Auto-DAS: Automated Proxy Discovery for Training-free Distillation-aware Architecture Search

Haosen Sun[1], Lujun Li[1]✉, Peijie Dong[2], Zimian Wei[3], and Shitong Shao[4]

[1] The Hong Kong University of Science and Technology
[2] The Hong Kong University of Science and Technology (Guangzhou)
[3] National University of Defense Technology
[4] Southeast University
hsunas@connect.ust.hk;lilujunai@gmail.com;pdong212@connect.hkust-gz.edu.cn;
weizimian16@nudt.edu.cn; shaoshitong@seu.edu.cn

Abstract Distillation-aware Architecture Search (DAS) seeks to discover the ideal student architecture that delivers superior performance by distilling knowledge from a given teacher model. Previous DAS methods involve time-consuming training-based search processes. Recently, the training-free DAS method (*i.e.*, DisWOT) proposes KD-based proxies and achieves significant search acceleration. However, we observe that DisWOT suffers from limitations such as the need for manual design and poor generalization to diverse architectures, such as the Vision Transformer (ViT). To address these issues, we present Auto-DAS, an automatic proxy discovery framework using an Evolutionary Algorithm (EA) for training-free DAS. Specifically, we empirically find that proxies conditioned on student instinct statistics and teacher-student interaction statistics can effectively predict distillation accuracy. Then, we represent the proxy with computation graphs and construct the proxy search space using instinct and interaction statistics as inputs. To identify promising proxies, our search space incorporates various types of basic transformations and network distance operators inspired by previous proxy and KD-loss designs. Next, our EA initializes populations, evaluates, performs crossover and mutation operations, and selects the best correlation candidate with distillation accuracy. We introduce an adaptive-elite selection strategy to enhance search efficiency and strive for a balance between exploitation and exploration. Finally, we conduct training-free DAS with discovered proxy before the optimal student distillation phase. In this way, our auto-discovery framework eliminates the need for manual design and tuning, while also adapting to different search spaces through direct correlation optimization. Extensive experiments demonstrate that Auto-DAS generalizes well to various architectures and search spaces (*e.g.*, ResNet, ViT, NAS-Bench-101, and NAS-Bench-201), achieving state-of-the-art results in both ranking correlation and final searched accuracy. Code at: https://github.com/lliai/Auto-DAS.

Keywords: Knowledge Distillation · Architecture Search · AutoML

✉ Corresponding author & contributed equally to this work.

1 Introduction

Knowledge Distillation (KD) transfers logits [17, 52], features [26, 32], or relation [40, 47] knowledge of heavy teacher models as additional supervised signals to augment lightweight student models. KD originally served as model compression technique for large models and has recently been used as a common strategy for performance improvement in different scenarios. For example, distilling a CNN teacher for a Vision Transformer (ViT) [49] can significantly enhance its performance, particularly when training from scratch in data-limited scenarios. However, the architecture gap between the teacher and student network creates a mismatch between the receptive field and the prediction reliability [38, 56]. These disparities contribute to distillation gains, *i.e.*, students with a smaller architecture gap with teachers will enjoy better final results.

To alleviate these problems, a new task, Distillation-aware Architecture Search (DAS), is presented to search for an optimal student architecture from the given teacher, which obtains the best performance with the help of neural architecture search. In contrast to the traditional NAS methods, the DAS approach needs to consider the effect of the teacher and employ distillation accuracy as the search evaluation metric. Some observations [7,36] indicate that the models searched by traditional NAS methods [9, 20, 35] are suboptimal in distillation, and there is a common performance ranking gap for obtained students between vanilla training and distillation. For the DAS task, some training-based methods leverage Reinforcement Learning (RL) [36] or gradient method [14] as search strategies to find the ideal student. However, these methods bring in high search costs (*e.g.*, 5 days with 200 TPUv2 in AKD [36]), which causes heavy burdens for application. Recently, DisWOT [7] first presents the training-free DAS framework with some KD-based proxies, exploiting the connection between training-free proxies and knowledge distillation loss. It achieves significant search acceleration by using statistics between random networks for predicting distillation accuracy without candidate training. However, some drawbacks still limit its extensive application, which could highlight from two aspects: **(1) Expert design and tuning costs.** These KD-based proxies are inspired by some distillation studies with extensive expert intuition and require extensive tuning. It is hard to state if these proxies are optimal for DAS as limited by the designer's experience. **(2) Generalization issue.** Handcrafted ones only perform well on some specific settings but are hard to adapt to different scenarios without changing/modification. As shown in Figure 1, existing handcrafted proxies perform well on the CNNs but do not generalize well to ViT models. Thus, two questions arise: ***(1) How to efficiently discover the proxies without expert knowledge? (2) How to find the optimal proxies for different search spaces?***

For the first problem, we observe that previous proxy expressions are usually constructed for typical inputs and common primitive operations. Inspired by Auto-Zero [42], we explore a from-scratch proxy discover framework alternative to handcrafted for DAS task. For the second problem, we can evolve various proxy candidates by directly taking their predictability for distillation accuracy as feedback. In addition, we find that proxies with inputs from student

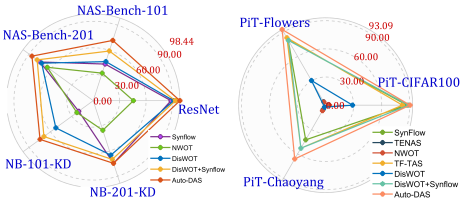

Figure 1: Spearman correlation (%) on CNN models(*left*) and ViT models(*right*).

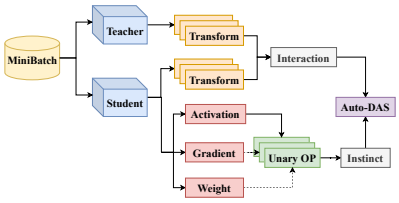

Figure 2: Illustration of Auto-DAS.

instinct statistics and teacher-student interaction statistics consistently achieve better results. For example, DisWOT+Synflow in Figure 1 achieves stable improvements on multiple search spaces. Based on these observations, we present Auto-DAS, a from-scratch framework for discovering proxies for DAS tasks are proposed as an alternative to manually designed ones. As shown in Figure 2, we first build the proxy search space, consisting of both instinct and interaction proxy. The instinct proxy is used with student's weights and gradient statistics as input and different unary and binary mathematical operations as candidate components. For interaction proxy, we select teacher-student features, embeddings, logits as inputs, different basic transformation operations, and distance functions to form the computation graph. Then, we randomly sample various proxies as the population and then perform mutation and crossover operations according to the ranking ability of different proxies. To improve generality, we directly treat the Kendall correlation between the prediction of the candidate proxy and the ground truth under distillation as the fitting objective of the evolutionary algorithm. To speed up the process, we employ an adaptive-elite selection strategy during the evolution search. With the automatic search framework, our Auto-DASsurpasses existing training-free NAS approaches by a large margin without involving too much prior knowledge. Finally, after getting the Auto-DAS proxy in a given search space, we search for the best student architectures with the highest Auto-DAS score and apply the searched students to distill the knowledge from the given teacher.

