Appendix for "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

Table 1. Metric for more task on the taskonomy dataset. The experiments section in the paper demonstrates what metric we use for each task.

Notation	Representation	Notation	Representation
X	Input	$\mathbf{A}_{\mathrm{F}}$	Forward Attention Map
P	Head Selector	$\mathbf{A}_{\mathrm{R}}$	ReverseAttention Map
M	Message Controller	N	number of patches
Q	Query	H	number of heads
K	Key	C	token dims
V	Value	C_h	token dims per head
W	Projections		

Overview

In this appendix, we supplement the main paper by providing more thorough evaluations and empirical analyses to back up our claims. We also include more detailed descriptions of our experiments to help readers better understand our paper.

This appendix is organized as follows.

- In Section A, we give the notations used in this work.
- In Section B, we benchmark our models on two dense prediction downstream tasks.
- In Section C, we introduce detailed analysis to our model, including the relationship to pruning-based transformers, the comparison between reversed attention and forward attention, possible applications on video recognition, and some ablation studies.
- In Section D, we detail the training configurations and implementation details for each downstream task.

A. Notations

We provide the notations shown in Table 1 for this work.

B. Downstream Tasks

We benchmark our models on two dense prediction downstream tasks. All the model training follows common practices and protocols, as in [26, 28].

Semantic segmentation. In Table 2, we show the performance of our models on ADE20K [36] against several powerful counterparts. Considering DeiT [26] is the baseline that can be apple-to-apple comparable to us, we pretrain DeiT and our models on ImageNet-1K and produce the results of them under the same setting. We can see that: our DependencyViT consistently outperforms its counterparts including Swin [18]; and even DependencyViT-Lite surpasses the baseline PVT [28] by a large margin. Notably, the backbone model for DependencyViT-Lite only costs $1/3$ computations (see the numbers in parentheses of the table) of our DependencyViT, showing its efficiency.

Object detection and instance segmentation. We benchmark our models on object detection with COCO 2017 [17] based on Mask R-CNN [9]. Table 3 show the detection and instance segmentation results. The results of DeiT and our models are implemented by us under the same setting. We observe substantial gains across all settings and metrics compared with several CNN and transformer baselines. Surprisingly, the backbone FLOPs consumption of DependencyViT-Lite-T is 3.5 GFLOPs, costing only 1.5% of the entire network.

Table 2. Comparison with SoTA methods for semantic segmentation on ADE20K [36] val set. Single-scale evaluation is used. FLOPs are measured by 512×2048. Considering the segmentation head UperNet [32] is heavy, while the network backbone occupies only a small part of the computation, we mark the GFLOPs of the backbone of our works in parentheses.

Backbone	Method	#Params (M)	FLOPs (G)	mIoU (%)
ResNet18	SemanticFPN [15]	15.5	128.8	32.9
PVT-Tiny [28]	SemanticFPN [15]	17.0	132.8	35.7
DeiT-Tiny [26]	UperNet [32]	10.7	142.8	37.8
DependencyViT-Lite-T	UperNet [32]	11.1	130.2 (7.8)	36.1
DependencyViT-T	UperNet [32]	11.1	145.1 (22.7)	**40.3**
ResNet50	SemanticFPN [15]	28.5	729.6	36.7
PVT-Small [28]	SemanticFPN [15]	28.2	712.0	39.8
DeiT-Small [26]	UperNet [32]	41.3	566.8	43.0
Swin-Tiny [18]	UperNet [32]	60.0	945.0	44.5
DependencyViT-Lite-S	UperNet [32]	43.1	515.2 (29.6)	41.2
DependencyViT-S	UperNet [32]	43.1	574.4 (88.8)	45.7

Table 3. COCO object detection and segmentation results with Mask R-CNN [10]. All models are trained with $1\times$ schedule and multi-scale inputs. FLOPs are measured by 800×640. The GFLOPs of the backbone of our DependencyViT and DependencyViT-Lite are marked in parentheses. The first three metrics are for object detection, while the last three for instance segmentation.

