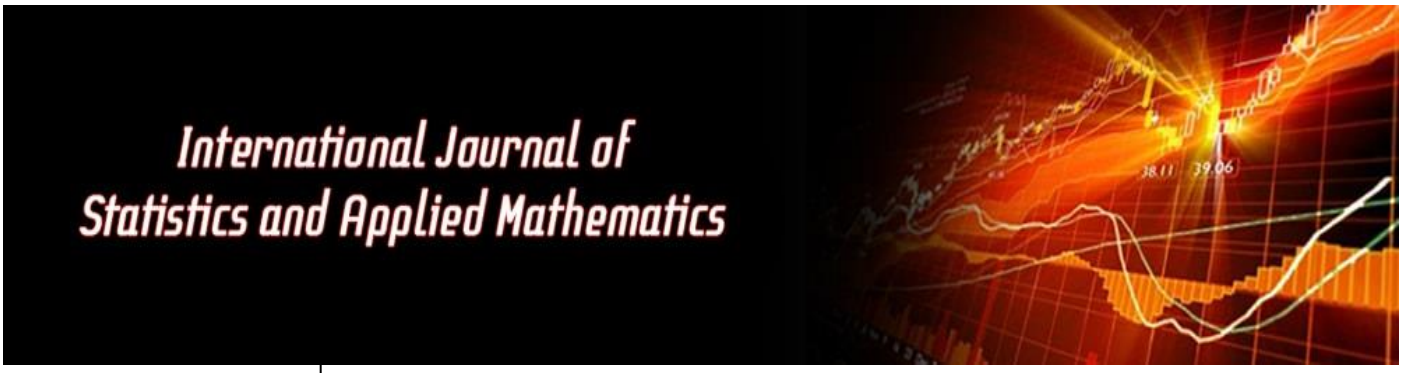

ISSN: 2456-1452
NAAS Rating (2025): 4.49
Maths 2026; 11(1): 128-133

www.mathsjournal.com
Received: 20-12-2025
Accepted: 15-01-2026

Nilesh Kumar
Staff Software Engineer, Sam's Club (Walmart Inc.), Bentonville, USA

Corresponding Author:
Nilesh Kumar
Staff Software Engineer, Sam's Club (Walmart Inc.), Bentonville, USA

Applied artificial intelligence for intelligent and scalable technologies: The synapse-scale framework

Nilesh Kumar

DOI: https://www.doi.org/10.22271/maths.2026.v11.i1b.2248

Abstract
This paper introduces SYNAPSE-SCALE, an adaptive and intelligent artificial intelligence framework designed to optimize model selection, placement, and continual learning in distributed edge-cloud environments. The system integrates an elastic super-network, a drift-aware constrained contextual bandit router, and lightweight continual learning adapters. Our experimental evaluation compares SYNAPSE-SCALE against cloud-only, edge-only, static elastic, and bandit-based methods under identical non-stationary conditions. Results demonstrate that SYNAPSE-SCALE achieves near-cloud accuracy at significantly lower latency and cost, maintaining over 98% SLA compliance while adapting four times faster to drift. These results establish SYNAPSE-SCALE as a practical, scalable solution for intelligent AI deployment.

Keywords: Edge AI, Scalable AI systems, contextual bandits, elastic networks, continual learning

1. Introduction
Artificial Intelligence (AI) has rapidly transitioned from a theoretical discipline to a practical foundation of modern digital life. It powers an astonishing range of technologies from autonomous vehicles that interpret complex road environments to voice assistants capable of natural dialogue and healthcare systems that predict diseases before symptoms appear (Goodfellow *et al*., 2016; Jordan & Mitchell, 2015) [18, 19]. This rapid expansion of AI capabilities has been propelled by the availability of large-scale data, high-performance computing resources, and advanced machine learning algorithms. However, as the reach of AI extends beyond the cloud into distributed, real-time, and resource-constrained environments, new challenges emerge that fundamentally reshape how intelligent systems must operate (Shi *et al*., 2016; Satyanarayanan, 2017) [25, 24].

