# AttnZero: Efficient Attention Discovery for Vision Transformers

Lujun Li[1], Zimian Wei[2✉], Peijie Dong[3], Wenhan Luo[1], Wei Xue[1], Qifeng Liu[1✉], and Yike Guo[1✉]

[1] The Hong Kong University of Science and Technology
[2] National University of Defense Technology
[3] The Hong Kong University of Science and Technology (Guangzhou)
lilujunai@gmail.com;weizimian16@nudt.edu.cn;pdong212@connect.hkust-gz.edu.cn;
whluo.china@gmail.com;weixue@ust.hk;liuqifeng@ust.hk;yikeguo@ust.hk

**Abstract** In this paper, we present AttnZero, the first framework for automatically discovering efficient attention modules tailored for Vision Transformers (ViTs). While traditional self-attention in ViTs suffers from quadratic computation complexity, linear attention offers a more efficient alternative with linear complexity approximation. However, existing hand-crafted linear attention suffers from performance degradation. To address these issues, our AttnZero constructs search spaces and employs evolutionary algorithms to discover potential linear attention formulations. Specifically, our search space consists of six kinds of computation graphs and advanced activation, normalize, and binary operators. To enhance generality, we derive results of candidate attention applied to multiple advanced ViTs as the multi-objective for the evolutionary search. To expedite the search process, we utilize program checking and rejection protocols to filter out unpromising candidates swiftly. Additionally, we develop Attn-Bench-101, which provides precomputed performance of 2,000 attentions in the search spaces, enabling us to summarize attention design insights. Experimental results demonstrate that the discovered AttnZero module generalizes well to different tasks and consistently achieves improved performance across various ViTs. For instance, the tiny model of DeiT|PVT|Swin|CSwin trained with AttnZero on ImageNet reaches 74.9%|78.1%|82.1%|82.9% top-1 accuracy. Codes at: https://github.com/lliai/AttnZero.

**Keywords:** Vision Transformer · Efficient Attention · AutoML

## 1 Introduction

In recent years, Vision Transformers (ViTs) demonstrate remarkable success in various vision tasks, including image classification [16, 19, 58, 62], object detection [60, 75], semantic segmentation [47]. ViTs draw inspiration from the transformative impact of Transformers in natural language processing, utilizing self-attention mechanisms to capture global dependencies. However, self-attention

✉ Corresponding authors.

has a quadratic complexity that scales with the square of the input sequence length. This quadratic complexity limits the scalability of ViTs for handling large image inputs, thereby limiting their practical applications.

To address this issue, researchers have been exploring the use of linear attention [57] as a more computationally efficient alternative. Unlike self-attention, linear attention has a linear complexity approximation, making it more suitable for handling large input sequences. However, existing linear attention methods [42, 54] suffer from several limitations: (1) Linear attention often leads to reduced performance than softmax attention due to its simplifying assumptions. The simple linear operators used in linear attention have limited flexibility and produce noisier activations than nonlinear functions [24]. These performance degradations challenge leveraging linear attention as a viable substitute for self-attention in ViTs. (2) Hand-crafted linear attention designs are often specific to certain tasks or architectures, lacking generality across different scenarios. Additionally, the manual design is labor-intensive, time-consuming, and heavily reliant on expert knowledge. Consequently, these drawbacks hinder the scalability and accessibility of linear attention in ViTs.

To address these challenges, we have made several observations. We notice that some large language models [4, 7] employ new operators in Transformers to improve performance. For example, LLaMA [59] currently uses RMSNorm and SiLU to prevent training crashes. These successes inspire us to assemble a series of advanced operators to enhance linear attention and explore their optimal combination formulations. However, the question remains: How can we efficiently solve this combination problem? Recent developments in AutoML, such as AutoML-Zero [50] and NAS methods [40, 46], offer a promising solution. These methods involve constructing an extensive search space encompassing various options and searching for optimal architectures. Nevertheless, existing NAS methods are not suitable for solving the formulation discovery problem in our case for several reasons. **(1) Different search space.** The search spaces in conventional NAS methods typically focus on network width/depth, block/cell types, and parametric operators [5, 56]. In contrast, our attention search space includes many non-parametric operators and different combinatorial types. This leads to an explosion in the search space and many unsuccessful candidates. **(2) Different search algorithms.** Most recent NAS methods employ weight-sharing strategies [14, 27] to reduce search costs. However, our search space is explosive and extremely sparse, meaning that using previous ways to find potential candidates is less efficient and effective. **(3) Different search goals.** Most NAS methods prioritize performance for a single task and may struggle to generalize well to different applications [65]. In contrast, our objective is to discover generalized attention mechanisms that can be applied to various ViT models and tasks. These challenges highlight that attention discovery is a new task that requires a unique search space and strategy compared to traditional NAS methods.

Based on the above observations, we propose AttnZero, a new framework that aims to discover superior linear attention formulations that achieve a better

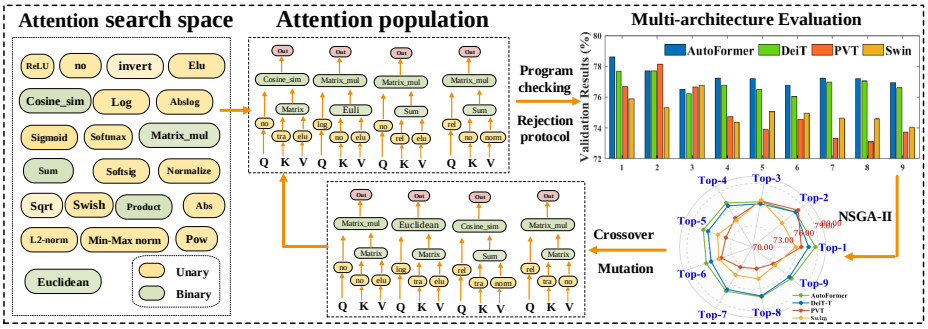


**Figure 1:** Overall pipeline of AttnZero. Our framework obtains candidate attention populations by sampling from attention search space. With program checking and rejection protocols, we conduct validation results of these attentions on multiple ViTs. Then, we use NSGA-II to obtain the optimal solution and cross-mutate for the next population.

accuracy-efficiency tradeoff. As shown in Figure 1, our framework can be divided into two parts: a comprehensive attention search space and a multi-objective evolutionary search. Based on the basic form of linear attention, we develop six computation graphs to represent attention candidates in the search space. Then, we combine unary and binary operators with advanced functions, including various activation functions, normalization techniques, and matrix arithmetic functions. This approach allows us to obtain flexible and effective attention candidates. Our goal for the search algorithm is to find attention formulations applicable across different ViT models. To achieve this, we model the search task as a multi-objective optimization problem and employ the NSGA-II evolution method to obtain optimal solutions. During the search process, we validate candidate attention in multiple ViT models through multi-architecture evaluations. Based on these evaluations, our search engine exploits crossover and mutation for better populations. To accelerate the search process, we perform a fast program check and a rejection protocol to eliminate unpromising candidates during the search process. In this way, our AttnZero improves at least 5× in search efficiency without the weight-sharing setting [22], making our method free of ranking issues in traditional NAS methods [9, 18].

