Supplementary Materials for *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

In this supplemental material, we present additional details on our proposed method. By providing these additional details, we aim to provide a deeper understanding by offering thoughtful explanations, detailed analyses, and technical specifics that were excluded from the main paper due to length limitations. Our intention is to facilitate complete replication and fair evaluation of the framework. The supplemental material is organized as follows:

- Section 1 delves into further experiments and discussions. It includes additional experiments conducted on high-resolution inputs, a comprehensive comparison of different methods, analysis of seeds and hyperparameters, quantitative results, and discussions.
- Section 2 presents a thorough analysis of AttnZero-Bench-101. It outlines the training settings and provides a detailed examination of candidate attention analysis.
- Section 3 provides comprehensive implementation details, covering various settings and hyperparameters.
- Section 4 offers a comprehensive description of the detailed model architectures utilized in our method.

1 More Experiments and Discussions

1.1 Transfer Experiments on High-Resolution Inputs

The experimental results in Table 1 show that our method, AttnZero, consistently achieves performance boosts even when applied to the Swin model with high-resolution inputs. Compared to the baseline Swin-B model, our Swin-B+AttnZero variant demonstrates improved accuracy. Specifically, with a 224^2 resolution, AttnZero achieves a Top-1 accuracy of 83.7, 0.2 higher than the baseline. Similarly, at a resolution of 384^2, AttnZero achieves a Top-1 accuracy of 84.8, showing a boost of 0.3 compared to the baseline. These results highlight our method's effectiveness in enhancing the Swin model's performance, even with high-resolution inputs.

✉ Corresponding authors.

Table 1: Results of AttnZero with high resolution inputs on the ImageNet classification task.

Method	**Reso**	**#Params**	**Flops**	**Top-1**
Swin-B	224^2	88M	15.4G	83.5
Swin-B+AttnZero	224^2	89M	15.3G	**83.7** (+0.2)
Swin-B	384^2	88M	47.0G	84.5
Swin-B+AttnZero	384^2	88M	43.5G	**84.8** (+0.3)

Table 2: Comparison of different linear attention designs on Swin-Tiny structures.

Linear Attention	FLOPs	#Param	Acc.
Hydra Attn [2]	4.5G	29M	80.7
Efficient Attn [38]	4.5G	29M	81.0
Linear Angular Attn [44]	4.5G	29M	79.4
Enhanced Linear Attn [3]	4.5G	29M	81.8
AttnZero (ours)	4.3G	28.2M	**82.1** (+0.8)

1.2 More Comparisons on ImageNet Datasets

Comparison results in Table 2 demonstrate that our proposed AttnZero method consistently outperforms other efficient linear attention designs on the Swin-Tiny structure. Despite having slightly fewer FLOPs and parameters compared to the existing techniques, AttnZero achieves the highest accuracy of 82.1%, surpassing the second-best Enhanced Linear Attention by a significant margin of 0.8%. Our method exhibits superior performance while maintaining computational efficiency even when compared to computationally expensive attention mechanisms like Hydra Attention, Efficient Attention, and Linear Angular Attention. This consistent boost in accuracy across different efficient attention methods highlights the effectiveness of our approach in leveraging linear attention for improved representation learning. Furthermore, as depicted in Figure 1 (right), our approach incorporating PVT and Swin architectures consistently outperforms various other models, including RegNet [35], T2T [45], ConT [11], and CvT [42]. This demonstrates our approach's superior performance and stability compared to these alternative models.

1.3 Multiple Repeated Trials and Sensitivity Analysis of Seeds

The experiment involving multiple repeated trials and sensitivity analysis of seeds aims to assess the robustness and stability of the proposed method's performance. Table 3 presents the results of training the searched attention on a small dataset with different initialization seeds. The observed variances in the results range between 0.1 and 0.2, indicating a relatively low degree of fluctuation.

Table 3: Multiple replicate experiments results of AttnZero on tiny datasets. We report top-1 "mean (std)" accuracies (%) over 3 runs.

Model	Top-1 Acc. (%)		
	CIFAR-100	Flowers [33]	Chaoyang [48]
DeiT-T	65.08	50.06	82.00
DeiT-T+AttnZero	**77.68** ±0.20	**57.89** ±0.30	**83.12** ±0.06
AutoFormer-T	66.58	54.98	82.84
AutoFormer-T+AttnZero	**78.61** ±0.24	**61.58** ±0.28	**83.68** ±0.10
PVT-T	67.42	58.57	82.46
PVT-T+AttnZero	**76.68** ±0.16	**64.30** ±0.24	**84.57** ±0.08
Swin-T	68.25	58.85	82.98
Swin-T+AttnZero	**75.90** ±0.18	**65.13** ±0.22	**85.11** ±0.09

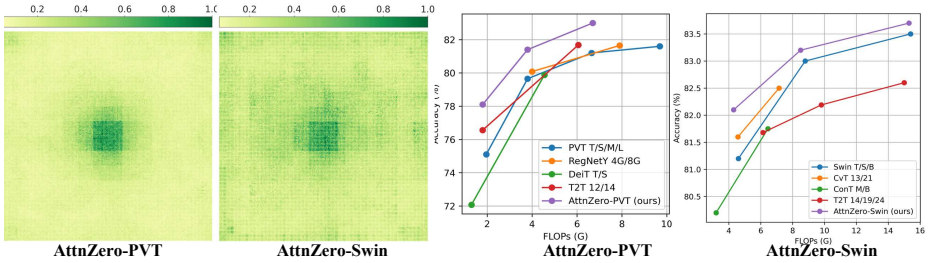

Figure 1: Effective Receptive Field (ERF) visualization (***Left***) and Accuracy (***Right***) of our proposed AttnZero.

