Isomorphic Pruning for Vision Models – Supplementary Materials –

Gongfan Fang[1], Xinyin Ma[1], Michael Bi Mi[2], and Xinchao Wang[1]

[1] National University of Singapore
[2] Huawei Technologies Ltd.
gongfan@u.nus.edu, xinchao@nus.edu.sg

A Details of Vision Transformer Pruning

Vision transformers, in contrast to Convolutional Neural Networks, encompass a more diverse composition of substructures within their network architecture. This section presents a detailed case study on vision transformers, elucidating the isomorphic pruning process. As depicted in Figure 1, a fundamental block of vision transformers, as described in [2], comprises a multi-head attention layer and a Multi-Layer Perceptron (MLP) layer. We annotate the dimensions of intermediate features and demarcate their isomorphic groups using different colors. Owing to their heterogeneous composition, vision transformers naturally form several groups:

The Embedding Group: This group encompasses parameters responsible for generating intermediate features between modules, of witch the dimension is marked as E. In the ViT-Base model, as specified in [2], the embedding size is typically 768. The presence of residual connections mandates uniformity in embedding sizes across different blocks, necessitating simultaneous pruning. Consequently, the embedding group in a ViT-Base model comprises $E = 768$ substructures.

The MLP Group: A vision transformer includes several MLP layers, each with an identical structure. This group maps $N \times E$ embeddings to $N \times M$ intermediate results before transforming them back into E-dimensional features. Dimension M is pruned within this group to effectively reduce the model size.

Head Dimension Group: Central to the vision transformer is the self-attention module, which aggregates information across tokens. A typical self-attention module maps embeddings to Query, Key, and Value, with the dimension such as $N \times H \times Q$ for the Query. The dimensions of Q and K must be identical, while the dimension of V can be set variably. However, many implementations, such as Pytorch-Image-Models [10], require identical QKV dimensions. Therefore, this work prunes Q, K, and V concurrently. Additionally, the input dimension of the subsequent projection layer is adjusted accordingly.

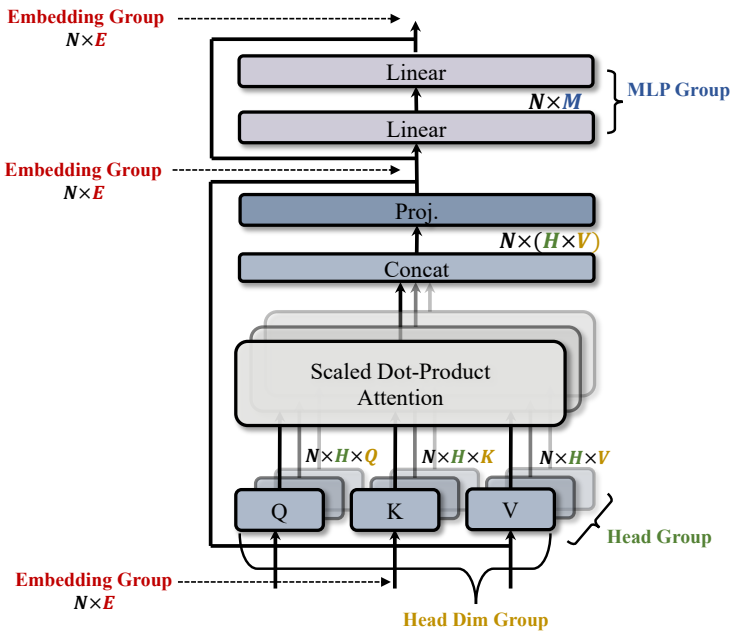

Fig. 1: The isomorphic groups in a vision transformer block. There are three groups for width pruning, which reduces the dimensions of embedding, MLP and attention. One special group works in the head level, which removes entire heads for acceleration. The shapes of intermediate features are hightlighted.

Head Group: In addition to the aforementioned groups for width pruning, the pruning of the attention head is also considered for further acceleration. This involves compressing the H dimension as shown in Figure 1.