Backbone	#Params (M)	FLOPs (G)	Mask R-CNN 1x					
			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet18 [10]	31.2	190.0	34.0	54.0	36.7	31.2	51.0	32.7
PVT-Tiny [28]	32.9	195.0	36.7	59.2	39.3	35.1	56.7	37.3
Deit-Tiny [26]	27.3	244.6	30.6	46.8	32.8	27.4	44.7	28.9
DependencyViT-Lite-T	27.8	238.1 (3.5)	35.2	58.8	38.6	34.1	56.2	36.1
DependencyViT-T	27.8	245.6 (11.0)	**37.8**	**62.1**	**41.4**	**36.0**	**59.3**	**38.6**
ResNet50 [10]	44.2	260.0	38.0	58.6	41.4	34.4	55.1	36.7
PVT-Small [28]	44.1	245.0	40.4	62.9	43.8	37.8	60.1	40.3
Deit-Small [26]	44.9	276.2	36.9	55.1	39.7	32.7	52.3	34.5
DependencyViT-Lite-S	46.85	249.9 (13.2)	38.1	62.5	41.8	36.2	59.4	38.4
DependencyViT-S	46.85	280.0 (43.3)	**42.4**	**66.5**	**46.4**	**38.5**	**62.7**	**41.9**

C. Analysis

In this section, we introduce detailed analysis to our model.

C.1. Relation to Pruning-based Methods

Our work is related to dynamic-merged [31, 33] or pruning-based [4, 14, 21, 34] vision transformers. For example, DynamicViT [21] is a pruning-based transformer by optimizing a learnable weight for each token through Gumbel-Softmax.

However, the above methods mainly focus on the image classification tasks. They can not perform dense predictions because the information of their pruned patches is lost. On the contrary, pruning in a tree structure preserves the information lost by explicitly learned structures. As shown in the main paper, the pruned nodes in our DependencyViT-Lite can be retrieved from their parents for dense predictions, showing the importance of dependency induction.

C.2. Reversed attention vs. Forward one

Though forward attention well models the information interaction between patches, it mainly focuses on the task-specific region rather than the entire image, *e.g.*, the foreground region for the image classification task. This is because forward self-attention works through "gathering information", thus the information in the background region that does not contribute to the recognition task is to a large extent suppressed and not gathered. The observation is evidenced by many previous works.

However, for our reversed self-attention, all the patches are get attended, *e.g.*, a subtree will be generated for the background area. The background information is kept because we do not prune any parent nodes. We then use the message controller to filter the useless information out for the final image recognition. Therefore, reversed attention has better generalization when extended to dense prediction tasks such as semantic segmentation, which is empirically validated by our experiments.

Table 4. Comparison of image classification on ImageNet-1K when different number of tokens are pruned.

Model	kept tokens	#Params (M)	FLOPs (G)	Top-1 (%)
DependencyViT-Lite-32	32	6.2	0.6	72.4
DependencyViT-Lite-64	64	6.2	0.8	73.7
DependencyViT-Lite-128	128	6.2	1.0	74.9
DependencyViT	196	6.2	1.3	75.4

Table 5. Video-level accuracy on the Kinetics-400 validation set.

Method	Top-1 (%)	Top-5 (%)	FLOPs (G)	Frames	Resolution
TimeSformer	76.9	92.7	0.20	8	224
TimeSformer-Lite	70.6	89.3	0.08	8	224
TimeSformer-HR	78.1	93.3	1.70	16	448
TimeSformer-HR-Lite	73.1	90.4	0.67	16	448
TimeSformer-L	79.8	94.1	2.38	96	224
TimeSformer-L-Lite	74.1	91.3	0.61	96	224

C.3. Pruning ratio

We also show DependencyViT-Lite with different pruning ratios by keeping the remaining token number as 32, 64, and 128. The results are shown in Table 4. We can see that when we keep 128 tokens, the performance drop is minor relative to the full DependencyViT. The performance gap could be larger when more tokens are pruned.

C.4. Dynamic Pruning on Video Recognition

We evaluate the models on the validation sets of Kinetics-400 (K400). Kinetics-400 consists of $240K$ training videos and $20K$ validation videos that span 400 human action categories. The results can be found in Table 5. Note that to use the pretrained model provided by TimesFormer [3], we only apply our dynamic pooling scheme on TimesFormer without the message controller. We perform dynamic pruning in the $2_{th}, 5_{th}, 8_{th}, 11_{th}$ layers, with 20% tokens pruned each time on both the temporal and spatial dimension. We can see under three different settings, the lite models still maintain a good performance while the FLOPs are reduced to 25%.

As shown in Figure 1, we show DependencyViT-Lite can learn the temporal dependency from videos. The sampled 8 frames are parsed into three subtrees (in gray boxes). And we use black lines to show the dependencies between two subtrees. We see that the root subtree contains keyframes and the root frame is the most informative frame.