As shown in Table 4, our method consistently outperforms the baseline model. This narrow range of variance suggests that the performance of the proposed method is consistent and reliable, and the improvement over other methods is stable across different initialization conditions. Consequently, the experiment demonstrates that the proposed approach is not overly sensitive to specific seed values, ensuring reproducible and trustworthy results. The consistent performance across multiple trials with varying initialization seeds provides confidence in the method's robustness and applicability in practical scenarios.

1.4 Analysis of Hyperparameters

Our framework is involved in search and training hyperparameters. For all the training hyperparameters, we use the same configurations with baselines and other methods [12, 29]. For a fair comparison, we do not change the training hyperparameters in all experiments. Therefore, we also can analyze the impact of this parameter without analyzing it. In our hyperparametric analysis, we investigate the impact of different hyperparameters on the performance of our method. Specifically, we analyze the effects of population size (N), crossover probability (p_c), mutation probability (p_m), and maximum generations (G) on our method.

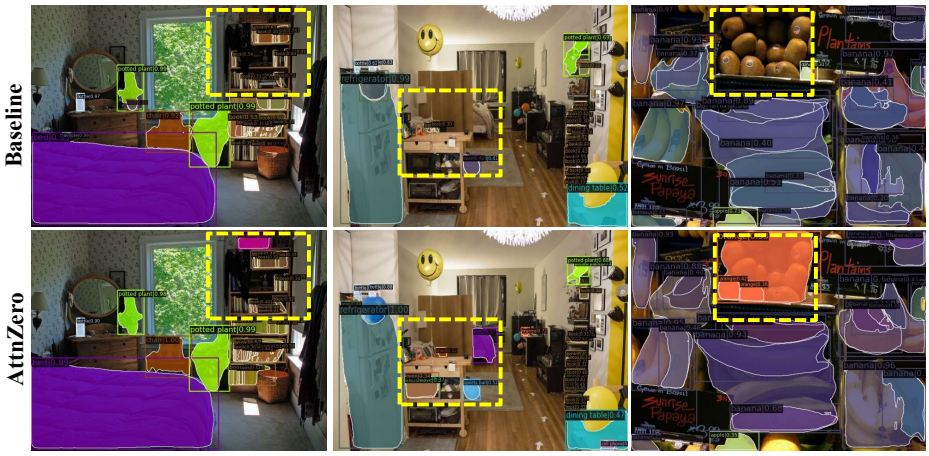

Figure 2: Qualitative analysis of baseline Mask RCNN-PVT-Tiny (***First Row***) and Mask RCNN-PVT-Tiny with our approach (***Second Row***) on COCO benchmarks.

Table 4: Different seeds on method with AutoFormer-T on CIFAR-100 dataset.

Method	random seed			AVG	STD
	0	1	2		
(Ours)	78.61	78.35	78.10	**78.35**	0.26

In our default setting, we configure these hyperparameters as $(20, 0.9, 0.1, 100)$. Table 5 presents the results of our method with different hyperparameter configurations on the CIFAR-100 dataset using AutoFormer-T. It can be observed that increasing the number of search rounds from 100 to 200 leads to a marginal improvement in performance, as seen in the comparison of $(20, 0.9, 0.1, 100)$ and $(20, 0.9, 0.1, 200)$. However, the other settings in our default configuration appear optimal, as variations in population size, crossover probability, and mutation probability do not significantly impact the performance. For instance, comparing $(20, 0.9, 0.1, 100)$ with $(10, 0.9, 0.1, 100)$ shows only a slight decrease in performance. Similarly, individually changing the crossover or mutation probability does not result in substantial performance differences. These findings suggest that our default hyperparameter settings are practical, and further adjustments to these parameters may not yield significant performance improvements.

1.5 Quantitative Results

The quantitative results, as shown in Figure 1 (left), demonstrate the effectiveness of our method in expanding the receptive fields. The perceptual field of our approach encompasses a larger central region, indicating that the underlying ViTs enhanced by our method possess a larger perceptual field. This global visual modelling capability empowers our model with enhanced capability for

Table 5: Different Hyperparameters (N, p_c, p_m, G) on method with AutoFormer-T on CIFAR-100 dataset.