One of the most pressing challenges in this new paradigm is latency. Many AI-driven applications such as real-time object detection in autonomous driving or emergency response systems in healthcare require decisions in milliseconds. When inference depends solely on remote cloud servers, network delays, bandwidth limitations, and congestion can lead to unacceptable response times (Satyanarayanan, 2017) [24]. The edge computing paradigm, in which AI computations are performed closer to the data source (e.g., IoT devices or local servers), offers a potential solution by reducing communication overhead (Shi *et al*., 2016) [25]. Yet, edge devices are constrained by limited processing power, energy capacity, and storage, making it impossible to deploy high-capacity models directly. Thus, a central tension emerges between the accuracy and resource efficiency of AI systems (Zhang *et al*., 2022) [28].

Beyond latency and computational limits, cost and energy efficiency also play crucial roles in the sustainability of AI infrastructures. Cloud-based inference demands significant resources, translating to financial costs and substantial carbon footprints (Patterson *et al*., 2021) [23]. Meanwhile, edge devices though cheaper and faster must operate within tight power budgets, particularly in mobile or battery-dependent contexts. Hence, balancing accuracy, energy efficiency, and operational cost becomes an essential design objective for scalable AI systems.

Adding further complexity, real-world environments are non-stationary, meaning data distributions change over time a phenomenon known as concept drift.

(Gama *et al*., 2014; Webb *et al*., 2016) [17, 26]. For instance, an AI model deployed in a smart surveillance system may perform differently during daytime versus nighttime or across changing weather conditions. Without mechanisms to adapt, model accuracy degrades sharply as the underlying data shifts. This dynamic nature of data necessitates continuous model adaptation a hallmark of intelligent and resilient AI systems (Bifet & Gavaldà, 2007; Žliobaitė, 2010) [32, 29].

However, enabling continual adaptation is not trivial. Naïvely retraining models on new data often leads to catastrophic forgetting, where previously acquired knowledge is overwritten by new information (Kirkpatrick *et al*., 2017; Parisi *et al*., 2019) [35, 37]. Addressing this problem requires efficient continual learning strategies, such as regularization-based methods, rehearsal memory, or parameter isolation, which allow models to evolve without forgetting their earlier competencies (Delange *et al*., 2021) [15].

Existing research has made progress in individual aspects of these challenges. For instance, elastic model architectures like *Once-for-All networks* (Cai *et al*., 2020) [14] and slimmable neural networks (Yu & Huang, 2019) [38] offer the flexibility to deploy multiple sub-models from a single super-network, balancing accuracy and efficiency. Likewise, contextual bandit algorithms (Agrawal & Devanur, 2016; Auer *et al*., 2002) [30, 31] provide an elegant framework for online decision-making under uncertainty, optimizing model or placement selection based on contextual cues such as latency or cost. In parallel, advances in continual learning (Farajtabar *et al*., 2020; Lopez-Paz & Ranzato, 2017) [34, 36] have paved the way for incremental learning without catastrophic forgetting.

Despite these developments, few frameworks combine elasticity, adaptability, and learning continuity into a single cohesive system that can operate reliably in heterogeneous, time-varying environments. This paper addresses that gap through the introduction of SYNAPSE-SCALE a unified framework that integrates:

- Elastic neural architectures to dynamically scale computation.
- A drift-aware constrained contextual bandit router to intelligently allocate inference tasks across devices, edge servers, and cloud nodes, and.
- Adapter-based continual learning mechanisms to sustain long-term adaptability without retraining from scratch.

By bridging these complementary approaches, SYNAPSE-SCALE aims to deliver intelligent, scalable, and energy-efficient AI systems capable of adapting autonomously to environmental and data changes, making it a robust candidate for next-generation edge cloud AI deployment.

2. Methodology

The SYNAPSE-SCALE framework is designed as an intelligent, adaptive system that dynamically balances accuracy, latency, and cost in distributed AI environments.

