


AMD: Automatic Multi-step Distillation of Large-scale Vision Models

Supplementary Material

Cheng Han^{1,2} , Qifan Wang³, Sohail A. Dianat², Majid Rabbani², Raghuveer M. Rao⁴, Yi Fang⁵, Qiang Guan⁶, Lifu Huang⁷, and Dongfang Liu^{*2}

¹ University of Missouri – Kansas City

² Rochester Institute of Technology

³ Meta AI

⁴ DEVCOM Army Research Laboratory

⁵ Santa Clara University

⁶ Kent State University

⁷ Virginia Tech

This supplementary contains additional experimental results and discussions of our ECCV 2024 submission: *AMD: Automatic Multi-step Distillation of Large-scale Vision Models*, organized as follows:

- §1 presents more details on the implementation of *Structural Pruning*.
- §2 provides additional experiments on the effectiveness of structural pruning.
- §3 considers the effect of joint optimization with parameter sharing during AMD, and further gathers an additional architectural design.
- §4 discusses the efficiency of optimal selection.
- §5 includes additional diagnostic experiments of AMD.
- §6 adds more discussions on the current multi-teacher distillation approach.
- §7 provides extensive results on traditional comparisons on convolutional neural networks. We further conduct additional experiments on semantic segmentation.
- §8 discusses related license, reproducibility, technical contributions, social impact, complexity, limitations and directions of our future work.

1 Details on Pruning

We follow common practices [3, 8], and arrange structural pruning based on the ranking of important parameters, defined by important scores. Specifically, we first initialize learnable vectors/masks to ones in self-attention head and feed-forward layer. Formally, for a self-attention head with $\hat{\mathbf{X}}$ as input, we have:

$$\begin{aligned} \mathbf{Z} &= \text{SelfAttention}(\hat{\mathbf{X}}) \\ &= \sum_i^I \psi_i \cdot \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d}}\right) V_h, \end{aligned} \quad (1)$$

where $Q_h = \hat{\mathbf{X}} \mathbf{W}_Q$, $K_h = \hat{\mathbf{X}} \mathbf{W}_K$, $V_h = \hat{\mathbf{X}} \mathbf{W}_V$. \mathbf{W}_Q , \mathbf{W}_K and \mathbf{W}_V are projection matrices. i represents the i -th head among I head, and ψ_i is the learnable mask. After

the self-attention layer, a feed-forward network is applied with two fully-connected layers [5, 7, 18], transforming the features along the embedding dimension. We insert learnable mask in feed-forward layer as:

$$\begin{aligned} \mathbf{Y} &= \text{FeedForward}(\mathbf{Z}) \\ &= \sum_n^N \delta_n \cdot f(\mathbf{Z}\mathbf{W}_1)\mathbf{W}_2, \end{aligned} \quad (2)$$

where $f(\cdot)$ is a non-linear activation function. n represents the n -th feed-forward layer among N layers, and δ_n is the learnable mask. The importance score is defined as the expected sensitivity of the model to the mask variables [8, 21]:

$$\begin{aligned} S_i^I &= \mathbb{E}_{x \sim \mathcal{D}_x} \left| \frac{\partial \mathcal{L}(x)}{\partial \psi_i} \right|, \\ S_n^N &= \mathbb{E}_{x \sim \mathcal{D}_x} \left| \frac{\partial \mathcal{L}(x)}{\partial \delta_n} \right|, \end{aligned} \quad (3)$$

where \mathcal{L} is the loss function, \mathcal{D}_x is the training data distribution [21], and \mathbb{E} is the expectation. The importance scores assigned to each self-attention and feed-forward layer reflect their contribution to the model performance. Specifically, a low importance score suggests a minor or even negative impact while a high importance score stands for the significance of corresponding structure.