We conduct comprehensive experiments to evaluate the transferability of our discovered attention mechanisms across different ViT models, datasets, and tasks without requiring additional searches. When applied to various ViT models, our AttnZero consistently outperforms the baseline method. For instance, our method achieves gains of 3.0%∼1.6% on PVT-T to PVT-L on the ImageNet dataset. Furthermore, ViT models with AttnZero demonstrate significant improvements when considering different data domains. For example, DeiT+AttnZero achieves gains of 6.2%, 7.8%, 1.1%, and 2.7% on the Tiny-ImageNet, Flowers, Chaoyang, and ImageNet datasets, respectively. Additionally, we compile a new benchmark named Attn-Bench-101, which consists of 2,000 attentions from our

search space. The benchmark highlights the preference for advanced normalization techniques, activation functions, and well-performing attentions. In conclusion, our contributions can be summarized as follows:

- To improve linear attention, we present a new task, efficient attention discovery, that generalizes to various ViT models and tasks. For this task, our AttnZero, the first framework, is developed to achieve effective attention search.
- We propose a comprehensive attention search space, including different types of computation graphs and advanced operators as options. We develop multi-objective searches with multiple ViTs evaluations to ensure generalisability.
- Our found attention can be well generalized to DeiT|PVT|Swin|CSwin and other ViTs. Extensive experiments on multiple datasets in classification, detection, and segmentation prove the superior performance of our AttnZero.
- We build Attn-Bench-101, providing many attentions and their results. Some attention design guidances are summarised based on these results.

## 2 Related Works

**Efficient Attention.** The application of self-attention in computer vision has made significant progress [2,8,26,68,69] with the introduction of ViT [19]. Then, PVT [60] introduces a hierarchical structure while Swin Transformer [41] proposes window-based attention to model dependencies efficiently. However, softmax attention's quadratic complexity poses challenges. Linear attention reduces this but often degrades performance [6, 30, 64]. Efficient Attention [54] applies softmax to Q and K while SOFT [42] uses matrix decomposition. Castling-ViT [66] employs softmax attention during training but linear attention at inference. FLatten [24] preserves diversity through focused functions. While effective, these methods still have performance limitations and require complex manual design. In contrast, our AttnZero automatically discovers novel formulations to enhance linear attention. Rather than replacing prior work, we aim to improve the performance of the existing method while reducing manual effort. We build our search space based on previous work and provide feedback on new patterns. Thus, AttnZero complements rather than competes with manual methods.
**Neural Architecture Search (NAS).** NAS methods automate model design through search spaces and algorithms [49,77]. Early NAS works adopt reinforcement learning with expensive costs [78]. Recent one-shot NAS [15] and zero-shot NAS [11–13,36,37,73,76] use techniques like weight sharing [23] and training-free proxy evaluation to speed up search. Current NAS methods on ViTs [5, 45, 72] (*e.g.*, AutoFormer [5]) search the number of depth/head of transformers. In contrast, AttnZero focuses on discovering attention mechanisms, not architectural dimensions. We also apply AttnZero to AutoFormer and observe consistent improvements, indicating our method is orthogonal to existing ViT-NAS methods. There are attempts to search entire attention blocks [67], SE-like attention [43,61], projection layers [21], and Transformers searches in other fields like

CTR [70] and NLP [20,55]. We clarify that our AttnZero is the first to search for efficient attention in visual tasks, which clearly differs in spaces/strategies/goals from previous methods. Inspired by general idea in AutoML-Zero [51], we achieve many new designs in search space/strategies for new attention discovery task, and opens up new possibilities for enhancing the capabilities of ViTs.

## 3 Methodology: Efficient Attention Discovery

In this section, we first review the linear attention form. Then, we present our search space and algorithms. The overall process is shown in Figure 1.

### 3.1 Recap of Self-Attention and Linear Attention

In self-attention, input tokens are projected into query ($Q$), key ($K$), and value ($V$) matrices, respectively. The general form of self-attention is given by:

$$O_i = \sum_{j=1}^{N} \frac{\mathrm{Sim}(Q_i, K_j)}{\sum_{j=1}^{N} \mathrm{Sim}(Q_i, K_j)} V_j, \tag{1}$$

where $O_i$ represents the output of the $i$-th token and Sim denotes the similarity function. The commonly used similarity function is the softmax function, $\exp\left(QK^T/\sqrt{d}\right)$, where $d$ is the dimension of the input tokens. However, the quadratic time and space complexity of softmax attention, $\mathcal{O}(N^2 d)$, becomes impractical for large numbers of tokens ($N$). To address this, linear attention is explored as an efficient alternative, which can be expressed as

$$O_i = \frac{\phi(Q_i)\left(\sum_{j=1}^{N} \phi(K_j)^T V_j\right)}{\phi(Q_i)\left(\sum_{j=1}^{N} \phi(K_j)^T\right)}, \tag{2}$$

where $\phi$ represents the kernel function. Linear attention reduces the computational complexity to $\mathcal{O}(N)$, making it scalable for large input sequences. This reduction is achieved by using carefully designed kernels to approximate the similarity function, enabling a change in the computation order from $(QK^T)V$ to $Q(K^TV)$, exploiting the associative property of matrix multiplication.

