Review

Token Pruning for Efficient NLP, Vision, and Speech Models

Yong Jianhong *

Posted Date: 26 February 2025

doi: 10.20944/preprints202502.2098.v1

Keywords: token pruning; model compression; efficient transformers; dynamic sequence reduction; neural network optimization; deep learning acceleration; NLP efficiency; computer vision pruning; speech processing optimization; adaptive pruning

Review

Token Pruning for Efficient NLP, Vision, and Speech Models

Yong Jianhong

School of Information Science And Technology, Xiamen University; yong.jianhong@xmu.edu.cn

Abstract: The rapid growth of Transformer-based architectures has led to significant advancements in natural language processing (NLP), computer vision, and speech processing. However, their increasing computational demands pose challenges for real-time inference, edge deployment, and energy efficiency. Token pruning has emerged as a promising solution to mitigate these issues by dynamically reducing sequence lengths during model execution while preserving task performance. This survey provides a comprehensive review of token pruning techniques, categorizing them based on their methodologies, such as static vs. dynamic pruning, early exit strategies, and adaptive token selection. We explore their effectiveness across various domains, including text classification, machine translation, object detection, and speech recognition. Additionally, we discuss the trade-offs between efficiency and accuracy, challenges in generalization, and the integration of token pruning with other model compression techniques. Finally, we outline future research directions, emphasizing self-supervised token selection, multimodal pruning, and hardware-aware optimization. By consolidating recent advancements, this survey aims to serve as a foundational reference for researchers and practitioners seeking to enhance the efficiency of deep learning models through token pruning.

Keywords: token pruning; model compression; efficient transformers; dynamic sequence reduction; neural network optimization; deep learning acceleration; NLP efficiency; computer vision pruning; speech processing optimization; adaptive pruning

1. Introduction

The rapid advancement of deep learning, particularly in the field of natural language processing (NLP), has led to the widespread adoption of large-scale Transformer models such as BERT, GPT, and T5. These models have demonstrated remarkable performance across various tasks, including text classification, machine translation, question answering, and text generation [1]. However, their ever-growing size and computational complexity pose significant challenges in terms of efficiency, deployment, and real-world applicability, particularly in resource-constrained environments such as mobile devices, edge computing, and real-time inference systems [2]. To address these challenges, a wide range of model compression techniques have been explored, including pruning, quantization, knowledge distillation, and low-rank approximation. Among these, pruning has gained particular attention due to its ability to systematically remove redundant parameters or operations while maintaining competitive performance. While traditional pruning techniques focus on weight sparsification or structured pruning of entire neurons and layers, the advent of Transformer-based architectures has given rise to a more specialized pruning strategy: token pruning. Token pruning is a form of dynamic model compression that aims to reduce the computational burden of Transformer models by selectively removing tokens during inference, thereby reducing the sequence length and, consequently, the number of operations required in self-attention layers. Unlike weight pruning, which primarily targets the parameter efficiency of models, token pruning directly optimizes inference-time efficiency by discarding less informative tokens, leading to significant reductions in computational cost without requiring retraining or extensive fine-tuning. This makes token pruning particularly attractive for latency-sensitive applications such as real-time translation, speech recognition, and interactive dialogue systems. The core idea behind token pruning is motivated by the observation that not all tokens contribute equally to the final model output. In many NLP tasks, a large proportion of tokens—especially function words, punctuation, or repetitive structures—carry limited semantic value in deep Transformer layers. By identifying and removing such tokens dynamically, token pruning can achieve a trade-off between efficiency and accuracy [3]. Several methodologies have been proposed to achieve this, including attention-based importance scores, reinforcement learning strategies, and saliency-based token selection mechanisms [4]. These techniques vary in terms of granularity, adaptiveness, and computational overhead, leading to a diverse landscape of token pruning approaches. Despite its promising advantages, token pruning introduces several challenges. One key issue is determining which tokens to prune in a manner that is both computationally efficient and effective in preserving model accuracy. Unlike static pruning, which involves removing parameters once during training, token pruning is inherently dynamic and requires lightweight yet reliable importance estimation mechanisms. Moreover, aggressively pruning tokens may lead to degraded performance, loss of contextual information, and instability in sequence-to-sequence tasks [5]. Recent research efforts have attempted to mitigate these issues through learnable pruning strategies, hybrid pruning approaches that combine token and weight pruning, and architectural modifications that inherently reduce token redundancy [6]. Given the growing importance of token pruning in efficient NLP model design, this survey aims to provide a comprehensive overview of the field. We begin by reviewing the fundamental concepts of Transformer-based models and their computational bottlenecks. Next, we categorize and analyze different token pruning techniques, highlighting their methodologies, advantages, and limitations. We also discuss the interplay between token pruning and other compression techniques, such as quantization and distillation. Finally, we outline open challenges and future research directions, emphasizing the need for more robust, adaptable, and interpretable token pruning strategies [7]. By systematically examining existing work and identifying key trends, this survey serves as a valuable resource for researchers and practitioners seeking to optimize Transformer models for real-world deployment [8]. As the demand for efficient deep learning models continues to grow, token pruning is poised to play a crucial role in bridging the gap between large-scale NLP models and practical, efficient AI applications.