We take a global ranking in Vision Transformer (ViT) [5], and a partial ranking within each stage for Swin Transformer [20] due to the unique design of self-attention layers. Before ranking the importance scores for the structures from the same type (*i.e.*, separate for self-attention and feed-forward layer), a l_2 normalization is taken for balanced pruning [23]. In our study, we adopt a pruning method that, while straightforward, has proven to be efficacious. Although more sophisticated pruning approaches [6, 36] might offer incremental improvements in performance, our primary aim lies in delineating the optimal training schedule for large-scale vision models. For CNN-based architectures, we follow common practices [17, 23] for pruning. Specifically, we utilize the greedy pruning stated in [17]. Consequently, we posit that the exploration of advanced pruning methods may represent a fruitful avenue for future research.

2 Effect of Structural Pruning

In order to verify the effectiveness of our proposed *Structural Pruning*, we further conduct experiments on comparing AMD and MMD with pruning via dropping layers and hidden dimensions (*i.e.*, by restricting the numbers of layers and hidden dimensions, one can significantly reduce the GPU overhead of models), denoted as ViT – Base_{2L}. The results are shown in Table 1. As seen, with a lower computational demand (*i.e.*, 2.64G FLOPS *vs* 2.93G FLOPS), traditional practices with structural pruning get noticeable performance advantages when comparing to pruning via dropping layers. For example, ViT – Base_{2L} DKD [42] is 1.16% lower in accuracy when comparing to ViT – Base_{15%} DKD [42]. Furthermore, both MMD and AMD present consistently

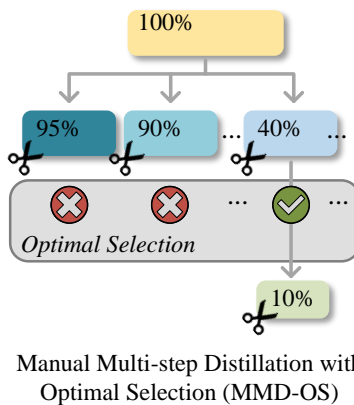


Fig. 1: Manual Multi-step Distillation with Optimal Selection (MMD-OS) can be considered as the intermediate stage between our proposed MMD and AMD.

superior performance when comparing to ViT – Base_{2L} practices. These two observations indicate that the simply drop from transformer layers could deteriorate the performance of knowledge distillation. Our structure pruning method, on the other hand, offers an efficient yet simple solution in §1.

3 Effect of Joint Optimization

A point of consideration regarding our proposed AMD is the effectiveness of the *Joint Optimization* proposed in §3.3. We thus devise an intermediate architectural design that strategically positions itself between our previously proposed MMD and AMD, named MMD-OS. The primary objective is to conduct a focused ablation study, isolating and evaluating the impact **only** from the *Joint Optimization* component. Specifically, it identifies a set of teacher-assistants with different scales and selects the optimal teacher assistant through *Optimal Selection* (Figure 1). We follow our experimental settings in §4.3 and present results on ViT-Base trained on CIFAR-10 and CIFAR-100 [15]. The results are shown in Table 2. As seen, while MMD-OS can demonstrate a marginally superior performance relative to AMD (*i.e.*, 95.53% *vs* 95.52% on CIFAR-10), its significant computational overhead associated with MMD-OS cannot be overlooked. MMD-OS incurs a high computational cost (*i.e.*, 10× *vs* 2.2×). Consequently, we argue that AMD represents the most efficient and effective approach among all the solutions proposed in our study.

4 Effect of Optimal Selection

We further replace the NPSD-based *Optimal Selection* with manual selection, *i.e.*, distill the student with all teacher-assistant candidates produced by the *Joint Optimization*, and

Table 1: We further investigate the impact of structural pruning techniques by comparing with pruning via dropping layers and hidden dimensions (see §2 on ViT-Base CIFAR-100 [15]). We apply 15% student model scale since it has similar computational cost *w.r.t.* 2-layer transformer student (*i.e.*, 2.64G FLOPS *vs* 2.93G FLOPS). The highest accuracy among all approaches are shown in **bold**. The second highest are shown in underline. Same for Table 3.

