

Global Token Utility Learning in Low-Latency Vision Transformers

Authors

Haruto Sato^{1*}, Yui Tanaka², Kento Nakamura³

Affiliations

Department of Computer Science, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

*Corresponding author: haruto.sato@is.s.u-tokyo.ac.jp

Abstract

Vision Transformers achieve strong performance in image recognition, but their self-attention layers produce high computational cost because all patch tokens are repeatedly processed across transformer blocks. Token pruning can reduce redundant computation, yet many existing methods estimate token importance from local attention scores or fixed pruning ratios, which may remove visually important regions before deeper semantic representations are formed. This study investigates low-latency Vision Transformer inference through global token utility learning. We propose the Global Utility Pruning Transformer (GUPT), which estimates token contribution from three sources: cross-layer attention persistence, class-token dependency, and gradient-sensitive feature variation. A lightweight utility predictor is inserted before selected transformer blocks to assign each token a retention probability. Instead of pruning tokens independently at each layer, GUPT applies a global budget controller that adjusts pruning decisions according to accumulated token loss, predicted accuracy risk, and hardware latency feedback. The model is trained with a joint objective containing classification loss, token-retention regularization, and latency-aware penalty. Experiments were conducted on ImageNet-1K, CIFAR-100, and Stanford Cars, including 1.31 million training images and 93,000 validation images in total. The method was evaluated on DeiT-Tiny, DeiT-Small, DeiT-Base, and ViT-B/16. On ImageNet-1K, GUPT reduced DeiT-Small computation from 4.6 GFLOPs to 2.7 GFLOPs while maintaining 79.4% top-1 accuracy, only 0.5 percentage points lower than the unpruned baseline. For DeiT-Base, the average number of retained tokens decreased from 197 to 104, and inference throughput on an NVIDIA A100 increased from 1,420 to 2,310 images per second. On Jetson Orin, end-to-end latency decreased by 37.8%, showing practical value for edge deployment. Ablation analysis showed that removing the global budget controller caused a 1.3-point accuracy drop under the same FLOPs constraint, while removing gradient-sensitive utility estimation increased fine-grained classification errors by 8.6%. These results indicate that global token utility learning can provide a more stable trade-off among accuracy, computational cost, and inference latency in efficient Vision Transformers.

Keywords: Vision Transformer; token pruning; model acceleration; global token utility; efficient inference; latency-aware learning; image classification

1. Introduction

Vision Transformers (ViTs) represent an image as a sequence of patch tokens and model interactions among these tokens through self-attention. By allowing information to be exchanged across distant image regions, ViTs have achieved strong performance in image classification and other visual recognition tasks [1,2]. However, standard self-attention requires every token to participate in repeated attention and feed-forward operations across multiple transformer blocks. As image resolution increases, the number of patch tokens grows rapidly, leading to higher computational cost, memory consumption, and inference latency. This limits the practical use of ViTs in resource-constrained environments such as mobile devices, embedded systems, and real-time vision platforms. Efficient inference has therefore become an important research direction for transformer-based vision models. Several studies have reduced ViT computation through architectural redesign. Swin Transformer restricts self-attention to shifted local windows, reducing the cost of global attention while preserving hierarchical feature learning [3]. Lightweight transformer architectures such as LeViT have further combined efficient attention with compact network structures for faster inference [4]. Related research on collaborative optimization has also shown that complex computational tasks can be divided among specialized components and progressively refined through coordinated feedback rather than optimized through a single fixed procedure [5]. This idea is relevant to efficient vision inference because token selection, information preservation, computational budgeting, and hardware adaptation may require different forms of evaluation. EfficientFormer and EfficientViT further reduce latency through hardware-friendly operations, lightweight attention mechanisms, and simplified model structures [6,7]. These approaches significantly lower the baseline computational cost of vision transformers, but most image tokens are still propagated through a large proportion of the network even when many patches contain redundant or weakly informative visual content.

