Adaptive Backbone Selection for Complexity-Aware and Energy-Efficient Visual Inference

Syed Amir Hamza[1] and Alexander Jesser[2]

[1] Forkit Dev, Heilbronn, Germany
sa.hamza@forkit.dev
[2] Institute for Intelligent Cyber-Physical Systems (ICPS), Heilbronn University (HHN), Heilbronn, Germany
alexander.jesser@hs-heilbronn.de

Abstract. Modern deep learning models for computer vision often deploy powerful, monolithic backbones to ensure high accuracy across diverse visual scenes. However, such static inference pipelines overlook the varying complexity of individual inputs, leading to unnecessary computation, increased energy consumption, and reduced efficiency—especially in real-time and resource-constrained environments. In this work, we propose a novel Adaptive Backbone Selection (ABS) framework that dynamically selects the most suitable CNN backbone on a per-image basis, guided by image complexity and optimized through reinforcement learning.
Our method integrates a lightweight complexity analyzer to estimate the edge and texture richness of input images and a policy network trained with a reward function that balances accuracy and inference latency. A memory-efficient Backbone Manager handles dynamic model switching with an LRU caching strategy to minimize GPU overhead. The resulting adaptive pipeline leverages a spectrum of pretrained models—from lightweight MobileNets to high-capacity DenseNets—selecting the minimal model required for each input to maintain accuracy.
We evaluate our approach on the ImageNet validation set and show that it achieves competitive accuracy while reducing average compute through adaptive routing, leading to lower latency and memory usage in our experimental setup. Our framework offers a promising direction for sustainable AI, enabling real-time, resource-efficient vision systems.

Keywords: Adaptive Inference · Dynamic Backbone Selection · Resource-Aware Deep Learning · Complexity-Aware Routing · Reinforcement Learning · Green AI · Sustainable AI · Real-Time Vision Systems · Energy-Efficient Deep Learning · CNN Model Switching · Multi-Backbone Inference

1 Introduction

Deep learning models have achieved state-of-the-art performance, but their growing size and complexity have led to significant computational and environmental

costs. From 2012 to 2018, the computational demand of deep learning increased by a factor of 300,000, leading to substantial energy usage and carbon emissions [1]. In contrast, the human brain—after which these models are loosely inspired—remains orders of magnitude more efficient, raising concerns about the resource sustainability of AI systems.

To address this imbalance, Schwartz et al. [1] proposed the concept of *Green AI*, advocating for efficiency and energy usage to be weighed alongside accuracy. Recent analyses have reinforced this view, with estimates showing that inference accounts for nearly 60% of AI energy consumption in large-scale deployments [2] [3].

Adaptive computation has emerged as a promising strategy to mitigate unnecessary inference cost. Approaches such as BlockDrop [4] dynamically skip layers based on input complexity, while early-exit architectures like BranchyNet [5] and MSDNet [6] allow predictions to be made earlier for simpler inputs.

Building on these ideas, we propose a novel framework that dynamically selects the most appropriate CNN backbone from a pool of pretrained models. Our system leverages a lightweight complexity analyzer and a reinforcement learning-based policy network to match inference effort to input difficulty—achieving a balance between accuracy and energy efficiency.

1.1 Contributions

- A fast complexity analysis module that estimates input difficulty in real-time.
- A policy network that selects the optimal pretrained backbone per input.
- A reinforcement learning setup that optimizes the policy based on accuracy and latency trade-offs.
- Extensive evaluation on ImageNet demonstrating significant reductions in energy usage and inference time with minimal accuracy loss [7].
- Estimated environmental impact, including potential CO_2 savings.
- Ablation studies on complexity estimation and reward balancing.

This work advances sustainable deep learning by tailoring computation to the complexity of each input.

2 Related Work

2.1 Static Models and the Need for Adaptivity

Deep convolutional neural networks (CNNs) such as ResNet [8], DenseNet [9], and EfficientNet [10] have become foundational in computer vision. Despite their success, these models apply a uniform, resource-intensive computation path to all inputs, regardless of complexity. This inefficiency contradicts the principles of Green AI [1], which emphasize energy-efficient computation alongside predictive accuracy. These limitations motivate the development of architectures that can tailor inference cost to task difficulty.

