MDP: Multidimensional Vision Model Pruning with Latency Constraint

Supplementary Material

6. Latency Modeling: Ours v.s. Prior Arts

In the main paper, we highlight that prior approaches [35, 61, 62] rely on imprecise latency estimation, leading to sub-optimal accuracy-latency trade-offs. These methods use a simplistic latency model that assumes a linear relationship between latency and the number of output channels. However, this approach is limited as it cannot simultaneously account for multiple prunable dimensions and is restricted to CNNs.

When pruning transformers, it is essential to model simultaneous variations across multiple dimensions, such as embedding size, query/key dimensions, value dimensions, number of heads, and MLP size, which cannot be captured by a linear model. Even for CNNs, such linear modeling lacks precision. To illustrate, we provide a visualization in Fig. 5, demonstrating the limitations of prior latency modeling. Specifically, these models only account for changes in output channels while neglecting concurrent variations in input channels caused by pruning in preceding layers.

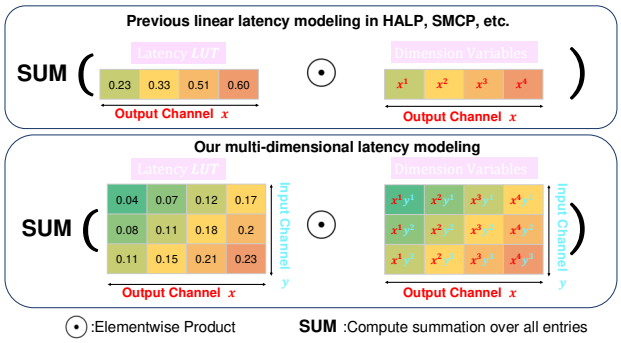

Figure 5. **Comparison in latency modeling between ours and prior arts [35, 61]**. Example with CNNs.

7. Solving MINLPs

In order to solve our MINLP program, we leverage the method called OA [18, 24] which decomposes the problem into solving an alternating finite sequence of NLP subproblems and relaxed versions of MILP master program. We also leverage a method called Feasibility Pump [5] to to expedite the process of finding feasible solutions within constraints.

The entire program could be efficiently solved on common CPUs for modern network sizes. For instance, when applied to a model like ResNet50 [30], the entire optimization problem can be solved in approximately 5 seconds on an Intel Xeon E5-2698 CPU. Moreover, in Table 4, we show the overhead of solving MINLPs for ResNet50 and DEIT-Base relative to their training times. As shown, solving the MINLP program is efficient, for example taking only $\mathbf{0.1}\%$ of DEIT-Base's total training time. Theoretically, as established in [41], the problem remains tractable and scalable as long as the objective and constraint functions are convex separable or gated with a switch variable. Here, block decision variables act as gates, simplifying the solving.

8. Efficiency of LUTs Preparation

In our framework, we leverage latency look-up tables (LUTs) to model the latency impacts from different pruning decisions. In LUTs, the inference latency for *all possible* local subnet structures are recorded. For CNN layers for example, this includes input and output channel counts from 0 up to their original values. While this may seem computationally expensive, the process is efficient in practice. Firstly, LUTs are generated *only once* to measure target hardware latency and can be reused for future pruning runs, with minimal overhead even without parallelization (Table 4). Moreover, our *model-specific decomposition* as shown in Eqn.3,4 significantly reduces the total number of LUT entries, and we further adopt channel grouping [35, 61, 69] to cluster channels of similar importance into single elements, minimizing overhead. Additionally, LUTs can be reused across related architectures (e.g., ResNet101 shares many LUT entries with ResNet50).

9. Adaptation to CPU

To demonstrate adaptability, we evaluated our method on the **CPU** platform using an Intel Xeon E5 processor. The results, summarized in Table 5, indicate substantial performance gains over prior work. Notably, compared to HALP [61], our method achieves more than double the FPS (**118.2** vs. 45.9) while also attaining a higher Top-1 accuracy (**75.2** vs. 74.5). This improvement is even more pronounced than the speedup observed on GPU. We attribute this to the reduced number of blocks and the smaller overall network depth, which make the network more CPU-friendly. These results highlight the effectiveness of MDP in generalizing across both CPU and GPU platforms.