2. Background and Preliminaries

2.1. Transformer Models and Their Computational Complexity

Transformer-based architectures have revolutionized the field of natural language processing (NLP) by introducing the self-attention mechanism, which allows models to capture long-range dependencies in text. The foundation of Transformer models is the multi-head self-attention (MHSA) mechanism, which enables contextualized token representations by attending to all other tokens in a sequence. Given an input sequence of length N, self-attention computes attention scores between all pairs of tokens, resulting in an $\mathcal{O}(N^2)$ complexity per layer in terms of both computation and memory [9]. The computational burden of self-attention grows quadratically with sequence length, making it one of the primary bottlenecks in large-scale models such as BERT, GPT, and T5. Although techniques like sparse attention, low-rank approximations, and memory-efficient attention mechanisms have been proposed, reducing the effective sequence length remains one of the most direct ways to improve inference efficiency. This motivates the need for token pruning as a means to dynamically shorten sequences while maintaining model performance.

2.2. Model Compression Techniques for Transformers

Before delving into token pruning specifically, it is essential to understand the broader landscape of model compression techniques used to improve the efficiency of Transformer architectures. The primary methods include:

- **Weight pruning:** This approach removes individual weights or entire neurons in a network to create sparse models [10,11]. Variants such as unstructured pruning and structured pruning have been explored extensively.
- **Quantization:** Reducing the precision of model parameters from 32-bit floating point to lower-bit representations (e.g., 8-bit, 4-bit) significantly decreases memory usage and accelerates inference.
- **Knowledge distillation:** A smaller student model is trained to mimic the behavior of a larger teacher model, achieving efficiency without significant loss in accuracy.
- **Low-rank factorization:** Decomposing weight matrices into low-rank components reduces the number of parameters and computational complexity [12].
- **Efficient architectures:** Specialized architectures such as Longformer, Linformer, and Reformer modify the Transformer structure to handle longer sequences with reduced computational overhead [13].

While these methods provide various ways to enhance efficiency, token pruning stands out as a dynamic technique that adapts sequence length per input instance, making it particularly useful for real-time applications [10].

2.3. Definition and Taxonomy of Token Pruning

Token pruning refers to the process of selectively removing tokens from an input sequence during model inference to improve computational efficiency [14]. Unlike static pruning methods that modify a model's architecture permanently, token pruning operates dynamically, allowing the model to adjust sequence lengths based on input characteristics. Broadly, token pruning methods can be categorized into the following types:

- **Hard token pruning:** Tokens are entirely removed from the sequence, preventing them from contributing to future computations [15]. This results in direct computational savings but may lead to information loss [16].
- **Soft token pruning:** Instead of completely discarding tokens, their contributions are downweighted or aggregated into fewer representations. This allows the model to retain some information while reducing the effective sequence length.
- **Adaptive token pruning:** The decision to prune is dynamically determined based on the importance of tokens, often using learnable gating mechanisms or attention-based importance scores [17].
- **Hybrid approaches:** Some methods combine token pruning with other compression techniques, such as knowledge distillation or weight pruning, to maximize efficiency.

Each approach offers a different trade-off between computational savings and accuracy, and the effectiveness of token pruning largely depends on the underlying pruning criterion and implementation strategy.

2.4. Challenges in Token Pruning

While token pruning has shown promising results in improving Transformer efficiency, several challenges remain:

- **Token importance estimation:** Accurately determining which tokens to prune in a computationally efficient manner is nontrivial. Many approaches rely on attention scores, gradient-based saliency measures, or reinforcement learning.
- **Preserving model accuracy:** Aggressive token pruning can lead to performance degradation, particularly for tasks requiring fine-grained token interactions such as named entity recognition and machine translation [18].
- **Generalizability:** Token pruning strategies that work well on one task or dataset may not generalize effectively to others [19]. Models need adaptive mechanisms that can adjust pruning strategies based on task requirements [20].

doi:10.20944/preprints202502.2098.v1

- **Implementation complexity:** Unlike weight pruning, which can be applied statically, token pruning requires runtime modifications to the model's execution graph, making deployment more complex.

Addressing these challenges is an active area of research, and recent innovations aim to develop more robust, adaptive, and interpretable token pruning techniques.

2.5. Scope and Organization of the Survey

This survey provides a comprehensive review of token pruning techniques for efficient Transformer-based models [21]. The subsequent sections are organized as follows:

- Section 3 discusses various token pruning methodologies, including rule-based, learning-based, and hybrid approaches [22].
- Section 4 provides an in-depth comparison of token pruning methods in terms of computational savings, accuracy trade-offs, and practical deployment considerations [23].
- Section 5 highlights real-world applications of token pruning in NLP tasks, computer vision, and speech processing.
- Section 6 outlines open research challenges and future directions in token pruning.