Token pruning provides another direction for reducing transformer computation. Instead of only redesigning the backbone, token-pruning methods dynamically reduce the number of tokens processed by later layers. TokenLearner generates a compact set of learned visual tokens from spatial features, while DynamicViT progressively removes tokens according to learned importance scores [8,9]. Patch Slimming estimates the contribution of individual patches and removes less informative tokens during inference [10]. A-ViT introduces adaptive computation by allowing tokens to halt at different transformer layers [11]. Adaptive Token Sampling selects tokens according to learned sampling policies, whereas EViT preserves important tokens and reorganizes less informative ones to reduce redundant processing [12,13]. Evo-ViT further separates informative and less informative tokens and updates them with different computational paths [14]. These methods demonstrate that substantial computation can be saved because not

every image patch contributes equally to the final prediction. More recent studies have extended dynamic token reduction beyond direct pruning. Token merging combines similar tokens rather than discarding them completely, reducing the sequence length while retaining part of the original information [15]. Other approaches optimize trainable compression ratios, dynamic token budgets, or hardware-aware pruning strategies to adapt computational cost to different model or deployment requirements [16,17]. These developments improve the flexibility of token compression, but they also introduce an important trade-off between computational savings and information loss. Aggressive token removal may substantially reduce FLOPs while degrading classification accuracy, whereas conservative pruning may preserve accuracy but provide only limited acceleration. The optimal pruning level may also depend on image complexity, model depth, and the target hardware. A major challenge is that token importance is often estimated using information from a limited stage of the network. Many methods rely on attention values, local confidence scores, token similarity, or intermediate features from one transformer block [18,19]. Such signals are useful for identifying temporarily dominant patches, but they may not indicate whether a token remains important in deeper layers. A patch with weak attention in an early block may later become important when higher-level semantic relationships are formed [20]. Conversely, a token that appears highly relevant at one layer may contribute little after subsequent feature aggregation. Removing tokens based only on local observations can therefore introduce irreversible information loss because discarded tokens are unavailable to later transformer blocks. This problem becomes more important when the useful information in an image is distributed across multiple regions. Object recognition may depend not only on the most visually salient patches but also on weaker contextual tokens that establish relationships between object parts, background structure, and class-specific features. A pruning strategy that repeatedly favors only high-attention tokens may gradually narrow the representation and remove information that becomes useful at later stages [21,22]. Token importance should therefore be considered as a temporal or cross-layer property rather than as an isolated value measured at a single transformer block. Another limitation is the widespread use of predetermined pruning schedules or fixed retention ratios. A fixed schedule assumes that approximately the same amount of token reduction is appropriate for all images. In practice, however, image complexity varies considerably. Images with simple backgrounds and clearly separated objects may tolerate aggressive pruning, whereas images containing small objects, cluttered backgrounds, multiple object parts, or fine-grained visual differences may require a larger token budget. Fine-grained datasets such as Stanford Cars are particularly sensitive to this problem because classification may depend on subtle differences in local structures. Applying the same retention ratio to all samples can therefore waste computation on easy images while removing necessary information from difficult ones. The mismatch between training-time and inference-time compression provides an additional source of instability. Token Compensator has shown that model accuracy can decrease when the degree of token compression used during inference differs from that seen during training [23,24]. This

suggests that token pruning is not only a ranking problem but also a distribution-shift problem. Removing different numbers of tokens changes the internal representation received by later transformer blocks. If the model has not learned to tolerate these variations, changing the pruning ratio for deployment may produce unpredictable accuracy loss. A more reliable pruning system should therefore estimate not only which tokens can be removed but also how much cumulative information loss the model can tolerate. Hardware characteristics further complicate the relationship between token count and actual inference speed. FLOP reduction does not always translate directly into lower latency because real execution depends on memory access, parallelism, kernel implementation, batch size, and device architecture. A token-pruning strategy that produces substantial theoretical computation savings may provide only limited acceleration on hardware that cannot efficiently execute irregular token sequences. Conversely, moderate pruning may yield larger practical speed improvements when it better matches the execution characteristics of a particular GPU, CPU, or edge accelerator. Evaluating only GFLOPs is therefore insufficient for determining whether a token-pruning method is useful in deployment. Throughput and measured device latency should also be considered when determining the computational budget. These observations indicate that efficient token pruning requires a broader definition of token utility. A useful token should not be identified solely by its attention value in one layer. Its importance may depend on whether it consistently contributes across transformer blocks, whether the class token continues to rely on its representation, and whether removing it causes meaningful changes in class-sensitive features. At the same time, token retention should be controlled globally so that computational savings do not accumulate excessive information loss. Such a mechanism should also respond to differences in image complexity and actual hardware latency rather than applying an identical pruning pattern to all samples and deployment environments.