$(20, 0.9, 0.1, 100)$	$(20, 0.9, 0.1, 200)$	$(20, 0.9, 0.1, 200)$	$(10, 0.9, 0.1, 100)$	$(10, 0.8, 0.1, 100)$	$(20, 0.9, 0.05, 100)$
78.61	78.82	78.88	78.12	78.42	78.46

global visual understanding. Moving to Figure 2, the visualizations comparing the detection results of our method and the baseline method in MS-COCO provide clear evidence of the significant superiority of our approach. Our method excels in accurately detecting small objects and handling object detection in complex scenes. These results highlight the robustness and capability of our method in addressing challenging object detection scenarios.

1.6 More Discussions

About comparisons of the AttnZero framework compared to existing NAS methods. We would like to address the comparisons made between the AttnZero framework and existing NAS methods [4, 8, 13, 24, 25, 32, 47, 49]: (1) General NAS methods: Our method is not a generic NAS approach. While general NAS methods focus on improving the efficiency of NAS and the method itself, our method has a different objective. Our primary purpose is to establish a new attention mechanism search explicitly tailored for Vision Transformers (ViTs) rather than enhancing the NAS aspects. Therefore, our method tends to have a different search space, strategy, and goal than these generic methods. (2) ViT-NAS approach: The ViT-NAS approach migrates generic NAS methods to search the width and depth of different parts of the ViT architecture. However, it is crucial to highlight that our method is not searching for architectural dimensions but rather the fine internal structure of attention. We have conducted extensive experiments on ViT-NAS to demonstrate that our method is orthogonal to this technique, as the search objects are fundamentally different. (3) Other Auto-Zero or Multi-Objective NAS methods: While we share some ideas with these approaches, such as Auto-Zero or Multi-Objective NAS, we have proposed a new search space and a specific search process based on our newly established task. Consequently, we are a specific application of these general ideas, but we differ in our unique search space and process. Our AttnZero framework is distinct from existing NAS methods. We focus on discovering efficient attention modules tailored for ViTs rather than generic NAS improvements or architectural dimensions. We have introduced a specific search space and search process to address the challenges specific to attention discovery in ViTs.

About comparisons of the AttnZero with other search strategies. While we acknowledge that a direct comparison with existing NAS methods may not be feasible at this stage, the fact that our method outperforms random search provides strong evidence of its effectiveness. Random searches are indeed strong baselines and can provide valuable insights into the effectiveness of our search operator for this new task. In our experiments, we conducted a random search

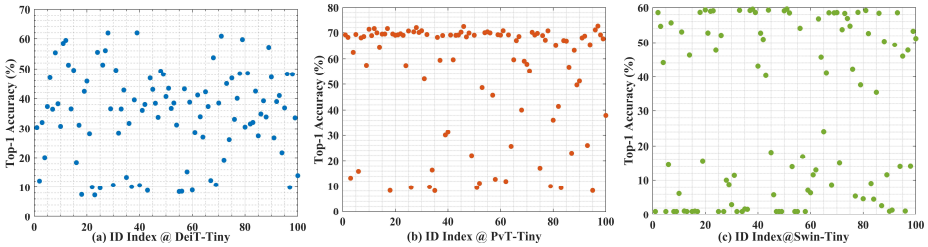

Figure 3: (a) Performance distribution of subsets of Attn-Bench-101 (AutoFormer). (b) Performance distribution of subsets of Attn-Bench-101 (PVT). (c) Performance distribution of subsets of Attn-Bench-101 (Swin). Here, we randomly select 100 attentions from our entire benchmark for comprehensive analysis.

in our search space and consistently observed that our method outperformed random search. This significant improvement demonstrates the effectiveness of our search operator for this new task. Our core contribution lies in proposing new search tasks and spaces beyond the existing preliminary search methods. By addressing the specific challenges of attention discovery in Vision Transformers, we have developed a framework that achieves superior performance compared to random search. We believe that this achievement demonstrates the efficacy of our search operator and its suitability for this task. We encourage future work to build upon our findings and further optimize the method [**?**, 6, 7, 9, 21, 34, 41, 47] with knowledge distillation methods [5, 16–20, 22, 23, 26, 37, 43] to continue improving the performance.

2 AttnZero-Bench-101

AttnZero-Bench-101 is a comprehensive evaluation platform for assessing the performance of different attention mechanisms on four ViT architectures. This benchmark provides a standardized evaluation for each attention mechanism and enables NAS algorithms to identify high-performing attention mechanisms. By evaluating and comparing the performance of different attention mechanisms on a standard set of architectures, AttnZero-Bench-101 facilitates fair comparisons. It provides valuable insights for the development and improvement of attention mechanisms.

2.1 Datasets and Training settings

Our study includes popular image classification datasets, such as CIFAR-100 [14], which consists of 50,000 training samples and 10,000 test samples across 100 classes. To train all our Vision Transformer (ViT) models, we adopt the training settings outlined in [28]. Specifically, we utilize the AdamW optimizer [31] with an initial learning rate of 5e-4 and a weight decay of 0.05. The learning rate schedule follows a cosine policy [30], gradually reducing the learning rate

to 5e-6. Each ViT model undergoes 100 training epochs, with a linear warm-up period of 20 epochs, employing a batch size of 128. Our training process involves images with an interpolated resolution of 224×224. These standardized settings ensure consistency and enable fair comparisons among our attention candidates.