C.5. Related Work in NLPs

Unsupervised dependency parsing is also a long-standing task in NLP. This task aims to induce dependency trees from raw corpora that do not have human-annotated tree structures. Traditional dependency grammar induction

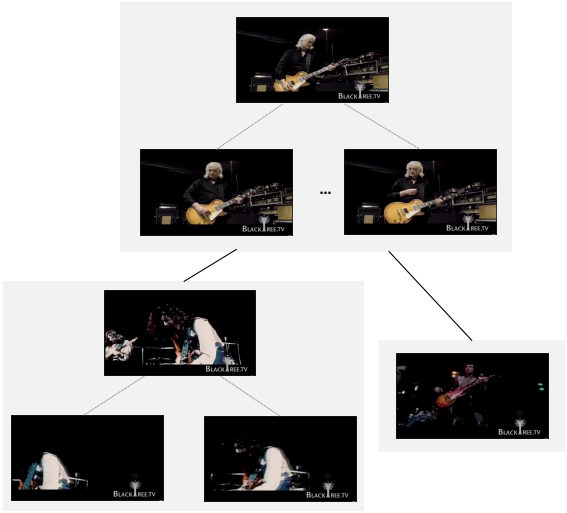

Figure 1. We show DependencyViT-Lite can learn the temporal dependency from videos. The sampled 8 frames are parsed into three subtrees (in gray boxes). And we use black lines to show the dependencies between two subtrees. We see that the root subtree contains keyframes and the root frame is the most informative frame. A few frames are enough for video recognition.

methods [1, 8, 25] are based on Dependency Model with Valence (DMV) [13]. DMV-based methods induce dependency from the statistical relation between tokens and their Part-of-Speech Tagging. Despite being very successful in the natural language domain, similar methods can not be directly applied to visual dependency induction due to two reasons: 1) DMV-based methods require discrete tokens as input, whereas visual inputs are continuous values; 2) they also heavily rely on the sequential order of input tokens, whereas visual inputs have at least two dimensions. In recent years, researchers proposed several transformer-

based unsupervised dependency parsing methods, including Structformer [23] and UDGN [22]. However, unsupervised vision dependency parsing using transformers is still very challenging because images are composed of pixels that contain no significant semantic or syntactic meaning. In contrast, natural language is composed of words expressing abstract concepts and belonging to specific syntactic roles. To overcome the challenge, DependencyViT adapts a progressive parsing schema that gradually composes low-level representations to high-level representations and makes progressive parsing decisions alongside the level of abstractness.

D. Training Details

D.1. Details of Model Configuration

In this work, we simply follow the design strategy suggested by the standard ViT (DeiT) [7, 26]. The non-overlapping patch embedding layer is implemented by stride convolution. The convolutional kernel and stride value are 16 and 16, respectively. We stack our dependency blocks with the resolution and feature dimension kept the same. We set the number of attention heads $H = 12$ and the number of dependency blocks $L = 12$ for all models. We set token dimensions $C = 192$ for the tiny model and $C = 384$ for the small model. In the head selector, we introduce a temperature hyper-parameter for the softmax function, which is set to 0.1 for all models.

For DependencyViT-Lite, similar to current hierarchical models that divide the entire architecture into four stages, we perform dynamic pruning in the $2_{th}, 5_{th}, 8_{th}, 11_{th}$ layers with a token kept number as 160, 128, 96, and 64, respectively. For dense prediction tasks, the tree architecture is still maintained by recording relationships (probability distributions) between the pruned nodes and their parents to form a complete tree. After the end of the network, we retrieve those pruned nodes by a soft aggregation from their parents, preserving the model capability and generating a dense representation. As a result, the proposed architecture can conveniently replace the backbone networks in existing methods for various vision tasks.

D.2. Image Classification on ImageNet

The ILSVRC 2012 classification dataset (ImageNet-1K) [5] consists of 1,000 classes, with a number of 1.2 million training images and 50,000 validation images.

We compare different methods on ImageNet-1K [5]. We implement our DependencyViT on the timm framework [29]. Following [6, 16, 18, 30, 35], we use the same set of data augmentation and regularization strategies used in [26] after excluding repeated augmentation [2, 11] and exponential moving average (EMA) [20]. We train all the models for 300 epochs with a batch size 2048 and use AdamW [19] as the optimizer. The weight decay is set to 0.05 and the maximal gradient norm is clipped to 1.0. We use a simple triangular learning rate schedule [24] as in [27]. The stochastic depth drop rates are set to 0.1 and 0.2 for our tiny and small models, respectively. During training, we crop images randomly to 224×224, while a center crop is used during evaluation on the validation set. For fair comparisons, neither token labeling [12] nor distillation [26] is used in all experiments.

D.3. Object Detection on COCO

The COCO dataset [17] contains over $200,000$ images labeled with object detection bounding boxes and instance segmenation masks. We evaluate our approach on the `val2017`, containing 5000 images.