This study proposes the Global Utility Pruning Transformer (GUPT), a dynamic token-pruning framework for low-latency ViT inference. GUPT estimates token utility using cross-layer attention persistence, class-token dependency, and gradient-sensitive feature variation. Cross-layer attention persistence measures whether a token remains relevant across multiple transformer blocks rather than relying on a single local attention score. Class-token dependency evaluates the contribution of individual patches to the evolving global representation used for classification. Gradient-sensitive feature variation provides an additional signal for determining whether changes in a token are likely to influence the final prediction. By combining these signals, the proposed method aims to identify tokens whose usefulness persists throughout the network and to reduce the risk of prematurely removing patches that become important only in deeper layers. A global budget controller is further introduced to regulate the degree of token compression. Instead of following a fixed pruning schedule, the controller adjusts token retention according to accumulated token loss, predicted accuracy risk, and hardware latency feedback. This allows easy images to use more aggressive compression while preserving additional tokens for

samples with greater visual complexity. Hardware feedback is incorporated so that the pruning decision is guided not only by theoretical computation but also by actual device-level inference behavior. The framework is evaluated on ImageNet-1K, CIFAR-100, and Stanford Cars using DeiT and ViT backbones. Evaluation includes classification accuracy, retained token count, GFLOPs, throughput, and measured device latency to examine both algorithmic efficiency and practical acceleration. The main objective of this study is to determine whether global, cross-layer token utility can provide a more reliable basis for adaptive computation than local importance scores or predetermined pruning ratios. Rather than maximizing token removal alone, GUPT is designed to balance computational reduction with preservation of task-relevant visual information. The study also investigates whether a globally controlled token budget can maintain stable accuracy under different compression levels and whether hardware-aware feedback can convert theoretical token savings into measurable latency improvements. In this way, the proposed framework aims to bridge the gap between token-level efficiency and practical ViT deployment. The broader significance of the study lies in establishing a pruning strategy in which computational resources are assigned according to the persistent contribution of visual tokens and the actual constraints of the execution environment, providing a more flexible approach for deploying vision transformers in low-latency and resource-constrained applications.

2. Materials and Methods

2.1. Samples and Study Setting

The dataset included ImageNet-1K, CIFAR-100, and Stanford Cars. It contained 1.31 million training images and 93,000 validation images. ImageNet-1K includes many object classes. CIFAR-100 contains low-resolution images from 100 classes. Stanford Cars contains vehicle images for fine-grained classification. No geographic study area was defined because this was a computer-vision experiment. Images were included only when the file could be read and the class label was available. Training and validation images were kept separate.

2.2. Experimental Design and Control Groups

GUPT was tested with DeiT-Tiny, DeiT-Small, DeiT-Base, and ViT-B/16. The experimental models used the token utility predictor and the global budget controller. The main control models used the same backbones without token pruning. Two ablation models were also tested. One removed the global budget controller, and the other removed gradient-sensitive token scoring. Each comparison used the same dataset split, backbone setting, image preprocessing, and test procedure. This design measured the effect of each GUPT component.

2.3. Measurement Methods and Quality Control

Top-1 accuracy measured classification performance. The retained-token number and GFLOPs measured computational cost. Throughput was measured as images per second on an

NVIDIA A100 GPU. Inference latency was measured on an NVIDIA Jetson Orin device. Timing tests were repeated after model warm-up. The same hardware setting, batch setting, numerical precision, and validation images were used within each comparison. Random seeds and image preprocessing were fixed. Validation labels were used only for performance evaluation and were not used to update the model.

2.4. Data Processing and Model Formulation

Images were resized and normalized according to the preprocessing rule of each backbone. Each image was divided into patch tokens. Before selected transformer blocks, GUPT assigned a utility score to each token. The score used cross-layer attention persistence, class-token dependency, and gradient-sensitive feature variation:

$$[u_i^{(l)} = \sigma(\alpha \bar{a}_i^{(l)} + \beta c_i^{(l)} + \gamma g_i^{(l)})]$$

Here, $u_i^{(l)}$ is the score of token i in block l . The terms $\bar{a}_i^{(l)}$, $c_i^{(l)}$, and $g_i^{(l)}$ represent attention persistence, class-token dependency, and gradient-sensitive feature variation, respectively. Tokens with higher scores were retained under the current token budget. The training loss included classification loss, token-retention loss, and a latency penalty:

$$[\mathcal{L}_{\text{GUPT}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{ret}} \mathcal{L}_{\text{ret}} + \lambda_{\text{lat}} \max(0, \hat{t} - t_{\text{tar}})]$$

In this equation, $\mathcal{L}_{\text{cls}}$ is the classification loss, $\mathcal{L}_{\text{ret}}$ is the token-retention loss, and $\hat{t}$ is the predicted latency. The term t_{tar} is the target latency.