### 3.2 Efficient Attention Search Space

**Attention Representation.** In Figure 2, we construct six types of computational graphs using (Q, K, V) as inputs to represent our candidate attentions. Among them, our i3n2 attentions follow the same process as standard linear attention, and we improve it by searching for different unary operations. Partially inspired by Castling-ViT [66], we introduce novel attention mechanisms with three binary operator nodes based on basic linear attention. With additional operators with either $\mathcal{O}(N \times d)$ or $\mathcal{O}(d \times d)$ computational complexity, the

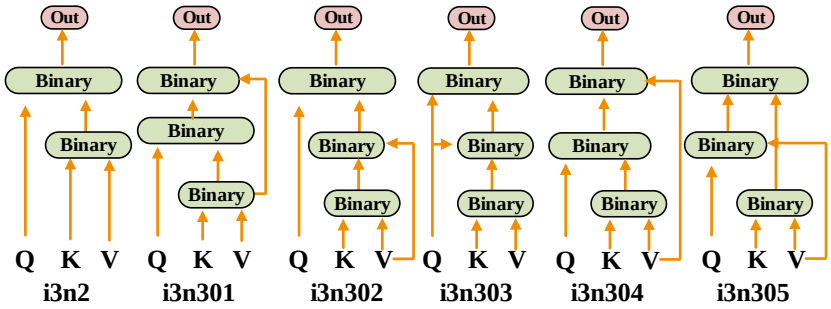


**Figure 2:** Schematic of the six computational graphs in our search space. We denote the graph with 3 inputs and 2 binary operations as i3n2. Similarly, i3n3 refers to a graph with 3 inputs and 3 binary operations. Since each binary operation takes two inputs, there are 5 possible graph structures for i3n3. We label these i3n301∼i3n305, respectively. For simplicity, we do not display unary operators here.

**Table 1:** Some operations in our attention search space. "UOP" and "BOP" denote the unary and binary operation, respectively. Full operations are in the Appendix.

| OP ID | OP Name | Expression |
|---|---|---|
| UOP00 | no_op | $x$ |
| UOP01 | relu | $\max(0, x)$ |
| UOP02 | scale | $\frac{x}{\sqrt{d_x}}$ |
| UOP03 | sqrt | $\sqrt{x}$ |
| UOP04 | invert | $\frac{1}{x+e^{-6}}$ |
| UOP05 | l2_norm | $\frac{x-\text{mean}(x)}{\text{std}(x)}$ |
| UOP06 | sigmoid | $\frac{1}{1+e^{-x}}$ |
| UOP07 | logsoftmax | $\ln \frac{e^x}{\sum_{i=1}^{n} e^{s_i}}$ |
| UOP08 | softmax | $\frac{e^x}{\sum_{i=1}^{n} e^{s_i}}$ |
| UOP09 | softsign | $\frac{x}{1+\lvert x\rvert}$ |

| OP ID | OP Name | Expression |
|---|---|---|
| UOP10 | elu | $\text{elu}(x) + 1$ |
| UOP11 | sigmoid_revert | $1 - \text{sigmoid}(x)$ |
| UOP12 | exp | $e^x$ |
| UOP13 | abslog | $\lvert\ln x\rvert$ |
| UOP14 | min_max_norm | $\frac{x-\min(x)}{\max(x)-\min(x)}$ |
| UOP15 | transpose | $x^T$ |
| UOP16 | swish | $x \times \text{sigmoid}\,(x)$ |
| BOP01 | sum | $x + y$ |
| BOP02 | cosine_sim | $cos(x, y)$ |
| BOP03 | product | $x \odot y$ |
| BOP04 | matrix_multi | $x \cdot y$ |
| BOP05 | euclidean_dis | $d(x, y)$ |

performance of these attention mechanisms can be enhanced without a significant increase in computational overhead. Based on the different binary nodes, these augmented attentions can be categorized into five types (i3n301 to i3n305), effectively enhancing the modeling capability. We divide the sub-search space according to these different types of attention. During the search process, we first search the solutions within the sub-search space in parallel and then combine them to obtain the final search result. Figure 4 (b) shows performance for different attention candidates. These designs allow our method to balance computational complexity and accuracy.

**Primitive Operations.** Table 1 presents some unary and binary operators in our search space. These operators encompass both commonly used operations found in existing linear attentions [3, 24, 54, 66] and our proposed ones, including: (1) The activation operations are responsible for enhancing the non-

**Algorithm 1** NSGA-II Evolution Search for AttnZero.

**Input**: Search space $\mathcal{A}$, population size $N$, crossover probability $p_c$, mutation probability $p_m$, max generations $G$.
**Output**: Set of Pareto-optimal attention formulations.

Initialize population $\mathcal{P}_0$ with $N$ individuals generated randomly;
Evaluate the fitness of each individual in $\mathcal{P}_0$ for all objectives;
**for** $g = 1$ to $G$ **do**
  Perform non-dominated sorting on $\mathcal{P}_{g-1}$ to classify into fronts $F_1, F_2, \ldots$;
  Calculate crowding distance for individuals in each front;
  $\mathcal{Q}$ := Select $N$ individuals for mating via front rank and crowding distance;
  Generate offspring $\mathcal{Q}'$ via crossover & mutation on $\mathcal{Q}$ with prob. $p_c$ and $p_m$;
  Evaluate the fitness of each offspring in $\mathcal{Q}'$ for all objectives;
  Apply rejection protocols to $\mathcal{Q}'$ to remove unpromising candidate and get $\mathcal{Q}''$
  $\mathcal{R}$ := Combine $\mathcal{P}_{g-1}$ and $\mathcal{Q}''$;
  Perform non-dominated sorting on $\mathcal{R}$;
  Update $\mathcal{P}_g$ by selecting top $N$ individuals via front rank and crowding distance;
**end for**
$\mathcal{P}^*$ := Extract the first front from $\mathcal{P}_G$ as the set of Pareto-optimal solutions;

linearity of attention. We include *relu* in FLatten [24] and introduce new activation functions such as *swish* and *elu*. (2) Normalization operations ensure a smoother distribution of attention scores. We provide options such as *l2_ norm* and *min_ max_ norm*. (3) The binary operators facilitate information interaction. Alongside *cosine_ similarity* in Hydra [3], we also consider *euclidean_ distance* as a simplified calculation option. As shown in Figure 4 (c), these operators enhance the versatility and adaptability of attention expressions, resulting in improved performance. Some complex operations such as decomposition [64] and Gaussian kernel [6] would result in a highly sparse search space or need hardware-specific supports. Hence they have not been included in our investigation. In future work, we will continue to involve more promising operators in our search space.