2.2 Conditional Computation for Efficient Inference

Conditional computation aims to resolve this inefficiency through input-aware, dynamic behavior.

Early-Exit Models. Approaches like BranchyNet [5] attach intermediate classifiers that enable early exits for simple inputs. While this reduces computation, it can degrade performance when early exits dominate or are misused.

Dynamic Skipping. Methods like BlockDrop [4] and SkipNet [11] learn data-dependent policies to skip blocks or layers within a network. These techniques offer fine-grained adaptivity but are restricted to a single network architecture and require joint optimization of skip paths and internal representations.

Mixture-of-Experts (MoE). MoE frameworks [12] use gating networks to route inputs to specialized subnetworks (experts). V-MoE [13] extends this idea to vision transformers by inserting conditional experts into transformer blocks. However, these systems often rely on expensive co-training schemes, careful regularization, and custom infrastructure to scale.

2.3 Backbone Selection and Model Routing

Our approach is inspired by the flexibility of these dynamic techniques but introduces key simplifications for real-world applicability. Rather than co-training experts or embedding routing within a single model, we treat each pretrained CNN as a frozen "expert" and perform routing at the model level. This coarse-grained decision process supports heterogeneity and reduces training and engineering overhead.

Compared to DynaSwitch [14], which uses reinforcement learning for model switching across fine-tuned networks, our system avoids backbone retraining and employs a lightweight REINFORCE-trained policy to guide selection. This increases interpretability, improves compatibility with off-the-shelf models, and simplifies deployment.

While Deep Elastic Networks [15] construct a single configurable network for multi-task learning, our method enables plug-and-play inference across existing backbones. This design facilitates Green AI benchmarking and supports transfer learning without architectural modifications.

3 Proposed Methodology

We propose an **Adaptive Backbone Selection (ABS)** framework that dynamically selects the most appropriate CNN backbone per input image to balance accuracy, latency, and energy efficiency. The system is composed of four key components: (1) a complexity analyzer, (2) a policy network, (3) a reward-guided training mechanism, and (4) an optional memory-aware inference manager.

3.1 Overview of the ABS Pipeline

Given an input image $x \in \mathbb{R}^{H \times W \times C}$, ABS performs the following steps:

1. Compute a scalar complexity score $S(x)$ using both structural and confidence-based cues.
2. Use a policy network π_θ to select a backbone b_i from a pool $\mathcal{B} = \{b_1, b_2, \ldots, b_N\}$.
3. Perform classification using the selected backbone to obtain prediction $\hat{y} = b_i(x)$.

The policy network π_θ is trained to optimize a reward function R that balances classification accuracy, computational latency, and routing diversity.

3.2 Complexity Analyzer

The complexity score $S(x)$ quantifies input difficulty and guides policy decisions. It is defined as:

$$S(x) = \alpha \cdot \text{Entropy}(x) + (1 - \alpha) \cdot (1 - \text{Confidence}(x))$$

where:

- **Entropy** is computed using edge density derived from Sobel-filter-based gradient histograms.
- **Confidence** is the highest softmax probability from a lightweight pretrained classifier.
- $\alpha \in [0, 1]$ is a tunable weighting factor; we set $\alpha = 0.6$ based on empirical validation.

This formulation ensures that images with high structural detail or classification ambiguity receive higher complexity scores.

3.3 Policy Network

The policy network π_θ is a lightweight two-layer MLP that maps $S(x)$ to a probability distribution over available backbones:

$$\pi_\theta(b_i \mid S(x)) = \text{Softmax}(W_2 \cdot \text{ReLU}(W_1 S(x) + b_1) + b_2)$$

where W_1, W_2 and b_1, b_2 are trainable parameters. The selected backbone b_{i^*} is:

$$i^* = \arg\max_i \pi_\theta(b_i \mid S(x))$$

3.4 Reward-Guided Policy Optimization

The policy is trained using the REINFORCE algorithm [23], which updates parameters to maximize the expected reward:

$$\nabla_\theta J(\theta) = \mathbb{E}_{b_i \sim \pi_\theta}[R \cdot \nabla_\theta \log \pi_\theta(b_i \mid S(x))]$$

The reward function R encourages efficient and diverse inference:

$$R = \lambda \cdot \mathbb{1}[\hat{y} = y] + (1 - \lambda)(1 - L_i) + \gamma \cdot \mathbb{1}[b_i \neq b_{t-1}]$$

where:

- $\hat{y}$ is the predicted label, y the ground truth.
- L_i is the normalized latency of backbone b_i.
- $\lambda \in [0, 1]$ balances accuracy and efficiency; γ promotes routing diversity.