In comparison to manual DAS techniques, our Auto-DASframework offers several advantages: (1) **Autonomy.** It facilitates the spontaneous discovery of more potent and efficient proxies that may elude human experts, thus mitigating bias and ensuring that the resultant architectures are tailored for the intended problem or dataset. (2) **Versatility.** The search mechanism of Auto-DASenhances adaptability by uncovering proxies that perform well across diverse search spaces. Our framework can evolve flexible proxies to generalize across architectures, based on the ultimate distillation outcomes for the search objectives. (3) **Enlightening.** Our approach enables the automatic identification of more expressive, efficient, and effective proxies for optimizing and customizing student architectures, paving the way for novel research and applications in DAS and KD tasks. (4) **Efficacy.** We have conducted comprehensive experiments on ResNet, NAS-Bench-101, NAS-Bench-201, and ViT-Bench search spaces to confirm the

superiority of our Auto-DAS. The results demonstrate that our Auto-DAScan surpass other zero-shot proxies in distillation accuracy and achieve state-of-the-art results in rank consistency, such as a 40% improvement in Kendall correlation over DisWOT. To summarize, our key contributions to Auto-DAS are:

- We introduce the pioneering framework for discovering proxies from scratch, Auto-DAS, to address handcrafted limitations and enhance generalizability in predicting distillation accuracy across various students.
- We devise a well-structured proxy search space, incorporating instinct and interaction statistics as inputs, along with diverse unary, binary, transforms, and distance functions as operations. We develop evolutionary algorithms with adaptive-elite selection for effective and efficient proxy search.
- We provide empirical evidence that Auto-DASachieves cutting-edge performance across ResNet, NAS-Bench-101, NAS-Bench-201, ViT-Bench search spaces and multiple datasets.

2 Related work

Distillation-aware architecture search. The increasing demand for practical applications of large-scale models [5, 12] has led to impressive advancements in Knowledge distillation (KD) across various domains [26–30, 32, 43], including image classification [17], object detection [51, 55], and NLP [21]. However, discrepancies between teacher and student architectures can substantially impact distillation outcomes. Consequently, distillation-aware student architecture search for a given teacher model has emerged as a crucial new area of research. Traditional reinforcement learning-based search methods [36] utilize KD-loss as feedback for DAS but incur significant additional costs during the search process. Subsequent research employs efficient gradient-based methods [14], albeit introducing new optimization challenges. Recently, DisWOT [7] introduced the first training-free DAS framework. However, DisWOT relies on a manually designed proxy, which lacks generalization across different search spaces. To address this limitation, we propose the first automated proxy discovery for training-free DAS tasks. Our method not only retains the advantages of DisWOT, such as efficiency and flexibility, but also automatically optimizes proxies without expert intervention. Auto-DAS paves the way for novel DAS research and offers valuable insights based on search result analysis for the research community.

Training-free architecture search. Conventional NAS [18, 60] involves designing search spaces, algorithms, and evaluations to automatically discover optimal architectures under given constraints. Training-based methods employ train-then-search or weight sharing processes [19, 41, 54, 60]. Training-free NAS, also known as zero-shot NAS [6, 8, 31, 33, 34, 57, 59], enables faster search by predicting performance without training. Zero-cost proxies are categorized as parameter-level or architecture-level. Parameter-level approaches rely on pruning and saliency values [23, 45, 50], while architecture-level methods evaluate expressiveness [34] or linear map correlations [37]. DisWOT proposed KD-based

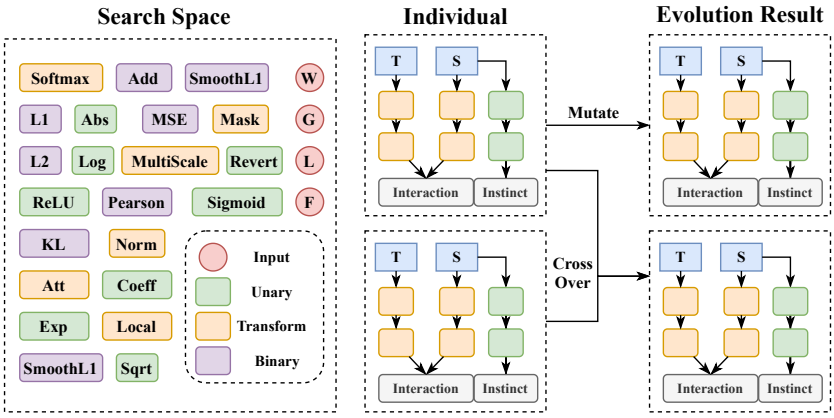

Figure 3: The overall illustration of Auto-DAS. Our search space includes unary, binary, transform, and distance options with different inputs. Then, we construct our intrinsic & interaction proxies and use crossover & mutation to search for the best proxy.

Table 1: Operations of Auto-DAS. More details are available in the Appendix.

Proxy	Input	Type	Primitive Operations
instinct	Activations	Unary	$exp, mish, leaky, relu, tanh, softmax, sigmoid, pow2, pow4, log, abslog, abssqrt$
	Gradients	Unary	$l1_{norm}, revert, min-maxto-mean, to-std, invert, ln_{norm}, normalized_{sum}, no-op$
	Weights	Binary	$sum, subtract, multiply, divide, dot.$
Interaction	Feature	Transform	$tanh, log, sqrt, batchnorm, min-max, norm, softmax, exp, mish, leaky, relu$
	Embed	Transform	$mask, no, bmm, mm, scale, multi-scale, scale, local, batch, channel, drop, satt, natt, catt$
	Logits	Distance	ℓ_1, ℓ_2, ℓ_{KL}, ℓ_{hard}, ℓ_{Cosine}, $\ell_{Pearson}$

proxies, but their fixed formulations limit generalization. To overcome these limitations, we consider student internal statistics and teacher-student similarities as inputs and evolve superior proxies from scratch for training-free DAS tasks. Inspired by Auto-Zero, our innovations include a novel proxy search space and a specialized search process, promoting automatic proxy design for distillation. Our approach differs from EZNAS [2] in terms of tasks (ours is customized for CNNs and ViTs), search algorithms (ours employs adaptive-elite selection), and search space (ours incorporates teacher-student knowledge). Unlike EZNAS and Auto-KD, which focus solely on vanilla classification, we directly search on the distillation task and include teacher-student knowledge and additional options in the search space.

