Automated Knowledge Distillation via Monte Carlo Tree Search

Lujun Li[1†*] Peijie Dong[2†] Zimian Wei[2] Ya Yang[3]
[1] The Hong Kong University of Science and Technology, [2] National University of Defense Technology
[3] City University of Hong Kong
[1]lilujunai@gmail.com[1]{dongpeijie, weizimian16}@nudt.edu.cn, [3] yya9@outlook.com

Abstract

In this paper, we present Auto-KD, the first automated search framework for optimal knowledge distillation design. Traditional distillation techniques typically require handcrafted designs by experts and extensive tuning costs for different teacher-student pairs. To address these issues, we empirically study different distillers, finding that they can be decomposed, combined, and simplified. Based on these observations, we build our uniform search space with advanced operations in transformations, distance functions, and hyperparameters components. For instance, the transformation parts are optional for global, intra-spatial, and inter-spatial operations, such as attention, mask, and multi-scale. Then, we introduce an effective search strategy based on the Monte Carlo tree search, modeling the search space as a Monte Carlo Tree (MCT) to capture the dependency among options. The MCT is updated using test loss and representation gap of student trained by candidate distillers as the reward for better exploration-exploitation balance. To accelerate the search process, we exploit offline processing without teacher inference, sparse training for student, and proxy settings based on distillation properties. In this way, our Auto-KD only needs small costs to search for optimal distillers before the distillation phase. Moreover, we expand Auto-KD for multi-layer and multi-teacher scenarios with training-free weighted factors. Our method is promising yet practical, and extensive experiments demonstrate that it generalizes well to different CNNs and Vision Transformer models and attains state-of-the-art performance across a range of vision tasks, including image classification, object detection, and semantic segmentation. Code is provided at https://github.com/lilujunai/Auto-KD.

1. Introduction

Various visual tasks [18, 34, 53] have been successfully tackled by Deep Neural Networks (DNNs). Despite the appealing performance, the prevailing DNNs usually have large numbers of parameters, leading to heavy costs of memory and computation. Conventional techniques such as pruning weights from networks [19, 36] and quantizing networks to use low-bit parameters [10, 57, 85] have proven to be effective for mitigating this computational burden. Recently, Knowledge Distillation (KD) [70, 24], another promising solution family to train compact yet accurate models, has attracted increasing attention. The objective of knowledge distillation (KD) is to transfer the acquired knowledge from a high-capacity DNN model (*i.e.*, teacher) to a lower-capacity target DNN model (*i.e.*, student), effectively balancing accuracy and efficiency during runtime.

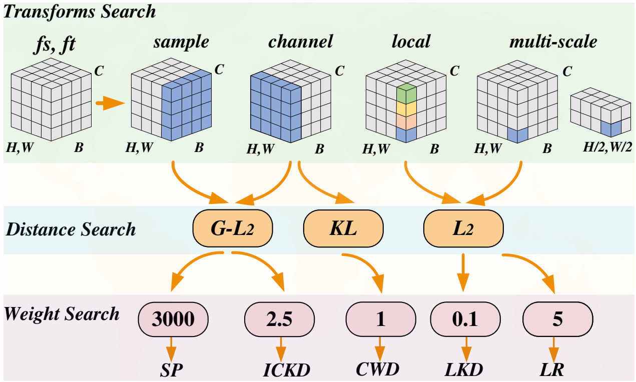

Figure 1. Illustration on distiller search space on intermediate features. Recent sadvancements in distillation methods (*e.g.*, SP [69], ICKD [49], CWD [64], LKD [42] and LR [55]) can be searched with various options of transforms, distances and weights search.

Problem Statement: While numerous KD methods [54, 66, 82] have been proposed, one major challenge is the sensitivity of their performance to hyperparameters and teacher-student architecture pairs. Different hyperparameters, such as the weighted factor used in the loss function, can have a significant impact on the final performance of the distilled student model (see Figure 2 (Left)). Similarly, the same distiller performs quite differently under various teachers (see Figure 2 (Right)). Therefore, the practical usage of KD

[*] Corresponding author, † equal contribution.

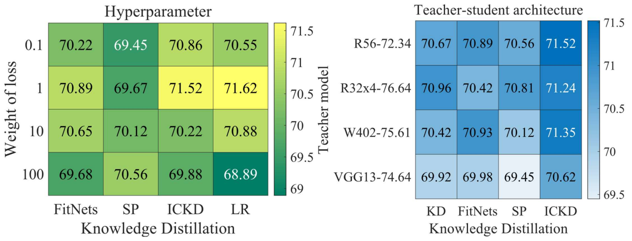

Figure 2. **Left:** Top-1 mean accuracy (%) achieved by KD methods via various loss weights for ResNet20 (69.06%) with teacher ResNet110 on CIFAR-100. **Right:** Top-1 mean accuracy (%) of KD methods with different teachers for ResNet20 on CIFAR-100.

always involves time-consuming tuning of specific hyperparameter settings. Another problem is that existing distillation methods depend on manual human design and expert knowledge. Handcrafted distillations can be highly task or dataset-specific, limiting their generalizability to new scenarios. For these issues, an intuitive solution is to explore automated tuning ways. However, it is not easy to implement such an automatic search framework because of the following aspects: (1) knowledge distillation involves various hyperparameter settings, loss functions, and transformation types, making it more complex than traditional hyperparameter optimization [21]. This complexity presents a significant obstacle in designing a unified search space and identifying optimal solutions. (2) In contrast to weight-sharing methods [27, 47] for acceleration, such a multi-variate joint search task needs to use the expensive multi-trial route to prevent weight-sharing errors and optimization collapse.

Our New Observations. To effectively build unified search spaces and optimize search costs, we conduct detailed analyses and experiments on different existing distillation methods. For the search space, we find that (1) **Decomposability.** Most advanced distillers can be decomposed into basic transformation and distance functions units. As shown in Figure 1, both SP [69] and ICKD [49] employ similar distance functions but differ in transformation. Conversely, ICKD [49] and CWD [64] share similar feature transformations but adopt distinct distance functions. (2) **Combinability** between different transformations and distance functions. As shown in Figure 3 (Left), the attention [82] and mask [78] transformations can be combined with multi-scale operation [55] with additional gains. Sample-wise transformations in SP with KL loss alternative to $G-L_2$ loss yield better performance than the original form (see Figure 3 (Right)). (3) **Simplifiability.** Some distillation options can be ignored in building the search space because of their consistently poorer results (*e.g.*, L_1 in Figure 3 (Right)). In addition, most distillers can obtain good results within limited hyperparameter selections (*e.g.*, four values of loss weight in Figure 2 (left)). These observations inspire us to build search spaces shown in Figure 1 that include varying key distillation operations in transformations, distance functions, and weights. Regarding efficiency, we find that offline storage of knowledge, sparse training for students, and advanced distillation properties, such as data efficiency and fast convergence, can be used to accelerate the distillation process. These findings offer valuable insights that contribute to the design of search space and the reduction of search budget.