Method	FLOPs	CIFAR-100 [15] $t_{\text{op-1}}$	GPU hours
ViT – Base _{100%} (teacher)	17.6G	98.01%	-
ViT – Base _{2L} KD [arXiv15] [12]	2.93G	59.85%	1×
ViT – Base _{2L} DKD [CVPR22] [42]	2.93G	69.10%	1×
ViT – Base _{2L} CRD [ICLR20] [31]	2.93G	74.15%	1×
ViT – Base _{2L} TAKD [AAAI20] [22]	2.93G	69.73%	20×
ViT – Base _{15%} KD [arXiv15] [12]	2.64G	60.13%	1×
ViT – Base _{15%} DKD [CVPR22] [42]	2.64G	70.26%	1×
ViT – Base _{15%} CRD [ICLR20] [31]	2.64G	78.40%	1×
ViT – Base _{15%} TAKD [AAAI20] [22]	2.93G	71.02%	20×
ViT – Base _{15%} MMD	2.64G	80.25%	20×
ViT – Base _{15%} AMD	2.64G	<u>80.23%</u>	2.2×

Table 2: Performance and GPU overhead *w.r.t.* different proposed architectures.

Method	Dataset	Performance	GPU hours
ViT – Base _{100%} (teacher)	CIFAR-10 [15]	98.01%	-
– MMD		95.54%	20×
– MMD-OS ($m = 9$)		95.54%	10×
– AMD ($m = 9$)		95.52%	2.2×
ViT – Base _{100%} (teacher)	CIFAR-100 [15]	89.33%	-
– MMD		80.11%	20×
– MMD-OS ($m = 9$)		80.10%	10×
– AMD ($m = 9$)		80.19%	2.2×

choose the best student. We concur that the derived baseline ensures no degradation in student performance with approximately 10× training time.

5 Additional Diagnostic Experiments

5.1 Impact of Different Loss Components.

To analyze the impact of different loss components, we further conduct ablation studies on three variants of AMD: ❶ AMD without cross-entropy loss \mathcal{L}_{ce} ; ❷ AMD without logit-based loss \mathcal{L}_{logit} ; ❸ AMD without feature-mimicking based \mathcal{L}_{feat} . As seen in Table 4, we observe a significant performance drop (80.19% \rightarrow 75.24%) by removing the supervision on hidden states (*i.e.*, \mathcal{L}_{feat}), which is consistent with our observations (*i.e.*, performance are suboptimal via logit-based methods). The removal of \mathcal{L}_{ce} and \mathcal{L}_{logit} also cause a marked performance degradation (*i.e.*, 80.19% \rightarrow 78.32% and

Table 3: Ablation study on various optimizers and learning rate. The experiments are conducted on ViT-Base [5] CIFAR-100 [15] using AMD.

Optimizer	Learning Rate	CIFAR-100 [15] top-1
SGD	3×10^{-1}	0.01% (Failed)
	3×10^{-2}	0.01% (Failed)
	3×10^{-3}	0.01% (Failed)
	3×10^{-4}	0.13%
AdamW	3×10^{-1}	0.01% (Failed)
	3×10^{-2}	42.48%
	3×10^{-3}	80.19%
	3×10^{-4}	75.38%

Table 4: Impact of different loss components, including three variants from original training objectives.

Method	Performance
AMD	80.19%
- w/o \mathcal{L}_{CE}	78.32%
- w/o \mathcal{L}_{logit}	78.01%
- w/o \mathcal{L}_{feat}	75.24%
- w/ \mathcal{L}_{dkd}	80.22%

80.19% \rightarrow 78.01%, respectively), underscoring the integral role both losses play in enhancing model efficacy. Note that the influence of getting rid of \mathcal{L}_{logit} has a higher impact on performance, which is consistent with our claim in the previous ablation study (*i.e.*, $\alpha = 1$). For completeness, we also conduct experiments on combining DKD [42] loss, which introduces target class knowledge distillation (*i.e.*, TCKD) and non-target class knowledge distillation (*i.e.*, NCKD) to further decompose \mathcal{L}_{ce} . Specifically, we have $\mathcal{L}_{dkd} = \zeta\text{TCKD} + \eta\text{NCKD}$, incorporating balancing parameters ζ and η . The result indicates that the DKD loss further improves the model performance (*i.e.*, 80.19% \rightarrow 80.22%). However, it is imperative to underscore that the incorporation of the DKD loss introduces additional hyper-parameters, which consequently engenders increased fluctuations in the quest to achieve optimal results. Therefore, we retain the original design as delineated in Eq. 2 for the purpose of maintaining stability in the model’s performance.