### 3.3 Multi-objective Evolutionary Search

After building the search space $\mathcal{A}$, we sample the attention candidates $a \in$ and apply it to the ViT model with weights $W$. For single-object search, the aim is to find the optimal attention $a^*_{single}$ by evaluating candidates using the trained ViT model ($W^*$):

$$a^*_{single} = \underset{a \in \mathcal{A}}{\text{argmax}} \text{ACC}_{\text{val}} \left( \mathcal{N} \left( a, W^*_{\mathcal{A}}(a) \right) \right) \tag{3}$$

However, $a^*_{single}$ may perform well on the search model but fail to other ViT models. For example, attention searched in Swin, while superior on Swin, performs poorly on DeiT, PVT in Figure 5 (b). Such attentions suffer from limited usefulness because they are difficult to transfer directly to different tasks.

**Multi-objective Evolution.** To improve generalization of attention candidates in different ViT models, our AttnZero framework employs a multi-objective evolutionary algorithm. This algorithm aims to maximize the results of multiple ViT models $_1$, $_2$, ..., $_n$ under predefined computational complexity constraints. We define a fitness function as follows:

$$a^*_{multi} = \underset{a \in \mathcal{A}}{\text{argmax}} \text{ACC}_{\text{val}} \left( _1 \left( a, W^*_1 {}_{\mathcal{A}}(a) \right), _2 \left( a, W^*_2 {}_{\mathcal{A}}(a) \right), ..., _n \left( a, W^*_n {}_{\mathcal{A}}(a) \right) \right). \quad (4)$$

Our framework incorporates validation results of attention in DeiT, AutoFormer, PVT, and Swin as objectives. This ensures that our attention module performs effectively in various ViT models, including pure ViT, ViT models with NAS search, ViT models with hierarchical design, and ViT models with local window configurations, respectively. Solving this multi-objective problem is challenging due to obstacles like conflicting optimals. To address this, we utilize NSGA-II method [10] in Alg. 1 as our search engine. Our search begins with an initial population of candidate attention modules and each candidate's fitness is evaluated based on performance across ViT models. Non-dominated sorting ranks the candidates and assigns them to fronts based on how they compare to others. Then. we use crowding distance [48] calculations to maintain diversity within the population. The selection process chooses candidates for the next generation, considering sorting and distance. Genetic operations like crossover and mutation are applied to the selected candidates. This introduces diversity and explores new areas of the search space. The least fit candidates are replaced with offspring, ensuring continuous gains. The algorithm iterates through these steps until it meets a termination criterion. During the search process, our algorithm takes FLOPs as the complexity constraint. Finally, we consider the real speed results in selecting final found attention from the Pareto-optimal solutions.

**Search Acceleration Strategies.** To improve the search efficiency, we employ several strategies for search acceleration: (1) Program checking. We perform program checking by verifying if the single candidate attention module can successfully generate outputs using random tensor inputs. This helps us identify and filter out candidates with arithmetic errors or dimension mismatches. (2) Rejection protocols. During training each population, we reject candidates with collapsed optimisation (*e.g.*, loss = nan) and historical average validation results falling far below baseline linear attention. These strategies allow more than 5× search acceleration.

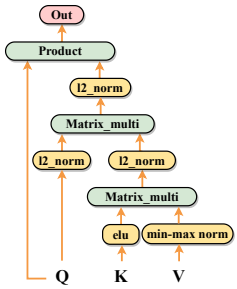


**Figure 3:** Discovered attention $a^*_{AttnZero}$.

### 3.4 Discovered AttnZero Analysis

**Formulas of discovered AttnZero.** Figure 3 illustrates the optimal attention $a^*_{AttnZero}$ in our search, whose detailed expressions are as follows:

$$a^*_{AttnZero} = Q \ \times \eta_2 \left( \eta_2 \left( Q \right) \eta_2 \left( \phi_1 \left( K \right)^T \eta_1 \left( V \right) \right) \right) \quad (5)$$

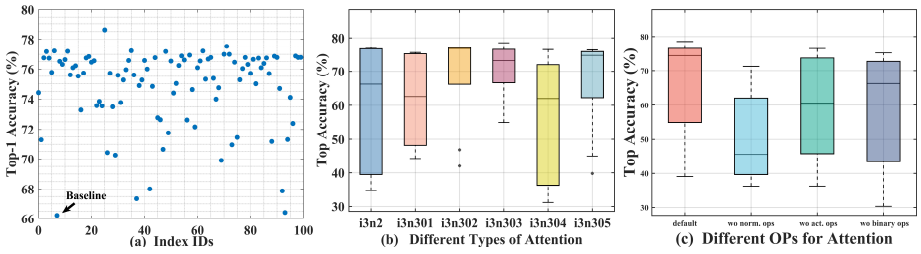


**Figure 4:** (a) Performance distribution of subsets of Attn-Bench-101 (AutoFormer). (b) Results of various types of Attention. (c) Results of our default search space, default search space without advanced normalization operations, activation operations, and binary operations that we have newly introduced. More details are in the Appendix.

where $\phi_1$ utilizes the Exponential Linear Unit (ELU) with a shift of 1, introducing nonlinearities into the attention computation. The $\eta_1$ performs min-max normalization, scaling the matrix values to a range between 0 and 1. This normalization step ensures that the attention weights are properly distributed and avoids potential issues with extreme values. Furthermore, the l2 normalization function $\eta_2$ guarantees that the attention weights have a unit norm, facilitating better interpretation and utilization of the attention mechanism. Regarding computational complexity, AttnZero requires triple matrix multiplications (*i.e.*, $\mathcal{O}(d \times d + N \times d + N \times d)$. Despite this, the overall time and memory complexity of AttnZero remains $\mathcal{O}(N)$, which ensures AttnZero to handle computations for sequences of varying lengths efficiently.