3 Methodology

This section delves into the intricacies of existing proxies, our proposed search space, the search algorithm, and an analysis of the results. Subsequently, we outline a training-free student search process utilizing the discovered zero-cost proxy. Figure 3 illustrates the workflow of our approach.

3.1 Examination of existing proxies

To provide context for our training-free method, we first examine current zero-cost proxies: SNIP [23] calculates a saliency metric at initialization using a single mini-batch of data to estimate the loss change when a specific parameter is eliminated. Synflow [45] introduces a modified version of synaptic saliency scores that prevents layer collapse during parameter pruning, while Fisher [46] computes the sum over all activation gradients in the network, applicable for channel pruning. Their detailed formulations are as follows:

$$\rho_{snip} = \left| \frac{\partial \mathcal{L}}{\partial \mathcal{W}} \odot \mathcal{W} \right|, \rho_{synflow} = \frac{\partial \mathcal{L}}{\partial \mathcal{W}} \odot \mathcal{W}, \rho_{fisher} = \left(\frac{\partial \mathcal{L}}{\partial \mathcal{A}} \mathcal{A} \right)^2, \tag{1}$$

where $\mathcal{L}$, $\mathcal{W}$, $\mathcal{A}$ are loss function, weight and activation. H is the Hessian matrix. These proxies consist of various unary, binary operations with statistics input from the candidate model, named instinct proxies in our paper. For the distillation scenario, DisWOT presents some KD-based proxies based on KD-loss design, as follows:

$$\rho_{DisWOT} = \mathcal{D}_{L2}(\mathcal{G}([\mathcal{A}S, \mathcal{A}T])) + \mathcal{D}_{L2}(\mathcal{G}([\mathcal{F}S, \mathcal{F}T])), \tag{2}$$

where $\mathcal{A}$, $\mathcal{F}$, $\mathcal{E}$, $\mathcal{P}$ are activation maps, feature maps, embeddings, and logits prediction. $[\cdot, \cdot]$ means to perform the same operation on two primitives. $\mathcal{G}$ is the gram-matrix. $\mathcal{D}_{L2}$, $\mathcal{D}_{KL}$ are ℓ_2 and ℓ_{KL} distance. These KD-based proxies use teacher-student pair-wise statistics as input and different transformations and distance functions to construct formations that we denote as interaction proxies. Based on these analyses, we build search spaces for instinct and interactive proxies in the next part.

3.2 Search Space for Proxy Discovery

Our search space structure comprises three primary components: input choices, primitive operations, and proxy representation. As illustrated in Table 1, for input choices in constructing zero-cost proxies, the instinct proxy extracts activations, gradients, and weights from the student model, while the interaction proxy selects features, embeddings, and logits of the teacher-student network. These inputs represent the most informative options common across traditional and KD-based proxies. For instance, features provide insights into data distribution and internal network representations, while gradients highlight weight sensitivity to loss functions. Following this observation, we register each student layer's activation, weights, and corresponding gradients as potential inputs for our zero-cost proxy. Regarding primitive operations, we consider unary and binary operations for instinct proxies, and transform and distance operations for interaction proxies. We include mathematical, normalization, and scaling operations for unary operations; matrix sums and products for binary operations; and activation, normalization, scaling, and other transformations, along with typical distance functions for interaction proxies. Our proxy is represented as a computation graph, where input nodes are instinct and interaction statistics, and

Algorithm 1 Evolutionary Strategy for Auto-DAS Proxy Discovery

Input: Architecture space $\mathcal{S}$, candidate set $\mathcal{P}$, iteration limit $\mathcal{T}$, selection proportion r, temporary set $\mathcal{R}$, elite count k.
Output: Highest-scoring Auto-DAS configuration.

Initialize candidate set $\mathcal{P}0$:= GenerateInitialCandidates(P_i);
Empty temporary set $\mathcal{R} := \emptyset$;
for $i = 1, 2, \ldots, \mathcal{T}$ **do**
    Reset temporary set $\mathcal{R} := \emptyset$;
    Populate $\mathcal{R}$ with random selections from $\mathcal{P}$;
    Elite group G_ik := SelectTopPerformers($\mathcal{R}$, k);
    Choose parent G_i^p := RandomlyPick(G_{ik});
    Generate offspring G_i^m := MUTATE(G_i^p);
    // Adaptive Elite Preservation
    Randomly generate baseline E;
    if $\rho(G_i^m) > \rho(E)$ **then**
        Include G_i^m in P;
    else
        Include E in P;
    end if
    Eliminate lowest-scoring proxy from population.
end for

intermediate nodes are primitive operations. The graph's output is the proxy score used for distillation ranking. Our comprehensive search space encompasses most zero-cost proxies, KD-based losses, and model similarity formulations, constituting our core innovation and contribution.

3.3 Evolution Procedure and Objective

Utilizing our proxy search space, we employ an Evolutionary Algorithm (EA) to identify optimal proxy expressions (refer to Figure 3 and Algorithm 1). Our EA initiates with a population of candidate proxies and iteratively evolves it over generations using genetic operators such as selection, crossover, and mutation to generate superior solutions. For objective fitting, each proxy is evaluated based on the ranking correlation between its output proxy $\mathcal{Q}$ scores and actual performances $\mathcal{D}$ to efficiently discover the optimal proxy ρ^* from search space $\mathcal{S}$, as follows:

$$\rho^*_{Auto-DAS} = \arg\max \rho \in \mathcal{S}(\tau(\mathcal{D}, \mathcal{Q})), \tag{3}$$

where Kendall's Tau τ serves as the correlation coefficient. Each evolution generation selects the top-k candidates with the highest Auto-DAS scores and randomly chooses a parent from these candidates for mutation. During mutation, a node in the computation graph is randomly selected and mutated with newly generated primitive operations.