2.2 Attention Search Space

Our search space includes six computation graphs and different operators. The primitive operations utilized in are classified into two categories based on their inputs. The unary operations are applied to a single input tensor, while the binary operations are conducted on two input tensors. The search space of the automatic proxy consists of 25 unary operations and 4 binary operations. Their detailed formulations are presented in Table 6.

2.3 Detailed Analysis of Benchmarks

The experimental results in Figure 3 demonstrate the performance of various attention candidates on the DeiT, PVT, and Swin models. Notably, the accuracy of different attention candidates appears to be more consistent in the DeiT model. Conversely, the PVT and Swin models exhibit a wider range of successful and failed candidates. This difference can be attributed to the additional complexities in layering and window design introduced by PVT and Swin, which make the performance of attention candidates more sensitive to these intricate architectural designs compared to the relatively simpler DeiT model. Additionally, Figure 4 in the main text presents the accuracy distribution of the AutoFormer. It is important to consider that the different types of computational graphs and candidate operators depicted in Figure 4 represent combined statistics across the four ViT subsets (DeiT, PVT, Swin, and AutoFormer). This comprehensive analysis offers insights into the variations in attention candidate performance across diverse vision transformer architectures, highlighting the impact of architectural choices on the effectiveness of attention mechanisms. It is worth noting that in Figure 4 (c), the label "wo norm.op" refers to the default search space without the inclusion of *l2_ norm* and *min_ max_ norm* operators. Similarly, "wo act. ops" indicates the default search space without the *swish* and *elu* activation functions. Lastly, "wo binary ops" represents the default search space without the *euclidean_ dis* and *cosine_ sim* operators. These labels specify the specific search space configurations used for the respective experiments, providing important context for understanding the results presented in Figure 4.

2.4 Detailed Attention Design Guidelines

In the detailed design of the attention mechanism, attention types i3n302 and i3n303 consistently outperform other types, as shown in Figure 4 (b). These attention types possess an augmented topology that strikes a favorable balance between performance and efficiency. When analyzing the ablations presented in

Figure 4 (c), it becomes evident that advanced activation and normalization operations play a more significant role in the overall performance of the attention mechanism compared to the inclusion of new binary operators. This emphasizes the importance of selecting and combining appropriate activation and normalization operations for optimal attention performance. Regarding advanced activation operations, Swish is a novel activation function introduced by Google researchers. It is defined as the element-wise product of the input and the sigmoid function of the input. Swish combines the smoothness of the sigmoid function with the linearity of the input, resulting in a more flexible and expressive activation. It has been found to outperform ReLU in certain cases, offering improved gradient flow and better handling of negative values. Another advanced activation function is ELU, which introduces non-linearity into the attention mechanism. It smoothly approaches negative values by using the exponential function. ELU mitigates the "dying ReLU" problem and allows negative values to have non-zero gradients, making it advantageous in certain scenarios. For advanced normalization operations, L2 normalization (also known as Euclidean normalization) scales the attention scores by dividing them by the L2 norm of the scores. This ensures that the attention scores have a unit length, leading to a smoother attention distribution and improved interpretability. L2 normalization prevents attention weights from becoming too concentrated on a few elements, promoting a more balanced attention distribution. Another option is min-max normalization, which scales the attention scores to a specific range, typically between 0 and 1. This normalization achieves this by subtracting the minimum value from each score and dividing it by the range (maximum value - minimum value). Min-max normalization ensures that the attention scores are within a consistent range, making them more interpretable and comparable across different contexts. By incorporating binary operators, information interaction within the attention mechanism is facilitated. In addition to the widely used *cosine_similarity* operator from Hydra [2], we also consider the *euclidean_distance* as a simplified calculation option. These binary operators provide alternative ways to measure the similarity or dissimilarity between attention elements, enabling more diverse and flexible attention patterns. To summarize, attention types i3n302 and i3n303 are preferred in the detailed design of the attention mechanism due to their consistent superior performance. Furthermore, advanced normalization and activation operations contribute more significantly to the attention mechanism's performance compared to new binary operators. These design choices enhance performance, flexibility, and adaptability of the attention mechanism in various applications and scenarios.

3 Experiments and Implementation Details

3.1 Details on Search Experiments

Search spaces and Parallel Search. We divide the overall search space into sub-search spaces according to different attention mechanisms. The search space $\mathcal{A}$ represents the set of all possible attention mechanisms or architectures that

can be explored. By dividing this space into sub-search spaces $\mathcal{A}_1$, $\mathcal{A}_2$, ..., $\mathcal{A}_6$, we group attention mechanisms or architectures based on their characteristics or types. Let $f_{\text{acc}}(\mathcal{A})$ and $f_{\text{comp}}(\mathcal{A})$ denote the accuracy and computational complexity of an attention mechanism $\mathcal{A}$, respectively. During the search process, we perform parallel searches within each sub-search space $\mathcal{A}_i$, $i = 1, 2, \ldots, 6$. This parallel exploration allows us to explore different attention mechanisms efficiently simultaneously, taking advantage of computational resources and potentially discovering diverse solutions. Within each sub-search space $\mathcal{A}_i$, we search for attention mechanisms a_i^* that optimize the accuracy and computational complexity objectives:

$$a_i^* = _{a \in \mathcal{A}_i} f_{\text{acc}}(\mathcal{A}), _{a \in \mathcal{A}_i} f_{\text{comp}}(\mathcal{A})$$

