

Token Pruning within the Attention Mechanism for Efficient Vision Transformers

Shuto Kusaki¹, Ryuto Ishibashi¹ and Lin Meng^{2,*}

¹Graduate School of Science and Engineering, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan

²College of Science and Engineering, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan

Abstract

In recent years, Vision Transformers (ViTs) have garnered significant attention in the field of image recognition for their superior performance, outperforming conventional Convolutional Neural Networks (CNNs). However, a major challenge with ViTs is their substantial computational cost and memory usage, which makes them difficult to deploy in resource-constrained environments. In particular, their application to devices with limited computational power and memory, such as Internet of Things (IoT) devices, faces numerous technical barriers. Against this backdrop, the efficient utilization of computational resources is essential to make ViTs practical for real-world use. Therefore, this research focuses on pruning as a means to reduce computational complexity. In this study, we propose a novel pruning method that diverges from conventional token pruning. Our approach prunes the attention mechanism itself—a core component of ViTs—to reduce the computational overhead generated within it. While traditional token pruning only reduces the number of tokens, our proposed method streamlines the attention mechanism, enabling a more significant reduction in computational complexity. This allows for further computational savings that are unattainable with token pruning alone, leading to a substantial decrease in resource consumption while maintaining overall performance. Experiments conducted on the CIFAR-10 dataset show that by applying our proposed attention mechanism pruning, we achieved a 47% reduction in computational complexity with only a 0.86% decrease in accuracy. This result is highly beneficial for running ViTs in computationally restricted settings and indicates the potential for their practical application on IoT and edge devices. Thus, we believe that our novel pruning method significantly enhances the computational efficiency of ViTs, contributing to the expansion of their applicability in resource-constrained environments.

Keywords

Vision Transformer, Token Pruning, Deep learning, Image Recognition

1. Introduction

Image recognition is a fundamental task in Computer Vision (CV), with widespread applications in diverse fields such as autonomous driving systems, medical image diagnostics, and security systems. Advances in this technology have dramatically improved the ability of machines to

The 7th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2025), September 10-11, 2025, Kusatsu, Shiga, Japan

*Corresponding author.

✉ ri0124xv@ed.ritsumei.ac.jp (S. Kusaki); gr0517rs@ed.ritsumei.ac.jp (R. Ishibashi); menglin@fc.ritsumei.ac.jp (L. Meng)

ORCID iD 0000-0003-4351-6923 (L. Meng)

© 2025 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

understand and analyze images in a human-like manner, underpinning modern technological innovation. Historically, Convolutional Neural Networks (CNNs) [1, 2, 3] have been the predominant model for image recognition. CNNs exhibit excellent performance in local feature extraction, establishing them as the standard model in the field [4, 5, 6]. However, CNNs face several limitations. In particular, they are constrained by their inability to capture the global context of an entire image, failing to fully reflect long-range dependencies. Consequently, there has been a demand for new approaches to enhance recognition capabilities by considering the broader context of an image.

In this context, the Vision Transformer (ViT) [7] has emerged, adapting the Transformer architecture [8] that has rapidly proliferated in the field of Natural Language Processing (NLP). ViT has garnered significant attention by achieving performance superior to that of CNNs in image recognition. Its architecture involves dividing an image into fixed-size patches and feeding them into the model as a sequence of tokens. This approach enables the model to effectively capture the global context of the entire image, overcoming the bias towards local features inherent in CNNs. Indeed, ViTs have produced results surpassing traditional CNNs in various image recognition tasks, and their continued development is highly anticipated.

On the other hand, ViTs present several practical challenges [9]. The most significant of these are their immense computational complexity and massive memory usage compared to CNNs. The core Self-Attention mechanism in ViT has a computational complexity that scales quadratically with the number of input tokens, N . This implies that as the input image resolution increases or the number of patches grows, the computational cost explodes. This makes it extremely difficult to deploy ViT in resource-constrained environments such as mobile and edge devices. Therefore, to resolve these issues and make ViT practical for a wide range of applications, methods to improve its computational efficiency are urgently needed.