10. Pruning Efficient CNNs

We include results for pruning efficient CNNs like MobileNet-V1 and MobileNet-V2 in Table 7. Our method consistently outperforms prior approaches. For example, on MobileNet-V2, compared to UPDP [50], we achieve

Train (mins)	Solve MINLP (mins)	LUT Prep. (mins)
ResNet50 trained for 90 epochs		
667(×1)	0.09(×0.01%)	152(×23%)
DEIT-Base trained for 300 epochs		
7614(×1)	7.53(×0.1%)	320(×4.2%)

Table 4. Overhead of solving MINLP and preparing the Look-up Table(LUT) for ResNet50 and DEIT-B. LUT only needs to be generated once. CPU in use is Intel Xeon E5-2698.

Method	Top-1↑	FPS↑
AutoSlim [84]	74.0	33.3
EagleEye [42]	74.2	31.3
MetaPrune [53]	73.4	33.3
HALP-70% [61]	74.5	45.9
Ours	**75.2**	**118.2**

Table 5. **Generalization of MDP on CPU Platform.** FPS measured on Intel CPU Xeon E5. Ours attains significant improvements from prior arts, specifically in speedups.

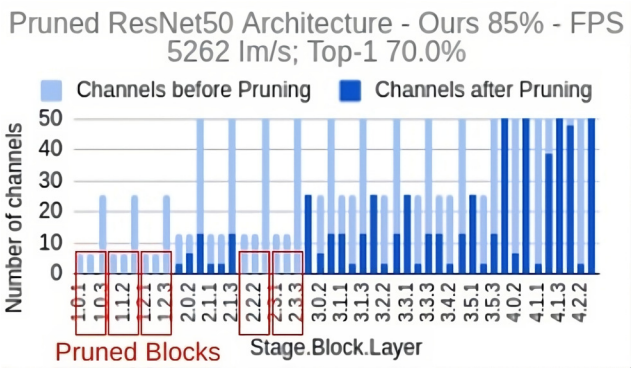

Figure 6. Pruned architecture of ResNet50 on ImageNet.

slightly higher Top-1 (**72.6** vs. 72.5) and improved speed (**2.63** vs. 2.50).

Furthermore, our approach focuses on aggressively pruning moderate or large models to derive efficient ones. This strategy offers superior accuracy-speed tradeoffs compared to directly training efficient models from scratch, as shown in Fig. 8.

11. Pruned Structure Analysis

To provide insights into our pruning algorithm, we present the pruned structure of ResNet50 on ImageNet in Fig. 6, targeting an 85% latency reduction. The figure shows that pruning is predominantly concentrated in the shallower layers, contrary to the common expectation of deeper layers collapsing due to smaller gradients. This indicates that when latency is strict constraint, the pruning pattern is influenced not only by importance ranking but also by latency considerations. Since earlier layers process larger feature maps and are generally more latency-intensive, they are pruned more aggressively than later stages.

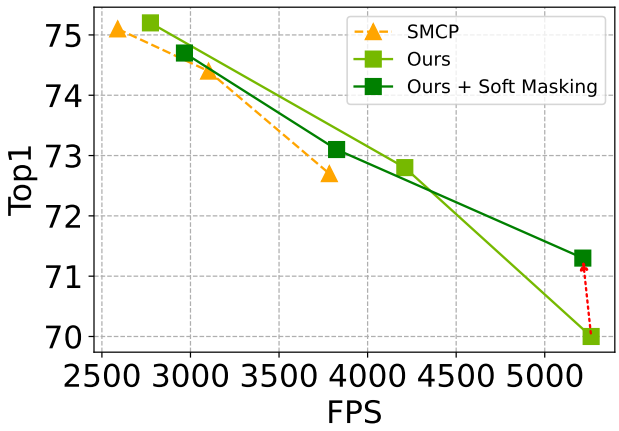

Figure 7. **Results of ours with soft masking on ImageNet with ResNet50**. We investigate the effectiveness of soft masking techniques in our method and observe improvement in Top1 at a high FPS level. Top-right is better.