**Analysis of Attn-Bench-101.** Our Attn-Bench-101 collects 2,000 attention configurations and their training results in DeiT/AutoFormer/PVT/Swin on CIFAR-100 (see Appendix for details). We analyze some results on AutoFormer to understand better our search space: (1) In Figure 4 (a), we observe that our search space contains many candidates with stronger performance than baseline model. (2) Among various computed graphs, we identify i3n302 and i3n303, superior to the other types (see in Figure 4 (b)). This is attributed to their augmented topology allowing for a good tradeoff between performance and efficiency. (3) For ablations in Figure 4 (c), advanced activation and normalization operations offer more contributions compared to new binary operators.

**Guidance for Efficient Attention Design.** Based on the above analysis, we can summarize that we prefer attention types of i3n302 & i3n303 and combinations of advanced normalization and activation operations in the detailed design.

## 4 Search Experiments

### 4.1 Search Implementation and Results

**Implementation.** Our attention search is performed on CIFAR-100, which are scaled to 224×224 resolution. We search using the validation results to ensure

**Table 2:** Top-1 accuracy (%) of hand-crafted attentions (*e.g.*, Linear [28] and FLatten [24]) and attention with Random NAS method in our search space on CIFAR-100.

| Models | Baseline | Linear [28] | FLatten [24] | Random (NAS) | **AttnZero (NAS)** |
|---|---|---|---|---|---|
| DeiT-T | 65.08 | 57.80 | 72.45 | 76.04 | **77.68** (+12.60) |
| AutoFormer | 66.58 | 64.58 | 75.42 | 76.94 | **78.61** (+12.03) |
| PVT-T | 67.42 | 66.58 | 74.83 | 73.12 | **76.68** (+9.26) |
| Swin-T | 68.25 | 69.45 | 74.31 | 74.58 | **75.90** (+7.65) |

fair comparisons, and our validation set does not overlap with the test set. For search settings, we configure ($N$, $p_c$, $p_m$, $G$) in Alg. 1 as $(20, 0.9, 0.1, 100)$ for parallel search on DeiT|AutoFormer|PVT|Swin. To obtain validation results, we adopt standard training settings [31] (*e.g.*, 300 epochs and AdamW optimizer). With our search acceleration strategies, the overall search only costs 1.05 days on the 8×V100. More details are available in the Appendix.

**Comparison Results with Hand-Crafted Attentions.** The results in Table 2 demonstrate the effectiveness of our AttnZero in consistently achieving stable boosts across different models. Firstly, when compared to the baseline, our AttnZero achieves significant improvements. For example, the DeiT-T model with our AttnZero improves from a baseline accuracy of 65.08% to 77.68%. Similar improvements are observed for the AutoFormer, PVT-T, and Swin-T models, indicating the consistent effectiveness of our approach. Furthermore, our AttnZero shows improvements over the recent state-of-the-art method. For example, the gain over FLatten for the DeiT-T model is 5.2%. These findings emphasize the potential and significance of our AttnZero approach in improving the performance of various models.

**Comparing Random NAS Method.** It is not easy to directly compare our method to other NAS methods because we use different search spaces and do not have weight sharing. To analyze our search strategy, we compare it to random search in our same search space. The results in Table 2 show that our AttnZero consistently improves model performance compared to random search. We achieve increases in accuracy ranging from 1.32% to 3.56% depending on the model. This demonstrates that our search strategy is more effective than a random search.

### 4.2 Ablation Study

**Multi-Objective Search (MOS) vs. Single-Objective Search (SOS).** We employ MOS to enable the discovery of attentions that exhibit strong generalization capabilities. Figure 5 (a) illustrates the optimal candidate resulting from our MOS process. The results demonstrate that the Top-3 candidates outperform the others consistently across various ViT models. In Figures 5 (b) and (c), we compare the generalization performance of MOS and SOS across different ViT models and datasets, respectively. The results indicate that while the SOS

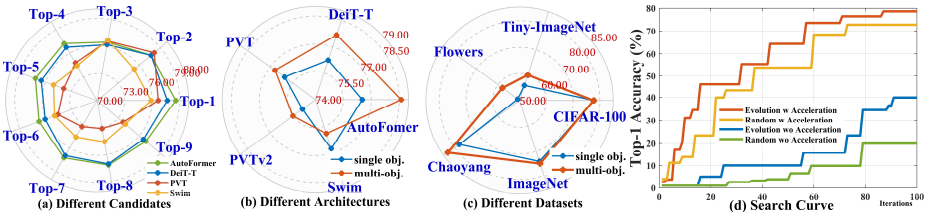


**Figure 5:** (a) Candidates in multi-objective search. (b) Comparisons of multi-objective vs. single-objective on multiple architectures; (c) Comparisons of multi-objective vs. single-objective on multiple datasets. (d) Search curves of various settings.

**Table 3:** Results of AttnZero for different ViT models and datasets.

| Model | Top-1 Acc. (%) | | |
|---|---|---|---|
| | Tiny ImageNet [1] | Flowers [44] | Chaoyang [74] |
| DeiT-T | 53.62 | 50.06 | 82.00 |
| **DeiT-T+AttnZero** | **59.91** (+6.29) | **57.89** (+7.83) | **83.12** (+1.12) |
| AutoFormer-T | 56.38 | 54.98 | 82.84 |
| **AutoFormer-T+AttnZero** | **62.56** (+6.18) | **61.58** (+6.60) | **83.68** (+0.84) |
| PVT-T | 58.47 | 58.57 | 82.46 |
| **PVT-T+AttnZero** | **63.10** (+4.63) | **64.30** (+5.73) | **84.57** (+2.11) |
| Swin-T | 60.22 | 58.85 | 82.98 |
| **Swin-T+AttnZero** | **64.55** (+4.33) | **65.13** (+6.28) | **85.11** (+2.13) |

method may yield better results on the searched models, MOS consistently outperforms SOS over a wider range of ViT architectures. Similarly, our AttnZero approach with MOS consistently demonstrates performance benefits across different dataset domains. For instance, our AttnZero with MOS achieves a 0.9% improvement over SOS on the ImageNet dataset.

**Ablation of the Search Algorithm.** In Figure 5 (d), we examine the evolutionary search and search acceleration strategies. The search results reveal the effectiveness of our acceleration strategy (*i.e.*, program checking and rejection protocols) in filtering out unpromising candidates and greatly expediting the algorithm's convergence. These search curves also indicate that the evolutionary algorithm outperforms random search regarding efficiency in finding high-quality solutions within our framework.