Efficient proxy search with existing benchmark. Obtaining pre-trained results $\mathcal{D}$ can be resource-intensive. To address this, we leverage pre-computed architecture-performance pairs from existing DisWOT benchmarks [7]. These

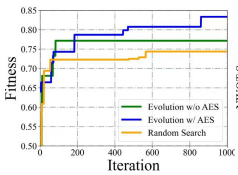

Figure 4: Search curve on ResNet.

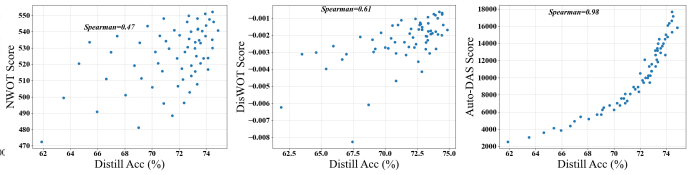

Figure 5: Correlation visualization of NWOT (*left*), DisWOT (*middle*), Auto-DAS (*right*) on ResNet search space.

benchmarks provide distillation results for various ResNet and NAS-Bench-101/201 search space models. Utilizing this benchmark eliminates the need to retrain performance ground-truths and significantly reduces the proxy search process time to **merely 0.05 GPU-day**. Due to its excellent generality (demonstrated in our extensive experiments), our searched proxy $\rho_{Auto-DAS}$ transfers well to similar search spaces and can be directly applied in most scenarios without additional search budget.

Adaptive-elite selection strategy. To mitigate population deterioration and premature convergence, we introduce an adaptive-elite selection strategy. This approach involves comparing the performance of mutation-generated offspring with that of a randomly generated individual. The individual with a higher Auto-DAS score is then added to the population. By maintaining or enhancing the overall population performance over time, this strategy accelerates convergence towards a high-performing solution (see Figure 4).

3.4 Searched Zero-cost Proxy

We present formulas of the searched proxies on ρ_{CNN} for CNN Search space (*e.g.*, ResNet, NAS-Bench) and ρ_{ViT} ViT search space as follows:

$$\rho_{CNN} = \mathcal{D}_{KL}(\text{sigmoid}([\mathcal{F}_S, \mathcal{F}_T])) + \frac{1}{M}\sum |\text{logsoftmax}(\frac{\partial L}{\partial W})|_F, \tag{4}$$

$$\rho_{VIT} = \mathcal{D}_{KL}\left(\text{mask}\left([\mathcal{F}_S, \mathcal{F}_T]\right)\right) + \frac{M}{\sum \text{sigmoid}\left(\left|\log\left|\frac{\partial L}{\partial W}\right|\right|\right)} \tag{5}$$

Where W is the weight parameter, $\partial\mathcal{L}/\partial W$ is the corresponding gradient. $\|\cdot\|_F$ means the Frobenius-norm. M denotes mean operation.

Analysis of searched proxy. The discovered formulations indicate that (1) mathematical operations like softmax, logsoftmax, sigmoid, and mask operations benefit proxy predictability. (2) Gradient and features are vital input statistics for instinct and interaction proxies. As shown in Figure 5, the discovered proxies of Auto-DAS achieve significant ranking improvement.

Understanding why our Auto-DAS surpasses other NAS methods. (1) Many training-based distillation NAS methods use weight sharing and proxies, often leading to inaccurate performance estimates. Our Auto-DAS searches proxies based on real performance, yielding one with strong ranking ability. (2)

Table 2: Spearman correlation (%) results on the NAS-Bench-101 and NAS-Bench-201 (NB-101/201) datasets. In this context, NB-101/201-KD denotes the task of distilling the accuracies of the architectures on the respective NAS-Bench-101/201 datasets.

Method	ResNet	NB-101	NB-201	NB-101-KD	NB-201-KD
FLOPs	72.92	30.81	63.38	15.56	64.55
Fisher	81.37	-38.81	35.91	-33.92	4.45
Grad_Norm	82.35	-39.23	58.70	-39.16	-10.01
SNIP	85.07	-29.01	58.17	-21.78	16.91
Synflow	88.30	43.69	74.61	20.36	74.63
NWOT	45.66	32.84	64.41	22.97	35.27
DisWOT(M_s)	77.24	49.61	65.74	50.16	53.88
DisWOT(M_r)	49.38	30.74	56.46	42.94	45.27
DisWOT	91.38	46.57	72.36	52.45	64.90
Auto-DAS	**98.44**	**71.84**	**87.73**	**85.95**	**74.66**

Other train-free NAS methods primarily target vanilla training, not distillation. They rely on weak correlation proxies. Ours exhibits a stronger correlation, ensuring more efficient model discovery. While DisWOT considers distillation in its proxy design, it is manually crafted and lacks direct optimization and feedback from distillation results. In contrast, our Auto-DAS method not only includes all distillation designs in the proxy search space settings but also optimizes our proxies directly based on the distillation results. Our approach naturally outperforms other methods in terms of distillation performance.

3.5 Training-free Student Search

The training-free student search enables us to efficiently explore numerous candidate architectures without the need for costly and time-consuming training. After obtaining an effective proxy with the aid of an evolutionary search, we employ it to conduct a training-free student search. Utilizing randomly initialized weights, denoted as $\mathcal{W}$, we seek the optimal student, represented as α^*, from the search space $\mathcal{A}$, as follows:

$$\alpha^* = \arg\max_{\alpha \in \mathcal{A}} \rho^*_{Auto-DAS}(\alpha, \mathcal{W}). \tag{6}$$

After the search phase, we use the pre-trained teacher model to distill the optimal student network α^*. We adopt the original knowledge distillation method [17] as the auxiliary loss function.

4 Experimental results

In this section, we present the ranking correlation and search accuracy results of our Auto-DAS on CNN, NAS-Bench, and ViT search space. For fair comparisons, we employ the same search, distillation settings, KD accuracy ground truths in DisWOT as the ground truth for correlation validation. We report mean results based on more than 3 repeated trials.

Table 3: Accuracy (%) of NDS-ResNet space on CIFAR-100.

Constraints	NWOT	Synflow	DisWOT	**Auto-DAS**
5M-FLOPs	63.19	64.28	65.98	**66.52**
100M-FLOPs	70.38	72.12	72.89	**73.25**
0.5M-Param.	70.38	71.58	72.89	**73.32**
1M-Param.	72.57	73.56	74.23	**74.66**

Table 4: Results (%) of other KD methods under 1M parameters.