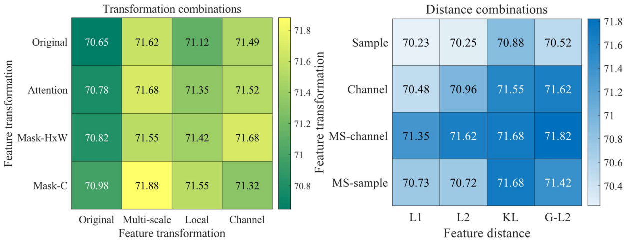

Figure 3. **Left:** Top-1 mean accuracy (%) of combinations of different transformations for distilling ResNet20 (69.06%) with teacher ResNet110 on CIFAR-100. **Right:** Top-1 mean accuracy (%) of combinations of transformations and distances for ResNet20 on CIFAR-100. Mask-C and Mask-H×W denote channel-wise mask and spatial-wise mask. MS-channel and MS-sample refer to our new combination transformations Multi-scale→Channel and Multi-scale→Sample.

Our New Search Framework. Based on the exciting observations described above, we present Auto-KD, an efficient and effective automated search framework that finds optimal knowledge distillation designs for distilling a given teacher-student model. Specifically, Auto-KD consists of three important components: the unified tree-like distiller search space, Monte Carlo tree search, and search acceleration strategies. We organize the search space for feature distillers into a tree-like structure consisting of different options from global, intra-spatial, and inter-spatial feature transformations, feature distance functions, and weight factors. The transformation options include attention [82, 83], mask [78], multi-scale [6], sample-wise [69], and channel-wise [49, 64, 86] operations. The distance functions include KL, L_2, $G-L_2$. Following most logits KD [24], we also further build a search space by extending the search for logits distance function and temperature values. To find the optimal candidate efficiently, we choose a powerful Monte Carlo tree search to select, expand, simulate, and reward the values of various nodes in the search tree. The reward determines the test loss of the student and the representation gap between teacher and student. To accelerate the search process, we use offline storage of knowledge to replace teacher inference, sparse training for student models with proxy training settings (*i.e.*, subsets and early stop). These strategies result in at least 40× faster training and 15× more training parameters and memory savings. Finally, we extend Auto-KD to multi-feature and multi-teacher distillation with train-free fine-grained weighted factors, which condition on teacher feature entropy and feature similarity of teacher-student.

Valuation and Evaluation In principle, our Auto-KD differs from previous hand-designed distillation methods and opens new doors to automated distillation designs. Its merits can be highlighted in three aspects: (1) **Effective.** Auto-KD solves the distiller's hyperparameter and architecture-sensitive problem and helps to obtain stable distillation gain in different scenarios. Extensive experiments on visual tasks and models demonstrate the leading performance of Auto-KD. On the CIFAR-100 dataset, Auto-KD achieves $3\% \sim 7\%$ gain for CNN and $2\% \sim 13\%$ gain for ViT models, surpassing other SOTA methods with significant margins. On the large-scale ImageNet dataset, ResNet18 and MobileNet with Auto-KD reach a 3% absolute gain over the baseline model. For downstream tasks, Auto-KD also improves the detector with 3.7 AP on MS-COCO and the segmenter with $3.1\% \sim 3.7\%$ mIOU in the cityscape. (2) **Efficiency.** Auto-KD employs bags of efficient training strategies based on the distillation properties and achieves significant speed-ups. This framework greatly benefits the following search methods and the application of distillation. (3) **Insightful.** Auto-KD in-depth analyzes existing advanced distillation designs and explores their combinations to generate many new distillers. Auto-KD not only provides guidelines for practical applications, but also develops a new research direction. We anticipate that our endeavors in automating the design of distillers will, to some degree, support and advance future research on automated knowledge distillation.

Main Contributions:

- By exploring the decomposability, combinability and simplifiability of distillation methods, we propose a new automated distillation search framework for optimal distiller design, which, to the best of our knowledge, is not achieved in the area of knowledge distillation.

- Auto-KD organizes the unified distiller search space as a Monte Carlo tree and performs Monte Carlo tree search. In addition, Auto-KD leverages bags of efficient strategies and achieves significant search acceleration.

- We perform thorough evaluations on classification, detection, and segmentation. Auto-KD achieves state-of-the-art performance across multiple datasets and architectures (*e.g.*, CNN and vision transformer). We also successfully extend Auto-KD in multi-layer and multi-teacher distillation.

2. Automated Knowledge Distillation

Figure 4 presents the search process in Auto-KD. In this section, we first specify its three key components: search space design, MCT search, and acceleration strategies. Then, we introduce its applications and extensions.

2.1. Search Space Design

Problem formulation The aim of KD is to train a smaller student model (S) to learn from the teacher model (T). More specifically, the teacher model's outputs, referred to as p_T and f_T, correspond to the logits and features, respectively. Meanwhile, the outputs of the student model, denoted as p_S, f_S, are trained to match those of the teacher by minimizing:

$$\mathcal{L}_{KD} = \mathcal{W}_f \times \mathcal{D}_f \big(\mathcal{T}_f \langle f_S, f_T \rangle\big) + \mathcal{W}_p \times \mathcal{D}_p \left(p_S/\tau, p_T/\tau\right), \tag{1}$$

where $\mathcal{W}_f$ is the loss weighted factor, $\mathcal{T}_f$ is feature transformations, $\mathcal{D}_f(\cdot,\cdot)$ and $\mathcal{D}_p(\cdot,\cdot)$ is distance function measuring the difference of feature representations and logits.

Unified tree-structured search space. For efficient search and optimal accuracies, we organize key operations in KD into a tree-like structure in Table 1. (1) For feature transforms, we observe that existing designs can be decomposed into three categories: global feature, intra-spatial, and inter-spatial transformations, respectively. For example, attention KD [82] generates attention factors to re-align teacher-student features. Then, these features can be pooled into multi-scale ones [6] and transformed by channel transforms to achieve better channel-wise alignment [64]. The combination of diverse operations allows our search space to capture the transformations in recent SOTA KDs and many new forms. For other transform parts, we employ a pooling layer to align feature scales and 1×1 Conv to align filter numbers. For converting logits KD, we apply intra-class and inter-class transform [29] to improve performance. (2) For the distance function, we select some potential distances for the feature KD and logits KD. Other distances are not included in our search space because of their poor performance. (3) For the hyperparameters of loss weights and temperature factors, we select common values as candidates for loss weights of feature KDs and temperature factors for logits KD. Our search space is highly uniform and tight, and we achieve advanced searches on it, detailed in the following sections.

Extended Search Space. In addition, we also extend the search space with some distillation designs that we have explored ourselves: (1) $G-L2$ and Renyi entropy for embedding feature distillation. (2) Logits normalization with a scaling factor for KL divergence in logits KD. (3) Additional options for setting warm-up and early-stop for loss weights.