5.2 Learning Rate Schedule

Table 3 reports the performance AMD with respect to different learning rates and optimizers on ViT-Base [5] CIFAR-100 [15]. As seen, the SGD optimizer exhibits a marked sensitivity to the learning rate, characterized by a notable incidence of failure cases (*i.e.*, we employed various learning rate schedulers, including linear, cosine, and cosine with restarts, to investigate their impact on model performance. However, it was observed that all these scheduling methods consistently resulted in low accuracy levels). AdamW optimizer with learning rate 3×10^{-3} in large-scale vision model, on the other hand,

Table 5: Impact of candidate scaling m .

Method	Performance	GPU hours
ViT – Base _{100%} (teacher)	89.33%	-
– MMD	80.11%	20×
– AMD ($m = 1$)	78.39%	2×
– AMD ($m = 3$)	79.46%	2×
– AMD ($m = 6$)	79.84%	2.1×
– AMD ($m = 9$)	80.19%	2.2×
– AMD ($m = 15$)	80.22%	2.6×

Table 6: The result of knowledge distillation compared to [2].

Method	CIFAR-100 top-1
[ICML2023] [2]	78.91%
AMD	79.17%

presents a robust and superior performance. We thus apply AdamW as the *default* optimizer for all methods.

5.3 Impact of Candidate Sampling Rate.

We further study the variation of candidate sampling rate by changing the number of sampled candidates $m \in \{1, 3, 6, 9, 15\}$. A higher value of m signifies a more refined granularity in the sampling rate. This increased granularity is directly correlated with an extended duration of training time. The GPU hours and their corresponding student performance are reported in Table 5. We set $m = 9$ for a satisfying tradeoff between performance and computational overhead. An increased sampling rate invariably leads to a longer training time, which yields marginal enhancements in performance. For example, when having $m = 15$, we observe 0.03% performance gain can be achieved with 18% GPU hour increment. We argue that this is inefficient for training schedule.

6 Discussion on Multi-teacher Distillation

We observed that a recent study by [2] comprehensively discusses the disparity gap in knowledge distillation. Specifically, they identify a method involving sequential distillation from multiple teachers organized into a curriculum, which notably enhances the effectiveness of knowledge distillation and mitigates the capacity gap between teacher and student models. It is important to note that [2] is orthogonal to ours, which can be distinguished by two primary facets:

- The most profound distinction lies in the fact that AMD is primarily oriented towards image classification, in contrast to [2], which endeavors to tackle the task

Table 7: The results of knowledge distillation methods on various CNN-based architectures on CIFAR-100 [15].

Teacher	WRN40×2	ResNet56	VGG13
Student	WRN16×2	ResNet20	VGG8
Teacher	76.46%	73.44%	75.38%
Student	73.64%	69.63%	70.68%
KD [12]	74.92	70.66	72.98
FitNet [27]	75.75%	71.60%	73.54%
AT [39]	75.28%	71.78%	73.62%
SP [33]	75.34%	71.48%	73.44%
VID [1]	74.79%	71.71%	73.96%
RKD [24]	75.40%	71.48%	73.72%
PKT [26]	76.01%	71.44%	73.37%
AB [11]	68.89%	71.49%	74.27%
FT [14]	75.15%	71.52%	73.42%
CRD [31]	76.04%	71.68%	74.06%
SSKD [37]	76.04%	71.49%	75.33%
TAKD [22]	75.04%	70.77%	73.67%
DGKD [30]	76.24%	71.92%	74.40%
AMD	<u>76.06%</u>	71.95%	<u>74.47%</u>

of object detection. We contend that image classification serves as a more foundational aspect and has the potential to engender broader social impacts, especially when considering the scarcity of knowledge distillation approaches applied to large vision foundation models.