By systematically exploring existing approaches and identifying key trends, this survey aims to serve as a valuable reference for researchers and practitioners seeking to develop efficient NLP models through token pruning [24].

3. Token Pruning Methodologies

Token pruning strategies can be broadly classified based on their approach to token selection, pruning granularity, and whether the pruning decisions are static or dynamic. In this section, we explore different methodologies for token pruning, categorizing them into three main groups: rule-based methods, learning-based approaches, and hybrid techniques [25].

3.1. Rule-Based Token Pruning

Rule-based token pruning methods rely on predefined heuristics or criteria to determine which tokens should be removed during inference. These approaches do not require additional learnable parameters and are typically computationally inexpensive. Some of the most common rule-based pruning techniques include:

3.1.1. Attention Score-Based Pruning

One intuitive way to prune tokens is to leverage the attention scores computed in Transformer layers. Since self-attention mechanisms assign varying importance to different tokens, tokens receiving consistently low attention can be considered less informative and safely removed. Several methods use a threshold-based strategy, where tokens with attention scores below a predefined threshold are pruned at each layer.

3.1.2. Entropy-Based Pruning

Entropy measures the amount of uncertainty or variability in a distribution [26]. In the context of token pruning, token representations with low entropy (i.e., those that do not change significantly across layers) are considered less important and can be removed [27]. This method assumes that tokens with stable representations contribute minimally to the model's final output.

3.1.3. Fixed-Length Pruning

A simpler approach involves reducing sequence length to a fixed number of tokens, either by truncation or by removing tokens based on predefined heuristics such as stop words or punctuation marks. While effective in reducing computational costs, this method lacks adaptability and may result in performance degradation for tasks requiring longer context windows [28].

3.2. Learning-Based Token Pruning

Unlike rule-based methods, learning-based approaches use trainable mechanisms to decide which tokens to prune. These methods typically rely on reinforcement learning, gating mechanisms, or auxiliary networks to dynamically select tokens [29].

3.2.1. Reinforcement Learning-Based Pruning

Reinforcement learning (RL) has been applied to token pruning by formulating it as a sequential decision-making problem. In this setup, a policy network determines which tokens to keep based on task-specific rewards such as model accuracy and efficiency gains [30]. RL-based pruning enables adaptive token selection but often requires additional training and fine-tuning [31].

3.2.2. Gated Token Pruning

Gated pruning methods introduce lightweight gating mechanisms that learn token importance during training [32]. A gating function, parameterized by a small neural network, is applied to each token representation to decide whether to retain or discard the token. These methods allow for dynamic, input-dependent pruning.

3.2.3. Saliency-Based Pruning

Saliency-based methods compute token importance scores using gradient-based techniques. Similar to saliency maps in computer vision, these methods measure how much removing a token affects the model's output. Tokens with minimal impact are considered redundant and pruned [33].

3.3. Hybrid Token Pruning Approaches

Hybrid token pruning strategies combine multiple techniques to balance efficiency and performance. These approaches often integrate rule-based heuristics with learning-based mechanisms for greater adaptability.

3.3.1. Progressive Token Pruning

Progressive pruning methods gradually reduce the number of tokens over multiple Transformer layers rather than removing a fixed number of tokens at once [34]. This allows for a smoother transition, preserving essential contextual information while improving efficiency [35].

3.3.2. Multi-Stage Pruning

Multi-stage pruning involves applying different pruning criteria at different layers of the Transformer. For example, early layers may use attention-based heuristics, while deeper layers leverage learned importance scores. This hierarchical pruning strategy helps mitigate performance loss by ensuring that only the least useful tokens are removed [36].

3.3.3. Integration with Other Compression Techniques

Token pruning is often used in conjunction with other model compression techniques such as quantization and weight pruning. By combining pruning with quantization-aware training or knowledge distillation, researchers have achieved further efficiency gains while maintaining accuracy.

3.4. Comparison of Token Pruning Strategies

Table 1 provides a comparison of the different token pruning methodologies based on criteria such as computational cost, adaptability, and ease of implementation.

Table 1. Comparison of token pruning strategies.

Method	Computational Cost	Adaptability	Implementation Complexity
Attention-Based	Low	Moderate	Low
Entropy-Based	Low	Low	Low
Fixed-Length	Low	Low	Low
Reinforcement Learning	High	High	High
Gated Pruning	Moderate	High	Moderate
Saliency-Based	High	High	High
Hybrid (Progressive)	Moderate	High	Moderate

3.5. Summary and Insights

Token pruning has emerged as a powerful technique for improving Transformer efficiency by reducing sequence length dynamically [37]. Rule-based methods offer simple and computationally efficient solutions but lack adaptability, while learning-based methods provide greater flexibility at the cost of additional computational overhead [38]. Hybrid approaches balance these trade-offs by integrating multiple pruning strategies [39]. The next section provides an in-depth analysis of the effectiveness of these methods, examining their impact on model accuracy, computational savings, and real-world applicability [40].