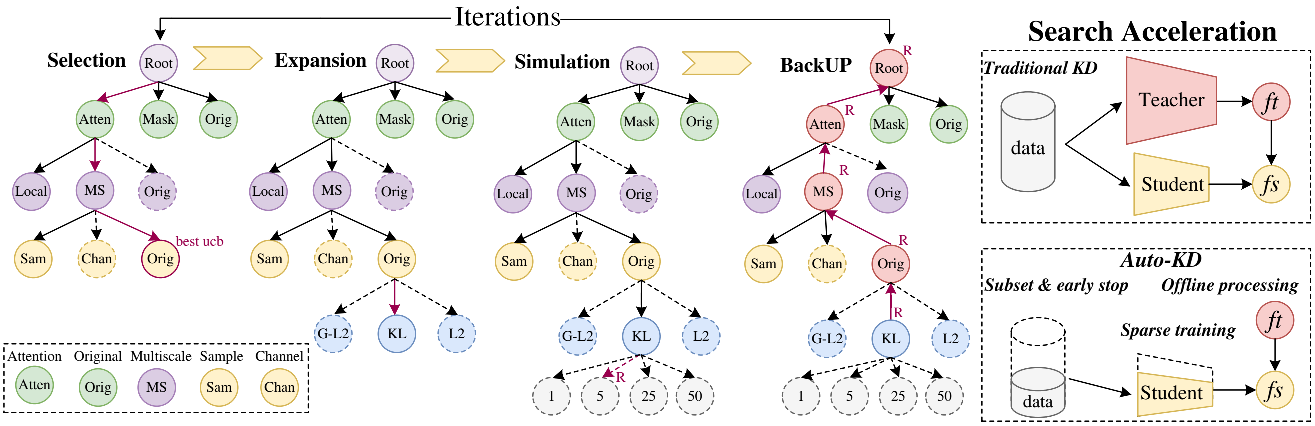

Figure 4. The overall framework of Auto-KD, which models the search space into a MCT, then searches the optimal design of knowledge distillation using Monte Carlo tree search (left) and search acceleration strategies (right).

Table 1. Various distillation operations and their forms in our search space, which will be detailed in the Appendix.

Type	Operation	Expression
Global $\mathcal{T}_f$	Attention	$\alpha \times f_S, \alpha \times f_T, \alpha \sim (f_S, f_T)$
	Mask	$M \times f_S, M \times f_T, M \in (0,1)$
	Original	f_S, f_T
Intra-spatial $\mathcal{T}_f$	Multi-scale	$f_S^{N,C,H/2,4,W/2,4}, f_T^{N,C,H/2,4,W/2,4}$
	Local	$f_S^{n^2 \times N,C,H/n,W/n}, f_T^{n^2 \times N,C,H/n,W/n}$
	Original	f_S, f_T
Inter-spatial $\mathcal{T}_f$	Sample	$f_S^{N,CHW}, f_T^{N,CHW}$
	Channel	$f_S^{C,NHW}, f_T^{C,NHW}$
	Original	f_S, f_T
Distance $\mathcal{D}_f$	$G - \mathcal{L}_2$	$\lvert\lvert f_S \cdot f_S^\intercal / \lvert\lvert f_S \cdot f_S^\intercal \rvert\rvert - f_T \cdot f_T^\intercal / \lvert\lvert f_T \cdot f_T^\intercal \rvert\rvert \rvert\rvert^2$
	$\mathcal{L}_{KL}$	$\sigma(f_T) \times log[\sigma(f_T)/\sigma(f_S)]$
	$\mathcal{L}_2$	$\lvert\lvert f_S - f_T \rvert\rvert^2$
Distance $\mathcal{D}_p$	$\mathcal{L}_{KL}$	$\sigma(p_T) \times log[\sigma(p_T)/\sigma(p_S)]$
	$\mathcal{L}_{Pearson}$	$1 - cov(p_S, p_T)/\big(std(p_S) \cdot std(p_T)\big)$
Weight $\mathcal{W}_f$	Constant	1,5,25,50
Weight $\mathcal{W}_p$	Constant	0.1, 0.5, 1, 5
Temperature τ	Constant	1,2,4,8

2.2. Monte Carlo Tree Search

We perform the Monte Carlo Tree Search (MCTS) [71] for the following reasons: (1) MCTS is a powerful and efficient sampling-based tree search method to solve complex decision problems [3, 65]. (2) MCTS could capture correlations of operation candidates in our distiller search space, improving interpretability and stability. The main steps of the algorithm can be summarized as follows.

Selection: In this step, the algorithm selects the best node from the current tree using an Upper Confidence Bound (UCB) formula. The UCB formula balances exploration and exploitation, allowing the algorithm to choose the node with the highest potential for improvement. For a node n_i, the UCB is computed by:

$$\nu(n_i) = R_i/N_i + C \cdot \sqrt{2 \cdot \ln N_b/N_i}, \tag{2}$$

where R_i represents the reward for node n_i, while N_i and N_b indicate the number of visits to node n_i and its parent node n_b, respectively. The control parameter C determines the extent of exploration. In our approach, the reward value R is dependent on the $\mathcal{L}\_CE(p_S, Y)$ loss of the student model and the similarity between the teacher and student on the validation set, as defined below:

$$R = 1 - (\mathcal{L}_{CE}(p_S, Y) + \mathcal{L}_{CKA}(f_S, f_T)), \tag{3}$$

where $\mathcal{L}_{CKA}(\cdot,\cdot)$ is Centered Kernel Alignment (CKA) metric [32] for representation similarity.

Expansion: During this step, the algorithm generates additional child nodes for the selected node, representing potential future states of the system.

Simulation: Following the expansion of the selected node, the newly added node undergoes evaluation through a trajectory of random actions until a terminal state is reached. The outcome of the simulation is then utilized to estimate the quality of the child node.

Backpropagation: In the final step, the algorithm updates the estimated quality of the parent nodes based on the simulation results. The updated quality values are used to influence future selections in the tree.

2.3. Search Acceleration Strategies

Offline processing and sparse training. Our approach employs offline processing to reduce computational costs without requiring teacher inference. Specifically, we store the feature maps generated by the teacher after a single forward pass and apply the same data augmentation techniques used during training to ensure spatial alignment. In addition, we introduce sparse training to reduce the memory budget and computation. In distillation, we first set up a random mask [50] to force certain weights of the student model to be zero and then configure a dynamic strategy [15] to preserve the distillation gains. The results in Table 2 indicate that offline processing effectively reduces inference parameters

Table 2. Ablation on acceleration techniques (*i.e.*, offline processing, student training with 50% sparsity, subsets & early-stop with 20% of original dataset & epochs) for ResNet-20 (69.09%) via teacher ResNet-110 on CIFAR-100. Training time (GPU-seconds) is measured on a single 2080Ti GPU and × represents the improving ratios than traditional KD. Acc-1. denotes student accuracy (%) of candidate distiller in the search phase and Acc-2. represents the final results (%) of our searched distiller in the distillation phase.

