M denotes the message controller. We use 'forward' and 'backward' to indicate the attention direction. We can see that the head selector brings smaller gains than the message controller. And the gains in reverse attention are larger than gains in forward attention. Both the head selector and the message controller are important to dependency induction and the dynamic pooling scheme, *i.e.*, DependencyViT-Lite.

More ablation studies and downstream experiments can be found in Appendix.

5. Conclusion

This paper studies patch-level visual dependency parsing using our proposed DependencyViT. We show that the reversed self-attention mechanism in transformers can naturally capture long-range visual dependencies between image patches. With reversed self-attention, a child node is trained to attend to its parent and send the information to the parent node, and a hierarchical dependency tree can be established automatically. Furthermore, dynamically image pooling is made possible by learned dependencies, *i.e.*, merging child nodes into their corresponding parent nodes, based on which we propose a lightweight model DependencyViT-Lite. Extensive experiments on both self- and weakly-supervised pretraining on ImageNet, as well as five downstream tasks, show the model's effectiveness.

Limitations. Although our work achieves good performance on many tasks by visual dependency induction, it is an initial study with a fixed patch size and efficient settings The current patch size limits its performance on small objects. We will explore more and further scale up our model. The proposed approach has no ethical or societal issues on its own, except those inherited from computer vision.

Acknowledgements. Ping Luo is partially supported by the National Key R&D Program of China No.2022ZD0161000 and the General Research Fund of HK No.17200622. Chuang Gan was supported by the MIT-IBM Watson AI Lab, DARPA MCS, DSO grant DSOCO21072, and gift funding from MERL, Cisco, Sony, and Amazon.

References

[1] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *NeurIPS*, 34, 2021. 3

[2] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. *arXiv:2106.08254*, 2021. 5

[3] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *European conference on computer vision*, pages 617–632. Springer, 2016. 6, 7

[4] Sandro Braun, Patrick Esser, and Björn Ommer. Unsupervised part discovery by unsupervised disentanglement. In *DAGM German Conference on Pattern Recognition*, pages 345–359. Springer, 2020. 2

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 5

[6] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *ICCV*, 2021. 8

[7] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 5

[8] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 5, 6

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 5

[10] Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee Kenneth Wong, Joshua B Tenenbaum, and Chuang Gan. Grounding physical concepts of objects and events through dynamic visual reasoning. In *International Conference on Learning Representations*. 1

[11] Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik Learned-Miller, and Chuang Gan. Mod-squad: Designing mixture of experts as modular multi-task learners. *arXiv preprint arXiv:2212.08066*, 2022. 3

[12] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *ICCV*, pages 589–598, 2021. 3

[13] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. *Advances in Neural Information Processing Systems*, 34:28104–28118, 2021. 2

[14] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *Arxiv preprint 2102.10882*, 2021. 5

[15] Yoeng-Jin Chu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400, 1965. 4, 6

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 5

[17] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 2

[18] Mingyu Ding, Zhenfang Chen, Tao Du, Ping Luo, Josh Tenenbaum, and Chuang Gan. Dynamic visual reasoning by learning differentiable physics models from video and language. *Advances In Neural Information Processing Systems*, 34:887–899, 2021. 1

[19] Mingyu Ding, Xiaochen Lian, Linjie Yang, Peng Wang, Xiaojie Jin, Zhiwu Lu, and Ping Luo. Hr-nas: Searching efficient high-resolution neural architectures with lightweight transformers. In *CVPR*, 2021. 3

[20] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. *arXiv preprint arXiv:2204.03645*, 2022. 3

[21] Mingyu Ding, Yan Xu, Zhenfang Chen, David Daniel Cox, Ping Luo, Joshua B Tenenbaum, and Chuang Gan. Embodied concept learner: Self-supervised learning of concepts and mapping through instruction following. In *Conference on Robot Learning*, pages 1743–1754. PMLR, 2023. 1

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3, 5

[23] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Visual grounding with transformers. *arXiv preprint arXiv:2105.04281*, 2021. 1

[24] Marzieh Edraki, Nazanin Rahnavard, and Mubarak Shah. Subspace capsule network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10745–10753, 2020. 2

[25] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 1, 2

[26] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *ICCV*, pages 12259–12269, 2021. 3

[27] Feng Han and Song-Chun Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):59–73, 2008. 1, 2

[28] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. *arXiv preprint arXiv:2204.07143*, 2022. 3

[29] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 5

[30] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 8

[32] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[33] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *ICCV*, pages 11936–11945, 2021. 3

[34] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *International conference on learning representations*, 2018. 2

[35] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021. 3

[36] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 869–878, 2019. 2

[37] Ayush Jaiswal, Wael AbdAlmageed, Yue Wu, and Premkumar Natarajan. Capsulegan: Generative adversarial capsule network. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 2

[38] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 7

[39] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 74–83, 2021. 1

[40] Adam Kosiorek, Sara Sabour, Yee Whye Teh, and Geoffrey E Hinton. Stacked capsule autoencoders. *Advances in neural information processing systems*, 32, 2019. 1, 2

[41] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

[42] Changlin Li, Tao Tang, Guangrun Wang, Jiefeng Peng, Bing Wang, Xiaodan Liang, and Xiaojun Chang. Bossnas: Exploring hybrid cnn-transformers with block-wisely self-supervised neural architecture search. In *ICCV*, pages 12281–12291, 2021. 3

[43] 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. *arXiv preprint arXiv:2201.09450*, 2022. 3

[44] Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. Deep hierarchical semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1246–1257, 2022. 2