4. Effectiveness and Trade-Offs of Token Pruning

Token pruning presents a compelling solution for enhancing the efficiency of Transformer-based models, yet it inherently involves trade-offs between computational savings and task performance [41]. In this section, we analyze the effectiveness of different token pruning approaches, examine their impact on model accuracy, and discuss factors influencing their real-world applicability [42].

4.1. Impact on Model Accuracy

The primary concern when pruning tokens is maintaining model accuracy, particularly for complex NLP tasks. The extent of accuracy degradation depends on several factors, including the pruning method, the pruning rate, and the nature of the task. Key observations from prior research include:

- **Effect on Classification Tasks:** In text classification tasks, token pruning has been found to work well since many input tokens contribute redundantly to the final decision [43]. Studies have shown that models can retain up to 95% of their accuracy while reducing sequence length by 50%.
- **Impact on Sequence Labeling:** For tasks such as named entity recognition (NER) and part-of-speech (POS) tagging, aggressive token pruning may lead to loss of fine-grained information, as every token contributes to the final output [44]. Adaptive pruning methods often perform better in such cases.
- **Challenges in Generative Tasks:** Tasks such as machine translation and text generation are particularly sensitive to token pruning, as the quality of generated text depends on maintaining long-range dependencies. In these cases, softer pruning strategies such as saliency-based approaches are often more effective [45].

4.2. Computational Efficiency Gains

Token pruning significantly reduces the computational cost of Transformer models by decreasing the number of tokens processed in each self-attention layer. The efficiency gains depend on the pruning rate and the depth at which pruning is applied [46–48]. Notable improvements include:

- **Reduction in FLOPs:** Studies have shown that token pruning can reduce floating point operations (FLOPs) by up to 60% while maintaining comparable accuracy [49].
- **Inference Speedup:** Models with token pruning achieve 1.5× to 3× speedups on real-world benchmarks without requiring additional hardware modifications [50].

- **Memory Savings:** Since self-attention layers have quadratic complexity with respect to sequence length, reducing the number of tokens directly decreases memory consumption, making models more feasible for deployment on edge devices.

4.3. Robustness and Generalization

A major challenge in token pruning is ensuring that the learned pruning strategies generalize well across different datasets and tasks. Some key insights include:

- **Task-Specific Adaptability:** Methods like reinforcement learning-based pruning can adapt pruning policies based on task requirements, improving robustness across datasets [51].
- **Sensitivity to Domain Shifts:** Token pruning strategies trained on one dataset may not generalize well to out-of-domain data. Hybrid approaches that combine multiple selection criteria tend to be more resilient to domain shifts.
- **Pruning Stability:** Applying token pruning in earlier layers generally results in more stable performance compared to late-layer pruning, as early layers encode redundant representations that can be safely removed.

4.4. Comparison with Other Compression Techniques

Token pruning is one of several model compression techniques used to enhance the efficiency of large-scale Transformers [52]. Table 2 compares token pruning with weight pruning, quantization, and knowledge distillation.

Table 2. Comparison of token pruning with other compression techniques.

Method	**Accuracy Retention**	**Computational Savings**	**Deployment Complexity**	**Adaptability**
Weight Pruning	Moderate	Moderate	High	Low
Quantization	High	High	Moderate	Moderate
Knowledge Distillation	High	High	High	Low
Token Pruning	Variable	High	Moderate	High

Token pruning offers a unique balance of computational savings and adaptability, making it particularly well-suited for real-time inference and energy-efficient NLP applications.

4.5. Summary and Key Insights

Token pruning has demonstrated significant promise in reducing the computational cost of Transformer models while maintaining task performance [53]. The key takeaways from this analysis are:

- Token pruning can achieve up to 60% computational savings with minimal impact on accuracy for classification and retrieval-based tasks.
- Generative and sequence labeling tasks require more careful pruning strategies to avoid loss of critical information.
- Adaptive and hybrid pruning approaches offer greater generalization and robustness compared to static rule-based methods.
- Token pruning is highly complementary to other model compression techniques and can be integrated into broader efficiency frameworks.

In the next section, we explore real-world applications of token pruning, highlighting its use in NLP, computer vision, and speech processing.

5. Applications of Token Pruning

Token pruning has been successfully applied across various domains, including natural language processing (NLP), computer vision, and speech processing. By dynamically reducing sequence lengths, token pruning enhances efficiency while maintaining task performance. This section explores its applications in different fields, highlighting real-world use cases and potential benefits.

5.1. Natural Language Processing (NLP)

Transformer models have become the backbone of modern NLP, powering tasks such as text classification, question answering, and machine translation. Token pruning enables these models to operate more efficiently, particularly for long-text processing and real-time inference [54].

5.1.1. Text Classification

Text classification models often process input sequences that contain redundant information [55]. Token pruning helps by selectively removing unimportant tokens, reducing computational costs while maintaining high classification accuracy [56]. Methods such as attention-based pruning have been shown to achieve up to a 2× speedup with minimal accuracy loss.