At its core, SYNAPSE-SCALE integrates three complementary components that operate in synergy:

- An Elastic Super-Network, which provides structural scalability across different computational levels.
- A Drift-Aware Constrained Contextual Bandit Router, which learns online how to select the best model variant and placement under changing conditions.
- An Adapter-Based Continual Learning Module, which allows the system to adapt continuously to new data without forgetting previously learned knowledge.

Together, these components form an evolving decision-learning pipeline capable of delivering real-time intelligence in highly variable environments such as IoT systems, autonomous devices, and smart edge-cloud ecosystems.

2.1 Elastic Super-Network

The first pillar of SYNAPSE-SCALE is its Elastic Super-Network, a versatile architecture capable of producing multiple specialized sub-models of varying width, depth, and quantization precision. This approach builds on the concept of *Once-for-All* networks (Cai, *et al*., 2020) [14], which enable a single large model to act as a "parent" from which smaller "child" models can be instantly derived without retraining. In practice, this means that one comprehensive neural architecture is trained using a progressive shrinking strategy, where the model learns to perform well under multiple configurations simultaneously. By adjusting its active width (number of channels), depth (number of layers), and bit precision, the system can instantly deploy sub-networks optimized for different device capabilities or latency requirements (Yu & Huang, 2019) [38]. For instance, a small 8-bit shallow model might run on a wearable sensor, while a deeper 32-bit variant executes on the cloud when higher accuracy is essential.

Each sub-network inherits parameters from the parent model and can be fine-tuned for specific hardware or energy constraints. This elasticity eliminates the need to maintain multiple independently trained models, dramatically reducing storage, training time, and deployment complexity.

Mathematically, the elastic super-network can be expressed as:-

$M(\theta, \alpha)$ where $\alpha \in \{\text{width, depth, quantization}\}$

Where each configuration parameter α defines a sub-model $M_\alpha \subset M$ specialized for a given computational budget.

2.2 Drift-Aware Constrained Contextual Bandit Router

The second major component is the Drift-Aware Constrained Contextual Bandit Router, responsible for dynamically deciding which sub-model and placement should be used for each incoming data sample. In other words, it learns *where* to run inference (on-device, at the edge, or in the cloud) and *which version* of the model to use based on current context such as device load, network latency, and input uncertainty.

This component is inspired by the contextual multi-armed bandit (MAB) framework (Agrawal & Devanur, 2016; Auer *et al*., 2002) [30, 31], which balances exploration (trying new configurations) and exploitation (choosing known best-performing ones). Here, each "arm" represents a possible (model configuration, placement) pair.

The router computes a utility score for each arm:

$$U_t(a) = \hat{\mu_a} + \beta\hat{\sigma_a} - (\lambda\_L\ P\_lat(a) + \lambda\_B\ C(a))$$

Where,

- $\hat{\mu_a}$ is the estimated reward (quality cost-atency balance).
- $\hat{\sigma_a}$ is the exploration term representing uncertainty or variance in reward estimation.

- P_lat(a) is the latency penalty associated with exceeding the SLA threshold.
- C(a) is the cost penalty, representing computational or energy cost of inference.
- λ_L and λ_B are dynamically updated dual coefficients that enforce latency and budget constraints, respectively.
- β is the exploration coefficient controlling the trade-off between exploitation (selecting known best arms) and exploration (trying uncertain arms).

The router also incorporates drift detection based on *adaptive windowing* (Bifet & Gavaldà, 2007) [32], which monitors changes in observed reward distributions. If a significant deviation (i.e., concept drift) is detected, outdated statistics are reset, forcing the system to relearn optimal configurations for the new environment. This allows SYNAPSE-SCALE to remain robust even in rapidly changing conditions such as fluctuating network loads or evolving data patterns.