As one of the most promising approaches to address this computational bottleneck, token pruning has gained significant attention in recent years [10, 11]. Token pruning is a technique for creating more efficient models by reducing the computational cost and memory usage. It achieves this by decreasing the number of tokens in the sequence processed by the ViT, specifically by identifying and removing those deemed to have a low contribution to the final prediction or to be redundant. In this method, after the input image is converted into tokens, low-importance tokens are identified and removed, which directly mitigates the load on the Self-Attention mechanism. This pruning process allows for the creation of lighter and faster models by reducing the required computations while minimizing accuracy degradation.

Although token pruning is a crucial technology for advancing the practical application of ViTs, existing research still leaves room for improvement. Many methods have potential for further optimization in the design of their importance criteria (scoring) and in the timing and method of applying the pruning. Therefore, motivated by this gap, this paper proposes an improved token pruning methodology. In this work, we aim to validate the effectiveness of our newly proposed method and to provide new insights into enhancing ViT efficiency.

Overall, our main contributions can be summarized below:

- **Integrated Attention Pruning:** We introduce a novel pruning mechanism that operates within the self-attention computation, enabling more efficient processing by reducing both token and attention computation simultaneously.

- **Superior Computational Efficiency:** Our method achieves significant computational savings while maintaining competitive accuracy. Specifically, ATPViT reduces FLOPs by up to 47% and memory usage by up to 36.4% compared to baseline models, with only minimal accuracy degradation (0.86–1.12%).
- **Enhanced Resource Utilization:** Compared to conventional Top-K and EViT methods, ATPViT achieves additional reductions of 0.9% in FLOPs and 3.6-4.3% in memory usage without further accuracy loss, demonstrating improved efficiency over existing approaches.
- **Practical Benefits for Edge Deployment:** The proposed method enables larger batch sizes within the same GPU memory constraints and reduces overall energy consumption, making it particularly suitable for resource-constrained mobile and edge devices.

Extensive experiments on standard benchmarks demonstrate that ATPViT consistently outperforms existing token pruning methods in terms of computational efficiency while maintaining comparable or superior accuracy. The results suggest that integrating pruning directly into the attention mechanism represents a promising direction for developing efficient Vision Transformers suitable for practical deployment scenarios.

2. Related Work

2.1. Vision Transformer

The Vision Transformer (ViT)[7] revolutionized computer vision by successfully adapting the Transformer architecture from natural language processing to image-related tasks. Unlike traditional Convolutional Neural Networks (CNNs)[1, 2, 3], ViT reimagines an image as a sequence of fixed-size patches. These patches are treated as tokens, similar to words in a sentence, allowing the model’s self-attention mechanism to weigh the importance of every patch in relation to all others. This enables the model to capture global context across the entire image from its earliest layers, fundamentally challenging the long-held dominance of convolutional approaches.

Initially proving its strength in image classification, the ViT architecture was quickly extended to more complex, dense prediction tasks like object detection [12] and semantic segmentation [13]. However, the scalability of the original design was a limitation, prompting significant architectural advancements. The most impactful of these have been hierarchical ViTs, such as the Swin Transformer [14]. By computing self-attention within local, non-overlapping windows that are shifted between layers, Swin Transformer efficiently builds a hierarchical feature representation. This design established ViTs as a powerful and versatile backbone for a wide array of vision applications.

Despite their success, ViTs are notoriously demanding in terms of computational resources. The self-attention mechanism’s complexity scales quadratically with the number of input patches, making ViTs computationally expensive for high-resolution images and difficult to deploy on resource-constrained devices. This efficiency challenge has become a major focus for the research community, driving the development of various optimization strategies. These include architectural redesigns, knowledge distillation[15], and, central to the work in this

paper, pruning methods designed to reduce computational load without a significant loss in performance.

2.2. Model acceleration

The fundamental idea behind Token Pruning is based on the observation that the token sequences processed by ViTs contain numerous redundant tokens that contribute little to the final prediction. For instance, in a typical image, background regions such as the sky, ground, or walls often contain less information and have uniform textures. The numerous tokens corresponding to these areas hold mutually similar information, and it is considered unnecessary to process all of them in detail. This issue becomes increasingly critical as model size grows because larger models divide an image into finer and more numerous tokens, which increases the number of redundant tokens and leads to wasted computational resources.