Method	Params (M)	Time (S)	Acc-1.	Acc-2.
Baseline	1.97	6,036	71.00	72.65
+ Offline	0.27 (7.29×)	4,654 (1.29×)	70.82	72.65
+ Sparse-50%	0.13 (15.15×)	3,816 (1.58×)	70.66	72.61
+ Subsets-20%	0.13 (15.15×)	793.2 (7.6×)	66.52	72.55
+ Early-stop-20%	0.13 (15.15×)	150.6 (40.07×)	48.98	72.52

and training time. Similarly, sparse student training substantially reduces training costs and marginally improves searched accuracy.

Proxy settings. With diverse and informative knowledge learned from a teacher, student models offer advantages in terms of data efficiency and faster training speeds. Based on these properties, we employ subsets and early stop the training process once the student model performs well enough to determine the quality of the candidate distillation and use the intermediate model to compute the reward signal. As shown in Table 2, adopting proxy settings reduces search overhead and ensures stability in the final searched accuracy.

2.4. Applications and Extensions

After the distiller search phase, we train the student network S using the discovered distiller, denoted by $\mathcal{L}_{Auto-KD}$. The optimization objectives are defined as follows, where $\mathcal{L}_{CE}$ is the cross-entropy loss:

$$\mathcal{L}_S = \mathcal{L}_{CE}(p_S, Y) + \mathcal{L}_{Auto-KD}, \tag{4}$$

Extension for multi-layer & multi-teacher distillation. Augmenting various features supervision from different layers of a single teacher or multiple teachers can enhance the quality of distillation, but this approach also presents challenges in weight tuning. To address this issue, we scale the weight of loss with Train-Free (TF) factors in multi-feature distillation based on the information bottleneck theory. The TF factors are determined by the information entropy of teacher features and the similarity of teacher-student feature pairs. Consider $\mathcal{Q}$ as a set comprising layer location pairs for feature distillation. The optimization objective function can be defined as follows:

$$\mathcal{L}_{Auto-KD+TF} = \mathcal{W}_f \sum_{q \in \mathcal{Q}} \mu_q \times \mathcal{D}_f\big(\mathcal{T}_f\langle f_S^q, f_T^q \rangle\big), \tag{5}$$

$$\mu_q = \mathcal{D}_{entropy}(f_T^q) \times \mathcal{D}_{CKA}(f_S^q, f_T^q). \tag{6}$$

where μ_q is the Train-Free (TF) fine-grained weighted factor, $\mathcal{D}_{entropy}$ is the standard entropy metric and $\mathcal{D}_{CKA}$ is the CKA feature distance metric, respectively. TF factor μ_q allows efficient adjustment of $\mathcal{W}_f$ in multi-layer, cross-layer, and multi-teacher KD and achieves comparable results.

2.5. Detailed Analysis of Searched Distillers

A thorough understanding of the specific task and the characteristics of the teacher-student network architecture should guide the choice of knowledge distillation operation. From the searched distillers of Auto-KD for different models, we can summarize some observations regarding the applicability of different knowledge distillation methods as:

1. Channel-wise distillation operation is recommended for wider teacher-student networks. By aligning the feature maps of the teacher and student networks, this method facilitates the transfer of knowledge from the teacher to the student network.

2. Multi-scale distillation operation is a useful method when there is a significant semantic gap between the teacher-student networks. It can help to gain more in most cases, especially for downstream tasks.

3. Mask distillation operation is more appropriate for teacher-student with relatively large distillation gaps, but it may also result in knowledge loss in the transfer process.

4. Attention distillation operation is more advantageous in heterogeneous teacher-student architectures, particularly for Vision Transformer models.

5. Local distillation operations are recommended for multi-label or local information-critical tasks, with average performance on datasets such as CIFAR-10/100.

6. Sample-wise distillation operation can benefit different models, but it is influenced by the corresponding task and batch size.

7. $\mathcal{L}_{KL}$ has better generalizability for different tasks, and $\mathcal{L}_{Pearson}$ is more practical for complex tasks.

8. Adjusting the temperature coefficient is useful for most tasks, and the optimal value of feature weights is generally between 1 and 25.

3. Experiments

In this section, we assess the efficacy of our proposed Auto-KD approach on classification, detection, and segmentation tasks, while also comparing its performance against other knowledge distillation methods. To ensure fair comparisons, we employ identical training settings and report the mean results obtained from multiple trials conducted throughout the experiments. For more comprehensive implementation details, please refer to the Supplementary Materials.

Table 3. Comparison of results on CIFAR-100. Most of the results of other methods refer to the original papers [6, 66]. W40-2, R32×4, R8×4, MV2, SV1 and SV2 stand for WRN-40-2, ResNet32×4, ResNet8×4, MobileNetV2, ShuffleNetV1 and ShuffleNetV2. We report top-1 "mean " accuracies (%) for Auto-KD over 3 runs.

Model		Same architectural style						Different architectural style			
	Teacher Student	W-40-2 W-16-2	R56 R20	R110 R20	R110 R32	R32×4 R8×4	VGG13 VGG8	VGG13 MNetV2	R32×4 SNetV1	R32×4 SNetV2	W-40-2 SNetV1
w/o logits KD	Teacher	75.61	72.34	74.31	74.31	79.42	74.64	74.64	79.42	79.42	75.61
	Student	73.26	69.06	69.06	71.14	72.50	70.36	64.60	70.50	71.82	70.50
	FitNets [61]	73.58	69.21	68.99	71.06	73.50	71.02	64.14	73.59	73.54	73.73
	AT [82]	74.08	70.55	70.22	72.31	73.44	71.43	59.40	71.73	72.73	73.32
	SP [69]	73.83	69.67	70.04	72.69	72.94	72.68	66.30	73.48	74.56	74.52
	RKD [54]	73.35	69.61	69.25	71.82	71.90	71.48	64.52	72.28	73.21	72.21
	CRD [66]	75.48	71.16	71.46	73.48	75.51	73.94	69.73	75.11	75.65	76.05
	LR [55]	76.12	71.89	71.82	73.89	75.63	74.84	70.37	77.25	77.18	77.14
	Auto-KD	**76.62**$_{\pm 0.18}$	**72.12**$_{\pm 0.17}$	**72.23**$_{\pm 0.19}$	**74.29**$_{\pm 0.18}$	**77.35**$_{\pm 0.21}$	**75.06**$_{\pm 0.16}$	**70.42**$_{\pm 0.12}$	**77.26**$_{\pm 0.17}$	**77.31**$_{\pm 0.34}$	**77.21**$_{\pm 0.36}$
w logits KD	KD [24]	74.92	70.66	70.67	73.08	73.33	72.98	67.37	74.07	74.45	74.83
	DIST [29]	75.35	71.78	71.68	73.86	75.79	73.86	69.17	75.23	76.08	75.85
	CRD+KD [67]	75.64	71.63	71.56	73.75	75.46	74.29	69.94	75.12	76.05	76.27
	ICKD-C [49]	75.57	71.69	71.91	74.11	75.48	73.88	69.53	74.86	75.86	76.12
	Auto-KD	**76.86**$_{\pm 0.23}$	**72.44**$_{\pm 0.15}$	**72.52**$_{\pm 0.22}$	**74.60**$_{\pm 0.18}$	**77.61**$_{\pm 0.36}$	**75.36**$_{\pm 0.15}$	**70.58**$_{\pm 0.18}$	**77.56**$_{\pm 0.21}$	**77.52**$_{\pm 0.16}$	**77.46**$_{\pm 0.32}$

Table 4. Accuracy results on ImageNet. Results of other methods quote the original papers report [6, 66].