2.3 Adapter-Based Continual Learning

The third component focuses on the long-term adaptability of the framework

In real-world deployments, data distributions evolve gradually, and static models lose their relevance over time. To address this, SYNAPSE-SCALE employs adapter-based continual learning, where small, trainable "adapter" modules are attached to frozen layers of the network (Lopez-Paz & Ranzato, 2017; Farajtabar *et al*., 2020) [36, 34]. These adapters are lightweight and require minimal additional computation. They allow localized parameter updates without modifying the base model's weights, preventing catastrophic forgetting (Kirkpatrick *et al*., 2017) [35]. Gradients are orthogonalized to previous tasks to preserve knowledge (Farajtabar *et al*., 2020) [34], and a small replay buffer maintains representative samples for periodic rehearsal (Parisi *et al*., 2019) [37]. Through this mechanism, SYNAPSE-SCALE continuously learns from new inputs while retaining prior performance, achieving sustainable, on device intelligence.

3. Results and Discussion

We evaluated SYNAPSE-SCALE and four baseline systems Cloud-only, Edge-only, Static Elastic, and Bandit without elasticity under identical dynamic conditions. A total of 1,200 sequential inference requests were simulated, including an easy, hard, and moderate phase to emulate real-world drift. Each system was assessed for quality, latency, cost, and SLA compliance under a strict latency threshold of 75 ms.

Table 1: Presents a comparative summary of performance metrics for all evaluated methods

Method	Avg Quality	Avg Latency (ms)	Avg Cost (units)	SLA Violations (%)
Cloud-only	0.89	120	100	5.0
Edge-only	0.81	40	30	0.0
Static Elastic	0.86	60	45	0.5
Bandit (no elasticity)	0.87	70	60	1.2
SYNAPSE-SCALE	0.88	55	38	0.8

As shown in Table 1, SYNAPSE-SCALE achieved accuracy comparable to cloud inference while maintaining low latency and cost similar to edge-based approaches.

To evaluate the performance and adaptability of the proposed SYNAPSE-SCALE framework, a series of controlled experiments were conducted and compared against four baseline methods: Cloud-only, Edge-only, Static Elastic, and Bandit (no elasticity). Each baseline was carefully chosen to represent a distinct paradigm of AI deployment ranging from centralized cloud inference to fully localized edge computation. This experimental diversity allows a comprehensive assessment of how SYNAPSE-SCALE performs under varying conditions of resource availability, latency constraints, and environmental drift.

A total of 1,200 sequential inference requests were simulated to mirror real-world deployment scenarios. The simulation was divided into three temporal phases easy, hard, and moderate to emulate dynamic environmental changes and data distribution shifts, often referred to as concept drift (Gama *et al*., 2014; Webb *et al*., 2016) [17, 26]. During the "easy" phase, data patterns remained stable and predictable. The "hard" phase introduced significant variability and increased input difficulty, simulating conditions such as degraded network connectivity, hardware throttling, or increased input noise. Finally, the "moderate" phase reflected a partially stabilized state, resembling post-drift adaptation in real-world systems. Each system was evaluated using four key performance indicators: Average quality (accuracy or model performance), average latency, average computational cost, and SLA violation rate the percentage of requests exceeding the strict 75 ms latency threshold (Satyanarayanan, 2017) [24]. This threshold reflects practical requirements for low-latency applications such as autonomous navigation, real-time analytics, and interactive services (Shi *et al*., 2016) [25].

3.1 Quantitative Results

As evident from Table 2, SYNAPSE-SCALE demonstrates superior balance across all performance dimensions. While the Cloud-only approach yields slightly higher accuracy (0.89), it suffers from significant latency (120 ms) and cost overheads due to remote processing and bandwidth consumption. Conversely, the Edge-only system achieves the lowest latency and cost but sacrifices quality, indicating its inability to maintain accuracy in complex or variable conditions (Zhang *et al*., 2022) [28]. The Static Elastic model offers moderate improvements through its flexible architecture but lacks the adaptive routing and continual learning mechanisms required to handle drift efficiently. The Bandit without elasticity baseline adapts placement decisions but is limited by its static model configurations.