2.2.1. Top-k Token Selection

Top-k token selection is a straightforward pruning approach that leverages attention weights to identify the most important tokens for retention. The method computes token importance scores based on the attention weights from the class token to patch tokens:

$$s_i = \frac{1}{H} \sum_{h=1}^H A_{cls,i}^{(h)} \quad (1)$$

where s_i represents the importance score of the i -th token, H is the number of attention heads, and $A_{cls,i}^{(h)}$ denotes the attention weight from the class token to the i -th patch token in the h -th head. The method then selects the top- k tokens with the highest importance scores:

$$\mathcal{K} = \text{TopK}(\{s_i\}_{i=1}^N, k) \quad (2)$$

where N is the total number of tokens and $k = N - p$ with p being the number of tokens to prune. This approach offers computational efficiency and requires minimal architectural modifications. However, it suffers from complete information loss of discarded tokens, potentially limiting performance when pruned tokens contain relevant information.

2.2.2. EViT (Efficient Vision Transformer)

EViT [16] addresses the information loss problem by introducing a token merging mechanism that preserves information from low-importance tokens. Instead of simply discarding tokens, EViT aggregates information from pruned tokens into a single representative token:

$$\mathbf{t}_{merged} = \sum_{i \in \mathcal{P}} w_i \mathbf{t}_i \quad (3)$$

where \mathcal{P} represents the set of tokens to be pruned, $w_i = \frac{s_i}{\sum_{j \in \mathcal{P}} s_j}$ are normalized importance weights, and \mathbf{t}_i is the feature vector of the i -th token. The final output maintains a fixed sequence length while preserving global information through the merged token. This weighted

aggregation approach demonstrates superior accuracy-efficiency trade-offs compared to simple pruning methods, showing that thoughtful information preservation enhances token reduction techniques in Vision Transformers.

3. Methodology

In this study, we propose the Attention-based Token Pruning in Vision Transformer (ATPViT), which performs pruning within the attention mechanism itself to dynamically remove low-importance tokens based on attention scores. This approach is designed to minimize information loss while simultaneously reducing the computational overhead within the attention mechanism.

3.1. ATPViT: TopK-Based

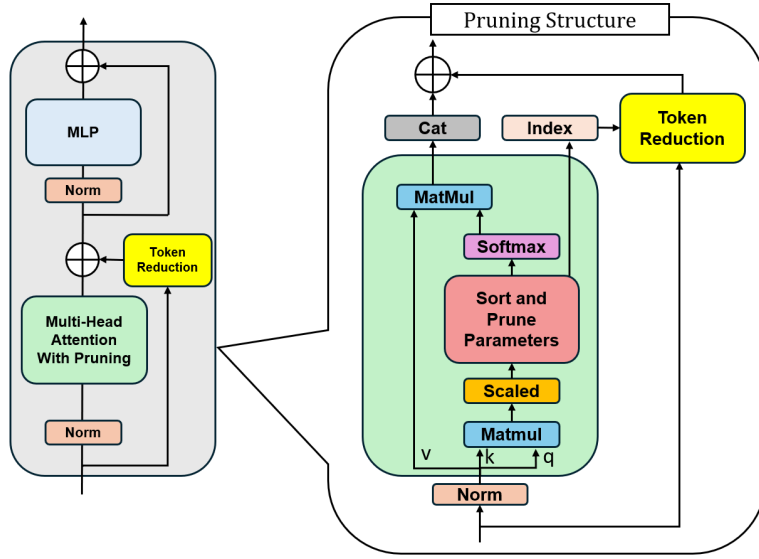


Figure 1: The structure of the proposed ATPViT. The right diagram provides a detailed view of the "Multi-Head Attention With Pruning" block, illustrating how pruning is integrated within the attention mechanism.

In conventional ViT pruning, one of the most common approaches, top-k pruning, inserts a pruning block between the attention block and the MLP block. Pruning is thus performed after the attention computation is complete. However, with this approach, the computational cost of the attention mechanism itself is not reduced. In contrast, rather than computing the full attention output and then pruning tokens, ATPViT optimizes the attention computation itself.

The detailed methodology of ATPViT is as follows. First, the computation within the Multi-Head Self-Attention (MHSA) of a Transformer Encoder can be described by the following equations.