Method	FitNets	AT	SP	RKD	CRD
Random NAS	70.12	70.16	70.46	71.19	71.59
DisWOT	74.85	74.5	74.95	74.62	75.25
Auto-GAS	**75.28**	**74.98**	**75.55**	**75.31**	**75.86**

Table 5: Top-1 accuracy (%) results on ImageNet.

Models	Baseline	Baseline+WSLD	Baseline+DIST	Random NAS+KD	Zen-NAS+KD	DisWOT	**Auto-DAS**
ResNet18	69.7	72.04	72.07	71.68	71.88	72.08	**72.58**
ResNet50	77.1	NA	78.45	77.48	77.80	78.62	**79.25**

4.1 Experiments on CNN models

Architecture space and methodology. For CNN validation, we focus on ResNet variants, encompassing both cifar-ResNet and NDS-ResNet domains. Our cifar-ResNet space, inspired by He et al. [15], comprises three phases with 1,3,5,7 block options, tailored for CIFAR-100. We also explore the NDS-ResNet realm for both CIFAR-100 and ImageNet-1k search trials. Across all models, we employ our discovered ρ_{CNN} without additional proxy exploration. During knowledge transfer, all identified student networks undergo training using CRD protocols [48], with ResNet56 serving as the mentor model. For ImageNet-1k, we investigate ResNet18/50-caliber students under equivalent parameter restrictions.

Findings on compact datasets. Our ranking analysis is rooted in individual knowledge transfer outcomes within the cifar-ResNet domain, as provided by DisWOT. Table 2 demonstrates that Auto-DAS achieves a notably higher Spearman coefficient compared to Fisher, SNIP, FLOPs, and NWOT. Our exploration experiments, conducted in the NDS-ResNet realm, pit our approach against Synflow, NWOT, and DisWOT, under 0.5M and 1M parameter limits, as well as 50M and 100M FLOPs thresholds. As illustrated in Table 3, Auto-DAS surpasses previous state-of-the-art methods by 0.9%∼2.0%↑ under identical constraints. Furthermore, we evaluate Auto-DAS across various KD techniques (refer to Table 4). The results affirm our method's efficacy, consistently outperforming DisWOT across different KD approaches.

ImageNet performance analysis. Table 5 presents the effectiveness of student models identified by Auto-DAS on ImageNet, with ResNet34/101 acting as teacher networks. For the ResNet18 variant, Auto-DAS attains 72.58% accuracy, outperforming both DisWOT and other KD+NAS methodologies. Similarly, with the ResNet50 model, Auto-DAS achieves a leading accuracy of 79.25%. These findings underscore the superior capability of our Auto-DAS approach in enhancing model precision across various architectures.

Table 6: Evaluation of Architectures on CIFAR-10, CIFAR-100, and ImageNet-16 in NAS-Bench-201 [11]. Distilled Accuracy (%) represents the classification accuracy of the discovered architecture after distillation training. Search Time (s) denotes the computational cost (in GPU-seconds) during the search phase. The performances of NWOT and TE-NAS are obtained from their respective publications. Our proposed Auto-DASapproach demonstrates superior accuracy with rapid search speed.

Type	Model	CIFAR-10			CIFAR-100			ImageNet-16-120		
		Dis. Acc(%)	Time (s)	Speed-up	Dis.Acc(%)	Time (s)	Speed-up	Dis. Acc(%)	Time (s)	Speed-up
Multi-trial	RS	93.63	216K	1.0×	71.28	460K	1.0×	44.88	1M	1.0×
	RL [3]	92.83	216K	1.0×	71.71	460K	1.0×	44.35	1M	1.0×
	BOHB [13]	93.49	216K	1.0×	70.84	460K	1.0×	44.33	1M	1.0×
	RSPS [25]	91.67	10K	21.6×	57.99	46K	21.6×	36.87	104K	9.6×
0ne-shot	GDAS [10]	93.39	22K	12.0×	70.70	39K	11.7×	42.35	130K	7.7×
	DARTS [35]	89.22	23K	9.4×	66.24	80K	5.8×	43.18	110K	9.1×
Zero-shot	NWOT [37]	93.73	2.2K	100×	73.31	4.6K	100×	45.43	10K	100×
	EZNAS [2]	93.63	2.2K	100×	69.82	4.6K	100×	43.47	10K	100×
Zero-shot DAS	DisWOT	93.55	1.2K	180×	74.21	9.2K	180×	47.30	20K	180×
	DisWOT(M_r)	93.49	0.72K	300×	73.62	18.4K	300×	45.63	40K	300×
	Auto-DAS	**93.96**	**1.8K**	**180×**	**74.73**	**13.8K**	**180×**	**47.50**	**30K**	**180×**

4.2 Experiments on NAS-Bench

Experimental setup and search space. The NAS-Bench-101 [53] and NAS-Bench-201 [11] datasets serve as crucial benchmarks for assessing the efficacy of various NAS algorithms. DisWOT offers knowledge distillation outcomes for architectures within the NAS-Bench-101/201 search spaces (referred to as NB-101-KD and NB-201-KD). Adhering to these benchmark protocols, we apply the identified ρ_{CNN} to model search and validate distillation rankings without additional proxy searches. Comprehensive implementation details are available in the Appendix.

Performance analysis. Table 2 illustrates that our Auto-DAS method achieves superior ranking performance, significantly outperforming DisWOT. Furthermore, we evaluate the standard correlation in NAS-Bench-101/201, with the ranking outcomes highlighting Auto-DAS's versatility through substantial improvements over existing techniques. Regarding search trials presented in Table 6, Auto-KD demonstrates remarkable gains in the accuracy-efficiency trade-off compared to multi-trial NAS approaches (such as Random Search (RS) and Reinforcement Learning (RL)) and Gradient-based One-shot NAS methods (like GDAS and DARTS), which typically incur substantial search costs. Our method also consistently enhances distillation accuracy when compared to other Zero-shot NAS techniques (including NWOT, TE-NAS, and DisWOT). These exceptional ranking and search results underscore the robust generalization capabilities of our approach across challenging NAS-Bench datasets.

4.3 Experiments on Vision Transformer

Architecture space and methodology. To assess the ranking consistency of zero-cost proxies for ViT structures, we develop a benchmark called ViT-Bench.

Table 7: Results (%) of Ranking Correlation on CIFAR-100, Flowers, and Chaoyang.