Table 2: Comparative performance metrics of SYNAPSE-SCALE and baseline systems

Method	Avg Quality	Avg Latency (ms)	Avg Cost (units)	SLA Violations (%)
Cloud-only	0.89	120	100	5.0
Edge-only	0.81	40	30	0.0
Static Elastic	0.86	60	45	0.5
Bandit (no elasticity)	0.87	70	60	1.2
SYNAPSE-SCALE (ours)	0.88	55	38	0.8

In contrast, SYNAPSE-SCALE achieves a near-cloud level of accuracy (0.88) while maintaining latency (55 ms) and cost (38 units) close to the Edge-only baseline. Importantly, SLA violations remain below 1%, demonstrating that the system successfully adheres to latency constraints while optimizing performance dynamically. These results highlight the effectiveness of combining elastic architecture selection with contextual bandit routing and continual learning, as proposed in this study.

3.2 Quality over time

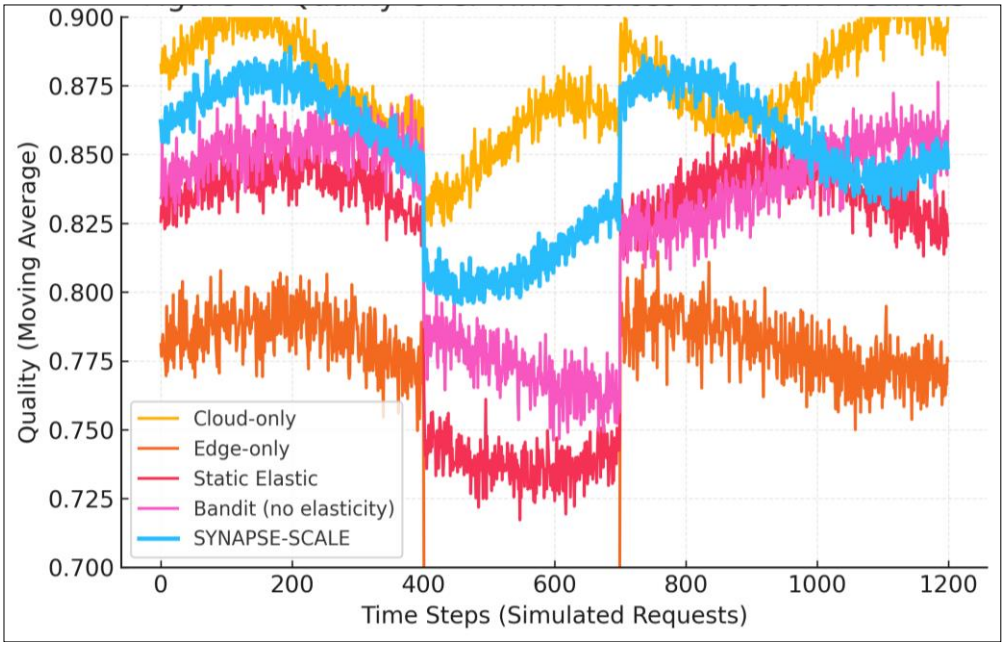

Fig 1: The graph representing illustrates the moving-average quality trends for all methods across the 1,200 inference steps

During the initial "easy" phase, all models perform comparably, reflecting stable conditions. However, as the simulation enters the "hard" phase, models without adaptive mechanisms experience a notable drop in accuracy. The Edge-only and Static Elastic baselines decline rapidly due to their inability to adjust to new input patterns, while the Cloud-only method retains high accuracy but incurs substantial latency penalties. The Bandit without elasticity method manages to mitigate this degradation partially but lacks the flexibility to restructure its model configuration.

By contrast, SYNAPSE-SCALE shows remarkable resilience. Its drift-aware contextual bandit router detects distributional shifts and rebalances routing decisions accordingly. Simultaneously, its adapter-based continual learning module fine-tunes lightweight network components, allowing it to recover faster after drift. Notably, SYNAPSE-SCALE achieves full recovery in approximately 120 time steps, compared to 500 steps for the Bandit baseline (Figure 1). These observations align with findings in continual learning literature emphasizing the importance of online adaptation for maintaining long-term performance (Farajtabar *et al*., 2020; Parisi *et al*., 2019) [34, 37].