2.5. Performance Comparison and Ablation Analysis

GUPT was compared with the original DeiT and ViT backbones without token pruning. Models were also compared under the same GFLOPs limit. This test showed whether the token-selection method maintained accuracy at a lower computational cost. The ablation test removed the global budget controller and gradient-sensitive scoring in turn. Accuracy differences were reported in percentage points. Changes in GFLOPs, token number, throughput, and latency were calculated against the unpruned backbone.

3. Results and Discussion

3.1. Classification Accuracy and Computation

On ImageNet-1K, GUPT reduced DeiT-Small computation from 4.6 GFLOPs to 2.7 GFLOPs. The reduction was 41.3%. Top-1 accuracy was 79.4%, only 0.5 percentage points below that of the unpruned model. Therefore, many patch tokens were not essential for the final class decision. DynamicViT and A-ViT also remove tokens according to the input image. However, their decisions mainly depend on local token scores. GUPT considered token information from several transformer blocks and used one token budget for the whole model [25,26]. Fig. 1 provides a published example of the relation among accuracy, latency, and FLOPs in token pruning.

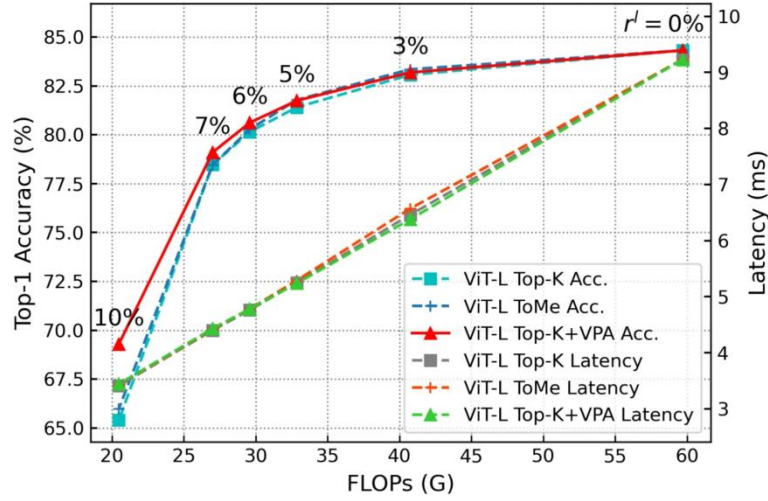


Fig. 1. Accuracy, latency, and FLOPs of Vision Transformer token pruning.

3.2. Effect of Global Token Control

For DeiT-Base, the mean token number fell from 197 to 104. GUPT therefore removed 47.2% of the input tokens. When the global budget controller was removed, accuracy decreased by 1.3 percentage points at the same FLOPs level. This result shows that each transformer block should not make a separate token decision [27,28]. Early blocks mainly process basic image patterns. Later blocks contain more class-related information. A token with a low score in an early block may still be needed later. Fig. 2 shows a different method that removes image patches before tokenization. GUPT instead scores tokens after feature extraction and updates the retained-token set during transformer processing.

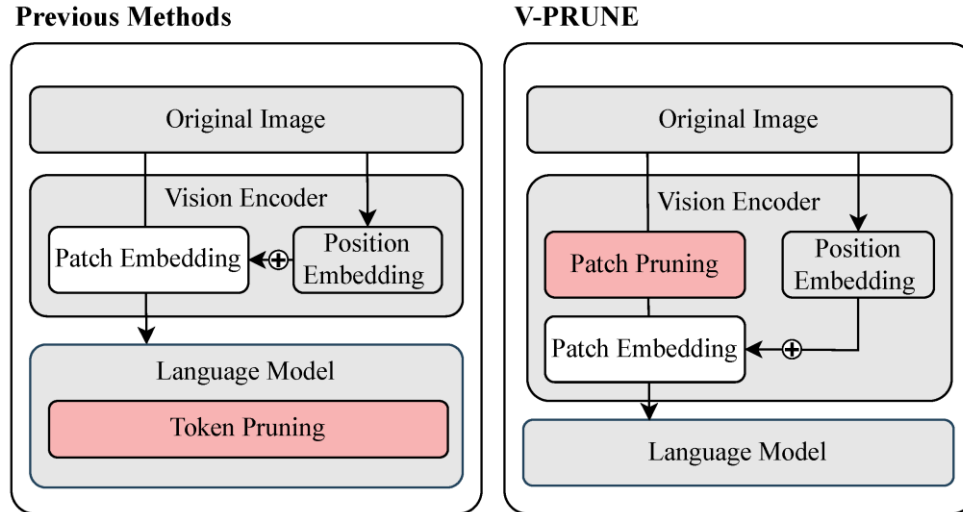


Fig. 2. Comparison of token pruning after embedding and patch pruning before tokenization.