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (4)$$

Our pruning process is introduced immediately after the initial operation in this calculation: the dot product of Q and the transposed K (K^T). At this stage, based on a criterion for identifying which tokens are redundant (importance scoring), we retain the top- K rows corresponding to the most important tokens and discard the rest. The indices of these top- K rows, which represent the tokens to be kept after the attention calculation, are then saved. While the importance scores in this study are determined by the methods we describe later, this step can be performed using any arbitrary method. Next, using the saved indices, we select the top- K tokens from the original input tokens to the MHSA block and discard the rest. A residual connection is then formed between these pruned input tokens and the output of the MHSA, thereby reducing the total number of tokens carried forward. By performing this operation at each layer of the Transformer block, our method achieves a much greater reduction in computational complexity compared to conventional approaches.

3.1.1. Computational Benefits

This approach provides several computational advantages:

- **Reduced Attention Computation:** By pruning attention matrix rows, we reduce the computational complexity from $O(N^2)$ to $O((N - p) \times N)$ for the attention-value multiplication.
- **Memory Efficiency:** The output tensor size is reduced from N to $N - p$ tokens, decreasing memory requirements for subsequent layers.
- **Cascading Speedup:** Token reduction in early layers accelerates computation in all subsequent transformer blocks.

3.1.2. Integration with Transformer Blocks

The pruning mechanism is seamlessly integrated into the transformer architecture. Each transformer block processes fewer tokens as the network progresses, maintaining the quality of representations while achieving significant computational savings. The method requires minimal modifications to the original Vision Transformer architecture and can be applied to any layer within the network.

The overall algorithm maintains the structural integrity of the transformer while providing adaptive token reduction based on learned attention patterns, making it particularly suitable for scenarios requiring both accuracy and efficiency.

3.2. ATPViT: EViT-Based

While ATPViT-Pruning achieves computational efficiency through direct token elimination, ATPViT-Merge addresses the inherent information loss limitation by introducing an intelligent token merging mechanism. This variant extends our attention-based pruning framework with EViT-inspired information preservation strategies, creating a hybrid approach that combines computational efficiency with enhanced accuracy retention. Similar to ATPViT-Pruning, the method first computes token importance scores using class token attention weights and identifies

Algorithm 1 Attention-based Token Pruning in Vision Transformer (ATPViT)

Require: Input tokens $\mathbf{X} \in \mathbb{R}^{B \times N \times C}$, number of tokens to prune p

Ensure: Pruned tokens $\mathbf{X}' \in \mathbb{R}^{B \times (N-p) \times C}$, selected indices \mathcal{I}

```
1: // Multi-Head Self-Attention Computation
2:  $\mathbf{Q}, \mathbf{K}, \mathbf{V} = \text{Linear}(\mathbf{X})$ 
3:  $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{B \times H \times N \times d_h}$ 
4:  $\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_h}}\right)$ 
5: // Token Importance Scoring
6:  $\mathbf{a}_{cls} = \mathbf{A}[:, :, 0, :] \in \mathbb{R}^{B \times H \times N}$ 
7:  $s_i = \frac{1}{H} \sum_{h=1}^H \mathbf{a}_{cls}[b, h, i]$  for  $i = 1, \dots, N$ 
8:  $s_0 = +\infty$ 
9: // Token Selection
10:  $k = N - p$ 
11:  $\mathcal{I} = \text{TopK}(\{s_i\}_{i=0}^{N-1}, k)$ 
12: // Attention Matrix Pruning
13:  $\mathbf{A}' = \mathbf{A}[:, :, \mathcal{I}, :]$ 
14:  $\mathbf{X}_{attn} = \mathbf{A}'\mathbf{V}$ 
15: // Token Pruning
16:  $\mathbf{X}' = \mathbf{X}[:, \mathcal{I}, :]$ 
17:  $\mathbf{X}' = \mathbf{X}' + \mathbf{X}_{attn}$ 
18: return  $\mathbf{X}', \mathcal{I}$ 
```

tokens for removal. However, instead of simply discarding these tokens, ATPViT-Merge employs a weighted aggregation mechanism that preserves their information content.