3.5 Training and Inference Procedure

The ABS framework is trained in two phases:

1. **Warm-up Phase:** For K epochs, backbones are sampled randomly to encourage exploration.
2. **Policy Optimization:** The policy π_θ is updated using REINFORCE based on rewards.

During inference, the complexity score $S(x)$ is computed, the policy selects the optimal backbone, and classification is performed using the frozen pretrained model.

3.6 Memory-Aware Routing (Optional)

To enhance runtime efficiency in deployment, we optionally employ an **LRU (Least Recently Used)** model caching strategy. This minimizes GPU memory swaps when consecutive inputs require different backbones, improving throughput in streaming scenarios.

4 Experimental Evaluation and Results

This section validates our central hypothesis: that an adaptive framework can intelligently manage computational resources to achieve a superior balance of accuracy and efficiency. All experiments were conducted on the ImageNet 2012 Validation set, comprising 50,000 images across 1,000 classes.

4.1 Performance Trade-Off and Visualizations

In the remainder of the section, we use internal visualizations to explain the behavior of the ABS framework—how it routes inputs, allocates models, and converges over time.

Figure 1 presents the distribution of accuracy versus inference latency across backbones within the ABS system. The adaptive system operates near this level with significantly reduced latency by intelligently selecting simpler models for less complex inputs. These results highlight ABS's Pareto-efficient behavior.

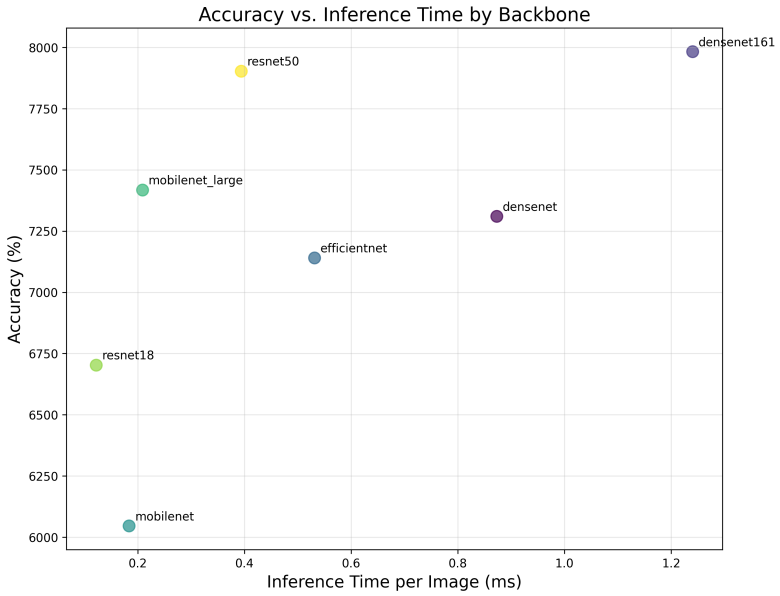

Fig. 1. Performance of individual backbones inside ABS. The system leverages the trade-off space to deliver near-optimal accuracy with reduced latency.

Figure 2 illustrates the memory footprint across inference runs. Since ABS dynamically selects lighter models for simpler inputs, average GPU memory usage is significantly reduced—making it ideal for edge deployment and energy-sensitive environments.