## 5 Transfer Experiments

### 5.1 Experimental Results on Tiny Datasets

**Implementation.** To check if our discovered attention generalizes beyond the CIFAR-100 dataset that we searched on, we it to multiple datasets (*i.e.*, Tiny ImageNet [1], Flowers [44], and Chaoyang [74]). We train DeiT|AutoFormer|PVT|Swin

**Table 4:** Results of AttnZero for DeiT & PVT on ImageNet (224×224).

| Method | Params | FLOPs | Top-1 (%) |
|---|---|---|---|
| DeiT-T [58] | 5.7M | 1.2G | 72.2 |
| **DeiT-T+AttnZero** | 5.7M | 1.0G | **74.9** (+2.7) |
| PVT-T [60] | 13.2M | 1.9G | 75.1 |
| **PVT-T+AttnZero** | 10.9M | 1.8G | **78.1** (+3.0) |
| PVT-S | 24.5M | 3.8G | 79.8 |
| **PVT-S+AttnZero** | 19.8M | 3.6G | **81.4** (+1.6) |
| PVT-M | 44.2M | 6.7G | 81.2 |
| **PVT-M+AttnZero** | 34.6M | 6.6G | **83.0** (+1.8) |
| PVT-L | 61.4M | 9.8G | 81.7 |
| **PVT-L+AttnZero** | 47.1M | 9.7G | **83.3** (+1.6) |

**Table 5:** Results of AttnZero on ImageNet with 224×224 resolution inputs.

| Method | Params | FLOPs | Top-1 (%) |
|---|---|---|---|
| AutoFormer-T [58] | 5.7M | 1.3G | 74.7 |
| **AutoF.-T+AttnZero** | 5.7M | 1.1G | **75.2** (+0.5) |
| Swin-T [41] | 29.0M | 4.5G | 81.3 |
| **Swin-T+AttnZero** | 28.2M | 4.3G | **82.1** (+0.8) |
| Swin-S | 50.0M | 8.7G | 83.0 |
| **Swin-S+AttnZero** | 49.5M | 8.5G | **83.2** (+0.2) |
| Swin-B | 88.0M | 15.4G | 83.5 |
| **Swin-B+AttnZero** | 89.0M | 15.3G | **83.7** (+0.2) |
| CSwin-T [17] | 23.0M | 4.3G | 82.7 |
| **CSwin-T+AttnZero** | 20.3M | 4.0G | **82.9** (+0.2) |

**Table 6:** Comparison of different efficient attention methods on ImageNet.

| Method | Reso | Params | FLOPs | Top-1 (%) |
|---|---|---|---|---|
| DeiT-T [58] | $224^2$ | 5.7M | 1.2G | 72.2 |
| DeiT-T+Hydra Attn [3] | $224^2$ | 5.7M | 1.1G | 68.3 |
| DeiT-T+Efficient Attn [54] | $224^2$ | 5.7M | 1.1G | 70.2 |
| DeiT-T+Linear Angular [66] | $224^2$ | 5.7M | 1.1G | 70.8 |
| DeiT-T+FLatten | $224^2$ | 6.1M | 1.1G | 74.1 |
| **DeiT-T+AttnZero** | $224^2$ | 5.7M | 1.0G | **74.9** (+2.7) |

using discovered attention from scratch. The training process follows standard settings [31], including 300 training epochs, a cosine learning rate scheduler, and the AdamW optimizer. Detailed settings are shown in the Appendix.

**Results on Various Datasets.** Table 3 highlights the effectiveness of AttnZero across various ViT models and datasets. It consistently surpasses the baselines, resulting in a 6.29% increase in accuracy for the DeiT-T model on Tiny-ImageNet. Similarly, AutoFormer and PVT-T models with AttnZero exhibit similar improvements. On the Flowers dataset, AttnZero achieves an accuracy of 57.89% for the DeiT-T model, surpassing the baseline by 7.83%. AutoFormer and PVT-T models also benefit from AttnZero. The results on the Chaoyang dataset further exemplify AttnZero's capability to enhance accuracies. On the DeiT-T model, AttnZero achieves an accuracy of 83.12%, compared to the 82% baseline. Moreover, the attention modules from AttnZero contribute to the improved performance of AutoFormer, PVT-T, and Swin-T models compared to their original versions. These findings demonstrate the promising ability of AttnZero across different datasets and models.

### 5.2 Experimental Results on Large-scale Datasets

**Implementation.** We conduct the experiment on the ImageNet [52] dataset, a widely used large-scale dataset consisting of 1000 classes and 1.2 million training images. Note that we do not perform additional searches or modify the exist-

**Table 7:** Results for Mask R-CNN on COCO dataset. We compute the FLOPs across the backbone, FPN, and detection head using an input resolution of 1280×800.

| Method | FLOPs | Sch. | $AP^b$ | $AP^b_{50}$ | $AP^b_{75}$ | $AP^m$ | $AP^m_{50}$ | $AP^m_{75}$ |
|---|---|---|---|---|---|---|---|---|
| PVT-T | 240G | 1x | 36.7 | 59.2 | 39.3 | 35.1 | 56.7 | 37.3 |
| PVT-T+FLatten | 244G | 1x | 38.2 | 61.6 | 41.9 | 37.0 | 57.6 | 39.0 |
| **PVT-T+AttnZero** | 209G | 1x | **40.0** | **62.6** | **43.1** | **37.4** | **59.4** | **40.1** |
| PVT-S | 305G | 1x | 40.4 | 62.9 | 43.8 | 37.8 | 60.1 | 40.3 |
| **PVT-S+AttnZero** | 248G | 1x | **41.0** | **63.4** | **45.0** | **38.2** | **60.4** | **41.1** |

ing classical models. Instead, we directly replace the self-attention mechanism with our discovered attention module. Furthermore, we ensure that our training settings remain consistent with the standard settings [17, 41, 58]. This includes training the models for 300 epochs, cosine learning rate scheduler, and AdamW optimizer. Specific training settings are shown in the Appendix.