The merging process operates directly within the attention computation. After identifying tokens to be removed (\mathcal{D}), the method extracts their corresponding attention rows from the attention matrix and computes weighted combinations based on their importance scores:

$$\mathbf{a}_{merged} = \sum_{i \in \mathcal{D}} \frac{s_i}{\sum_{j \in \mathcal{D}} s_j} \mathbf{a}_i \quad (5)$$

where s_i represents the importance score of token i and \mathbf{a}_i is its attention vector. This creates a single representative attention row that encapsulates information from all removed tokens.

The merged attention row is then concatenated with the attention rows of retained tokens, resulting in a reduced but information-preserving attention matrix. This approach maintains a fixed output sequence length while ensuring that no information is completely lost during the pruning process.

Compared to ATPViT-Pruning, ATPViT-Merge typically achieves better accuracy retention at the cost of slightly increased computational overhead due to the merging operations. The method provides a valuable trade-off option for applications where accuracy preservation is prioritized over maximum computational reduction, making it particularly suitable for scenarios requiring high-quality outputs while still benefiting from significant efficiency improvements.

This dual approach demonstrates the flexibility of our attention-based pruning framework,

allowing practitioners to choose between aggressive pruning (ATPViT-Pruning) and information-preserving reduction (ATPViT-Merge) based on their specific requirements.

4. Experiment

In this section, we evaluate the computational complexity and performance of various ViT pruning methods in order to compare our proposed method with existing approaches.

4.1. Implementation details

For our Vision Transformer models, we use `vit_small_patch_16_224` and `vit_base_patch_16_224` from the `timm` library. These are pretrained models configured for an input image size of 224x224 pixels and a patch size of 16x16. The `vit_small` model consists of 22.05M parameters, 6 attention heads, and 12 Transformer layers, while the `vit_base` model consists of 86.57M parameters, 12 attention heads, and 12 Transformer layers. The learning rate is set according to the formula ($\frac{Batch\ Size}{1024} \times 0.001$), where the batch size is adaptively adjusted for each model based on the available GPU memory.

All experiments were conducted on a platform equipped with an NVIDIA GeForce RTX 4090 GPU, an AMD Ryzen 9 7900X3D 12-Core Processor, and running Ubuntu 20.04.6 LTS as the operating system. The deep learning framework used was PyTorch, with Python 3.9 and CUDA 12.1.

Dataset For our experiments, we use two datasets: CIFAR-10 [17] and the Oxford-IIIT Pet Dataset.

The CIFAR-10 dataset consists of 50,000 training images and 10,000 test images, each being a 32x32 pixel RGB (3-channel) image. The images are categorized into 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Due to its small image size, CIFAR-10 is well-suited for addressing fundamental image recognition tasks while keeping computational costs manageable.

The Oxford-IIIT Pet Dataset was created through a joint effort by the Visual Geometry Group at the University of Oxford and the International Institute of Information Technology (IIIT), Hyderabad. It is an image dataset featuring 37 distinct breeds: 25 dog breeds and 12 cat breeds. The dataset comprises a total of 7,349 images, with approximately 200 images available for each breed. The annotations for each image include a precise category label for the breed, species information (dog or cat), and pixel-level segmentation masks that define the exact outline of each pet. This segmentation data is crucial for tasks that require a clear distinction between the subject and the background. Consequently, this dataset is used for Fine-Grained Visual Categorization (FGVC), an advanced classification task that involves distinguishing between highly similar sub-categories beyond simple labels like "dog" or "cat".

4.1.1. Evaluation metrics

In this study, to conduct a multifaceted evaluation of the performance of token pruning methods in ViT models, we performed a comprehensive analysis using the following three key metrics

in addition to accuracy.

FLOPs FLOPs represents the number of floating-point operations executed during one forward inference pass, indicating the computational complexity of the model. In this study, we used the FlopCountAnalysis from the fvcare library to analyze the number of operations in each layer in detail, reporting results in Giga-FLOPs (GFLOPs) units.