Teacher	Student	Acc.	Teacher	Student	KD [24]	AT [82]	OFD [22]	SRRL [30]	CRD [66]	KR [55]	MGD [78]	**Auto-KD**
ResNet-34	ResNet-18	Top-1	73.40	69.75	70.66	70.69	70.81	71.73	71.17	71.61	71.58	**72.45**
		Top-5	91.42	89.07	89.88	90.01	89.98	90.60	90.13	90.51	90.35	**90.69**
ResNet-50	MobileNet	Top-1	76.16	70.13	70.68	70.72	71.25	72.49	71.37	72.56	72.35	**73.26**
		Top-5	92.86	89.49	90.30	90.03	90.34	90.92	90.41	91.00	90.71	**91.17**

3.1. Experiments on CIFAR-100

Dataset and Implementation. CIFAR-100 dataset [33] is a widely evaluated classification benchmark in distillation. During the distiller search phase, we adopt a basic tree structure search space and training acceleration settings, including 48 early-stop training epochs, 20% training data subsets, 50% sparsity of student model training, and offline storage of teacher's knowledge. Our MCT search performs 100 iterations for each teacher-student pair. In the distillation phase, the teacher-student networks are trained with standard training settings, employing a training epoch of 200. A mini-batch size of 128 and a standard SGD optimizer are utilized. The learning rate follows a multi-step schedule, starting at 0.1 and decaying by 0.1 at 100 and 150 epochs.

Comparison results. Table 3 presents a comparative analysis of our Auto-KD$_f$ method (without logits KD) with state-of-the-art (SOTA) feature distillations and Auto-KD with other KD methods. For teacher-student pairs with the same architectural style, Auto-KD$_f$ and Auto-KD outperform the baselines by margins ranging from $3\% \sim 5\%$ and $3\% \sim 5\%$, respectively. Compared with other KDs, our approach achieves absolute accuracy gains of $1\% \sim 3\%$ for Auto-KD$_f$ and $1\% \sim 3\%$ for Auto-KD. Notably, our approach exhibits even stronger performance when dealing with different architectural styles. At the same time, most of the other KD methods suffer from a noticeable reduction in accuracy compared to the same architecture. Specifically, Auto-KD outperforms the baseline by margins of $5\% \sim 7\%$ and the random search results by margins of $1\% \sim 3\%$, demonstrating the effectiveness of our design for different structures. Compared with other SOTA multi-layer KD methods, our method achieves an additional gain of $0.3\% \sim 2\%$ with only a single layer of feature knowledge. Finally, when combined with the distillation of output logits, Auto-KD provides additional improvements and clearly outperforms complex methods like CRD+KD, and ICKD. These results show that Auto-KD can improve each student model with simple settings under different teacher-student pairs.

3.2. Experiments on ImageNet

Dataset and Implementation. We additionally perform experiments on the ImageNet dataset (ILSVRC12)[62]. Due to computational limitations, it is difficult to search directly on the original ImageNet. Consequently, we address this issue by searching on a subset of ImageNet, namely ImageNet-100. This subset is randomly selected from the original training set and consists of 500 instances of 100 categories. Following experiments on the CIFAR-100 dataset, we employ similar MCT search settings to identify optimal distillers for ImageNet. Subsequently, we conduct full student model training on the entire ImageNet dataset using standard architectures such as ResNet-18[20] and MobileNet[26]. The training configuration aligns with the majority of distillation methods, entailing a 100-epoch training duration with a multi-step learning rate. The learning rate commences at 0.1 and undergoes decay by a factor of 0.1 at 30, 60, and 90 epochs.

Comparison results. In Table 3, we present the performance results of our auto-kd on the ImageNet dataset. Our Auto-KD method significantly improves over baseline models, as we observe gains of $2\% \sim 3\%$ in Top-1 accuracy for ResNet-18 and MobileNet, respectively. Notably, Auto-KD performs better than other KD methods and outperforms CKD with margins ranging from $1\% \sim 2\%$. These results demonstrate the superiority of Auto-KD over other methods on a large-scale dataset. These results demonstrate the effectiveness of Auto-KD for improving the performance of DNNs on the ImageNet dataset.

Table 5. Top-1 mean accuracy (%) comparison on CIFAR-100.

Student	Vanilla	KD [25]	AT [82]	SP [69]	LG [37]	**Auto-KD**
DeiT-Ti [68]	65.08	73.25	73.51	67.36	78.15	**78.58** ± 0.32
T2T-ViT-7 [81]	69.37	74.15	74.01	72.26	78.35	**78.62** ± 0.18
PiT-Ti [23]	73.58	75.47	76.03	74.97	78.48	**78.51** ± 0.21
PVT-Ti [72]	69.22	73.60	74.66	70.48	77.07	**77.48** ± 0.35
PVTv2-B0 [73]	77.44	78.81	78.64	78.33	79.30	**79.37** ± 0.24

3.3. Experiments on Vision Transformer

Implementation. Distillation techniques allow Vision Transformer (ViT) to be trained from scratch easily with CNNs as teachers. To assess the efficacy of Auto-KD, we search ViT-based distillation strategies on the CIFAR-100 dataset. We perform the MCT search with the same settings as the CNN experiment. Subsequently, we train the ViT with the optimal distiller obtained and ResNet-56 as CNN teacher. The training process involves images of 224×224 resolution and spans 300 epochs. The initial learning rate is set to 5e-4, and a weight decay of 0.05 is applied using the AdamW optimizer.

Comparison results. Table 5 presents the results of the vanilla and distillation models employing different distillation methods. The results indicate that Auto-KD can significantly improve the performance of vision transformers with $2\% \sim 13\%$ margins and consistently yields superior performance than other methods. In addition, it is noteworthy that our proposed method applies to various ViT architectures, thereby validating its effectiveness.

3.4. Experiments on Object Detection

Datasets and implementation. We perform experiments on the MS-COCO dataset [46], which consists of 80 object categories. The training set comprises 120k images, while the testing set consists of 5k validation images. To evaluate the optimal distiller of Auto-KD on the MS-COCO dataset, we utilize the widely used open-source framework MMDetection citemmdetection as the strong baseline. Our application of Auto-KD includes two-stage detectors such as Faster R-CNN [60] and one-stage detectors like RetinaNet [45], both well-established object detection frameworks. Following established practices [45], all models are trained using a $2\times$ learning schedule spanning 24 epochs. The models are trained using the SGD optimizer with a momentum of 0.9 and a weight decay of 0.0001.