After exploring the sub-search spaces in parallel, we combine the discovered solutions a_i^* to obtain the final search result a^*. This combination can involve selecting the best-performing attention mechanisms from each sub-search space or combining different attention mechanisms to create hybrid or ensemble models. The performance metrics can include accuracy $f_{\text{acc}}(\mathcal{A})$ and computational complexity $f_{\text{comp}}(\mathcal{A})$, or other relevant evaluation criteria. By exploring a diverse set of attention mechanisms and architectures, our method aims to identify solutions that balance computational complexity and accuracy, ensuring efficient and effective attention mechanisms for various applications.

Multi-Objective Evolution Search. The multi-objective evolutionary algorithm used in our search process is the NSGA-II, a popular algorithm for solving multi-objective optimization problems. The algorithm has the following configuration in our setup: The number of Population sizes (N) is set to 20. The crossover probability (p_c) between two parent individuals to generate offspring is set to 0.9. Mutation probability (p_m) set to 0.1. The algorithm runs for a maximum of 100 generations. In our parallel search process, we simultaneously apply this multi-objective evolutionary algorithm on the DeiT, AutoFormer, PVT, and Swin architectures using the specified configuration. This parallel search approach allows us to explore a diverse set of concurrent attention modules and architectures, potentially leading to more effective and efficient attention mechanisms.

Evaluation Settings. Our attention search is performed on the CIFAR-100 dataset, where the images are scaled to a resolution of 224×224 pixels. We search using the validation results to ensure fair comparisons, and our validation set does not overlap with the test set. To obtain validation results, we adopt standard training settings from the literature [15], such as 300 epochs and the AdamW optimizer. The training process for obtaining validation results can be formulated as follows: Let $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ denote the training and validation datasets, respectively. The model parameters θ are optimized using the AdamW optimizer, a variant of the Adam optimizer with weight decay regularization. The objective function is the cross-entropy loss $\mathcal{L}_{\text{CE}}$ between the predicted class probabilities $p(y|x, \theta)$ and the ground truth labels y for the input images x:

$$\mathcal{L}_{\text{CE}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}} [\log p(y|x, \theta)]$$

The training process is carried out for 300 epochs and the validation performance is evaluated on $\mathcal{D}_{\text{val}}$ using the same cross-entropy loss function.

Search Costs. We have implemented effective strategies to mitigate these costs and make the search more efficient: 1. We have employed a divide-and-conquer strategy for parallel search and subsearch spaces. This approach significantly reduces the overall search time required. Our experiments have shown that the entire search process takes approximately 1.05 days on the 8xV100 platform, making it efficient and feasible. 2. we have also implemented program checking and rejection strategies to filter out unpromising candidates swiftly. These strategies have proven effective in accelerating the search process by more than five times, further reducing the overall cost. 3. Furthermore, it is worth mentioning that the cost of our search can be further reduced by modifying the dataset and training setup of the agent. In our approach, we have deliberately avoided using a proxy setup to mitigate potential sorting problems and ensure the discovery of the best-performing attention modules. This thoughtful consideration helps reduce unnecessary costs and streamline the search process.

Program Checking. Program checking is conducted to verify the functionality of the single candidate attention module by assessing its ability to generate outputs using random tensor inputs. This process allows us to identify and exclude candidates that exhibit arithmetic errors or dimension mismatches. To achieve this, we apply random tensor inputs $X \in \mathbb{R}^{B\times S\times D}$ to the attention module, where B represents the batch size, S denotes the sequence length, and D is the input feature dimension. The output of the attention module, $Y \in \mathbb{R}^{B\times S\times D}$, is then verified to ensure that it has the correct dimensions and that the computed values are within a reasonable range, without any computational discrepancies such as NaN (Not a Number) or Inf (Infinity) values. By applying a range of random tensor inputs and verifying the outputs, we can identify and exclude candidates that exhibit arithmetic errors or dimension mismatches, ensuring the attention module's functionality before proceeding with further evaluations.