**Results on Various Backbones.** Table 4 and Table 5 provides the performance results of our AttnZero on ImageNet. The results show that our method can transfer effectively to different ViT models. When applied to DeiT-T, AttnZero achieves a significant 2.7% gain over baseline. Similarly, AttnZero improves PVT-T accuracy to 78.1%, surpassing the baseline by 3.0%. This demonstrates that AttnZero can seamlessly migrate and consistently enhance performance across ViT architectures. AttnZero also provides stable gains across models of varying scales. Paired with PVT-S, it boosts accuracy by 1.6%. Applied to larger models like PVT-M, PVT-L, Swin-T, Swin-S, and CSwin-T, improvements of 1.8%, 1.6%, 0.8%, 0.2% and 0.2% respectively are observed. These findings highlight AttnZero's stability in boosting the performance of ViT models from small to large scale. These results highlight the stability of our AttnZero in enhancing the performance of ViT models across different scales.

**Comparison with Other Linear Attention.** Table 6 compares various linear attention methods. The results demonstrate that AttnZero surpasses other linear attention methods and baseline when incorporated into the DeiT-T model. Notably, compared to the Softmax baseline, AttnZero achieves a 2.7% improvement. In contrast, other linear attention methods, such as Hydra Attn, Efficient Attn, Linear Angular, and FLatten, exhibit lower accuracies. These findings highlight the superiority of AttnZero in preserving or surpassing the performance achieved by Softmax attention. On the other hand, other linear attention methods tend to result in notable performance degradation.

### 5.3 Transfer Experiments on Detection and Segmentation

**Object Detection.** We evaluate our methods for object detection and instance segmentation tasks on the COCO dataset [39]. Table 7 provides results on Mask R-CNN detector [25] with 1× schedules. Our proposed PVT-T+AttnZero model achieves 2.3 box AP gains on PVT-T and significantly outperforms FLatten.

**Table 8:** Results of semantic segmentation on ADE20K. We compute the FLOPs for both the encoders and decoders using input images at a resolution of 512×2048.

| Backbone | Method | FLOPs | #Params | mIoU | mAcc |
|---|---|---|---|---|---|
| PVT-T | SemanticFPN | 158G | 17M | 36.57 | 46.72 |
| PVT-T+FLatten | SemanticFPN | 169G | 16M | 37.21 | 48.95 |
| **PVT-T+AttnZero** | SemanticFPN | 126G | 15M | **39.16** | **50.30** |
| PVT-S | SemanticFPN | 225G | 28M | 41.95 | 53.02 |
| **PVT-S+AttnZero** | SemanticFPN | 165G | 23M | **42.62** | **54.02** |

These enhancements demonstrate the benefits of incorporating our found attention module within the popular ViT model for object detection tasks.
**Semantic Segmentation.** We assess the performance of our model using the ADE20K dataset [71]. Table 8 provides semantic segmentation results on SemanticFPN [29]. Among them, the Flatten module yields slight gains on the baseline PVT-T model. However, our AttnZero model achieves even higher mIoU (39.16) and mAcc (50.30) scores while reducing the FLOPs and parameters. Similarly, AttnZero achieves clear gains on PVT-S. These results highlight the compatibility of our AttnZero with different backbone architectures and its consistent improvements across various scenarios.

## 6 Conclusion

In this paper, we introduce AttnZero, the first method that automates the discovery of efficient attention modules that can be applied to various ViT models. AttnZero constructs a comprehensive search space consisting of different unary and binary operators as potential attention mechanisms. It utilizes a multi-objective genetic algorithm to search within this design space efficiently. Additionally, we introduce Attn-Bench-101, which provides attention profile results to summarize the insights gained from the search. Extensive experiments demonstrate that the attention modules discovered by AttnZero can be successfully applied to advanced ViTs for image classification, segmentation, and detection tasks. For future work, we will consider extending the AttnZero with knowledge distillation methods [32–35, 38, 53, 63]. **Iimitation.** We present search strategies but still involve some search costs following most AutoML methods. But these costs are worthwhile because our discovered attention can be effectively applied to different ViTs and tasks without additional searching (proven by our extensive transfer experiments). **Social impact.** Our AttnZero focuses on technology improvement without social and ethical implications.

## Acknowledgements

The research was supported by Theme-based Research Scheme (T45-205/21-N) from Hong Kong RGC, and Generative AI Research and Development Centre from InnoHK.

## References


1. http://tiny-imagenet.herokuapp.com/
2. Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al.: Xcit: Cross-covariance image transformers. NIPS (2021)
3. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Hoffman, J.: Hydra attention: Efficient attention with many heads. In: ECCVW (2022)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
5. Chen, M., Peng, H., Fu, J., Ling, H.: Autoformer: Searching transformers for visual recognition. In: ICCV. pp. 12270–12280 (2021)
6. Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al.: Rethinking attention with performers. In: ICLR (2021)
7. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. JMLR (2023)
8. Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., Shen, C.: Conditional positional encodings for vision transformers. Arxiv preprint 2102.10882 (2021), https://arxiv.org/pdf/2102.10882.pdf
9. Chu, X., Zhang, B., Xu, R.: Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In: ICCV (2021)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation (2002)
11. 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)
12. Dong, P., Li, L., Wei, Z.: Diswot: Student architecture search for distillation without training. In: CVPR (2023)
13. 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)
14. Dong, P., Niu, X., Li, L., Tian, Z., Wang, X., Wei, Z., Pan, H., Li, D.: Rd-nas: Enhancing one-shot supernet ranking ability via ranking distillation from zero-cost proxies. ICASSP (2023)
15. 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)
16. Dong, P., Niu, X., Tian, Z., Li, L., Wang, X., Wei, Z., Pan, H., Li, D.: Progressive meta-pooling learning for lightweight image classification model. In: ICASSP (2023)
17. Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. ArXiv **abs/2107.00652** (2021)
18. Dong, X., Yang, Y.: Nas-bench-201: Extending the scope of reproducible neural architecture search (2020)
19. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