Throughput Throughput is defined as the number of images processed per unit time (images/second), serving as a metric to evaluate the practical processing performance of models. For measurement, we employed high-precision time measurement using CUDA event timers with GPU synchronization processing. Specifically, after 20 warm-up executions, we measured the inference execution time for 100 iterations and calculated throughput from the average value.

$$\text{Throughput} = \frac{\text{Batch Size}}{\text{Average Inference Time}}$$

Memory Usage Memory Usage measures GPU memory consumption (MB) during inference execution, evaluating memory efficiency. Using PyTorch’s CUDA memory statistics functionality, we calculated the difference between peak memory usage before and after inference execution.

$$\text{Memory Usage} = \text{Peak Memory} - \text{Initial Memory}$$

By employing these evaluation metrics, we achieved comprehensive performance evaluation of token pruning methods with emphasis on practical applicability, while considering the trade-off relationship with accuracy.

4.2. Main result

The main results for ATPViT on the CIFAR-10 dataset are presented in Table 1 and Table 2, while the results on the Oxford-IIIT Pet Dataset are shown in Table 3. We compare our proposed method, ATPViT, with the baseline methods it is derived from, Top-K and EViT. The accuracy measurements in Table 1 and Table 3 were conducted after 50 epochs of training. The measurements in Table 2 were obtained by applying each respective pruning model to the pretrained base models.

For both ViT-Small and ViT-Base, the proposed ATPViT reduces computational cost while incurring only a minor degradation in accuracy, achieving greater savings compared to conventional methods. Specifically, on CIFAR-10 with ViT-Small, our ATPViT-Topk reduces FLOPs by 47% and memory usage by 36.4% compared to the baseline, with only a 1.12% drop in accuracy. This represents an additional reduction of 0.9% in FLOPs and 3.6% in memory usage over the conventional Top-K method, without any further loss in accuracy. Furthermore, on ViT-Small, our ATPViT-EViT reduces FLOPs by 43.7% and memory usage by 31.5% compared to the baseline, with an accuracy drop of only 0.86%. This achieves an additional 0.9% reduction in

Table 1
Performance of ViT-small on CIFAR-10

Model	Top-1 Acc(%)	GFLOPs	Throughput(images/s)	Memory Usage(MB)
Baseline(ViT-Small)	98.09	4.250	3001	1815
Pruning Rate r=10				
Top-K	97.43	3.14 (-26.1%)	3763	1675 (-7.6%)
EViT	97.62	3.28 (-22.8%)	3446	1771 (-2.4%)
ATPViT(topk)	97.46	3.11 (-26.9%)	3696	1611 (-11.3%)
ATPViT(emit)	97.60	3.25 (-23.4%)	3317	1705 (-6.1%)
Pruning Rate r=16				
Top-K	96.94	2.29 (-46.1%)	4850	1219 (-32.8%)
EViT	97.25	2.43 (-42.8%)	4480	1321 (-27.2%)
ATPViT(topk)	96.97	2.25 (-47%)	4795	1154 (-36.4%)
ATPViT(emit)	97.23	2.39 (-43.7%)	4321	1243 (-31.5%)

Table 2
Performance of off-the-shelf ViT-base on CIFAR-10

Model	Top-1 Acc(%)	GFLOPs	Throughput(images/s)	Memory Usage(MB)
Baseline(ViT-base)	97.56	16.87	1010	3625
Pruning Rate r=10				
Top-K	96.87	12.02 (-28.7%)	1357	3337 (-7.9%)
EViT	97.00	12.57 (-25.4%)	1278	3531 (-2.5%)
ATPViT(topk)	96.88	11.93 (-29.3%)	1339	3211 (-11.4%)
ATPViT(emit)	96.96	12.49 (-26.0%)	1232	3401 (-6.1%)
Pruning Rate r=16				
Top-K	95.38	8.75 (-48.1%)	1804	2405 (-33.6%)
EViT	95.66	9.29 (-44.8%)	1697	2614 (-27.8%)
ATPViT(topk)	95.40	8.623 (-48.9%)	1794	2288 (-36.9%)
ATPViT(emit)	95.73	9.170 (-45.7%)	1648	2470 (-31.8%)

FLOPs and 4.3% in memory usage compared to the conventional EViT method, again without further degrading accuracy. When comparing ATPViT-EViT with ATPViT-TopK, the former exhibits improved accuracy at the cost of increased computation, indicating a trade-off between accuracy and computational cost.

Moreover, even when using the more challenging Oxford-IIIT Pet Dataset, ATPViT reduces computational cost while maintaining high accuracy. This suggests that ATPViT can be used as a general-purpose method across various datasets.