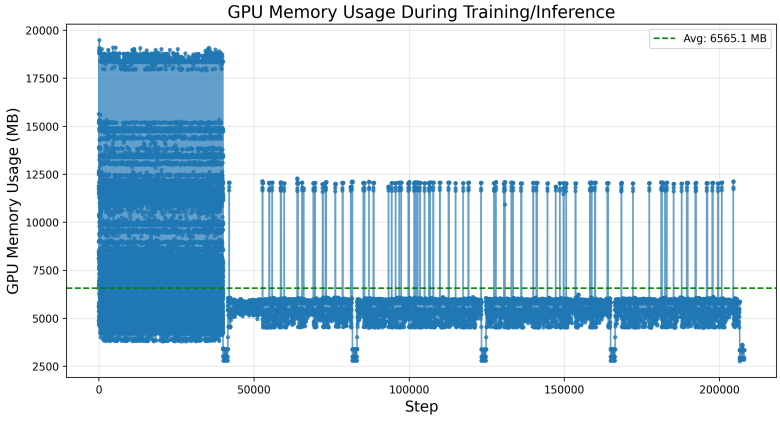

Fig. 2. ABS reduces GPU memory footprint by routing low-complexity inputs to efficient backbones.

The routing behavior learned by the policy is shown in Figure 3. Lightweight backbones such as MobileNetV2 and ResNet18 are frequently selected for common cases, while deeper models like DenseNet and EfficientNet are triggered selectively—highlighting ABS's ability to conserve resources without degrading performance.

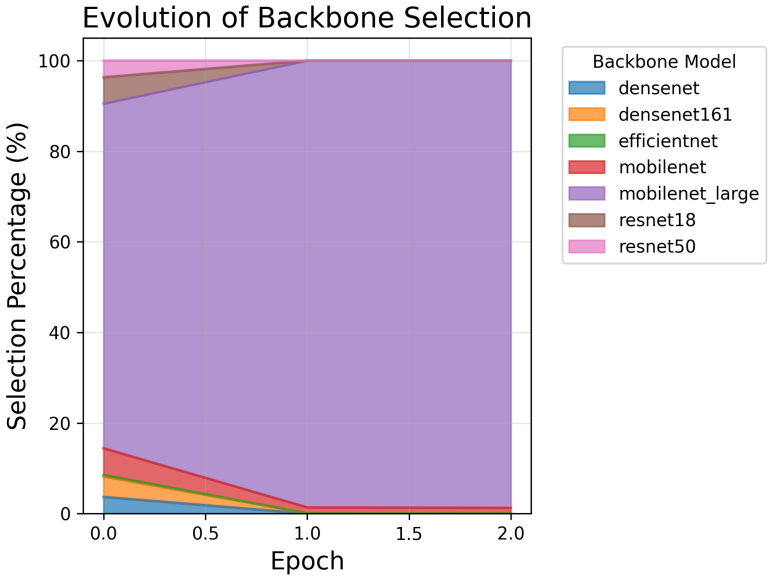

Fig. 3. Dynamic routing favors lightweight backbones for typical inputs, invoking heavier models only for high-complexity samples.

Figure 4 visualizes how input complexity correlates with model depth. As complexity increases, the policy tends to select deeper backbones, proving that ABS leverages difficulty-aware decisions to maintain accuracy while minimizing cost.

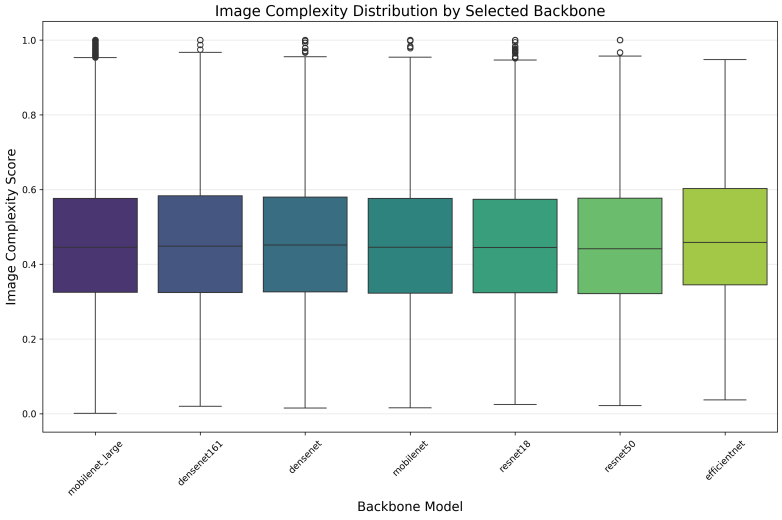

Fig. 4. ABS assigns deeper backbones to complex inputs and lighter ones to simpler cases, showcasing complexity-aware routing.