12. Training Detail

For reproducibility, we provide detailed hyperparameters and fine-tuning optimization settings in Table 6, adhering to the baseline configurations. Specifically, when fine-tuning the pruned DEIT-Base model on ImageNet, we incorporate the distillation loss from the convolutional RegNetY160 model, as described in the original paper.

13. Integration with Soft Masking

Recent advances in pruning [31, 35, 36, 39, 87] have increasingly adopted soft masking techniques to retain the learning capacity of pruned models by not directly removing the pruned weights. Notably, SMCP [35] integrates this method into the HALP hardware-aware pruning framework, resulting in an enhanced accuracy-latency tradeoff for pruned models. Here, we explore the potential of soft masking to enhance our model's performance.

We conduct this study on ImageNet with ResNet50 and depict the Pareto frontier of FPS versus Top-1 in Figure 7. For clarity, we also include the performance of SMCP [35] and ours. The results reveal that soft masking offers limited advantages at lower FPS levels with modest pruning ratios and latency reduction. Nonetheless, targeting higher FPS levels leads to notable improvements in Top-1 accuracy. This outcome may be attributed to the Taylor channel importance score we employed [59], which gauges parameter significance based on its impact on loss. Though it maintains precision with minor parameter deletions, its reliability may

Model	Dataset	Epochs	Optimizer, Momentum, WeightDecay	Learning Rate
ResNet50	ImageNet	90	SGD, 0.875, $3e-5$	Init=1.024, LinearDecay
DEIT-Base	ImageNet	300	AdamW, 0.9, 0.05	Init=$2e-4$, CosineAnneal
SSD512	PascalVOC	800	SGD, 0.9, $2e-3$	Init=$8e-3$, StepDecay
StreamPetr	NuScenes	60	AdamW, 0.9, 0.01	Init=$6e-4$, CosineAnneal

Table 6. Training Detail.

Model	Top-1	Speedup
MobileNet-V1	72.6	×1.00
MetaPrune[49]	66.1	×2.06
HALP-42%[57]	68.3	×2.32
SMCP-40%[34]	68.3	×2.39
Ours	**68.5**	**×2.41**
MobileNet-V2-1.4	75.3	×1.00
MBV2-1.4-DS-E	72.2	×2.50
UPDP-P11 [50]	72.5	×2.50
Ours	**72.6**	**×2.63**

Table 7. Results on pruning efficient CNNs.

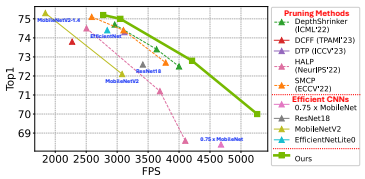

Figure 8. Top-right is better. Recent pruning works surpass efficient CNNs in speed-accuracy tradeoffs on ImageNet, with ours achieving the best.

diminish when a larger number of parameters are pruned concurrently. The iterative reassessment inherent to the soft masking technique may counteract this shortcoming.

References

[1] Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems*, pages 2270–2278, 2016. 2

[2] David E Bernal, Qi Chen, Felicity Gong, and Ignacio E Grossmann. Mixed-integer nonlinear decomposition toolbox for pyomo (mindtpy). In *Computer Aided Chemical Engineering*, pages 895–900. Elsevier, 2018. 3, 6

[3] David E Bernal, Stefan Vigerske, Francisco Trespalacios, and Ignacio E Grossmann. Improving the performance of dicopt in convex minlp problems using a feasibility pump. *Optimization Methods and Software*, 35(1):171–190, 2020. 3

[4] Pierre Bonami, Lorenz T Biegler, Andrew R Conn, Gérard Cornuéjols, Ignacio E Grossmann, Carl D Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas Sawaya, et al. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete optimization*, 5(2):186–204, 2008.