Comparison results. As shown in Table 6, Auto-KD demonstrates its effectiveness and generality by surpassing other state-of-the-art methods [64, 77, 83] for both object detectors, improving the average precision (AP) by 3.7 on RetinaNet and 4.0 on Faster R-CNN. This success in tackling challenging object detection tasks showcases the broad applicability and efficacy of Auto-KD.

Table 6. Results comparison of object detection on MS-COCO. T: teacher; S: student. CM RCNN: Cascade Mask RCNN.

Model	**AP**	**AP**$_L$	**AP**$_M$	**AP**$_S$
Two-stage detectors				
CM RCNN-X101[T]	45.60	26.20	49.60	60.00
Faster RCNN-R50[S]	38.40	21.50	42.10	50.30
KD [24]	39.70	23.20	43.30	51.70
FKD [83]	41.50	23.50	45.00	55.30
CWD [64]	41.70	23.30	45.50	55.50
DIST [29]	40.40	23.90	44.60	52.60
FGD [77]	42.00	23.80	46.40	55.50
MGD [78]	42.10	23.70	46.40	56.10
Auto-KD	**42.40**	**24.20**	**46.70**	**55.90**
One-stage detectors				
RetinaNet-X101[T]	41.00	23.90	45.20	54.00
RetinaNet-R50[S]	37.40	20.00	40.70	49.70
KD [24]	37.20	20.40	40.40	49.50
FKD [83]	39.60	22.70	43.30	52.50
CWD [64]	40.80	22.70	44.50	55.30
DIST [29]	39.80	22.00	43.70	53.00
FGD [77]	40.70	22.90	45.00	54.70
MGD [78]	41.00	23.40	45.30	55.70
Auto-KD	**41.10**	**23.30**	**45.50**	**55.80**

3.5. Experiments on Semantic Segmentation

Datasets and implementation. We conduct an evaluation of Auto-KD on the CityScapes dataset [9], which comprises 2,975 training images, 500 validation images, and 1,525 testing images. Following previous research, we employ PSPNet-ResNet101 [84] as the teacher model, while the student models consist of PSPNet and DeepLabV3 [5] with the ResNet18 backbone. During the distillation process, we use a batch size of 8 and train the models for 40,000 iterations using the SGD optimizer with a momentum of 0.9 and a weight decay of 0.0005. The results are reported based on the mean Intersection-over-Union (mIoU) under the single-scale evaluation setting.

Comparison results. As shown in Table 7, the student PspNet and DeepLabV3 get 3.1 and 3.7 mIoU improvement by adding our Auto-KD loss. The obtained results clearly demonstrate that our method outperforms the current state-of-the-art distillation approach for semantic segmentation. This finding provides strong evidence that the searched distillers effectively enhance the learning process of the student model.

Visualizations. Figure 5 showcases the visualization results of DeepLabV3-ResNet18 trained with Auto-KD and traditional KD methods. The Auto-KD approach results in more consistent dense-pixel classification, as demonstrated through the superior segmentation performance of the resulting segmenter. Accurate and consistent pixel labeling is of utmost importance for downstream tasks like object recognition and tracking, making this particularly crucial for image segmentation tasks [56]. The results of the study indicate that the proposed Auto-KD approach is more suitable for distilling knowledge from a teacher to a student model, resulting in enhanced segmentation performance.

Table 7. mIoU (%) results of CityScapes segmentation.

Teacher	DeepLabV3-R101(78.07)	
Student	DeepLabV3-R18(74.21)	PSPNet-R18(72.55)
SKD [52]	75.42 (1.21↑)	73.29 (0.74↑)
IFVD [74]	75.59 (1.38↑)	73.71 (1.16↑)
CWD[64]	75.55 (1.34↑)	74.36 (1.81↑)
CIRKD [75]	76.38 (2.17↑)	74.73 (2.18↑)
Auto-KD	**77.35 (3.14↑)**	**76.25 (3.70↑)**

3.6. Ablation Study

In this section, We analyze the impact of each component of Auto-KD in isolation and explore different variations of these components.

Search space & algorithm. Our well-designed search space with a tree-based structure reduces the problem's complexity, enabling the MCT search to explore promising distillers more efficiently. Results in Table 8 illustrate the clear advantages of our search method over naïve organizations. In addition, we involve other advanced distillation operations into the extended space, resulting in additional gains. For the search algorithm, our MCT search obtains stable benefits compared to random search.

Table 8. Ablation of search space for ResNet-20 on CIFAR-100.

Space	Naïve	Ours	Ours+ Extension
Random	69.82%±0.44	71.55%±0.51	71.98%±0.42
MCTS	70.32%±0.24	72.52%±0.22	72.65%±0.16

Table 9. Ablation on reward for ResNet-20 on CIFAR-100.

Reward	Acc.	$\mathcal{L}_{CE}$	$\mathcal{L}_{CE}$ +$\mathcal{L}_{L1}$	$\mathcal{L}_{CE}$ +$\mathcal{L}_{CKA}$
Top-1 (%)	72.25± 0.25	72.36±0.21	72.45±0.18	72.52±0.22

Comparing different reward functions. The design of the reward function is crucial as it guides the search for optimal solutions for the MCT search. In Table 9, we compare the performance of using accuracy, $\mathcal{L}_{CE}$, and other losses of the student models on the validation set. The results indicate that employing the loss value is better than direct accuracy and involving the CKA distance or L_1 distance of the teacher model results in stable improvements.

3.7. Multi-Layer & Multi-Teacher Extensions

Multi-layer distillation. As a generic framework, Auto-KD can be naturally used in multi-layer and cross-layer scenarios with our Train-Free factor (TF). To explore its potential, we choose KR and DistPro [11] as the references and train the student model with the same setting. Table 10 shows the results from which we can observe Auto-KD combined with TF can achieve better performance than KR+DistPro. In addition, the TF strategy with KD also achieves competitive gain over DistPro and is superior in efficiency by avoiding the meta-optimization process.

Table 10. Top-1 accuracy (%) of different multi-layer distillations.

Method	KR [55]	KR+DistPro [11]	KR+TF	Auto-KD+TF
Multi-layer	71.89±0.05	71.93±0.26	72.05±0.12	72.55±0.08
Cross-layer	71.92±0.16	72.03±0.28	72.18±0.18	72.62±0.17

Multi-Teacher distillation. Our Auto-KD with a train-free factor can also be employed for multi-teacher distillation incorporating intermediate features. The results in Table 11 demonstrate that TF and Auto-KD+TF consistently outperform AVEG and AEKD. The experimental results validate the applicability of Auto-KD in multi-teacher KD.

Table 11. Top-1 accuracy (%) ofdifferent multi-teacher distillations.