[45] Nianyi Li, Bilin Sun, and Jingyi Yu. A weighted sparse coding framework for saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5216–5223, 2015. 6, 7

[46] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv: Computer Vision and Pattern Recognition*, 2021. 3

[47] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 3

[48] Zhiheng Li, Wenxuan Bao, Jiayang Zheng, and Chenliang Xu. Deep grouping model for unified perceptual parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4053–4063, 2020. 2

[49] Xiaodan Liang, Hongfei Zhou, and Eric Xing. Dynamic-structured semantic propagation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 752–761, 2018. 2

[50] 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 *ECCV*, pages 740–755. Springer, 2014. 5, 7

[51] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Unsupervised part segmentation through disentangling appearance and shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8355–8364, 2021. 2

[52] 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 *ICCV*, 2021. 3, 8

[53] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 6

[54] Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. Deepusps: Deep robust unsupervised saliency prediction via self-supervision. *Advances in Neural Information Processing Systems*, 32, 2019. 6, 7

[55] Zizheng Pan, Bohan Zhuang, Jing Liu, Haoyu He, and Jianfei Cai. Scalable visual transformers with hierarchical pooling. *arXiv preprint arXiv:2103.10619*, 2021. 3

[56] Kitsuchart Pasupa, Phongsathorn Kittiworapanya, Napasin Hongngern, and Kuntpong Woraratpanya. Evaluation of deep learning algorithms for semantic segmentation of car parts. *Complex & Intelligent Systems*, pages 1–13, May 2021. 5, 6

[57] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 8

[58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[59] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *NeurIPS*, 34, 2021. 3

[60] Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8

learned tokens do for images and videos? *arXiv: Computer Vision and Pattern Recognition*, 2021. 3

[61] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017. 1, 2

[62] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 6, 7

[63] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):717–729, 2015. 5, 6, 7

[64] Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3971–3980, 2022. 2

[65] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *CVPR*, pages 16519–16529, 2021. 3

[66] Erik B Sudderth, Antonio Torralba, William T Freeman, and Alan S Willsky. Learning hierarchical models of scenes, objects, and parts. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1331–1338. IEEE, 2005. 1, 2

[67] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. *arXiv preprint arXiv:2201.02767*, 2022. 3

[68] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 3, 5, 6, 7, 8

[69] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022. 8

[70] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of computer vision*, 63(2):113–140, 2005. 1, 2

[71] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *CVPR*, pages 12894–12904, 2021. 3

[72] Bo Wan, Wenjuan Han, Zilong Zheng, and Tinne Tuytelaars. Unsupervised vision-language grammar induction with shared structure modeling. In *International Conference on Learning Representations*, 2021. 1

[73] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 136–145, 2017. 5, 6, 7

[74] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021. 8

[75] 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 *ICCV*, 2021. 3, 8

[76] Wenguan Wang, Zhijie Zhang, Siyuan Qi, Jianbing Shen, Yanwei Pang, and Ling Shao. Learning compositional neural information fusion for human parsing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5703–5713, 2019. 2

[77] Wenguan Wang, Hailong Zhu, Jifeng Dai, Yanwei Pang, Jianbing Shen, and Ling Shao. Hierarchical human parsing with typed part-relation reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8929–8939, 2020. 2

[78] Yangtao Wang, Xi Shen, Shell Hu, Yuan Yuan, James Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. *arXiv preprint arXiv:2202.11539*, 2022. 6, 7

[79] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 3, 8

[80] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *ICCV*, pages 10033–10041, 2021. 3

[81] Tianfu Wu and Song-Chun Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *International journal of computer vision*, 93(2):226–252, 2011. 1, 2

[82] Edgar Xi, Selina Bing, and Yang Jin. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*, 2017. 2

[83] Canqun Xiang, Lu Zhang, Yi Tang, Wenbin Zou, and Chen Xu. Ms-capsnet: A novel multi-scale capsule network. *IEEE Signal Processing Letters*, 25(12):1850–1854, 2018. 2

[84] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *NeurIPS*, 34, 2021. 3

[85] Hongwei Xue, Yupan Huang, Bei Liu, Houwen Peng, Jianlong Fu, Houqiang Li, and Jiebo Luo. Probing intermodality: Visual parsing with self-attention for vision-and-language pre-training. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[86] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1155–1162, 2013. 6, 7

[87] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013. 5, 6, 7

[88] Jinyu Yang, Peilin Zhao, Yu Rong, Chaochao Yan, Chunyuan Li, Hehuan Ma, and Junzhou Huang. Hierarchical graph capsule network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10603–10611, 2021. 2

[89] Qihang Yu, Yingda Xia, Yutong Bai, Yongyi Lu, Alan L Yuille, and Wei Shen. Glance-and-gaze vision transformer. *NeurIPS*, 34, 2021. 3

[90] Tan Yu, Gangming Zhao, Ping Li, and Yizhou Yu. Boat: Bilateral local attention vision transformer. *arXiv preprint arXiv:2201.13027*, 2022. 3

[91] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112*, 2021. 3

[92] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv: Computer Vision and Pattern Recognition*, 2021. 3

[93] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *ICCV*, 2021. 3

[94] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 7

[95] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021. 3

[96] Jingkai Zhou, Pichao Wang, Fan Wang, Qiong Liu, Hao Li, and Rong Jin. Elsa: Enhanced local self-attention for vision transformer. *arXiv: Computer Vision and Pattern Recognition*, 2021. 3

[97] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 5, 6

[98] Song-Chun Zhu, David Mumford, et al. A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision*, 2(4):259–362, 2007. 1, 2

[99] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2814–2821, 2014. 6, 7