[5] Pierre Bonami, Gérard Cornuéjols, Andrea Lodi, and François Margot. A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming*, 119(2):331–352, 2009. 3, 6, 1

[6] Samuel Burer and Adam N Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012. 2, 3

[7] Michael R Bussieck, Armin Pruessner, et al. Mixed-integer nonlinear programming. *SIAG/OPT Newsletter: Views & News*, 14(1):19–22, 2003. 2, 3

[8] Michael L Bynum, Gabriel A Hackebeil, William E Hart, Carl D Laird, Bethany L Nicholson, John D Siirola, Jean-Paul Watson, David L Woodruff, et al. *Pyomo-optimization modeling in python*. Springer, 2021. 3, 6

[9] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 6, 7

[10] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 3, 7

[11] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12270–12280, 2021. 7

[12] Shi Chen and Qi Zhao. Shallowing deep networks: Layer-wise pruning based on feature representations. *IEEE transactions on pattern analysis and machine intelligence*, 41(12): 3048–3056, 2018. 2, 3

[13] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34:19974–19988, 2021. 3, 7

[14] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *ICML*, pages 2285–2294. PMLR, 2015. 3

[15] Ting-Wu Chin, Ruizhou Ding, Cha Zhang, and Diana Marculescu. Towards efficient model compression via learned global ranking. In *CVPR*, pages 1518–1528, 2020. 2

[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. 6, 7

[17] 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. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3, 4

[18] Marco A Duran and Ignacio E Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36:307–339, 1986. 3, 4, 1

[19] Claudia D'Ambrosio and Andrea Lodi. Mixed integer nonlinear programming tools: an updated practical overview. *Annals of Operations Research*, 204:301–320, 2013. 3

[20] Sara Elkerdawy, Mostafa Elhoushi, Abhineet Singh, Hong Zhang, and Nilanjan Ray. To filter prune, or to layer prune, that is the question. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2, 3, 6, 7

[21] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 6, 7

[22] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16091–16101, 2023. 2

[23] Gongfan Fang, Xinyin Ma, Michael Bi Mi, and Xinchao Wang. Isomorphic pruning for vision models. In *European Conference on Computer Vision*, pages 232–250. Springer, 2025. 1, 2, 3, 7

[24] Roger Fletcher and Sven Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical programming*, 66:327–349, 1994. 3, 4, 1

[25] Oktay Günlük and Jeff Linderoth. Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical programming*, 124:183–205, 2010. 3

[26] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. 2015. 2

[27] Adam Harley, Yang You, Yang Zheng, Xinglong Sun, Nikhil Raghuraman, Sheldon Liang, Wen-Hsuan Chu, Suya You, Achal Dave, Pavel Tokmakov, et al. Tag: Tracking at any granularity. *ArXiv Preprint*, 3, 2024. 1

[28] Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. *NeurIPS*, 5, 1992. 2

[29] Haoyu He, Jianfei Cai, Jing Liu, Zizheng Pan, Jing Zhang, Dacheng Tao, and Bohan Zhuang. Pruning self-attentions into convolutional layers in single path. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 4, 6

[31] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. In *IJCAI*, 2018. 2

[32] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *CVPR*, pages 4340–4349, 2019. 2

[33] Yang He, Yuhang Ding, Ping Liu, Linchao Zhu, Hanwang Zhang, and Yi Yang. Learning filter pruning criteria for deep convolutional neural networks acceleration. In *CVPR*, pages 2009–2018, 2020. 2

[34] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 8

[35] Ryan Humble, Maying Shen, Jorge Albericio Latorre, Eric Darve, and Jose Alvarez. Soft masking for cost-constrained channel pruning. In *European Conference on Computer Vision*, pages 641–657. Springer, 2022. 2, 3, 4, 5, 6, 7, 8, 1

[36] Jangho Kim, Jayeon Yoo, Yeji Song, KiYoon Yoo, and Nojun Kwak. Dynamic collective intelligence learning: Finding efficient sparse model via refined gradients for pruned weights. *arXiv preprint arXiv:2109.04660*, 2021. 2

[37] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[38] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European Conference on Computer Vision*, pages 620–640. Springer, 2022. 3

[39] Aditya Kusupati, Vivek Ramanujan, Raghav Somani, Mitchell Wortsman, Prateek Jain, Sham Kakade, and Ali Farhadi. Soft threshold weight reparameterization for learnable sparsity. In *ICML*, pages 5544–5555. PMLR, 2020. 2