The above analysis of substructures within vision transformers presents 4 unique isomorphic groups, associated with the dimensions E, QKV, H, and M, in which all elements have the same architecture and computational topology. In isomorphic pruning, ranking is applied within each isomorphic group for a reliable comparison. For vision transformers, the above analysis is feasible since all sub-structures are aligned with the modular design. However, for CNNs like ConvNext, ResNet, the substructures can be more complicated. Thus, our implementation automate the identification of substructures with dependency analysis [1, 3, 5] as discussed in the main paper.

Pruning Ratios for Each Isomorphic Group: The established grouping facilitates the allocation of different pruning ratios to different isomorphic groups. In our experiments, we employed the pruning ratios detailed in Table 5 to accelerate transformer models. For the DeiT-S and DeiT-T models, we keep the number of heads and embedding dimensions after pruning the same as the official models [9], while modifying the head dimensions to achieve further model compression. DeiT-2.6G uses scaled pruning ratios based on the DeiT-S configurations.

Architecture	Base Model	Emb%	Head%	Dim%
DeiT-S	DeiT-B	50%	50%	25%
DeiT-2.6G	DeiT-B	60%	60%	30%
DeiT-T	DeiT-S	50%	50%	10%
DeiT-0.6G	DeiT-T	25%	30%	30%

Table 1: The target architecture and corresponding configurations for pruning. Specifically, "Emb", "Head", and "Dim" refer to the pruning ratios for the number of heads, the size of embeddings, and the size of head dimensions.

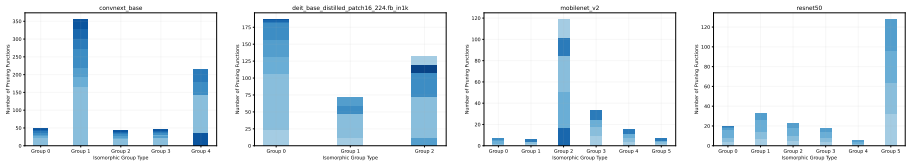

Fig. 2: The number of pruning functions in each isomorphic group. The types of pruning functions are highlighted with different colors. In isomorphic pruning, we perform ranking and pruning within each isomorphic group separately.

Swin Transformers Our method can be directly applied to Swin Transformers [6], which introduces local attention for better performance. We prune the Swin-Base to obtain 6G and 4.5G models and compare them to several baselines such as WDPruning [13], NViT-H [12] and X-Pruner [14]. The proposed method achieves better results compared to existing methods and pre-trained baselines.

B Details of CNN Pruning

This section elaborates on the pruning strategies applied to ConvNext, ResNet, and MobileNet-v2 networks. Compared to Transformers, Convolutional Neural Networks used in our experiments are more irregular, with intricated internal connections. We model the substructures are modeled as graphs, which facilitates automation of the pruning process, thereby obviating the need for cumbersome manual analysis. Figure 2 visually illustrates the statistics of the detected isomorphic groups. In isomorphic pruning, we seperately eliminate parameters from each distinct group. These groups encompass multiple sub-structures, varying in the number of layers. As elucidated in the main paper, each layer can be subject to two types of pruning functions. Assessing the relative significance of these substructures presents a challenge. Figure 2 illustrates the number of different pruning functions across various isomorphic groups. The x-axis represents the group ID, denoting isomorphic groups with identical graph structures. The y-axis quantifies the total number of pruning functions for each group, corresponding to the number of total nodes in the graph modeling. Different pruning functions are colored for better illustration. For instance, pruning functions targeting the

Method	#Params (M)	MACs (G)	Acc (%)
Swin-B†	87.77	15.48	83.42
Swin-T†	28.29	4.51	81.19
Swin-T (Ours)	24.66	4.47	81.32
X-Pruner [14]	N/A	3.20	80.70

Table 2: Pruning results for Swin Transformers pre-trained on ImageNet-1K.