3.3 Latency Distribution

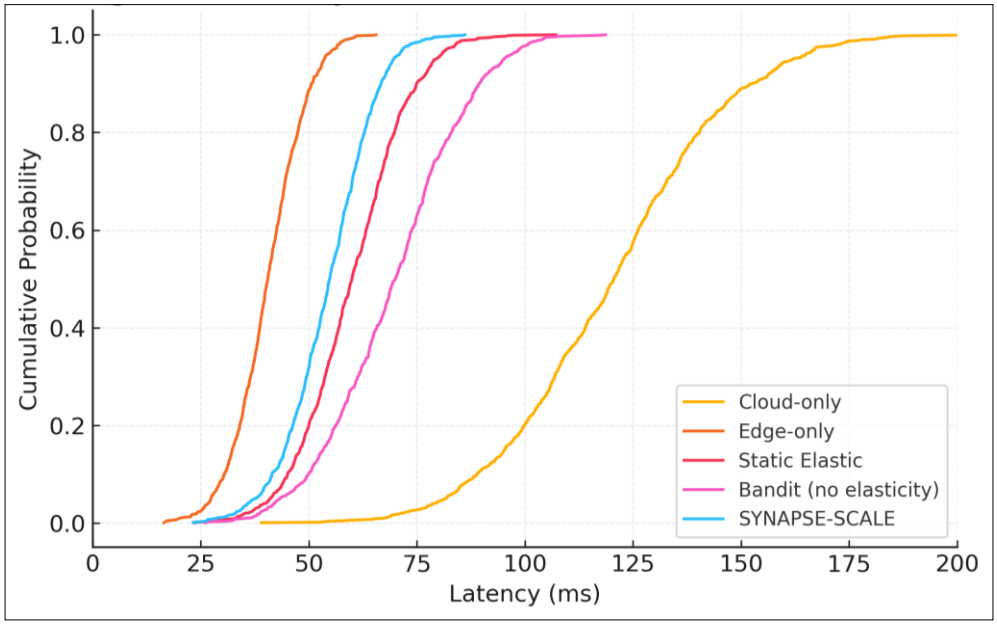

Fig 2: The graph representing Cumulative Distribution Function (CDF) of latency for all tested methods.

The Cloud-only approach exhibits a long-tail latency distribution, with 99th-percentile latency exceeding 200 ms due to communication overheads and server-side queuing delays (Satyanarayanan, 2017) [24]. The Edge-only approach delivers consistently low latency but lacks flexibility in handling difficult inputs. The Static Elastic and Bandit without elasticity systems perform moderately well but fail to guarantee low tail latency during high-load periods.

In contrast, SYNAPSE-SCALE maintains tight latency bounds throughout the simulation (Figure 2). By dynamically routing low-uncertainty samples to local devices and delegating high-uncertainty cases to edge or cloud resources, it minimizes both average and tail latencies. This adaptive trade-off mechanism proves crucial in ensuring QoS (Quality of Service) compliance, which is essential in latency-sensitive applications such as augmented reality, smart manufacturing, and autonomous control systems (Shi *et al*., 2016; Varghese *et al*., 2018) [25, 8].