[40] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *NeurIPS*, pages 598–605, 1990. 2

[41] Jon Lee and Sven Leyffer. *Mixed integer nonlinear programming*. Springer Science & Business Media, 2011. 2, 3, 1

[42] Bailin Li, Bowen Wu, Jiang Su, and Guangrun Wang. Eagleeye: Fast sub-net evaluation for efficient neural network pruning. In *ECCV*, pages 639–654, 2020. 2, 3, 6

[43] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *ICLR*, 2017. 2

[44] Yunqiang Li, Jan C van Gemert, Torsten Hoefler, Bert Moons, Evangelos Eleftheriou, and Bram-Ernst Verhoef. Differentiable transportation pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16957–16967, 2023. 6

[45] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022. 7

[46] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. Hrank: Filter pruning using high-rank feature map. In *CVPR*, pages 1529–1538, 2020. 2, 4

[47] Mingbao Lin, Bohong Chen, Fei Chao, and Rongrong Ji. Training compact cnns for image classification using dynamic-coded filter fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10478–10487, 2023. 6

[48] Shaohui Lin, Rongrong Ji, Yuchao Li, Yongjian Wu, Feiyue Huang, and Baochang Zhang. Accelerating convolutional networks via global & dynamic filter pruning. In *IJCAI*, page 8. Stockholm, 2018. 2

[49] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d v2: Recurrent temporal fusion with sparse model. *arXiv preprint arXiv:2305.14018*, 2023. 8

[50] Ji Liu, Dehua Tang, Yuanxian Huang, Li Zhang, Xiaocheng Zeng, Dong Li, Mingjie Lu, Jinzhang Peng, Yu Wang, Fan Jiang, et al. Updp: A unified progressive depth pruner for cnn and vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13891–13899, 2024. 1, 3

[51] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 1, 6, 7

[52] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Aqi Gao, Tiancai Wang, and Xiangyu Zhang. Petrv2: A unified framework for 3d perception from multi-camera images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3262–3272, 2023. 8

[53] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *ICCV*, pages 3296–3305, 2019. 6, 2

[54] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3, 7

[55] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1

[56] Jian-Hao Luo and Jianxin Wu. Neural network pruning with residual-connections and limited-data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1458–1467, 2020. 4

[57] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *NeurIPS*, 32:14014–14024, 2019. 2, 3

[58] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *ICML*, pages 2498–2507. PMLR, 2017. 2

[59] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *CVPR*, pages 11264–11272, 2019. 1, 2, 4

[60] Pavlo Molchanov, Jimmy Hall, Hongxu Yin, Jan Kautz, Nicolo Fusi, and Arash Vahdat. Lana: latency aware network acceleration. In *European Conference on Computer Vision*, pages 137–156. Springer, 2022. 3

[61] Maying Shen, Hongxu Yin, Pavlo Molchanov, Lei Mao, Jianna Liu, and Jose Alvarez. Structural pruning via latency-saliency knapsack. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[62] Maying Shen, Lei Mao, Joshua Chen, Justin Hsu, Xinglong Sun, Oliver Knieps, Carmen Maxim, and Jose M Alvarez. Hardware-aware latency pruning for real-time 3d object detection. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2023. 2, 3, 5, 8, 1

[63] Prabhakant Sinha and Andris A Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, 1979. 3

[64] Xinglong Sun. Pruning for better domain generalizability. *arXiv preprint arXiv:2306.13237*, 2023. 2

[65] Xinglong Sun and Humphrey Shi. Towards better structured pruning saliency by reorganizing convolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2204–2214, 2024. 2, 4

[66] Xinglong Sun, Ali Hassani, Zhangyang Wang, Gao Huang, and Humphrey Shi. Disparse: Disentangled sparsification for multitask model compression. In *CVPR*, pages 12382–12392, 2022. 2

[67] Xinglong Sun, Jean Ponce, and Yu-Xiong Wang. Revisiting deformable convolution for depth completion. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1300–1306. IEEE, 2023. 1