Model	Teacher-3 ResNet8×4	Teacher-2 ResNet20×4	Teacher-3 ResNet32×4	Student VGG8
Vanilla Acc	72.79	78.39	79.31	70.74±0.40
KD [24]	AVEG [63]	AEKD [63]	TF	Auto-KD-TF
KD Acc.	74.55±0.24	74.69±0.29	75.38±0.25	76.52±0.15

4. Related Work

Knowledge Distillation. Inspired by pioneering studies [1, 4], the original Knowledge Distillation (KD) [24] leverages soft logits from a pre-trained teacher as additional supervision to guide the training of the student, alongside ground truth labels. Subsequently, various feature distillation techniques [61, 79, 82, 51] are proposed, which focus on the intermediate feature representations. Additionally, relation distillation methods demonstrate that capturing relations [54]

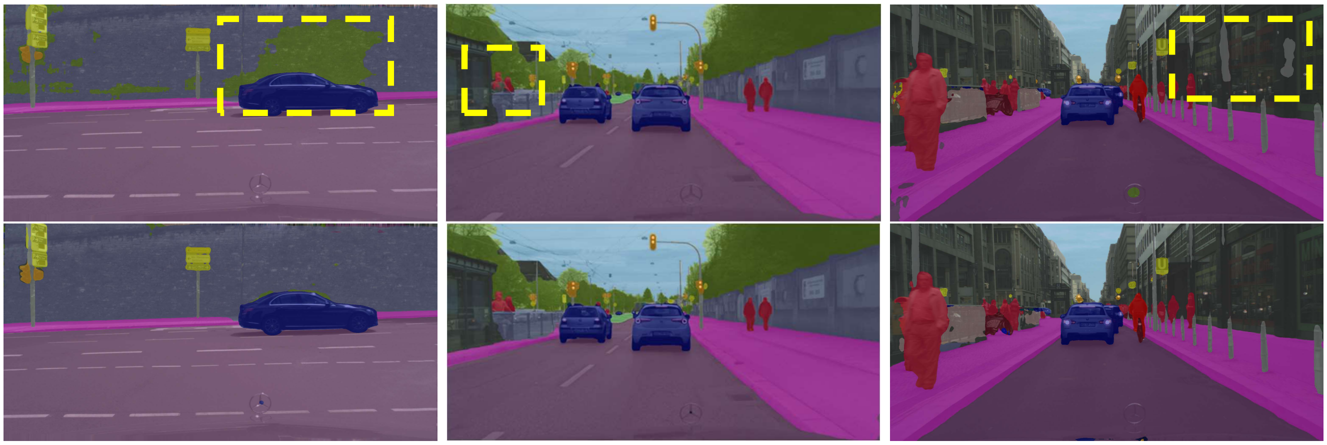

Figure 5. Qualitative segmentation results on the validation set of Cityscapes using the DeepLabV3-ResNet18 network: (first row) the original student network with channel-wise distillation, (second row) the original student network with our method. The yellow box highlights some of the incorrect classifications for comparison methods.

and higher-order dependencies [66] between logits or intermediate feature representations learned by the teacher provides valuable structural information. Furthermore, KD has been extended to multi-teacher models [63, 39], self-KD scenarios [2, 16, 38, 41, 40]), and diverse applications [17, 31]). Researchers have designed advanced transformations [29], distance functions [64], and weight-tuning strategies [48] to further enhance the effectiveness of distillation. However, these KD designs heavily rely on expert knowledge and manual tuning, leading to performance variations across different settings. Auto-KD addresses these challenges by introducing automated searches, ushering in a new era for KD research and applications.

Compared to Meta-KDs. These works [11, 48] only focus on hyperparameters tuning and involve complex optimization challenges. In contrast, our approach not only searches for hyperparameters but also for specific KD designs, resulting in additional performance improvement.

Automated Machine Learning. AutoML [21] is presented to automate Neural Network Architecture Search (NAS) and HyperParameter Optimization (HPO) and make them accessible to non-experts. NAS exploits gradient [47, 8], one-shot [27, 80, 14, 28, 13], and train-free strategies [7, 44, 12] to choose architecture rather than KD designs.

Compared to HPO methods. HPOs [58, 76] build search spaces and utilize search strategies to select the optimal values for the hyperparameters that control the behavior of the model. Recent methods search for loss formation [43, 35]. In contrast, our search space is complex, consisting of hyperparameters like loss weights and other searches for transformations & distances. In addition, our method employs advanced MCT search [3] and first explores the organization of search space and cost optimization in KD.

Compared to Auto-Zero. Auto-Zero methods [35, 59] opt to search from scratch and typically yield marginal gains due to the search space's sparsity and the search's inefficiency. To address these issues, the Auto-KD search starts with successful distillation operations, greatly improving search efficiency and generalizability.

5. Conclusion

In this paper, we present the Auto-KD framework, a novel method for automatically designing distillers. Auto-KD involves constructing a versatile and cohesive distiller search space, incorporating successful search operators derived from a thorough understanding of the decomposability, combinability, and simplicity exhibited in existing distillation methods. To enhance the efficiency of distiller search and enhance the performance of student models during distillation training, we employ Monte Carlo tree search and search acceleration strategies. Through extensive experimentation on three benchmarks, including CNNs, Vision Transformer models, object detection, and semantic segmentation, we demonstrate the effectiveness and broad applicability of the Auto-KD framework. We aspire that our work serves as an inspiration for future research in the design of knowledge distillation methods.

Limitations. Following general KD and AutoML methods, we search on classification tasks and transfer searched distiller to downstream tasks to verify generalization. In future work, we will make efforts to expand the Auto-KD with task-specific search space design for object detection, semantic segmentation and other downstream tasks.

References

[1] Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NIPS*, 2014.

[2] Hessam Bagherinezhad, Maxwell Horton, Mohammad Rastegari, and Ali Farhadi. Label refinery: Improving imagenet

classification through label progression. *arXiv preprint arXiv:1805.02641*, 2015.

[3] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 2012.

[4] Cristian Bucilă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, 2006.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *CVPR*, 2021.

[7] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. In *ICLR*, 2020.

[8] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[10] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NIPS*, 2015.

[11] Xueqing Deng, Dawei Sun, Shawn Newsam, and Peng Wang. Distpro: Searching a fast knowledge distillation process via meta optimization. 2022.

[12] Peijie Dong, Lujun Li, and Zimian Wei. Diswot: Student architecture search for distillation without training. In *CVPR*, 2023.

[13] Peijie Dong, Xin Niu, Lujun Li, Zhiliang Tian, Xiaodong Wang, Zimian Wei, Hengyue Pan, and Dongsheng Li. Rd-nas: Enhancing one-shot supernet ranking ability via ranking distillation from zero-cost proxies. *arXiv preprint arXiv:2301.09850*, 2023.

[14] Peijie Dong, Xin Niu, Lujun Li, Linzhen Xie, Wenbin Zou, Tian Ye, Zimian Wei, and Hengyue Pan. Prior-guided one-shot neural architecture search. *arXiv preprint arXiv:2206.13329*, 2022.

[15] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *Proceedings of Machine Learning and Systems 2020*. 2020.

[16] Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *ICML*, 2018.

[17] Nuno C Garcia, Pietro Morerio, and Vittorio Murino. Modality distillation with multiple stream networks for action recognition. In *ECCV*, 2018.