- While [2] constructs the teacher sequence through the application of the heuristic algorithm BGS, predicated on the representation similarities among diverse models, our work introduces AMD as a more sophisticated alternative. We elegantly reduced the capacity gap between models by designing *Joint Optimization*, enabling the direct inheritance of parameters by student models from their teacher counterparts within the same, singular architectural design. Our approach is therefore characterized by an accelerated convergence rate, and a more efficient training schedule, thereby presenting a more refined solution.

For completeness, we try our best to accommodate [2] into image classification task and report the top-1 performance on ViT-Small CIFAR-100 [15] dataset in Table 6. As seen, AMD reaches superior performance when compared to [2].

7 Extension to CNNs

For fairness and completeness, we further extend our AMD into traditional CNN-based architectures. We follow common practices with teacher-assistant design [22, 30], and compare AMD to other competitive methods. The results are shown in Table 7, comparing on various CNN architectures, *i.e.*, WideResNet [40], ResNet [9] and VGG [29]. The results show that our approach is also a promising approach with respect to CNN-based architectures. Specifically, AMD reaches the best performance among all the other competitive methods on ResNet distillation, and gets competitive results under

Table 8: Semantic segmentation results on Cityscapes [4] val dataset.

Method	mIOU
DeepLabV3-R101 (teacher)	78.07%
SKD [CVPR19] [19]	75.42%
IFVD [ECCV20] [34]	75.59%
CWD [ICCV21] [28]	75.55%
CIRKD [CVPR22] [38]	76.38%
DIST [NeurIPS22] [13]	77.10%
AMD	77.23%

the other two settings. Acknowledging the fact that AMD is not specifically designed for CNN-based architectures, an avenue for the development of a unified solution in the realm of knowledge distillation is viable, applicable to both transformer-based and CNN-based architectures.

We then explore and compare AMD with latest knowledge distillation methods on semantic segmentation task [10, 32] in Table 8 (using the same setting, *i.e.*, AMD replaces ResNet18 student with 15% ResNet101 on Cityscapes [4] val dataset). It can be seen that AMD achieves consistently competitive results *without* task-specific design, demonstrating its generalization capability.

8 Discussion

8.1 Asset License and Consent

Vision transformers are available under separate license terms: [huggingface/transformers](#) is licensed under [Apache-2.0](#); [Swin-Transformer](#) [20] and [ViT-pytorch](#) [5] are licensed under [MIT](#).

8.2 Reproducibility

AMD is implemented in Pytorch [25]. Experiments are conducted on 16 NVIDIA A100-40GB GPUs. To guarantee reproducibility, our full implementation shall be publicly released upon paper acceptance. We further provide the pseudo code of our proposed AMD in Algorithm 1.

8.3 Technical Contributions

The main contributions of AMD is of threefold. *First*, the motivation behind this work stems from our empirical observation of the optimal performance of the teacher assistant, prompting the introduction of the new NPSD metric. *Second*, although the individual pieces are not new, the design of the joint optimization framework with certain approximations (incremental property via structure pruning) has not been attempted