Finally, to confirm the robustness of the learned policy, we track its reward signal during training. As shown in Figure 5, the policy converges to a stable regime where routing decisions consistently optimize the trade-off between inference cost and accuracy.

Fig. 5. Reward signal stabilizes over time, indicating that the REINFORCE-based policy reliably learns optimal routing strategies.

4.2 Controller Overhead (Routing Stage)

A central question for adaptive inference is whether the controller introduces prohibitive overhead. Table 1 therefore isolates the cost of the *routing stage only* (complexity estimation + policy decision + any model-switch/caching overhead), excluding the CNN backbone forward pass.

Model	**Accuracy (%)**	**Routing latency (ms/img)**	**Routing energy (J/img)**	**Total routing CO_2 (**
MobileNetV2	82.78	0.2406	0.0662	0.37
ResNet18	79.15	0.2228	0.0613	0.34
EfficientNet-B0	90.50	0.3000	0.0825	0.46
DenseNet121	83.69	0.6953	0.1912	1.06
ResNet50	87.37	0.6336	0.1742	0.97
Adaptive (Ours)	**84.04**	**0.2950**	**0.0800**	**0.60**

Table 1. Accuracy (end-to-end) and *routing-stage overhead only* on ImageNet validation (50,000 images). Routing latency/energy include complexity estimation, routing decision, and any model-switch/caching overhead, but exclude the selected backbone forward pass; these values quantify controller overhead rather than total inference cost. Total CO_2 is computed from routing energy aggregated over 50,000 images using a fixed electricity carbon intensity.

While a monolithic EfficientNet-B0 achieves the highest top-1 accuracy, it does so at a constant, high computational cost. Table 1 shows that the routing stage itself is lightweight (sub-millisecond), so the controller is unlikely to dominate end-to-end runtime. Combined with per-instance backbone selection (Figures 1–4), ABS yields a favorable accuracy–efficiency trade-off by allocating heavier models only when needed. These outcomes are particularly noteworthy given that no fine-tuning was performed, underscoring ABS's practicality and deployability for real-world, green AI applications.

5 Conclusion and Future Work

We introduced an Adaptive Backbone Selection (ABS) framework that dynamically aligns model complexity with input difficulty—offering a resource-efficient alternative to static CNN architectures. By integrating a lightweight complexity analyzer with a reinforcement learning-based policy network, ABS learns to allocate inference effort intelligently. This leads to notable reductions in energy usage and latency without retraining any backbone.

Extensive evaluation on the ImageNet dataset suggests that ABS can reduce average energy consumption compared to heavy static models like ResNet50 (up to 58% in our experimental setup), while maintaining competitive top-1 accuracy. These results underscore ABS's relevance in the broader context of Green AI [1], where energy-efficient, sustainable model deployment is critical to minimizing carbon impact and infrastructure load [2].

5.1 Real-World Impact and Deployment Potential

ABS is designed for practical integration across diverse settings—from edge devices to cloud-based APIs. Because model weights remain frozen and routing is performed before inference, the system is reproducible, hardware-compatible, and easy to maintain. Its modular design also supports plug-and-play backbone replacement, making it ideal for deployment pipelines that must adapt to device constraints, user preferences, or real-time workloads.

5.2 Limitations

While ABS demonstrates promising trade-offs in efficiency and accuracy, several limitations remain. First, the framework currently relies on pretrained CNNs without task-specific fine-tuning, which may cap maximum achievable accuracy. Second, the complexity analyzer, while lightweight, may oversimplify certain visual patterns—resulting in occasional suboptimal routing. Third, our evaluation is restricted to classification on the ImageNet dataset; further testing on diverse domains such as segmentation or real-time video is needed to assess generalizability. Finally, hardware-dependent energy estimates are approximate and may vary across deployment environments.