20. Gao, J., Xu, H., Shi, H., Ren, X., Philip, L., Liang, X., Jiang, X., Li, Z.: Autobert-zero: Evolving bert backbone from scratch. In: AAAI (2022)
21. Guan, C., Wang, X., Zhu, W.: Autoattend: Automated attention representation search. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 3864–3874. PMLR (18–24 Jul 2021), http://proceedings.mlr.press/v139/guan21a.html
22. Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J.: Single path one-shot neural architecture search with uniform sampling. arXiv preprint arXiv:1904.00420 (2019)
23. Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J.: Single path one-shot neural architecture search with uniform sampling. In: ECCV. vol. abs/1904.00420, pp. 544–560. Springer (2019)
24. Han, D., Pan, X., Han, Y., Song, S., Huang, G.: Flatten transformer: Vision transformer using focused linear attention. In: ICCV (2023)
25. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
26. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. In: ICCV (2021)
27. Hu, Y., Wang, X., Li, L., Gu, Q.: Improving one-shot nas with shrinking-and-expanding supernet. Pattern Recognition (2021)
28. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are rnns: Fast autoregressive transformers with linear attention. In: ICML (2020)
29. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: CVPR (2019)
30. Kitaev, N., Kaiser, Ł., Levskaya, A.: Reformer: The efficient transformer. In: ICLR (2020)
31. Li, K., Yu, R., Wang, Z., Yuan, L., Song, G., Chen, J.: Locality guidance for improving vision transformers on tiny datasets. In: ECCV. pp. 110–127. Springer (2022)
32. Li, L.: Self-regulated feature learning via teacher-free feature distillation. In: ECCV (2022)
33. Li, L., Dong, P., Li, A., Wei, Z., Yang, Y.: Kd-zero: Evolving knowledge distiller for any teacher-student pairs. NeuIPS (2024)
34. Li, L., Dong, P., Wei, Z., Yang, Y.: Automated knowledge distillation via monte carlo tree search. In: ICCV (2023)
35. Li, L., Jin, Z.: Shadow knowledge distillation: Bridging offline and online knowledge transfer. In: NeuIPS (2022)
36. Li, L., Sun, H., Dong, P., Wei, Z., Shao, S.: Auto-das: Automated proxy discovery for training-free distillation-aware architecture search. In: ECCV (2024)
37. 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)
38. Li, L., Wang, Y., Yao, A., Qian, Y., Zhou, X., He, K.: Explicit connection distillation (2020)
39. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
40. 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)
41. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)

42. Lu, J., Yao, J., Zhang, J., Zhu, X., Xu, H., Gao, W., Xu, C., Xiang, T., Zhang, L.: Soft: Softmax-free transformer with linear complexity. In: NeurIPS (2021)
43. Nakai, K., Matsubara, T., Uehara, K.: Att-darts: Differentiable neural architecture search for attention. In: IJCNN. IEEE
44. 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)
45. Peng, H., Du, H., Yu, H., Li, Q., Liao, J., Fu, J.: Cream of the crop: Distilling prioritized paths for one-shot neural architecture search. NIPS (2020)
46. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. In: ICML (2018)
47. Qin, J., Wu, J., Xiao, X., Li, L., Wang, X.: Activation modulation and recalibration scheme for weakly supervised semantic segmentation. In: AAAI (2022)
48. Raquel, C.R., Naval Jr, P.C.: An effective use of crowding distance in multiobjective particle swarm optimization. In: CGEC (2005)
49. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: AAAI (2019)
50. Real, E., Liang, C., So, D., Le, Q.: Automl-zero: Evolving machine learning algorithms from scratch. In: ICML (2020)
51. Real, E., Liang, C., So, D., Le, Q.: Automl-zero: Evolving machine learning algorithms from scratch. In: ICML (2020)
52. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Li, F.F.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision (2015)
53. 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)
54. Shen, Z., Zhang, M., Zhao, H., Yi, S., Li, H.: Efficient attention: Attention with linear complexities. In: WACV (2021)
55. So, D.R., Mańke, W., Liu, H., Dai, Z., Shazeer, N., Le, Q.V.: Primer: Searching for efficient transformers for language modeling. arXiv preprint arXiv:2109.08668 (2021)
56. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML (2019)
57. Tay, Y., Dehghani, M., Bahri, D., Metzler, D.: Efficient transformers: A survey. arXiv preprint arXiv:2009.06732 (2020)
58. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)
59. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. corr, abs/2302.13971, 2023. doi: 10.48550. arXiv preprint arXiv.2302.13971
60. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: ICCV (2021)
61. Wang, X., Xiong, X., Neumann, M., Piergiovanni, A., Ryoo, M.S., Angelova, A., Kitani, K.M., Hua, W.: Attentionnas: Spatiotemporal attention cell search for video classification. In: ECCV (2020)
62. Wei, Z., Pan, H., Li, L.L., Lu, M., Niu, X., Dong, P., Li, D.: Convformer: Closing the gap between cnn and vision transformers. arXiv preprint arXiv:2209.07738 (2022)

63. Xiaolong, L., Lujun, L., Chao, L., Yao, A.: Norm: Knowledge distillation via n-to-one representation matching (2022)
64. Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., Singh, V.: Nyströmformer: A nyström-based algorithm for approximating self-attention. In: AAAI (2021)
65. Yang, A., Esperança, P.M., Carlucci, F.M.: Nas evaluation is frustratingly hard. arXiv preprint arXiv:1912.12522 (2019)
66. You, H., Xiong, Y., Dai, X., Wu, B., Zhang, P., Fan, H., Vajda, P., Lin, Y.C.: Castling-vit: Compressing self-attention via switching towards linear-angular attention at vision transformer inference. In: CVPR (2023)
67. You, S., Huang, T., Yang, M., Wang, F., Qian, C., Zhang, C.: Greedynas: Towards fast one-shot nas with greedy supernet. In: CVPR (2020)
68. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: Metaformer is actually what you need for vision. In: CVPR (2022)
69. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: ICCV (2021)
70. Zheng, Z., Gao, X., Pan, J., Luo, Q., Chen, G., Liu, D., Jiang, J.: Autoattention: Automatic field pair selection for attention in user behavior modeling. In: 2022 IEEE International Conference on Data Mining (ICDM). pp. 803–812. IEEE (2022)
71. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. IJCV (2019)
72. Zhou, Q., Sheng, K., Zheng, X., Li, K., Sun, X., Tian, Y., Chen, J., Ji, R.: Training-free transformer architecture search. In: CVPR. pp. 10894–10903 (2022)
73. Zhu, C., Li, L., Wu, Y., Sun, Z.: Saswot: Real-time semantic segmentation architecture search without training. In: AAAI (2024)
74. Zhu, C., Chen, W., Peng, T., Wang, Y., Jin, M.: Hard sample aware noise robust learning for histopathology image classification. TMI
75. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
76. 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)
77. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. In: ICLR (2017)
78. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: CVPR (2018)