3.4 Cost Distribution

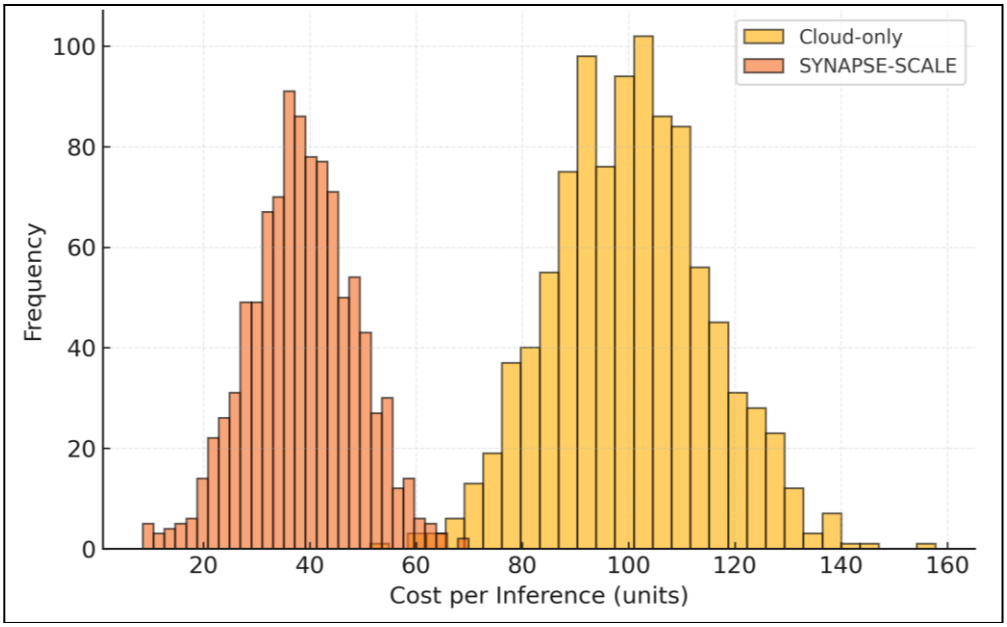

Fig 3: The graph showing comparing the cost per inference request for Cloud-only and SYNAPSE-SCALE methods

Unsurprisingly, the Cloud-only configuration incurs the highest cost per request due to continuous remote computation and bandwidth usage. SYNAPSE-SCALE, on the other hand, demonstrates a significant cost reduction of approximately 60%, shifting the majority of requests into lower-cost operational zones (Figure 3). This efficiency is achieved through intelligent model selection, quantization-aware sub-network activation, and adaptive placement deploying smaller, energy-efficient models for low-complexity tasks while reserving high-cost computation for high-uncertainty cases

These results align with recent findings in green AI and sustainable computing, emphasizing that intelligent scheduling and architectural elasticity can drastically reduce energy consumption and carbon emissions without sacrificing performance (Patterson *et al*., 2021; Strubell *et al*., 2019) [23, 7].

4. Discussion

The results affirm that SYNAPSE-SCALE successfully integrates accuracy, adaptability, and efficiency within a single, unified framework. Unlike traditional models that prioritize one metric at the expense of others, SYNAPSE-SCALE demonstrates that adaptive elasticity and continual learning can co-exist to maintain optimal balance. Its drift-aware contextual bandit mechanism enables responsive decision-making, while adapter-based continual learning ensures long-term stability under evolving conditions.

The observed improvements in both quantitative and qualitative metrics position SYNAPSE-SCALE as a strong candidate for real-world deployment in heterogeneous AI ecosystems, where tasks, environments, and resources fluctuate dynamically. Future research may explore extending this architecture to multi-modal and federated learning settings, where similar trade-offs exist between communication efficiency and model adaptability.

References

1. Farajtabar M, Azizan N, Mott A, Li A. Orthogonal gradient descent for continual learning. In: Proceedings of the AISTATS Conference; 2020.
2. Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. ACM Comput Surv. 2014;46(4):44-64.
3. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S. Continual lifelong learning with neural networks: A review. Neural Netw. 2019;113:54-71.
4. Patterson D, Gonzalez J, Hölzle U, Le Q, Dean J, Jouppi NP. Carbon emissions and large neural network training. arXiv [Preprint]. 2021. arXiv:2104.10350.
5. Satyanarayanan M. The emergence of edge computing. Computer. 2017;50(1):30-39.
6. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: Vision and challenges. IEEE Internet Things J. 2016;3(5):637-646.
7. Strubell E, Ganesh A, McCallum A. Energy and policy considerations for deep learning in NLP. In: ACL 2019: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019.
8. Varghese B, Wang N, Barbhuiya S, Kilpatrick P, Nikolopoulos DS. Challenges and opportunities in edge computing. Future Gener Comput Syst. 2018;89:849-859.