Rejection Protocol. We conduct a Rejection Protocol for newly generated Candidate Attentions to reduce the number of invalid proxies during the search process. Specifically, we check whether the loss of Attention belongs to a set of invalid scores, which includes values such as NaN (Not a Number) and Inf (Infinity). This step helps identify and filter out candidates with unreliable or inappropriate attention representations. Let $\mathcal{L}(\theta)$ denote the loss function of a Candidate Attention parameterized by θ, evaluated on a validation dataset. We define the set of invalid scores $\mathcal{I}$ as:

$$\mathcal{I} = \{\mathcal{L}(\theta) | \mathcal{L}(\theta) \in \{\text{NaN}, \pm\infty\}\}$$

If $\mathcal{L}(\theta) \in \mathcal{I}$, the Candidate Attention is immediately rejected as it exhibits unreliable or inappropriate attention representations. In addition, we track a running historical average μ and standard deviation σ of validation performance for previous Candidate Attention. If the current Candidate Attention's loss $\mathcal{L}(\theta)$ scores more than one standard deviation below the mean of this distribution, i.e., $\mathcal{L}(\theta) > \mu + \sigma$, it is also rejected. This filters out proposals that are inferior

to the linear attention baseline. By implementing these strategies within the Rejection Protocol, we can achieve an acceleration of more than five times during the search process. This allows us to efficiently filter out invalid or underperforming candidates and focus on those with the potential for better attention representations.

3.2 Detailed Analysis on Discovered Attention

The optimal attention $a^*_{AttnZero}$ in our search, whose detailed expressions are as follows:

$$a^*_{AttnZero} = Q \ \times \eta_2 \left(\eta_2 \left(Q \right) \eta_2 \left(\phi_1 \left(K \right)^T \eta_1 \left(V \right) \right) \right) \tag{1}$$

where $Q, K, V \in \mathbb{R}^{B \times S \times D}$ are the query, key, and value tensors, respectively, obtained by linearly projecting the input $X \in \mathbb{R}^{B \times S \times D}$, with B representing the batch size, S denoting the sequence length, and D being the input feature dimension. The function ϕ_1 utilizes the Exponential Linear Unit (ELU) with a shift of 1, introducing nonlinearities into the attention computation. The ELU activation function is defined as:

$$\phi_1(x) = \begin{cases} x & \text{if } x > 0 \\ e^x - 1 & \text{otherwise} \end{cases}$$

The η_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. The min-max normalization function is given by:

$$\eta_1(X) = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Furthermore, the l2 normalization function η_2 guarantees that the attention weights have a unit norm, facilitating better interpretation and utilization of the attention mechanism. The l2 normalization is defined as:

$$\eta_2(X) = \frac{X}{\|X\|_2}$$

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 handles computations for sequences of varying lengths efficiently.

3.3 Details on Transfer Experiments on Tiny Datasets

Datasets. We include popular image classification datasets in our study, such as Tiny ImageNet [1], Flowers [33], and Chaoyang [48]. These datasets provide

Table 6: The unary operations and binary operations in the proxy search space. "UOP" denotes the unary operation, and "BOP" denotes the binary operation.

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	$\frac{x}{1+\lvert x\rvert}$
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	$x \times \text{sigmoid}(x)$
UOP17	logsigmoid	$\log\left(\frac{1}{1+e^{-x}}\right)$
UOP18	neg	$-x$
UOP19	norm_d	$\text{normalize}(x, \text{dim} = d)$
UOP20	leaky_relu	$\max(0.1x, x)$
UOP21	mish	$x \times \tanh(\ln 1 + \exp(x))$
BOP01	sum	$x + y$
BOP02	cosine_sim	$cos(x, y)$
BOP03	product	$x \odot y$
BOP04	matrix_multi	$x \cdot y$
BOP05	euclidean_dis	$d(x, y)$
BOP06	mse	$\text{mse}(x, y)$

diverse and representative samples for evaluating the performance of our method across different domains. Tiny ImageNet [1] is a subset of the ImageNet dataset, consisting of 200 different classes with 500 training images and 50 validation images per class. It serves as a compact version of the original ImageNet dataset, allowing for faster experimentation and benchmarking of computer vision models. Flowers [33] is a dataset specifically designed for flower classification tasks. It comprises 102 categories of various flower species, with each class containing between 40 and 258 images. The dataset presents a challenging task due to the inherent similarities among different flower types, requiring models to capture subtle visual cues for accurate classification. Chaoyang dataset [48] consists of colon slide image patches from the Chaoyang hospital, labeled by three professional pathologists. The patches with consensus labels from all pathologists formed the testing set, while the remaining patches constituted the training set. For samples in the training set with inconsistent labels, one pathologist's opinion was randomly chosen. The final dataset comprised normal, serrated, adenocarcinoma, and adenoma samples for training and testing, totaling several thousand images.

Table 7: Architectures of AttnZero-DeiT models.

stage	output	AttnZero-DeiT-T	
		AttnZero	DeiT Block
res1	14×14	$\begin{bmatrix} \text{win } 14\times14 \\ \text{dim } 192 \\ \text{head } 3 \end{bmatrix} \times 12$	None

Table 8: Architectures of AttnZero-PVT models.

stage	output	AttnZero-PVT-M		AttnZero-PVT-L	
		AttnZero	PVT Block	**AttnZero**	PVT Block
res1	56×56	Conv1×1, stride=4, 64, LN			
		$\begin{bmatrix} \text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1 \end{bmatrix} \times 3$	None
res2	28×28	Conv1×1, stride=2, 128, LN			
		$\begin{bmatrix} \text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2 \end{bmatrix} \times 3$	None
res3	14×14	Conv1×1, stride=2, 320, LN			
		$\begin{bmatrix} \text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5 \end{bmatrix} \times 6$	None
res4	7×7	Conv1×1, stride=2, 512, LN			
		$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8 \end{bmatrix} \times 3$	None