5.1.2. Question Answering (QA)

QA systems, such as those based on BERT and T5, require processing lengthy passages to extract relevant information. Token pruning helps focus on the most informative tokens, reducing the number of computations per query. Adaptive pruning strategies have improved inference speed in QA benchmarks while preserving answer accuracy [57].

5.1.3. Machine Translation

Sequence-to-sequence models for translation must retain important context across long input sequences [58]. Aggressive pruning can degrade translation quality, making soft or adaptive pruning strategies more suitable. Techniques such as saliency-based pruning have been employed to selectively retain tokens critical for generating fluent and accurate translations.

5.2. Computer Vision

Token pruning is increasingly being adopted in vision transformers (ViTs), which process image patches as tokenized inputs. Since many image regions contain redundant information, pruning less informative tokens significantly accelerates vision models.

5.2.1. Image Classification

ViTs split an image into patches and process them as tokens [59]. Studies have shown that dynamically pruning uninformative patches results in substantial computational savings without compromising classification accuracy [60].

5.2.2. Object Detection

Object detection models based on transformers process dense sets of object proposals [61]. Token pruning reduces the number of tokens considered in each stage, leading to improved inference speed while preserving detection quality.

5.2.3. Video Understanding

For video tasks, the input sequences consist of multiple frames, leading to extremely long token sequences [62]. Token pruning selectively removes redundant temporal information, making long-range video analysis more computationally feasible.

5.3. Speech Processing

Speech recognition and processing systems have benefited from token pruning by reducing the sequence length of spectrogram features or intermediate transformer representations.

5.3.1. Automatic Speech Recognition (ASR)

In ASR, self-attention is applied over frame sequences extracted from raw audio [63]. Pruning methods discard less informative frames, improving inference efficiency with minimal impact on word error rates [64].

5.3.2. Speaker Identification

Speaker recognition models process long audio streams to extract speaker embeddings. Token pruning reduces computational costs by discarding redundant segments without affecting speaker verification performance [65].

5.4. Deployment in Real-World Systems

Token pruning has been integrated into various real-world AI systems, demonstrating its practical value in optimizing deep learning models for production environments [66].

5.4.1. Edge AI and Mobile Applications

Deploying large-scale transformers on edge devices and mobile processors is challenging due to limited computational resources [67]. Token pruning has enabled efficient inference on mobile devices for tasks such as real-time speech recognition and text generation [68].

5.4.2. Cloud-Based NLP Services

Token pruning has been incorporated into cloud-based NLP services, reducing operational costs for inference-heavy applications such as chatbot systems and document summarization services [69].

5.4.3. Energy-Efficient AI

Reducing unnecessary computations via token pruning lowers energy consumption, making AI models more sustainable and environmentally friendly [70]. This has implications for deploying AI in low-power settings, such as wearable devices and IoT applications.

5.5. Summary and Key Insights

Token pruning has demonstrated effectiveness in a wide range of applications across NLP, computer vision, and speech processing. Key takeaways include:

- Token pruning significantly accelerates NLP models, particularly for classification, question answering, and retrieval tasks [71].
- Vision transformers benefit from pruning redundant image patches, leading to faster and more efficient object detection and classification [72].
- Speech models use token pruning to reduce audio sequence lengths, improving real-time processing efficiency [73].
- Token pruning is highly beneficial for deploying deep learning models on edge devices, mobile platforms, and cloud environments.

The next section discusses open challenges and future research directions in token pruning, addressing limitations and opportunities for further improvements.

6. Challenges and Future Directions in Token Pruning

Despite its promising benefits, token pruning faces several challenges that limit its widespread adoption in deep learning applications [74]. In this section, we discuss key obstacles and outline potential future directions for research and development in token pruning.

6.1. Challenges in Token Pruning

6.1.1. Accuracy vs. Efficiency Trade-Off

One of the fundamental challenges in token pruning is balancing efficiency gains with minimal loss in model accuracy [75]. While pruning redundant tokens accelerates computation, excessive or poorly designed pruning strategies can degrade task performance. Achieving an optimal trade-off remains an open problem, particularly for complex tasks such as machine translation and generative modeling.

6.1.2. Dynamic vs. Static Pruning

Most pruning techniques fall into two categories: static pruning, which applies fixed rules regardless of input, and dynamic pruning, which adapts pruning decisions based on contextual token importance. While dynamic pruning offers greater flexibility, it introduces additional computational overhead, often requiring auxiliary models or gating mechanisms. Balancing adaptability with computational efficiency remains an active research challenge [76].

6.1.3. Pruning Granularity

Token pruning can be applied at different levels of granularity, including word-level, subword-level, or feature-level pruning [77]. While coarse-grained pruning is computationally efficient, fine-grained pruning can preserve more information [78]. Finding the optimal granularity level is still an open question.

6.1.4. Generalization Across Tasks and Domains

Pruning strategies trained on a specific dataset may not generalize well to different tasks or domains [79]. For instance, a pruning policy optimized for text classification may not work effectively for sequence labeling tasks [80]. Designing task-agnostic or adaptive pruning mechanisms that generalize across multiple domains is a crucial research direction.