Search Space	Proxy	CIFAR-100			Flowers			Chaoyang		
		Kendall	Spearman	Pearson	Kendall	Spearman	Pearson	Kendall	Spearman	Pearson
PiT	GraSP	-42.02	-58.71	-31.53	-50.66	-69.64	-40.90	-16.00	-22.94	-19.07
	SynFlow	69.79	87.05	70.80	62.22	79.98	71.66	30.96	42.66	39.24
	TENAS	-2.13	-3.21	-1.68	-2.86	-4.23	-3.33	-3.34	-5.04	-3.55
	NWOT	-2.61	-4.13	-0.52	2.67	3.69	0.71	4.73	6.87	4.43
	TF-TAS	63.83	82.20	58.37	64.48	82.91	67.23	37.99	52.92	42.68
	DisWOT	-19.05	-28.83	-28.56	-19.57	-30.33	-18.01	-3.36	2.40	15.22
	Auto-DAS	**72.81**$_{\pm 1.68}$	**90.32**$_{\pm 1.79}$	**77.76**$_{\pm 2.58}$	**76.42**$_{\pm 1.12}$	**93.09**$_{\pm 1.85}$	**87.63**$_{\pm 1.36}$	**48.36**$_{\pm 2.46}$	**66.13**$_{\pm 1.58}$	**66.58**$_{\pm 1.86}$

Table 8: Comparing the performance of our proposed Auto-DAS approach across three diverse datasets - CIFAR-100, flowers, and Chaoyang - using the PiT search space, we have achieved competitive results while incurring the lowest search cost, measured in time cost on a single GPU.

Search Space	Proxy	CIFAR-100			Flowers			Chaoyang		
		Param(M)	Dis.Acc(%)	Search Cost	Param(M)	Dis.Acc(%)	Search Cost	Param(M)	Dis.Acc(%)	Search Cost
PiT	Random	5.33	75.84	N/A	4.88	65.30	N/A	5.24	82.94	N/A
	GraSP	4.53	76.03	1.24 h	3.72	66.58	1.85 h	4.63	83.87	0.86 h
	SynFlow	11.05	77.13	1.08 h	5.23	68.12	0.99 h	4.93	83.73	0.70 h
	TENAS	6.93	76.09	5.14 h	4.26	68.03	5.14 h	6.76	83.64	5.07 h
	NWOT	5.21	76.64	3.02 h	10.77	67.72	3.09 h	6.37	83.31	3.08 h
	TF-TAS	16.07	77.06	1.21 h	10.30	68.21	0.95 h	4.32	84.34	0.71 h
	DisWOT	10.38	75.82	0.88 h	8.76	67.58	0.88 h	5.85	83.42	0.88 h
	Auto-DAS	12.42	**77.88**	1.28 h	10.52	**69.32**	1.28 h	4.88	**84.83**	1.28 h
	Auto-DAS (G-free)	11.35	77.72	**0.82h**	10.88	69.45	**0.82h**	5.25	84.98	**0.82h**

This benchmark provides ground-truth accuracy for ViTs on compact datasets (CIFAR-100 [22], Flowers [39], and Chaoyang [58]). Inspired by [24], which shows ViTs gain significantly on small datasets when distilled from an efficient CNN teacher, we concentrate on the distillation accuracy of student ViTs on these datasets with a predetermined teacher. We randomly sample ViTs from PiT [16] search spaces and train them individually using the distillation settings from [24]. ResNet56 serves as the CNN teacher. ViT-Bench encompasses 5k KD accuracy ground truths per dataset. We divide the benchmark into validation and test sets at a 4:3 ratio. For the proxy search phase, we configure EA parameters ($\mathcal{P}$, $\mathcal{T}$, r, k) in Alg. 1 as $(20, 100, 0.9, 5)$ and evaluate proxy candidates' ranking performance in the validation set each iteration. Post-search, we equitably compare the Auto-DAS proxy with conventional ones by random sampling in the test set. Our validation and test sets are distinct to ensure unbiased proxy comparisons.

Findings on compact datasets. Table 7 presents the rank consistency results of existing proxies across three datasets. Our proposed Auto-DAS surpasses other notable proxies in metrics like Kendall's tau [1], Spearman's rho [44], and Pearson's correlation coefficient [4], showcasing its efficacy and superiority in achieving consistent and precise ViT architecture search. To further demonstrate Auto-DAS's effectiveness, Table 8 displays search trial results. Our approach achieves superior distillation outcomes and faster search speeds across all three datasets. These empirical findings highlight the importance of effective zero-cost proxies and the role of our proposed Auto-DAS in optimizing ViT architecture search performance.

Table 9: Results of different ViT models on ImageNet.

Model	Method	Param.	Top-1 Acc.	Model	Method	Param.	Top-1 Acc.
DeiT-Tiny	Baseline+KD	5.7M	74.5%	Swin-Tiny	Baseline+KD	29.0M	81.7%
	TF-TAS+KD	5.7M	74.7%		TF-TAS+KD	29.2M	82.0%
	DisWOT	5.8M	75.3%		DisWOT	29.0M	82.2%
	Auto-DAS	**6.1M**	**75.9%**		**Auto-DAS**	**29.2M**	**82.9%**
PiT-Tiny	Baseline+KD	4.9M	73.2%	PiT-Small	Baseline+KD	23.5M	79.9%
	TF-TAS+KD	4.9M	73.5%		TF-TAS+KD	24.2M	80.1%
	DisWOT	4.6M	73.6%		DisWOT	23.8M	80.3%
	Auto-DAS	**4.8M**	**74.2%**		**Auto-DAS**	**24.5M**	**81.0%**

Table 10: Spearman correlation (%) on optimal proxy in our search space with vanilla inputs or operations.