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[19] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. In *NIPS*, 2015.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *arXiv preprint arXiv:1908.00709*, 2019.

[22] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *ICCV*, 2019.

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

[24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *arXiv:1503.02531*, 2015.

[26] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint, arXiv:1704.04861*, 2017.

[27] Yiming Hu, Yuding Liang, Zichao Guo, R. Wan, X. Zhang, Y. Wei, Q. Gu, and J. Sun. Angle-based search space shrinking for neural architecture search. In *ECCV*, 2020.

[28] Yiming Hu, Xingang Wang, Lujun Li, and Qingyi Gu. Improving one-shot nas with shrinking-and-expanding supernet. *Pattern Recognition*, 2021.

[29] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge distillation from a stronger teacher. *arXiv preprint arXiv:2205.10536*, 2022.

[30] Adrian Bulat Georgios Tzimiropoulos Jing Yang, Brais Martinez. Knowledge distillation via softmax regression representation learning. In *ICLR2021*, 2021.

[31] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *EMNLP*, 2016.

[32] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In *ICML*, 2019.

[33] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[35] Hao Li, Tianwen Fu, Jifeng Dai, Hongsheng Li, Gao Huang, and Xizhou Zhu. Autoloss-zero: Searching loss functions from scratch for generic tasks. In *CVPR*, 2022.

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

[37] Kehan Li, Runyi Yu, Zhennan Wang, Li Yuan, Guoli Song, and Jie Chen. Locality guidance for improving vision transformers on tiny datasets. *arXiv preprint arXiv:2207.10026*, 2022.

[38] Lujun Li. Self-regulated feature learning via teacher-free feature distillation. In *ECCV*, 2022.

[39] Lujun Li and Zhe Jin. Shadow knowledge distillation: Bridging offline and online knowledge transfer. In *NeuIPS*, 2022.

[40] Lujun Li, Liang Shiuan-Ni, Ya Yang, and Zhe Jin. Boosting online feature transfer via separable feature fusion. In *IJCNN*, 2022.

[41] Lujun Li, Liang Shiuan-Ni, Ya Yang, and Zhe Jin. Teacher-free distillation via regularizing intermediate representation. In *IJCNN*, 2022.

[42] Xiaojie Li, Jianlong Wu, Hongyu Fang, Yue Liao, Fei Wang, and Chen Qian. Local correlation consistency for knowledge distillation. In *ECCV*, 2020.

[43] Zelong Li, Jianchao Ji, Yingqiang Ge, and Yongfeng Zhang. Autolossgen: Automatic loss function generation for recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1304–1315, 2022.

[44] Ming Lin, Pichao Wang, Zhenhong Sun, Hesen Chen, Xiuyu Sun, Qi Qian, Hao Li, and Rong Jin. Zen-nas: A zero-shot nas for high-performance image recognition. 2021.

[45] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017.

[46] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[47] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *7th ICLR, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

[48] Jihao Liu, Boxiao Liu, Hongsheng Li, and Yu Liu. Meta knowledge distillation. *arXiv preprint arXiv:2202.07940*, 2022.

[49] Li Liu, Qingle Huang, Sihao Lin, Hongwei Xie, Bing Wang, Xiaojun Chang, and Xiaodan Liang. Exploring inter-channel correlation for diversity-preserved knowledge distillation. In *ICCV*, 2021.

[50] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Li Shen, Decebal Constantin Mocanu, Zhangyang Wang, and Mykola Pechenizkiy. The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training. *arXiv preprint arXiv:2202.02643*, 2022.

[51] Xiaolong Liu, Lujun Li, Chao Li, and Anbang Yao. Norm: Knowledge distillation via n-to-one representation matching. In *ICLR*, 2023.

[52] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *CVPR*, 2019.

[53] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[54] Wonpyo Park, Yan Lu, Minsu Cho, and Dongju Kim. Relational knowledge distillation. In *CVPR*, 2019.

[55] Chen Pengguang, Liu Shu, Zhao Hengshuang, and Jiaya Jia. Distilling knowledge via knowledge review. In *CVPR*, 2021.

[56] Jie Qin, Jie Wu, Xuefeng Xiao, Lujun Li, and Xingang Wang. Activation modulation and recalibration scheme for weakly supervised semantic segmentation. In *AAAI*, 2022.

[57] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Joseph. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, 2016.

[58] Esteban Real, Chen Liang, David So, and Quoc Le. Automl-zero: Evolving machine learning algorithms from scratch. In *ICML*, 2020.

[59] Esteban Real, Chen Liang, David R. So, and Quoc V. Le. Automl-zero: Evolving machine learning algorithms from scratch, 2020.

[60] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint, arXiv:1506.01497*, 2015.

[61] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[62] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[63] Xiaojie Li Jianlong Wu Fei Wang Chen Qian Shangchen Du, Shan You and Changshui Zhang. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. In *NeurIPS*, 2020.

[64] Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5311–5320, 2021.

[65] Maciej Swiechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mandziuk. Monte carlo tree search: A review of recent modifications and applications. *Artificial Intelligence Review*, pages 1–66, 2022.

[66] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

[67] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

[68] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers amp; distillation through attention. In *ICML*, 2021.

[69] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, 2019.

[70] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[71] Linnan Wang, Yiyang Zhao, Yuu Jinnai, Yuandong Tian, and Rodrigo Fonseca. Alphax: exploring neural architectures with deep neural networks and monte carlo tree search. *arXiv preprint arXiv:1903.11059*, 2019.

[72] 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*.

[73] 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. *Computational Visual Media*, 2022.

[74] Yukang Wang, Wei Zhou, Tao Jiang, Xiang Bai, and Yongchao Xu. Intra-class feature variation distillation for semantic segmentation. In *ECCV*, 2020.

[75] Chuanguang Yang, Helong Zhou, Zhulin An, Xue Jiang, Yongjun Xu, and Qian Zhang. Cross-image relational knowledge distillation for semantic segmentation. In *CVPR*, 2022.

[76] Jie Yang et al. Automatically labeling video data using multi-class active learning. In *Proceedings Ninth IEEE international conference on computer vision*, pages 516–523. IEEE, 2003.

[77] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. *arXiv preprint arXiv:2111.11837*, 2021.

[78] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. *arXiv preprint arXiv:2205.01529*, 2022.

[79] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 2017.

[80] Shan You, Tao Huang, Mingmin Yang, Fei Wang, Chen Qian, and Changshui Zhang. Greedynas: Towards fast one-shot nas with greedy supernet. In *CVPR*, 2020.

[81] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, 2021.

[82] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[83] Linfeng Zhang and Kaisheng Ma. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *International Conference on Learning Representations*, 2020.

[84] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[85] Shuchang Zhou, Wu Yuxin, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

[86] Zaida Zhou, Chaoran Zhuge, Xinwei Guan, and Wen Liu. Channel distillation: Channel-wise attention for knowledge distillation. *arXiv preprint, arXiv:2006.01683*, 2020.