5.3 Future Work

This study opens several promising directions for advancing adaptive and environmentally aware AI systems:

- **Task-Aware Extension**: Generalize ABS to multi-task inference (e.g., object detection, depth estimation) by conditioning routing on both task type and input complexity [14] [16].
- **Layer-Level Adaptivity**: Incorporate fine-grained early exits, dynamic pruning, or token skipping [4] [17] [18] to further reduce computation within selected backbones.
- **Hardware-Aware Learning**: Train policies to factor in device-specific signals such as thermal thresholds, battery level, or compute congestion [19], enabling context-sensitive deployment in embedded or mobile settings [20].
- **Interpretability-Aware Routing**: Enhance decision transparency through saliency maps or attention visualizations [21], especially important in safety-critical domains like healthcare or autonomous navigation [22].
- **Sustainability-Aware Objectives**: Develop reward functions that incorporate environmental budgets, carbon caps, or user-defined energy ceilings—moving toward AI systems that learn to self-regulate under real-world constraints [3].

In conclusion, the ABS framework lays the foundation for a new class of intelligent, interpretable, and sustainability-aligned computer vision systems. We believe this work marks a critical step toward Green AI deployment—where models adapt not just to inputs, but to the environment and operational context in which they serve.

References

1. Schwartz, R., Dodge, J., Smith, N. A., Etzioni, O.: Green AI. Communications of the ACM, 63(12), 54–63 (2020)
2. Verdecchia, R., Sallou, J., Cruz, L.: A Systematic Review of Green AI. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 13(1), e1507 (2023)
3. Yarally, T., Cruz, L., Feitosa, D., Sallou, J., van Deursen, A.: Uncovering Energy-Efficient Practices in Deep Learning Training: Preliminary Steps Towards Green AI. arXiv preprint arXiv:2303.13972 (2023)
4. Wu, Z., Nagarajan, T., Kumar, A., Rennie, S., Feris, R., Farhadi, A.: BlockDrop: Dynamic Inference Paths in Residual Networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
5. Teerapittayanon, S., McDanel, B., Kung, H. T.: BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks. 23rd International Conference on Pattern Recognition (ICPR) (2016)
6. Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., Weinberger, K. Q.: Multi-Scale Dense Networks for Resource Efficient Image Classification. International Conference on Learning Representations (ICLR) (2018)
7. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. CVPR (2016)
9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q.: Densely Connected Convolutional Networks. CVPR (2017)
10. Tan, M., Le, Q.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. International Conference on Machine Learning (ICML) (2019)
11. Wang, Y., Xu, C., Xu, C., Tao, D.: SkipNet: Learning Dynamic Routing in Convolutional Networks. European Conference on Computer Vision (ECCV) (2018)
12. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. International Conference on Learning Representations (ICLR) (2017)
13. Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jen-Hsun, C., Zhai, A., Houlsby, N.: Scaling Vision with Mixture of Experts. ICML (2021)
14. Li, C., He, X., Gong, Y., Zhang, H.: DynaSwitch: Learning to Switch Among Backbones for Adaptive Inference. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
15. Ahn, S., Kwak, S., Zhang, H., Torr, P.H.S.: Deep Elastic Networks with Model Selection for Multi-Task Learning. International Conference on Computer Vision (ICCV) (2019)
16. Pang, J., et al.: Dynamic Vision Transformers: A Survey. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
17. Yu, J., Huang, T.: Slimmable Neural Networks. International Conference on Learning Representations (ICLR) (2019)
18. Elbayad, M., Besacier, L., Verbeek, J.: Token-level Adaptive Training for Neural Machine Translation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2020)
19. Yang, Y., Li, Y., Tao, D.: Optimus-CC: A Causal-Based Task Scheduler for Cloud Computing Systems. IEEE Transactions on Parallel and Distributed Systems, 34(11), 3043–3057 (2023)

20. Zhang, Z., Wang, C., Zhang, M., Jin, R., Xu, C., Yu, J.: TranSTuner: A Transformer-based Framework for Hardware-aware Neural Architecture Search. Proceedings of the AAAI Conference on Artificial Intelligence, 37(9), 10314–10322 (2023)
21. Chefer, H., Gur, S., Wolf, L.: Transformer Interpretability Beyond Attention Visualization. CVPR (2021)
22. Wang, Z., Liu, Y., Xu, X., Zhang, Q., Zhang, L.: Explanatory Visuals: A Survey of Techniques for Visualizing and Explaining AI. IEEE Transactions on Visualization and Computer Graphics, 29(10), 6060–6080 (2023)
23. Williams, R. J.: Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. Machine Learning, 8, 229–256 (1992)