Methods	Settings	ResNet	NB-101	NB-201	NB-101-KD	NB-201-KD	PiT-CIFAR-100	PiT-Flowers	PiT-Chaoyang
Auto-DAS	**Auto-DAS**	**98.44**	**71.84**	**87.73**	**85.95**	**74.66**	**90.32**	**93.09**	**66.13**
Auto-DAS (instinct)	Activations	70.28	65.89	78.96	66.35	60.85	74.86	76.85	52.86
	Gradients	77.53	72.68	86.92	73.55	66.38	80.56	82.75	56.67
	weights	60.35	52.75	70.25	58.67	55.25	60.89	65.82	41.25
	Unary	78.85	74.25	88.25	77.42	66.53	82.83	84.38	58.69
	Binary	62.52	58.87	72.89	64.35	58.96	65.45	70.25	48.46
	All	80.52	75.86	89.62	78.56	68.55	83.25	86.52	60.12
Auto-DAS (interaction)	Feature	90.35	47.96	62.68	76.33	66.38	81.95	85.39	59.68
	Embed	88.65	46.55	60.75	74.85	63.28	80.25	83.27	56.88
	Logits	85.47	42.58	55.86	70.53	58.62	76.83	78.96	52.45
	Transform	92.35	48.88	64.66	78.95	68.65	87.25	76.55	61.35
	Distance	89.63	46.53	63.85	75.82	65.86	82.69	84.56	58.75
	All	92.82	50.88	65.88	80.56	70.48	85.42	88.73	63.38

Computational cost analysis. (1) Like most train-free NAS methods, Auto-DAS involves minor additional gradient computations compared to DisWOT. To address this, we explore a gradient-free version of Auto-DAS in Table 8 (labeled Auto-DAS (G-free)). Results indicate that our gradient-free Auto-DAS achieves better accuracy-efficiency trade-offs than DisWOT. (2) While DisWOT is efficient and yields good results on certain datasets, it lacks generalizability across various models and tasks. In contrast, Auto-DAS is automated, versatile, and effective, better suiting diverse application needs. Moreover, given the low cost of training-free approaches, the small additional time investment in searching is justified by the consistent performance gains achieved.

ImageNet performance analysis. Examining the results in Table 9, it's clear that Auto-DAS outperforms other methods across various models. For DeiT-Tiny, Auto-DAS achieves 75.9% accuracy, surpassing Baseline+KD (74.5%), TF-TAS+KD (74.7%), and DisWOT (75.3%). Similarly, Auto-DAS excels with the Swin-Tiny model. This trend persists for PiT-Tiny and PiT-Small models, where Auto-DAS consistently outperforms other methods. These findings underscore the effectiveness and adaptability of our Auto-DAS method in enhancing accuracy across different ViT architectures.

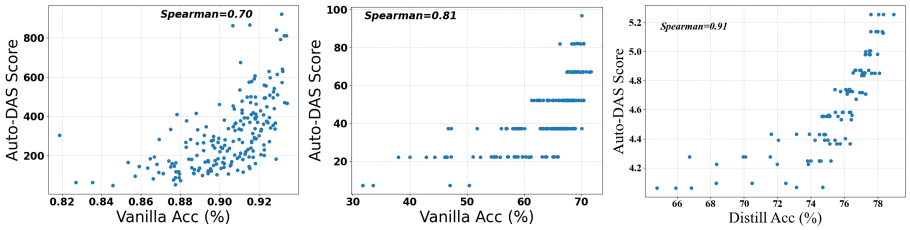

Figure 6: Correlation visualization for Auto-DAS on NAS-Bench-101, NAS-Bench-201 and ViT-Bench *(from left to right)*.

4.4 Ablation Studies

Proxy search space. We conduct experiments in Table 10 to analyze different components in our proxy search space. The results show that Auto-DAS (interaction) outperforms Auto-DAS (instinct) overall. The feature input and transform operations are particularly important for Auto-DAS (interaction), while weight input and binary operations contribute weakly. On the other hand, gradient input and unary operations show promising results for Auto-DAS (instinct). Different ops have varying impacts on the proxies' accuracy, and combining them enhances the performance of Auto-DAS approaches. These findings highlight the significance of considering different ops and their interactions in the search space for optimal performance.
Search algorithm. We use the EA with adaptive-elite selection for proxy search. As shown in Figure 4, the EA adaptive-elite selection obtains better final search results than EA and random search in the proxy search process.
Correlation visualization.. To intuitively observe the proxy's predictability, we visualize the score of Auto-DAS and the ground-truth performance in Figure 6. These visualizations demonstrate that Auto-DAS effectively detects the true distillation results.

5 Conclusion

We present Auto-DAS, a novel training-free framework for distillation-aware student architecture search. It includes proxy search and training-free student search. With discovered proxies, Auto-DAS enables efficient search without training costs. This significantly outperforms handcrafted proxies in distillation performance. Comprehensive results on NAS benchmarks, ViT, and CNN spaces over multiple datasets illustrate our method's superior ranking and distillation abilities. We hope our novel investigations will give more insight and new directions for the knowledge distillation and NAS research communities.
Iimitation. We mainly evaluate the same benchmarks as other NAS methods for fair comparison. We will continue to expand Auto-DAS for more tasks in future work. **Social impact.** Our Auto-DAS focuses on technology improvement without social and ethical implications.

References

1. Abdi, H.: The kendall rank correlation coefficient. Encyclopedia of measurement and statistics **2**, 508–510 (2007)
2. Akhauri, Y., Munoz, J.P., Jain, N., Iyer, R.: EZNAS: Evolving zero-cost proxies for neural architecture scoring. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) NeurIPS (2022), https://openreview.net/forum?id=lSqaDG4dvdt
3. Baker, B., Gupta, O., Naik, N., Raskar, R.: Designing neural network architectures using reinforcement learning. In: ICLR (2017)
4. Bowley, A.: The standard deviation of the correlation coefficient. Journal of the American Statistical Association **23**(161), 31–34 (1928)
5. Brown, T.B., Mann, B., Subbiah, N.R.M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., M, D., Ziegler, Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. arXiv preprint, arXiv:2005.14165 (2020)
6. Dong, P., Li, L., Tang, Z., Liu, X., Pan, X., Wang, Q., Chu, X.: Pruner-zero: Evolving symbolic pruning metric from scratch for large language models. In: ICML (2024)
7. Dong, P., Li, L., Wei, Z.: Diswot: Student architecture search for distillation without training. In: CVPR (2023)
8. Dong, P., Li, L., Wei, Z., Niu, X., Tian, Z., Pan, H.: Emq: Evolving training-free proxies for automated mixed precision quantization. In: ICCV. pp. 17076–17086 (2023)
9. Dong, P., Niu, X., Li, L., Xie, L., Zou, W., Ye, T., Wei, Z., Pan, H.: Prior-guided one-shot neural architecture search. arXiv preprint arXiv:2206.13329 (2022)
10. Dong, X., Yang, Y.: Searching for a robust neural architecture in four gpu hours. CVPR (2019)
11. Dong, X., Yang, Y.: Nas-bench-201: Extending the scope of reproducible neural architecture search. In: ICLR (2019)
12. 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 (2020)
13. Falkner, S., Klein, A., Hutter, F.: Bohb: Robust and efficient hyperparameter optimization at scale. In: ICML (2018)
14. Gu, J., Tresp, V.: Search for better students to learn distilled knowledge. arXiv preprint arXiv:2001.11612 (2020)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
16. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. In: ICCV (2021)
17. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
18. Hu, S., Xie, S., Zheng, H., Liu, C., Shi, J., Liu, X., Lin, D.: Dsnas: Direct neural architecture search without parameter retraining. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
19. Hu, Y., Liang, Y., Guo, Z., Wan, R., Zhang, X., Wei, Y., Gu, Q., Sun, J.: Angle-based search space shrinking for neural architecture search. In: ECCV (2020)