Implementation. In the training process, we train DeiT, AutoFormer, PVT, and Swin models using the discovered attention from scratch. The training follows standard settings [15], including 300 training epochs, a cosine learning rate scheduler, and the AdamW optimizer. Specifically, we utilize the AdamW optimizer [31] with an initial learning rate of 5e-4 and a weight decay of 0.05. The learning rate schedule follows a cosine policy [30], gradually reducing the learning rate to 5e-6. Each ViT model undergoes 100 epochs of training, with a linear warm-up period of 20 epochs, employing a batch size of 128. The training process involves images with an interpolated resolution of 224×224. These standardized settings ensure consistency and enable fair comparisons among our attention candidates.

3.4 Details on Transfer Experiments on ImageNet Datasets

Datasets. ImageNet [36] is a widely used large-scale dataset in computer vision research. It contains 1.2 million training images and 50,000 validation images across 1,000 categories. . The dataset covers a wide range of object categories such as animals, plants, vehicles, and everyday objects. ImageNet serves as a benchmark for evaluating the performance of various computer vision models and algorithms.

Table 9: Architectures of AttnZero-Swin models.

stage	output	AttnZero-Swin-T		AttnZero-Swin-S		AttnZero-Swin-B	
		AttnZero	Swin Block	**AttnZero**	Swin Block	**AttnZero**	Swin Block
res1	56×56	concat 4×4, 96, LN		concat 4×4, 96, LN		concat 4×4, 128, LN	
		[win 56×56, dim 96, head 3] $\times 2$	None	[win 56×56, dim 96, head 3] $\times 2$	None	[win 56×56, dim 128, head 3] $\times 2$	None
res2	28×28	concat 4×4, 192, LN		concat 4×4, 192, LN		concat 4×4, 256, LN	
		[win 28×28, dim 192, head 6] $\times 2$	None	[win 28×28, dim 192, head 6] $\times 2$	None	[win 28×28, dim 256, head 6] $\times 2$	None
res3	14×14	concat 4×4, 384, LN		concat 4×4, 384, LN		concat 4×4, 512, LN	
		None	[win 7×7, dim 384, head 12] $\times 6$	None	[win 7×7, dim 384, head 12] $\times 18$	None	[win 7×7, dim 512, head 12] $\times 18$
res4	7×7	concat 4×4, 768, LN		concat 4×4, 768, LN		concat 4×4, 1024, LN	
		None	[win 7×7, dim 768, head 24] $\times 2$	None	[win 7×7, dim 768, head 24] $\times 2$	None	[win 7×7, dim 1024, head 24] $\times 2$

Implementation. We conduct the experiment on the ImageNet [36] with the standard settings [10,29,39] and the input images are resized to a size of 224x224 pixels using bicubic interpolation. For the training settings, we train the model for 300 epochs with a warm-up period of 20 epochs. The weight decay is set to 0.05. The base learning rate is 5e-4, the warm-up learning rate is 5e-7, and the minimum learning rate is 5e-6. The learning rate scheduler is set to "cosine" with a decay interval of 30 epochs and a decay rate of 0.1. We use the AdamW optimizer with an epsilon value of 1e-8 and betas set to (0.9, 0.999). The SGD momentum is set to 0.9. In terms of augmentation, we employ the AutoAugment policy and set the random erase probability to 0.25. We also use mixup with an alpha value of 0.8. These augmentation techniques enhance the model's ability to generalize and improve performance.

3.5 Details on Transfer Experiments on Downstream Tasks

Object Detection. MS-COCO dataset [27] is a widely used dataset for object detection, segmentation, and captioning tasks in computer vision. It contains over 330,000 diverse images with detailed annotations such as bounding boxes, segmentation masks, and captions. The dataset covers various object categories and includes images with complex scenes and diverse backgrounds. MS-COCO is a valuable resource for training and evaluating models in object recognition and scene understanding. The training setup for our detector is configured with settings similar to the FLatten [12]. Specifically, we use the 1x learning rate schedule (12 epochs). The learning rate configuration is set to a step policy, with a linear warm-up period of 500 iterations and a warm-up ratio of 0.001. The learning rate is set to 2e-4. The learning rate is adjusted at epochs 8 and 11. For the optimizer, we use the AdamW optimizer with a learning rate of lr. The weight decay is set to 0.0001.