We benchmark our models on object detection with COCO 2017 [17]. The pre-trained models are used as visual backbones and then plugged into two representative pipelines, RetinaNet [16] and Mask R-CNN [9]. All models are trained on the 118k training images and results reported on the 5K validation set. We follow the standard to use two training schedules, $1\times$ schedule with 12 epochs and $3\times$ schedule with 36 epochs. The same multi-scale training strategy as in [18] by randomly resizing the shorter side of the image to the range of $[480, 800]$ is used. During training, we use AdamW [19] for optimization with initial learning rate 10^{-4} and weight decay 0.05. We use 0.1 and 0.2 stochastic depth drop rates to regularize the training for our tiny and small models, respectively.

D.4. Semantic Segmentation on ADE20k

Besides the instance segmentation results above, we further evaluate our model on semantic segmentation, a task that usually requires high-resolution input and long-range interactions. ADE20K [36] is a scene-centric containing 20 thousands images annotated with 150 object categories.

We benchmark our method on ADE20K [36]. Specifically, we use UperNet [32] as the segmentation method and our DependencyViT as the backbone. For all models, we use a standard recipe by setting the input size to 512×512 and train the model for 160k iterations with batch size 16.

References

[1] W. Ammar, C. Dyer, and N. A. Smith. Conditional random field autoencoders for unsupervised structured prediction. *Advances in Neural Information Processing Systems*, 27, 2014. 3

[2] M. Berman, H. Jégou, A. Vedaldi, I. Kokkinos, and M. Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019. 4

[3] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021. 3

[4] B. Chen, P. Li, B. Li, C. Li, L. Bai, C. Lin, M. Sun, J. Yan, and W. Ouyang. Psvit: Better vision transformer via token pooling and attention sharing. *arXiv preprint arXiv:2108.03428*, 2021. 2

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 4

[6] M. Ding, B. Xiao, N. Codella, P. Luo, J. Wang, and L. Yuan. Davit: Dual attention vision transformers. *arXiv preprint arXiv:2204.03645*, 2022. 4

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 4

[8] J. He, G. Neubig, and T. Berg-Kirkpatrick. Unsupervised learning of syntactic structure with invertible neural projections. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1292–1302, 2018. 3

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1, 4

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[11] E. Hoffer, T. Ben-Nun, I. Hubara, N. Giladi, T. Hoefler, and D. Soudry. Augment your batch: Improving generalization through instance repetition. In *CVPR*, pages 8129–8138, 2020. 4

[12] Z.-H. Jiang, Q. Hou, L. Yuan, D. Zhou, Y. Shi, X. Jin, A. Wang, and J. Feng. All tokens matter: Token labeling for training better vision transformers. *NeurIPS*, 34, 2021. 4

[13] D. Klein and C. D. Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, pages 478–485, 2004. 3

[14] Z. Kong, P. Dong, X. Ma, X. Meng, W. Niu, M. Sun, B. Ren, M. Qin, H. Tang, and Y. Wang. Spvit: Enabling faster vision transformers via soft token pruning. *arXiv preprint arXiv:2112.13890*, 2021. 2

[15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 2

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 4

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1, 4

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1, 2, 4

[19] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[20] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 4

[21] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 2

[22] Y. Shen, S. Tan, S. Alessandro, L. Peng, J. Zhou, and A. Courville. Unsupervised dependency graph network. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022. 4

[23] Y. Shen, Y. Tay, C. Zheng, D. Bahri, D. Metzler, and A. Courville. Structformer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7196–7209, 2021. 4

[24] L. N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019. 4

[25] V. I. Spitkovsky, H. Alshawi, A. Chang, and D. Jurafsky. Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1281–1290, 2011. 3

[26] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 1, 2, 4

[27] A. Trockman and J. Z. Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022. 4

[28] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 1, 2

[29] R. Wightman. Pytorch image models. *URL https://github.com/rwightman/pytorch-image-models.(cited on p.)*, 2019. 4

[30] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 4

[31] S. Wu, T. Wu, H. Tan, and G. Guo. Pale transformer: A general vision transformer backbone with pale-shaped attention. *arXiv preprint arXiv:2112.14000*, 2021. 2

[32] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 2, 4

[33] T. Yu, G. Zhao, P. Li, and Y. Yu. Boat: Bilateral local attention vision transformer. *arXiv preprint arXiv:2201.13027*, 2022. 2

[34] W. Zeng, S. Jin, W. Liu, C. Qian, P. Luo, O. Wanli, and X. Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. *arXiv preprint arXiv:2204.08680*, 2022. 2

[35] P. Zhang, X. Dai, J. Yang, B. Xiao, L. Yuan, L. Zhang, and J. Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *ICCV*, 2021. 4

[36] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 1, 2, 4