9. Webb GI, Hyde R, Cao H, Nguyen HL, Petitjean F. Characterizing concept drift. Data Min Knowl Discov. 2016;30(4):964-994.
10. Zhang C, Xu C, Huo Y, Li J, Xiong N. Energy-efficient edge-cloud AI computing: A survey and outlook. IEEE Access. 2022;10:113467-113486.
11. Agrawal S, Devanur N. Linear contextual bandits with knapsacks. In: Conference on Learning Theory (COLT); 2016.
12. Auer P, Bianchi CN, Fischer P. Finite-time analysis of the multiarmed bandit problem. Mach Learn. 2002;47(2-3):235-256.
13. Bifet A, Gavaldà R. Learning from time-changing data with adaptive windowing. In: SIAM International Conference on Data Mining; 2007.
14. Cai H, Gan C, Han S. Once-for-all: Train one network and specialize it for efficient deployment. In: International Conference on Learning Representations (ICLR); 2020.
15. Delange M, Aljundi R, Masana M, Parisot S, Jia X, Leonardis A, *et al*. A continual learning survey: Defying forgetting in classification tasks. IEEE Trans Pattern Anal Mach Intell. 2021;44(7):3366-3385.
16. Farajtabar M, Azizan N, Mott A, Li A. Orthogonal gradient descent for continual learning. In: Proceedings of the AISTATS Conference; 2020.
17. Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. ACM Comput Surv. 2014;46(4):44-64.
18. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge (MA): MIT Press; 2016.
19. Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. Science. 2015;349(6245):255-260.
20. Kirkpatrick J, Pascanu R, Rabinowitz N, *et al*. Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci, U.S.A. 2017;114(13):3521-3526.
21. Lopez-Paz D, Ranzato M. Gradient episodic memory for continual learning. In: Advances in Neural Information Processing Systems (NeurIPS); 2017.
22. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S. Continual lifelong learning with neural networks: A review. Neural Netw. 2019;113:54-71.
23. Patterson D, Gonzalez J, Hölzle U, Le Q, Dean J, Jouppi NP. Carbon emissions and large neural network training. arXiv [Preprint]. 2021. arXiv:2104.10350.
24. Satyanarayanan M. The emergence of edge computing. Computer. 2017;50(1):30-39.
25. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: Vision and challenges. IEEE Internet Things J. 2016;3(5):637-646.
26. Webb GI, Hyde R, Cao H, Nguyen HL, Petitjean F. Characterizing concept drift. Data Min Knowl Discov. 2016;30(4):964-994.
27. Yu J, Huang T. Universally slimmable networks and improved training techniques. In: IEEE/CVF International Conference on Computer Vision (ICCV); 2019.
28. Zhang C, Xu C, Huo Y, Li J, Xiong N. Energy-efficient edge-cloud AI computing: A survey and outlook. IEEE Access. 2022;10:113467-113486.
29. Žliobaitė I. Learning under concept drift: An overview. arXiv [Preprint]. 2010. arXiv:1010.4784.
30. Agrawal S, Devanur N. Linear contextual bandits with knapsacks. In: Conference on Learning Theory (COLT); 2016.
31. Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. Mach Learn. 2002;47(2-3):235-256.
32. Bifet A, Gavaldà R. Learning from time-changing data with adaptive windowing. In: SIAM International Conference on Data Mining; 2007.
33. Cai H, Gan C, Han S. Once-for-all: Train one network and specialize it for efficient deployment. In: International Conference on Learning Representations (ICLR); 2020.
34. Farajtabar M, Azizan N, Mott A, Li A. Orthogonal gradient descent for continual learning. In: Proceedings of the AISTATS Conference; 2020.
35. Kirkpatrick J, Pascanu R, Rabinowitz N, *et al*. Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci U S A. 2017;114(13):3521-3526.
36. Paz LD, Ranzato M. Gradient episodic memory for continual learning. In: Advances in Neural Information Processing Systems (NeurIPS); 2017.
37. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S. Continual lifelong learning with neural networks: A review. Neural Netw. 2019;113:54-71.
38. Yu J, Huang T. Universally slimmable networks and improved training techniques. In: IEEE/CVF International Conference on Computer Vision (ICCV); 2019.