Semantic Segmentation. The ADE20K dataset [46] is a comprehensive and widely used dataset for semantic segmentation in computer vision. It contains a diverse collection of images that cover a wide range of indoor and outdoor

scenes. The dataset consists of over 20,000 images, making it a significant resource for training and evaluating models. ADE20K provides detailed pixel-level annotations for each image, labeling each pixel with a corresponding object or scene class. This level of annotation allows for fine-grained understanding of image content and enables models to accurately segment objects and scenes in images. The training setup for our method follows the same settings as FLatten. Specifically, we use a maximum of 40,000 iterations and a learning rate schedule designed for 40,000 iterations. Evaluation is performed every 4,000 iterations, using the mean Intersection over Union (mIoU) as the evaluation metric. For the optimizer, we employ AdamW with a learning rate of 0.0002 and a weight decay of 0.0001 . The learning rate policy is set to polynomial, with a power of 0.9 and a minimum learning rate of 0.0, applied on an iteration basis rather than an epoch basis.

4 Model Architectures Details

We summarize the architectures of Transformer models, namely DeiT [39], PVT [40], and Swin [29], as detailed in Tables 7 to 9. To ensure a fair comparison, we augment our implementation by adding Depthwise convolution as same as FLatten [12] for fair comparison. Furthermore, we replace the original self-attention blocks with our focused linear attention block at all stages of DeiT and PVT models. However, in the case of the Swin model, we incorporate our module only in the early stages while maintaining the original structure (width and depth) of the model. This approach enables us to assess the performance of our module in comparison to the original attention blocks across different stages of the models.

References

1. `http://tiny-imagenet.herokuapp.com/`
2. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Hoffman, J.: Hydra attention: Efficient attention with many heads. In: ECCVW (2022)
3. Cai, H., Gan, C., Han, S.: Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. arXiv preprint arXiv:2205.14756 (2022)
4. Chen, K., Yang, L., Chen, Y., Chen, K., Xu, Y., Li, L.: Gp-nas-ensemble: a model for the nas performance prediction. In: CVPRW (2022)
5. Dong, P., Li, L., Wei, Z.: Diswot: Student architecture search for distillation without training. In: CVPR (2023)
6. 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)
7. 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)
8. 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)

9. 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)
10. 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)
11. d'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. In: ICML. pp. 2286–2296. PMLR (2021)
12. Han, D., Pan, X., Han, Y., Song, S., Huang, G.: Flatten transformer: Vision transformer using focused linear attention. In: ICCV (2023)
13. Hu, Y., Wang, X., Li, L., Gu, Q.: Improving one-shot nas with shrinking-and-expanding supernet. Pattern Recognition (2021)
14. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech Report (2009)
15. 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)
16. Li, L.: Self-regulated feature learning via teacher-free feature distillation. In: ECCV (2022)
17. 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)
18. Li, L., Dong, P., Li, A., Wei, Z., Yang, Y.: Kd-zero: Evolving knowledge distiller for any teacher-student pairs. NeuIPS (2024)
19. Li, L., Dong, P., Wei, Z., Yang, Y.: Automated knowledge distillation via monte carlo tree search. In: ICCV (2023)
20. Li, L., Jin, Z.: Shadow knowledge distillation: Bridging offline and online knowledge transfer. In: NeuIPS (2022)
21. Li, L., Li, A.: A2-aug: Adaptive automated data augmentation. In: CVPRW (2023)
22. Li, L., Shiuan-Ni, L., Yang, Y., Jin, Z.: Boosting online feature transfer via separable feature fusion. In: IJCNN (2022)
23. Li, L., Shiuan-Ni, L., Yang, Y., Jin, Z.: Teacher-free distillation via regularizing intermediate representation. In: IJCNN (2022)
24. 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)
25. 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)
26. Li, L., Wang, Y., Yao, A., Qian, Y., Zhou, X., He, K.: Explicit connection distillation (2020)
27. 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)
28. Liu, S., Ni'mah, I., Menkovski, V., Mocanu, D.C., Pechenizkiy, M.: Efficient and effective training of sparse recurrent neural networks. Neural Computing and Applications pp. 1–12 (2021)
29. 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)
30. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2017)

32. Lu, L., Chen, Z., Lu, X., Rao, Y., Li, L., Pang, S.: Uniads: Universal architecture-distiller search for distillation gap. In: AAAI (2024)
33. 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)
34. Qin, J., Wu, J., Xiao, X., Li, L., Wang, X.: Activation modulation and recalibration scheme for weakly supervised semantic segmentation. In: AAAI (2022)
35. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollar, P.: Designing network design spaces. In: CVPR (2020)
36. 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)
37. 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)
38. Shen, Z., Zhang, M., Zhao, H., Yi, S., Li, H.: Efficient attention: Attention with linear complexities. In: WACV (2021)
39. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)
40. 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)
41. 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)
42. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. In: ICCV. pp. 22–31 (2021)
43. Xiaolong, L., Lujun, L., Chao, L., Yao, A.: Norm: Knowledge distillation via n-to-one representation matching (2022)
44. 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)
45. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.H., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: ICCV (2021)
46. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. IJCV (2019)
47. Zhu, C., Li, L., Wu, Y., Sun, Z.: Saswot: Real-time semantic segmentation architecture search without training. In: AAAI (2024)
48. Zhu, C., Chen, W., Peng, T., Wang, Y., Jin, M.: Hard sample aware noise robust learning for histopathology image classification. TMI
49. 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)