Algorithm 1 Pseudo-code of AMD in a PyTorch-like style.

```

# numTA: number of teacher assistants (default=1)
# numWarmupSteps: number of warmup steps
# numTrainSteps: number of training steps
# numTrainEpochs: number of training epochs
# teacher_model_path: model path of teacher/teacher assistant

def AMD(numTA, teacher_model_path, numWarmupSteps, numTrainSteps,
        numTrainEpochs):
    for iteration in range(1, numTA):
        t_config = config_class.from_pretrained(teacher_model_path)
        t_model = model_class.from_pretrained(teacher_model_path,
            config=t_config)
        t_model = t_model.to(device)

        s_config = config_class.from_pretrained(teacher_model_path)
        s_model = model_class.from_pretrained(teacher_model_path,
            config=s_config)
        s_model = s_model.to(device)

        sandwich_sparsities = [s for s in s_config.sparsity_map]

        optimizer = AdamW(s_model.named_parameters(), lr=learning_rate)

        scheduler_name = "cosine_with_restarts"
        lr_scheduler = get_scheduler(
            scheduler_name, optimizer, numWarmupSteps, numTrainSteps)

        t_model.eval()
        base_model = s_model.module

        for epoch in range(numTrainEpochs):
            for step, batch in enumerate(train_loader):
                s_model.train()
                with torch.no_grad():
                    t_output = t_model(batch)

                loss_item = 0.
                for sparsity in sandwich_sparsities:
                    base_model.sparsify(sparsity)
                    s_output = s_model(batch)
                    loss = model_class.loss_fn(t_output, s_output) / len(
                        sandwich_sparsities)

                    loss_item += loss.item()
                loss.backward()
                train_losses.update(loss_item)
                optimizer.step()
                lr_scheduler.step()
                s_model.zero_grad()

```

before, leading to much efficient distillation. *Third*, we focus on the overlooked problem of scale reduction in Transformer-based vision models. Our comprehensive studies demonstrate competitive results with efficient distillation.

8.4 Complexity

One thing should be noted that multi-step distillation introduces additional complexity. However, the multi-step design proves particularly advantageous when a substantial

capacity gap exists between the teacher and student model. With the progressive scaling of vision models, this capacity gap becomes increasingly pronounced. While the proposed NPSD method does introduce additional computation, our efficient joint optimization approach allows for the effective identification of the optimal teacher-assistant, ultimately resulting in significantly faster overall training speeds compared to existing multi-step methods.

8.5 Social Impact and Future Works

In the evolving landscape of vision models, the trend towards increasingly expansive sizes has rendered their deployment a matter of pressing significance. Current approaches have predominantly centered on CNN architectures of a modest scale, overlooking the potential advantages and performance improvements that large-scale vision models could offer to low-power processors and mobile devices. This narrow focus on smaller architectures may inadvertently constrain the exploration of more efficient or powerful models that could enhance the computational capabilities of devices with limited processing power. As such, there is a pressing need to expand the scope of research to include and optimize large-scale vision models. In acknowledgment of this fact, our research introduces an innovative method for knowledge distillation applied to large-scale vision models. AMD is meticulously designed to optimize the efficiency of the training regimen whilst simultaneously enhancing the model’s performance when comparing to competitive methods. We believe our work bring fundamental insights into related fields (*e.g.*, object detection [16, 41, 43]). For potential limitations, our method as well as common practices [35] requires two hyper-parameters (*i.e.*, α and β for the overall training objective, Eq. 3), which needs further tune on datasets. Though in practice, we find both hyper-parameters are relatively stable (See §4.3 in our paper), and are sufficient enough to outperform *all* current methods, there is still possible integration of generating optimal combinations or having less number of hyper-parameters. This indicates a possible direction for our future research. Also, as stated in §1, we apply a straightforward pruning method, more studies on complex pruning methods can be developed for future research.

References

1. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: CVPR (2019) 7
2. Cao, S., Li, M., Hays, J., Ramanan, D., Wang, Y.X., Gui, L.: Learning lightweight object detectors via multi-teacher progressive distillation. In: ICML (2023) 6, 7
3. Chen, Z., Yang, Y., Qifan, W., Jiahao, L., Jingang, W., Yunsen, X., Wei, W., Dawei, S.: Minidisc: Minimal distillation schedule for language model compression. arXiv preprint arXiv:2205.14570 (2022) 1
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016) 8
5. 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. In: ICLR (2021) 2, 5, 8