On the other hand, the throughput of our proposed ATPViT was lower than that of the conventional methods, despite its reduction in GFLOPs. We attribute this to the implementation overhead introduced by our more intricate pruning process. Specifically, ATPViT performs

Table 3Performance of ViT-small on Oxford-IIIT Pet Dataset (Pruning Rate $r=10$)

Model	Top-1 Acc(%)	GFLOPs	Throughput(images/s)	Memory Usage(MB)
Baseline(ViT-Small)	92.91	4.250	3027	1815
Top-K	90.92	3.14 (-26.1%)	3713	1675 (-7.6%)
EViT	91.23	3.28 (-22.8%)	3524	1771 (-2.4%)
ATPViT(topk)	90.94	3.11 (-26.9%)	3691	1611 (-11.3%)
ATPViT(evit)	91.22	3.25 (-23.4%)	3482	1705 (-6.1%)

several sequential operations within the attention computation itself: 1) calculating importance scores, 2) identifying the top-K indices, and 3) gathering the corresponding data from the attention matrix and input tokens. Optimizing this overhead remains a key challenge for future work.

These results indicate that ATPViT holds significant advantages for applications on resource-constrained mobile and edge devices. Its benefits also include improving training efficiency by enabling larger batch sizes within the same GPU memory and reducing overall energy consumption.

5. Conclusion

In this study, we have proposed a new pruning methodology for Vision Transformers. Through a pruning strategy for the attention matrix that simultaneously reduces both tokens and the attention computation itself, our proposed ATPViT achieves greater reductions in computational cost and memory usage compared to conventional methods, without sacrificing accuracy. Furthermore, this method can be easily adapted to any ViT architecture as it requires no additional parameters or specialized training procedures. We therefore conclude that ATPViT offers significant potential for the application of Vision Transformers in resource-constrained environments, such as mobile and edge devices.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

- [1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems* 25 (2012).
- [2] K. Simonyan, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* (2014).

- [3] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
- [4] H. Li, L. Meng, Hardware-aware approach to deep neural network optimization, *Neuro-computing* 559 (2023) 126808.
- [5] Z. Li, Y. Yan, X. Wang, Y. Ge, L. Meng, A survey of deep learning for industrial visual anomaly detection, *Artificial Intelligence Review* 58 (2025) 279.
- [6] Z. Li, Y. Ge, L. Meng, A multi-scale information fusion framework with interaction-aware global attention for industrial vision anomaly detection and localization, *Information Fusion* 124 (2025) 103356.
- [7] A. Dosovitskiy, An image is worth 16x16 words: Transformers for image recognition at scale, *arXiv preprint arXiv:2010.11929* (2020).
- [8] A. Vaswani, Attention is all you need, *Advances in Neural Information Processing Systems* (2017).
- [9] R. Ishibashi, L. Meng, Automatic pruning rate adjustment for dynamic token reduction in vision transformer, *Applied Intelligence* 55 (2025) 342.
- [10] Y. Ge, Z. Li, X. Yue, H. Li, L. Meng, Dataset purification-driven lightweight deep learning model construction for empty-dish recycling robot, *IEEE Transactions on Emerging Topics in Computational Intelligence* (2025) 1–16.
- [11] X. Yue, L. Meng, Yolo-sm: A lightweight single-class multi-deformation object detection network, *IEEE Transactions on Emerging Topics in Computational Intelligence* 8 (2024) 2467–2480. doi:10.1109/TETCI.2024.3367821.
- [12] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, 2020. URL: <https://arxiv.org/abs/2005.12872>. arXiv:2005.12872.
- [13] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, H. Xia, End-to-end video instance segmentation with transformers, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 8741–8750.
- [14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, 2021. URL: <https://arxiv.org/abs/2103.14030>. arXiv:2103.14030.
- [15] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015. URL: <https://arxiv.org/abs/1503.02531>. arXiv:1503.02531.
- [16] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, P. Xie, Not all patches are what you need: Expediting vision transformers via token reorganizations, *arXiv preprint arXiv:2202.07800* (2022).
- [17] A. Krizhevsky, Learning multiple layers of features from tiny images (2009). URL: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.