6.1.5. Compatibility with Other Efficiency Techniques

Token pruning is often used alongside other efficiency techniques such as quantization, knowledge distillation, and weight pruning [81]. However, combining these methods in a synergistic manner without compounding accuracy losses is non-trivial. Future research should focus on designing integrated compression frameworks that maximize efficiency while preserving performance [82].

6.1.6. Robustness to Adversarial Attacks

Recent studies suggest that pruning techniques may introduce vulnerabilities to adversarial attacks, where slight modifications to input data lead to unexpected performance degradation [49]. Ensuring robustness against such adversarial manipulations is an emerging concern in token pruning research.

6.2. Future Research Directions

6.2.1. Neural Architecture Search for Token Pruning

Neural architecture search (NAS) has shown promise in designing efficient deep learning models. Applying NAS to automatically discover optimal pruning policies could lead to more effective and adaptive token pruning strategies.

6.2.2. Self-Supervised Learning for Token Importance Estimation

Self-supervised learning has demonstrated the ability to learn meaningful token representations without labeled data. Future work could explore self-supervised techniques to predict token importance dynamically, reducing the need for task-specific fine-tuning in pruning methods.

6.2.3. Multimodal Token Pruning

With the rise of multimodal models that process text, images, and audio simultaneously, token pruning methods must extend beyond single-modal data [83]. Research into cross-modal pruning strategies could further enhance efficiency in multimodal transformers [84].

6.2.4. Hardware-Aware Pruning Strategies

Most existing pruning techniques focus on reducing theoretical computational complexity rather than optimizing for real-world hardware constraints. Future research should explore pruning methods that are optimized for specific hardware architectures, such as GPUs, TPUs, and edge AI processors, ensuring that pruning translates into real-world speedup [85].

6.2.5. Energy-Efficient and Green AI Pruning

With growing concerns about AI's carbon footprint, energy-efficient AI research is gaining traction. Token pruning can play a critical role in reducing the environmental impact of large-scale models. Future work should explore pruning strategies designed explicitly for sustainability, optimizing energy consumption in both training and inference.

6.3. Summary and Key Takeaways

Token pruning has emerged as a crucial technique for enhancing the efficiency of Transformer models, but several challenges remain to be addressed. Key takeaways from this discussion include:

- The trade-off between efficiency and accuracy remains a core challenge in token pruning research [86].
- Dynamic pruning methods offer adaptability but introduce additional computational complexity.
- Generalization across tasks and domains is a major concern, requiring more flexible pruning mechanisms.
- Future research should explore NAS-based pruning, self-supervised token importance estimation, and multimodal pruning strategies.
- Hardware-aware pruning and energy-efficient AI are promising areas for real-world deployment of token pruning techniques.

The next section concludes the survey by summarizing the key contributions of token pruning research and outlining its implications for the future of deep learning [87].

7. Conclusion

Token pruning has emerged as a powerful technique for improving the efficiency of Transformer-based models by dynamically reducing sequence lengths while maintaining task performance. This survey has provided a comprehensive overview of token pruning, covering its methodologies, effectiveness, real-world applications, challenges, and future research directions [88].

7.1. Key Contributions of Token Pruning

Through a detailed examination of existing pruning techniques and their applications, we highlight the following key contributions of token pruning to deep learning:

- **Efficiency Gains:** Token pruning significantly reduces computational overhead, leading to faster inference times and lower memory consumption, making Transformer models more scalable for real-world deployment.
- **Task-Specific Adaptability:** Various pruning strategies have been developed to cater to different NLP, vision, and speech processing tasks, demonstrating the versatility of token pruning across multiple domains.
- **Integration with Other Efficiency Methods:** Token pruning complements other model compression techniques, such as weight pruning, quantization, and knowledge distillation, enabling holistic model optimization.
- **Real-World Impact:** Token pruning has been successfully deployed in edge computing, cloud-based AI services, and mobile applications, proving its practical value in reducing latency and energy consumption [89].

7.2. Challenges and Open Questions

Despite its promising advantages, token pruning is still an evolving research area with several challenges to address:

- The accuracy-efficiency trade-off remains a crucial factor, as excessive pruning may degrade model performance.
- Dynamic pruning techniques introduce additional computational complexity, necessitating more efficient selection mechanisms [90].
- Ensuring robustness and generalization across different datasets and domains is an ongoing challenge in pruning research [91].
- The integration of token pruning with hardware-aware optimizations is an important future direction for maximizing real-world performance gains.

7.3. Future Outlook

Looking ahead, we anticipate several promising directions in token pruning research:

- Advances in self-supervised learning and neural architecture search may lead to more intelligent and adaptive token pruning strategies.
- Multimodal pruning techniques can enable efficient processing in models that handle text, images, and speech simultaneously.
- Green AI initiatives will likely drive research toward more energy-efficient token pruning methods, reducing the environmental impact of large-scale deep learning models.
- Hardware-aware token pruning methods tailored for specialized accelerators (e.g., GPUs, TPUs, edge processors) will enhance the deployability of pruned models in real-world applications.