Training Configs	DeiT	ConvNext	ResNet-50	MobileNet-v2	Swin
optimizer	AdamW	AdamW	SGD	SGD	AdamW
base learning rate	0.0005	0.001	0.08	0.036	0.0005
weight decay	0.05	0.05	1e-4	4e-5	0.05
optimizer momentum	(0.9, 0.999)	(0.9, 0.999)	0.9	0.9	(0.9, 0.999)
batch size	2048	1024	1024	4096	2048
training epochs	300	300	100	300	300
learning rate schedule	cosine	cosine	30,60,90	cosine	cosine
warmup epochs	0	0	0	0	0
layer-wise lr decay	0	0	0	0	0
randaugment	✓	✓	None	None	✓
mixup	0.2	0.2	None	None	0.2
cutmix	1.0	1.0	None	None	1.0
random erasing	0.25	0.25	None	None	0.25
label smoothing	0.1	0.1	None	None	0.1
stochastic depth	0	0.1 (S) / 0.4 (T)	None	None	0.1
layer scale	None	None	None	None	None
gradient clip	5	None	None	None	5
exp. mov. avg. (EMA)	None	0.9999	None	None	None

Table 3: Pruning Plain Vision Transformers. All pruned models are only fine-tuned on ImageNet-1K.

input and output dimensions of a linear layer are distinguished by unique colors, since the pruning is performed on different dimensions.

Observations in Figure 2 reveal significant variability in the composition of each isomorphic group, potentially leading to unreliable rankings if a straightforward global pruning approach is employed. Additionally, the varying sizes of the isomorphic groups suggest that independent pruning within each group achieves a more balanced acceleration across different substructures, akin to local pruning. This approach also retains the flexibility to dynamically adjust pruning ratios for different layers.

C Experimental Details

Training. This section further details the training process and hyper-parameters in our experiments. We report the training configurations including optimizer, learning rate, and augmentation in Table 3. All models are fine-tuned with 8 RTX

A5000 GPUs, with Automatic Mixed Precision implemented by PyTorch [7]. For DeiT and ConvNext, we use strong augmentations as mentioned in the original paper [9, 11], but did not deploy warmup, layer-wise lr decay and layer scale for training. ResNet and MobileNet-v2 were trained with weak augmentation described in [4, 8].

Latency Test For the Latency test on GPU, we forward the model a batch size of 256 for 20-step warmup and 100-step experiments. We report the average execution time of the 100 rounds. For CPU test, we deploy a batch size of 8 and follow the same principle as GPU testing.

D Limitations

In this study, we empirically examine the impact of isomorphic sub-structures on pruning. Structural similarity might not be the sole determinant of importance distribution. Factors such as the training methodology, regularization techniques, and network depth can also play significant roles in shaping different distributions. While our experiments demonstrate the effectiveness of isomorphic pruning, further investigation in the future into these additional factors is necessary for a more comprehensive framework.

References

1. Chen, T., Liang, L., Ding, T., Zhu, Z., Zharkov, I.: Otov2: Automatic, generic, user-friendly. arXiv preprint arXiv:2303.06862 (2023) 2
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 1
3. Fang, G., Ma, X., Song, M., Mi, M.B., Wang, X.: Depgraph: Towards any structural pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16091–16101 (2023) 2
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 5
5. Liu, L., Zhang, S., Kuang, Z., Zhou, A., Xue, J.H., Wang, X., Chen, Y., Yang, W., Liao, Q., Zhang, W.: Group fisher pruning for practical network compression. In: International Conference on Machine Learning. pp. 7021–7032. PMLR (2021) 2
6. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021) 3
7. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019) 5
8. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018) 5
9. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. vol. 139, pp. 10347–10357 (July 2021) 2, 5
10. Wightman, R.: Pytorch-image-models. https://github.com/huggingface/pytorch-image-models 1
11. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: Convnext v2: Co-designing and scaling convnets with masked autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16133–16142 (2023) 5
12. Yang, H., Yin, H., Shen, M., Molchanov, P., Li, H., Kautz, J.: Global vision transformer pruning with hessian-aware saliency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18547–18557 (2023) 3
13. Yu, F., Huang, K., Wang, M., Cheng, Y., Chu, W., Cui, L.: Width & depth pruning for vision transformers. In: AAAI Conference on Artificial Intelligence (AAAI). vol. 2022 (2022) 3
14. Yu, L., Xiang, W.: X-pruner: explainable pruning for vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24355–24363 (2023) 3, 4