20. Hu, Y., Wang, X., Li, L., Gu, Q.: Improving one-shot nas with shrinking-and-expanding supernet. Pattern Recognition (2021)
21. Kim, Y., Rush, A.M.: Sequence-level knowledge distillation. In: EMNLP (2016)
22. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)
23. Lee, N., Ajanthan, T., Torr, P.: Snip: Single-shot network pruning based on connection sensitivity. In: ICLR (2018)
24. Li, K., Yu, R., Wang, Z., Yuan, L., Song, G., Chen, J.: Locality guidance for improving vision transformers on tiny datasets. In: European Conference on Computer Vision. pp. 110–127. Springer (2022)
25. Li, L., Talwalkar, A.S.: Random search and reproducibility for neural architecture search. ArXiv (2019)
26. Li, L.: Self-regulated feature learning via teacher-free feature distillation. In: ECCV (2022)
27. Li, L., Bao, Y., Dong, P., Yang, C., Li, A., Luo, W., Liu, Q., Xue, W., Guo, Y.: Detkds: Knowledge distillation search for object detectors. In: ICML (2024)
28. Li, L., Dong, P., Li, A., Wei, Z., Yang, Y.: Kd-zero: Evolving knowledge distiller for any teacher-student pairs. NeuIPS (2024)
29. Li, L., Dong, P., Wei, Z., Yang, Y.: Automated knowledge distillation via monte carlo tree search. In: ICCV (2023)
30. Li, L., Jin, Z.: Shadow knowledge distillation: Bridging offline and online knowledge transfer. In: NeuIPS (2022)
31. Li, L., Sun, H., Li, S., Dong, P., Luo, W., Xue, W., Liu, Q., Guo, Y.: Auto-gas: Automated proxy discovery for training-free generative architecture search. In: ECCV (2024)
32. Li, L., Wang, Y., Yao, A., Qian, Y., Zhou, X., He, K.: Explicit connection distillation. In: ICLR (2020)
33. Li, L., Wei, Z., Dong, P., Luo, W., Xue, W., Liu, Q., Guo, Y.: Attnzero: Efficient attention discovery for vision transformers. In: ECCV (2024)
34. Lin, M., Wang, P., Sun, Z., Chen, H., Sun, X., Qian, Q., Li, H., Jin, R.: Zen-nas: A zero-shot nas for high-performance image recognition. ICCV (2021)
35. Liu, H., Simonyan, K., Yang, Y.: DARTS: differentiable architecture search. In: 7th ICLR, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. vol. abs/1806.09055 (2019)
36. Liu, Y., Jia, X., Tan, M., Vemulapalli, R., Zhu, Y., Green, B., Wang, X.: Search to distill: Pearls are everywhere but not the eyes. In: CVPR (2020)
37. Mellor, J., Turner, J., Storkey, A., Crowley, E.J.: Neural architecture search without training. In: ICML (2021)
38. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: AAAI (2020)
39. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing. pp. 722–729. IEEE (2008)
40. Park, W., Lu, Y., Cho, M., Kim, D.: Relational knowledge distillation. In: CVPR (2019)
41. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. In: ICML. pp. 4092–4101 (2018)
42. Real, E., Liang, C., So, D.R., Le, Q.V.: Automl-zero: Evolving machine learning algorithms from scratch (2020)
43. Shao, S., Dai, X., Yin, S., Li, L., Chen, H., Hu, Y.: Catch-up distillation: You only need to train once for accelerating sampling. arXiv preprint arXiv:2305.10769 (2023)

44. Stephanou, M., Varughese, M.: Sequential estimation of spearman rank correlation using hermite series estimators. Journal of Multivariate Analysis **186**, 104783 (2021)
45. Tanaka, H., Kunin, D., Yamins, D.L., Ganguli, S.: Pruning neural networks without any data by iteratively conserving synaptic flow. NeurIPS (2020)
46. Theis, L., Korshunova, I., Tejani, A., Huszár, F.: Faster gaze prediction with dense networks and fisher pruning. ArXiv **abs/1801.05787** (2018)
47. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: ICLR (2020)
48. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: ICLR (2020)
49. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers amp; distillation through attention. In: ICML (2021)
50. Wang, C., Zhang, G., Grosse, R.: Picking winning tickets before training by preserving gradient flow. arXiv preprint arXiv:2002.07376 (2020)
51. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: CVPR (2019)
52. Xiaolong, L., Lujun, L., Chao, L., Yao, A.: Norm: Knowledge distillation via n-to-one representation matching (2022)
53. Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K., Hutter, F.: Nas-bench-101: Towards reproducible neural architecture search. In: ICML (2019)
54. You, S., Huang, T., Yang, M., Wang, F., Qian, C., Zhang, C.: Greedynas: Towards fast one-shot nas with greedy supernet. In: CVPR (2020)
55. Zhang, L., Ma, K.: Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In: ICLR (2020)
56. Zhou, H., Song, L., Chen, J., Zhou, Y., Wang, G., Yuan, J., Zhang, Q.: Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective (2021)
57. Zhu, C., Li, L., Wu, Y., Sun, Z.: Saswot: Real-time semantic segmentation architecture search without training. In: AAAI (2024)
58. Zhu, C., Chen, W., Peng, T., Wang, Y., Jin, M.: Hard sample aware noise robust learning for histopathology image classification. TMI (2021)
59. Zimian Wei, Z., Li, L.L., Dong, P., Hui, Z., Li, A., Lu, M., Pan, H., Li, D.: Autoprox: Training-free vision transformer architecture search via automatic proxy discovery. In: AAAI (2024)
60. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: CVPR (2018)