7.4. Final Remarks

As Transformer models continue to scale in size and complexity, token pruning offers a crucial mechanism for ensuring their efficiency and accessibility. By addressing current challenges and exploring new research directions, token pruning has the potential to reshape the landscape of deep learning, enabling powerful AI systems that are both high-performing and computationally efficient.

We hope this survey serves as a valuable resource for researchers and practitioners interested in token pruning, fostering further innovation in efficient deep learning architectures.

References

1. Shixing Yu, Tianlong Chen, Jiayi Shen, Huan Yuan, Jianchao Tan, Sen Yang, Ji Liu, and Zhangyang Wang. Unified visual transformer compression. *ArXiv*, abs/2203.08243, 2022.
2. Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/goyal20a.html.
3. Chris J.C. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical report, June 2010. URL https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/.
4. Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3916–3925, 2022.
5. Sucheng Ren, Zhengqi Gao, Tianyu Hua, Zihui Xue, Yonglong Tian, Shengfeng He, and Hang Zhao. Co-advise: Cross inductive bias distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 16773–16782, 2022.
6. *Learning Deep Structured Semantic Models for Web Search using Clickthrough Data*, October 2013. ACM International Conference on Information and Knowledge Management (CIKM). URL https://www.microsoft.com/en-us/research/publication/learning-deep-structured-semantic-models-for-web-search-using-clickthrough-data/.
7. Kavindu Chamith Hans Thisanke, Chamli Deshan. Semantic segmentation using vision transformers: A survey. *arXiv preprint arXiv:2305.03273*, 2023.

8. Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.
9. Ellen M. Voorhees and Angela Ellis, editors. *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*, volume 1250 of *NIST Special Publication*, 2019. National Institute of Standards and Technology (NIST). URL https://trec.nist.gov/pubs/trec28/trec2019.html.
10. Fedor Moiseev Elena Voita, David Talbot. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.
11. Yassine Zniyed, Thanh Phuong Nguyen, et al. Efficient tensor decomposition-based filter pruning. *Neural Networks*, 178:106393, 2024.
12. Bohan Zhuang, Jing Liu, Zizheng Pan, Haoyu He, Yuetian Weng, and Chunhua Shen. A survey on efficient training of transformers. pages 6823–6831, 08 2023. doi: 10.24963/ijcai.2023/764.
13. Yury A. Malkov and D. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:824–836, 2020.
14. Nico Messikommer Yifei Liu, Mathias Gehrig. Revisiting token pruning for object detection and instance segmentation. *arXiv preprint arXiv:2306.07050*, 2023.
15. Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.
16. Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793, 2020.
17. Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7838–7847, 2021.
18. Qiming Zhang Yufei Xu, Jing Zhang. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*, 2022.
19. Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022.
20. Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.
21. Sebastian Hofstätter and Allan Hanbury. Let's measure run time! extending the ir replicability infrastructure to include performance aspects. *SIGIR Open-Source IR Replicability Challenge (OSIRRC)*, 2019.
22. Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 668–685. Springer, 2022.
23. Yuxin Fang, Bencheng Liao, Xinggang Wang, and Fang. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34: 26183–26197, 2021.
24. Wenliang Zhao Yongming Rao, Zuyan Liu. Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. *arXiv preprint arXiv:2207.01580*, 2022.
25. Joel Mackenzie, Zhuyun Dai, Luke Gallagher, and Jamie Callan. Efficiency implications of term weighting for passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1821–1824, 2020.
26. Xianing Chen, Qiong Cao, Yujie Zhong, Jing Zhang, Shenghua Gao, and Dacheng Tao. Dearkd: data-efficient early knowledge distillation for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12052–12062, 2022.
27. Yixing Fan, Jiafeng Guo, Yanyan Lan, Jun Xu, Chengxiang Zhai, and Xueqi Cheng. Modeling diverse relevance patterns in ad-hoc retrieval. *CoRR*, abs/1805.05737, 2018. URL http://arxiv.org/abs/1805.05737.
28. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.
29. Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. Complementing lexical retrieval with semantic residual embedding, 2020.
30. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11): 613–620, November 1975. ISSN 0001-0782. doi: 10.1145/361219.361220. URL http://doi.acm.org/10.1145/361219.361220.

31. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, 2012. URL http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.
32. Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2017.
33. Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.
34. Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. Prop: Pre-training with representative words prediction for ad-hoc retrieval, 2020.
35. Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005. ISBN 0262220733.
36. Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
37. Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *Computer Vision – ECCV 2022 Workshops*, pages 35–49, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-25082-8.
38. Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval, 2020.
39. Shin-Jae Lee, Minsoo Jeon, Dongseung Kim, and Andrew Sohn. Partitioned parallel radix sort. *Journal of Parallel and Distributed Computing*, 62(4):656–668, 2002.
40. Zhiwei Hao, Jianyuan Guo, Ding Jia, Kai Han, Yehui Tang, Chao Zhang, Han Hu, and Yunhe Wang. Learning efficient vision transformers via fine-grained manifold distillation. *Advances in Neural Information Processing Systems*, 35:9164–9175, 2022.
41. Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. Distilling dense representations for ranking using tightly-coupled teachers, 2020.
42. Ross Girshick Yanghao Li, Hanzi Mao. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022.
43. Yukun Zheng, Zhen Fan, Yiqun Liu, Cheng Luo, Min Zhang, and Shaoping Ma. Sogou-qcl: A new dataset with click relevance label. In *The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '18, pages 1117–1120, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5657-2. doi: 10.1145/3209978.3210092. URL http://doi.acm.org/10.1145/3209978.3210092.
44. Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.
45. Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. Pseudo-relevance feedback for multiple representation dense retrieval. In Faegheh Hasibi, Yi Fang, and Akiko Aizawa, editors, *ICTIR '21*, pages 297–306. ACM, 2021. doi: 10.1145/3471158.3472250. URL https://doi.org/10.1145/3471158.3472250.
46. Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.
47. Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.
48. Yassine Zniyed, Thanh Phuong Nguyen, et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
49. Kenton Lee J Devlin, M Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
50. Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017.
51. Bhaskar Mitra and Nick Craswell. An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval*, pages 1–117, April 2018. URL https://www.microsoft.com/en-us/research/publication/introduction-neural-information-retrieval/.
52. Artem Babenko and Victor Lempitsky. The inverted multi-index. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1247–1260, 2014.

53. Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *CoRR*, abs/1904.08375, 2019. URL http://arxiv.org/abs/1904.08375.
54. Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019.
55. Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with bert, 2019.
56. Daniël Rennings, Felipe Moraes, and Claudia Hauff. An Axiomatic Approach to Diagnosing Neural IR Models. In Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra, editors, *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 489–503, Cham, 2019. Springer International Publishing. ISBN 978-3-030-15712-8. doi: 10/ggcmnb. ZSCC: NoCitationData[s0].
57. Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NIPS*, 2014.
58. Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. *Click Models for Web Search*. Morgan & Claypool, 2015. ISBN 9781627056489. doi: 10.2200/S00654ED1V01Y201507ICR043.
59. Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.
60. Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.
61. Hang Li. *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers, 2011. ISBN 1608457079, 9781608457076.
62. Chong Yu, Tao Chen, Zhongxue Gan, and Jiayuan Fan. Boost vision transformer with gpu-friendly sparsity and quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22658–22668, 2023.
63. Paul Pu Liang, Manzil Zaheer, Yuan Wang, and Amr Ahmed. Anchor & transform: Learning sparse embeddings for large vocabularies, 2020.
64. Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401075. URL https://doi.org/10.1145/3397271.3401075.
65. Nils Reimers and Iryna Gurevych. The curse of dense low-dimensional information retrieval for large index sizes, 2020.
66. Zhengkai Tu, Wei Yang, Zihang Fu, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. Approximate nearest neighbor search and lightweight dense vector reranking in multi-stage retrieval architectures. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, pages 97–100, 2020.
67. Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17302–17313, 2023.
68. Tharun Medini, Beidi Chen, and Anshumali Shrivastava. {SOLAR}: Sparse orthogonal learned and random embeddings. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=fw-BHZ1KjxJ.
69. Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021.
70. N Goyal Y Liu, M Ott. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
71. Massih-Reza Amini and Gaussier Eric. *Recherche d'Information - applications, modèles et algorithmes*. Algorithmes. Eyrolles, April 2013. URL https://hal.archives-ouvertes.fr/hal-00881257. I-XIX, 1-233.
72. Luyu Gao and Jamie Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval, 2021.

73. Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023.
74. Francois Chollet. *Deep Learning with Python*. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2017. ISBN 1617294438, 9781617294433.
75. S. Robertson. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.
76. Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20321–20330, 2023.
77. Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlnC1rKPB.
78. Weijun Hong, Guilin Li, Weinan Zhang, Ruiming Tang, Yunhe Wang, Zhenguo Li, and Yong Yu. Dropnas: Grouped operation dropout for differentiable architecture search. *arXiv preprint arXiv:2201.11679*, 2022.
79. Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022.
80. Grace Chu Andrew Howard, Mark Sandler. Searching for mobilenetv3. *arXiv preprint arXiv:1905.02244*, 2019.
81. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
82. Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.
83. Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pages 191–207. Springer, 2022.
84. Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.
85. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.
86. Yingqi Qu Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering, 2020.
87. Leonid Boytsov and Eric Nyberg. Flexible retrieval with NMSLIB and FlexNeuART. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 32–43, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlposs-1.6. URL https://aclanthology.org/2020.nlposs-1.6.
88. Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states, 2019.
89. Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, October 2002. ISSN 1046-8188. doi: 10.1145/582415.582416. URL http://doi.acm.org/10.1145/582415.582416.
90. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
91. Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.