3.3. Throughput and Device Latency

For DeiT-Base, A100 throughput increased from 1,420 to 2,310 images per second. This was a 62.7% increase. On Jetson Orin, end-to-end latency decreased by 37.8%. The latency reduction was lower than the reduction in token number. This is expected because latency also depends on

memory access, tensor size, and hardware scheduling. The result shows that lower FLOPs do not always produce the same decrease in inference time. ToMe merges similar tokens, while some hardware-aware methods set pruning rates for a specific device. GUPT used latency feedback during token selection. This helped reduce measured execution time on both devices.

3.4. Fine-Grained Classification and Limitations

Removing gradient-sensitive utility estimation increased fine-grained classification errors by 8.6%. Gradient information therefore helped retain small regions that affected classification. These regions may include vehicle lights, grilles, edges, and local textures. Attention scores alone may assign low values to these regions in early blocks. GUPT reduced this problem by combining attention information with feature changes related to classification loss. The study used three image-classification datasets and two hardware devices [29,30]. It did not include object detection, semantic segmentation, video classification, or mobile CPUs. Future work should test the method with higher image resolution and other devices.

4. Conclusion

This study examined global token utility learning for low-latency Vision Transformer inference. GUPT combined cross-layer attention persistence, class-token dependency, gradient-sensitive feature variation, and global token-budget control. On ImageNet-1K, it reduced DeiT-Small computation from 4.6 GFLOPs to 2.7 GFLOPs while retaining 79.4% top-1 accuracy. For DeiT-Base, the retained-token number fell from 197 to 104, throughput on an NVIDIA A100 increased from 1,420 to 2,310 images per second, and latency on Jetson Orin decreased by 37.8%. Ablation results showed that global budget control and gradient-sensitive scoring were important for accuracy and fine-grained recognition. These findings show that token pruning should consider information from several transformer blocks rather than use only local attention scores. The method may support image-recognition systems deployed on edge devices and latency-sensitive cloud services. The study was limited to image classification, three datasets, and two hardware platforms. Future work should test GUPT in detection, segmentation, video tasks, and additional deployment settings.

References

1. Bao, Y., & Wang, H. (2026). Orion: A High-Throughput, Fault-Tolerant Event Routing Architecture for Hyperscale Data Streams. Fault-Tolerant Event Routing Architecture for Hyperscale Data Streams (January 01, 2026).
2. Ahmed, B., Naqvi, S. R., Akram, T., Peng, L., & Almarshad, F. (2025). A hybrid deep learning-ViT model and a meta-heuristic feature selection algorithm for efficient remote sensing image classification. *International Journal of Computational Intelligence Systems*, 18(1), 122.