[68] Xinglong Sun, Adam W Harley, and Leonidas J Guibas. Refining pre-trained motion models. *arXiv preprint arXiv:2401.00850*, 2024. 1

[69] Xinglong Sun, Maying Shen, Hongxu Yin, Lei Mao, Pavlo Molchanov, and Jose M Alvarez. Advancing weight and channel sparsification with enhanced saliency. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025. 1, 2, 6

[70] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, pages 2820–2828, 2019. 2

[71] Hui Tang, Yao Lu, and Qi Xuan. Sr-init: An interpretable layer pruning method. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 2, 3

[72] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 3, 6, 7

[73] Huan Wang, Can Qin, Yulun Zhang, and Yun Fu. Neural pruning via growing regularization. In *ICLR*, 2021. 2, 6

[74] Haoxuan Wang, Pengyang Ling, Xin Fan, Tao Tu, Jinjin Zheng, Huaian Chen, Yi Jin, and Enhong Chen. All-in-one hardware-oriented model compression for efficient multi-hardware deployment. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 6

[75] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. *arXiv preprint arXiv:2303.11926*, 2023. 1, 2, 6, 8

[76] Wenxiao Wang, Shuai Zhao, Minghao Chen, Jinming Hu, Deng Cai, and Haifeng Liu. Dbp: Discrimination based block-level pruning for deep model acceleration. *arXiv preprint arXiv:1912.10178*, 2019. 2, 3, 6, 7

[77] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 7

[78] Cheng-En Wu, Azadeh Davoodi, and Yu Hen Hu. Block pruning for enhanced efficiency in convolutional neural networks. *arXiv preprint arXiv:2312.16904*, 2023. 2, 3

[79] Yu-Cheng Wu, Chih-Ting Liu, Bo-Ying Chen, and Shao-Yi Chien. Constraint-aware importance estimation for global filter pruning under multiple resource constraints. In *CVPR Workshops*, pages 686–687, 2020. 2, 3

[80] Pengtao Xu, Jian Cao, Fanhua Shang, Wenyu Sun, and Pu Li. Layer pruning via fusible residual convolutional block

for deep neural networks. *arXiv preprint arXiv:2011.14356*, 2020. 2, 3, 6, 7

[81] Huanrui Yang, Hongxu Yin, Maying Shen, Pavlo Molchanov, Hai Li, and Jan Kautz. Global vision transformer pruning with hessian-aware saliency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18547–18557, 2023. 1, 2, 3, 4, 7

[82] Tien-Ju Yang, Andrew Howard, Bo Chen, Xiao Zhang, Alec Go, Mark Sandler, Vivienne Sze, and Hartwig Adam. Netadapt: Platform-aware neural network adaptation for mobile applications. In *ECCV*, pages 285–300, 2018. 2, 3

[83] Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. *NeurIPS*, 32, 2019. 2

[84] Jiahui Yu and Thomas Huang. Autoslim: Towards one-shot architecture search for channel numbers. *NeurIPS Workshop*, 2019. 2, 6

[85] Shixing Yu, Tianlong Chen, Jiayi Shen, Huan Yuan, Jianchao Tan, Sen Yang, Ji Liu, and Zhangyang Wang. Unified visual transformer compression. *arXiv preprint arXiv:2203.08243*, 2022. 3

[86] Chuanyang Zheng, Kai Zhang, Zhi Yang, Wenming Tan, Jun Xiao, Ye Ren, Shiliang Pu, et al. Savit: Structure-aware vision transformer pruning via collaborative optimization. *Advances in Neural Information Processing Systems*, 35:9010–9023, 2022. 3, 7

[87] Aojun Zhou, Yukun Ma, Junnan Zhu, Jianbo Liu, Zhijie Zhang, Kun Yuan, Wenxiu Sun, and Hongsheng Li. Learning n: m fine-grained structured sparse neural networks from scratch. *ICLR*, 2021. 2

[88] Mingjian Zhu, Yehui Tang, and Kai Han. Vision transformer pruning. *arXiv preprint arXiv:2104.08500*, 2021. 3