6. Fang, G., Ma, X., Song, M., Mi, M.B., Wang, X.: Depgraph: Towards any structural pruning. In: CVPR (2023) [2](#)
7. Geva, M., Schuster, R., Berant, J., Levy, O.: Transformer feed-forward layers are key-value memories. arXiv preprint arXiv:2012.14913 (2020) [2](#)
8. Han, C., Wang, Q., Cui, Y., Cao, Z., Wang, W., Qi, S., Liu, D.: E²vpt: An effective and efficient approach for visual prompt tuning. In: ICCV (2023) [1, 2](#)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) [7](#)
10. He, T., Shen, C., Tian, Z., Gong, D., Sun, C., Yan, Y.: Knowledge adaptation for efficient semantic segmentation. In: CVPR (2019) [8](#)
11. Heo, B., Lee, M., Yun, S., Choi, J.Y.: Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In: AAAI (2019) [7](#)
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) [4, 7](#)
13. Huang, T., You, S., Wang, F., Qian, C., Xu, C.: Knowledge distillation from a stronger teacher. NeurIPS (2022) [8](#)
14. Kim, J., Park, S., Kwak, N.: Paraphrasing complex network: Network compression via factor transfer. NeurIPS (2018) [7](#)
15. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) [3, 4, 5, 7](#)
16. Li, C., Cheng, G., Wang, G., Zhou, P., Han, J.: Instance-aware distillation for efficient object detection in remote sensing images. IEEE TGRS **61**, 1–11 (2023) [10](#)
17. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710 (2016) [2](#)
18. Li, Y., Zhang, K., Cao, J., Timofte, R., Van Gool, L.: Localvit: Bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021) [2](#)
19. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: CVPR (2019) [8](#)
20. 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: ICCV (2021) [2, 8](#)
21. Michel, P., Levy, O., Neubig, G.: Are sixteen heads really better than one? In: NeurIPS (2019) [2](#)
22. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: AAAI (2020) [4, 7](#)
23. Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440 (2016) [2](#)
24. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: CVPR (2019) [7](#)
25. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019) [8](#)
26. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR (2016) [7](#)
27. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014) [7](#)
28. Shu, C., Liu, Y., Gao, J., Yan, Z., Shen, C.: Channel-wise knowledge distillation for dense prediction. In: ICCV (2021) [8](#)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) [7](#)
30. Son, W., Na, J., Choi, J., Hwang, W.: Densely guided knowledge distillation using multiple teacher assistants. In: ICCV (2021) [7](#)

31. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: ICLR (2019) 4, 7
32. Tian, Z., Chen, P., Lai, X., Jiang, L., Liu, S., Zhao, H., Yu, B., Yang, M.C., Jia, J.: Adaptive perspective distillation for semantic segmentation. IEEE TPAMI 45(2), 1372–1387 (2022) 8
33. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: ICCV (2019) 7
34. Wang, Y., Zhou, W., Jiang, T., Bai, X., Xu, Y.: Intra-class feature variation distillation for semantic segmentation. In: ECCV (2020) 8
35. Wu, S., Chen, H., Quan, X., Wang, Q., Wang, R.: Ad-kd: Attribution-driven knowledge distillation for language model compression. In: ACL (2023) 10
36. Xia, M., Zhong, Z., Chen, D.: Structured pruning learns compact and accurate models. In: ACL (2022) 2
37. Xu, G., Liu, Z., Li, X., Loy, C.C.: Knowledge distillation meets self-supervision. In: ECCV (2020) 7
38. Yang, C., Zhou, H., An, Z., Jiang, X., Xu, Y., Zhang, Q.: Cross-image relational knowledge distillation for semantic segmentation. In: CVPR (2022) 8
39. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016) 7
40. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016) 7
41. Zhang, L., Ma, K.: Structured knowledge distillation for accurate and efficient object detection. IEEE TPAMI (2023) 10
42. Zhao, B., Cui, Q., Song, R., Qiu, Y., Liang, J.: Decoupled knowledge distillation. In: CVPR (2022) 2, 4, 5
43. Zhixing, D., Zhang, R., Chang, M., Liu, S., Chen, T., Chen, Y., et al.: Distilling object detectors with feature richness. NeurIPS (2021) 10