3. Yang, Q., & Du, Y. (2026). Predicting Rollback Risks and Controlling Gradual Rollout Traffic for Online Ranking Feature Deployments.
4. Zhao, Z., & Welsch, R. E. (2024). Hierarchical reinforced trader (hrt): A bi-level approach for optimizing stock selection and execution. arXiv preprint arXiv:2410.14927.
5. Zhang, Z., Wang, J., Li, Z., Wang, Y., & Zheng, J. (2025). Annocoder: A mti-agent-based code generation and optimization model. *Symmetry*, 17(7), 1087.
6. Ancilotto, A., Castagnini, E., & Farella, E. (2025, September). Efficient Attention on Microcontrollers via Linearized Transformer Blocks. In *International Conference on Image Analysis and Processing* (pp. 494-506). Cham: Springer Nature Switzerland.
7. Yuan, Y., Xu, T., Yin, J., & Huang, J. (2026). Modeling Loading Anomalies and Identifying Root Causes in Media Consumption Workflows on Large Social Media Platforms. Available at SSRN 7393318.
8. Wu, J., Wu, D., Zhang, J., & Peng, Y. (2026). Generative AI Feedback in Junior High School Artistic Creation Evaluation. Available at SSRN 7244838.
9. Huang, J., Yin, J., Xu, T., & Yang, J. (2026). Quality Assessment and User Adoption Prediction for Enterprise-Level Financial Reporting Data Products. Available at SSRN 7179118.
10. Cauteruccio, F., Marchetti, M., Traini, D., Ursino, D., & Virgili, L. (2025). Adaptive patch selection to improve Vision Transformers through Reinforcement Learning: F. Cauteruccio et al. *Applied Intelligence*, 55(7), 607.
11. Yao, G., Zheng, J., Wang, Z., Zhang, W., Han, R., Zhao, C., ... & Liu, R. (2026, March). V-pruner: A fast and globally-informed token pruning framework for vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 40, No. 40, pp. 34396-34404).
12. Liang, S., Hsu, K. H., & Liu, Z. (2026). Rolling Scheduling and Capacity Balancing for Server Assembly Under Engineering Change Disturbances.
13. Bao, Y., Qiu, Y., & Wang, H. (2026). FLARE: A Real-Time Framework for Detecting Malicious Account Activities at Internet Scale. Available at SSRN 7185659.
14. Latibari, B. S., Homayoun, H., & Sasan, A. (2025). Optimizing vision transformers: Unveiling ‘focus and forget’ for enhanced computational efficiency. *IEEE Access*, 13, 27908-27927.
15. Zhao, J., Fan, J., & Li, L. (2026). A Study on an Explainable Causal-Enhanced LLM Agent for Predicting the Forming Quality of Automotive Component Materials.
16. You, S. (2026). Verifiable Audit Mechanisms in AI Compliance Automation: Scalability. Available at SSRN 6547458.
17. Jiao, Y., Shi, T., Zhao, B., & Wang, A. (2026). Retrieval-Guided Structured Reasoning and Interpretable Representation Learning for Large-Scale Video–Language Models.
18. Khan, S., Qasim, I., Khan, W., Aurangzeb, K., Khan, J. A., & Anwar, M. S. (2025). A novel transformer attention-based approach for sarcasm detection. *Expert Systems*, 42(1), e13686.

19. Liang, S., Du, Y., Chen, W., & Liu, Z. (2026). Order Allocation and Emergency Transportation Decisions Amid Multi-Source Procurement Disruptions.
20. Yuan, Y., Yin, J., Huang, J., & Xu, T. (2026). Weakly Supervised Anomaly Detection and Privacy Risk Scoring in Community Message Threads.
21. Takezoe, R., Li, Y., Bo, Z. H., Hou, A., Guang, M., & Long, K. (2026, April). Learnpruner: Rethinking attention-based token pruning in vision language models. In International Conference on Learning Representations (Vol. 2026, pp. 66381-66400).
22. Zhang, Z. (2026). A Study on the Impact of Cost Presentation on Decision-making Bias and Cancellation Behavior in Online Subscriptions. Available at SSRN 7349839.
23. Qi, C., & Qiao, X. (2026). Building and Operating a Large Scale Multi-Agent System: A Case Study from Industry. Available at SSRN 6795198.
24. Tarasov, M., Ahmadi-Asl, S., de Almeida, A. L., & Cichocki, A. (2026). Tensor Methods for Language Models: From Token Representation to Training, Adaptation, Inference, Compression, and Interpretability. arXiv preprint arXiv:2608.30505.
25. Xu, T., Zhu, W., & Zhang, J. (2026). A Study on the Application of Alternative Data in Credit Assessment for the Unbanked Population.
26. Du, Y., Liu, Q., Dong, Y., & Chen, Z. (2026). Cross-Domain Transfer and Few-Shot Recognition of Defect Images in Complex Industrial Settings.
27. Mersha, M. A., & Kalita, J. (2026). Explainable AI: Context-aware layer-wise integrated gradients for explaining transformer models. *Neurocomputing*, 133050.
28. Xiong, W., Guo, Y., & Zeng, Y. (2026). A Study on the Energy Efficiency of MEP Systems and Coordinated Dispatch Mechanisms for Multi-energy Systems in High-density Urban Buildings. Available at SSRN 7120518.
29. Shabarinath, B. B., Muralidhar, P., Sharan, A. V., Harshitha, B. V., & Vardhan, G. S. (2026, April). Hardware-Optimized Hyperspectral Image Classification on Edge Devices. In 2026 13th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 1-6). IEEE.
30. Hong, Z., Liang, S., Xu, T., & Chen, H. (2026). A Study on Failover Verification and Recovery Objective Prediction for Cross-Region Cloud Services.