Adaptive Compute Efficient Learning via Conceptual-Criticality (Student Abstract)

Iñigo Parra*[1], Mano Bharathi M*[2], Mayank Kumar*[3], Pushpa Kumar Balan*[4], Priyadarsi Mishra*[5]

[1]Department of Linguistics, UC Berkeley, Berkeley, CA, USA
[2] Research and Development, Opentext, Bangalore, India
[3] SCSET, Bennett University, Greater Noida, India
[4]Department of Computer Science and Cybersecurity, University of Central Missouri, Warrensburg, MO, USA
[5]Department of Engineering, Texas A&M University, College Station, TX, USA
iparra@berkeley.edu, immanobharathi21@gmail.com, mayankdhruv42@gmail.com, pushpakumarbalan@gmail.com, priyadarsimishra@tamu.edu

Abstract

The computational cost of large language models (LLMs) is a primary obstacle to sustainable deployment. Static resource allocation is inefficient, as not all inputs require the same depth of processing. We propose a framework for adaptive, compute-efficient learning via conceptual criticality, which dynamically tailors computation to the assessed difficulty of an input. A lightweight criticality prediction module estimates conceptual complexity on a continuous scale, and this score governs the LLM's inference pathway, selectively activating token pruning, layer skipping, and quantization. Simple inputs are processed with minimal FLOPs and latency, while complex inputs use the model's full capacity to preserve accuracy. We benchmark our framework and introduce metrics to quantify sensitivity to input criticality and per-sample computational savings. Results demonstrate an improved accuracy-efficiency trade-off, paving the way for more resource-aware systems.

Code — https://github.com/ManoBharathi93/Adaptive-Compute-Efficient-Learning-via-Conceptual-Criticality

Introduction

Large language models (LLMs) and other foundation models have achieved state-of-the-art performance across a wide spectrum of tasks, but their growing scale has brought mounting concerns about computational efficiency, energy consumption, and inference latency. Recent analyses suggest that inference often expends unnecessary resources on inputs that do not require the full capacity of the model (Pope et al. 2023; Schwartz et al. 2020). In practice, this inefficiency limits deployment in resource-constrained settings, increases environmental costs, and complicates real-time applications. Designing systems that allocate computation adaptively and spending more resources on "hard" inputs and fewer on "easy" ones has therefore become an important goal in efficient machine learning.

A range of strategies has emerged to address this challenge. We propose to extend this line of work by introducing the notion of conceptual criticality as a driver for adaptive compute allocation. Rather than viewing difficulty solely at the token or structural level, we frame it as an input-level property reflecting the conceptual complexity of a task instance. Intuitively, some questions or prompts demand deep reasoning or multiple inference steps, while others can be answered with shallow processing. By predicting criticality in advance, using lightweight classifiers or early-layer representations, we can guide dynamic resource allocation during inference, routing "easy" inputs through cheap paths and reserving expensive computation for "hard" ones.

This perspective aligns with cognitive science findings that human effort allocation is sensitive to task difficulty (Kool et al. 2010; Shenhav et al. 2017), but it has not yet been systematically incorporated into ML efficiency research. Our contribution is therefore twofold: (i) we operationalize conceptual criticality as a measurable property of inputs and (ii) we demonstrate how criticality-aware routing can improve efficiency/accuracy trade-offs in LLM inference. This approach complements existing token-level methods and introduces a more task-aligned principle for adaptive computation.

Background

At the architectural level, methods such as dynamic token routing (Alizadeh et al. 2024) and early exiting (Liu et al. 2022) selectively allocate computation within models based on learned signals of difficulty.

At the token level, approaches like SelfBudgeter (Li et al. 2025) and TokenButler (Zhou et al. 2025) predict input-specific token budgets, while ToSA (Wang et al. 2024) prunes uninformative tokens through selective attention mechanisms. These works demonstrate that adaptive allocation can yield substantial FLOP savings without severely

*These authors contributed equally.

compromising accuracy, yet they primarily rely on syntactic or token-level heuristics.

Methodology

Our proposed framework employs a two-stage process to achieve adaptive computation. First, a lightweight **pre-compute criticality module**, implemented as an LSTM classifier, assesses the conceptual difficulty of an input. We validated this approach on the `ag_news` dataset, using entropy as a proxy for criticality to establish a proof-of-concept. This criticality score then informs the second stage, acting as a direct routing signal that determines the computational budget and inference path for the input in an adaptive inference model based on a BranchyNet architecture (Teerapittayanon, McDanel, and Kung 2016).

We modified a standard 6-layer Transformer to include three early exit points after layers 1, 3, and 5. During inference, the model can exit at these points if its prediction confidence exceeds a given threshold (τ), bypassing the remaining layers. The *a priori* criticality score enhances this mechanism by allowing for distinct policies for different inputs; for instance, "easy" inputs can be configured to use a more lenient confidence threshold, encouraging an earlier exit, while "hard" inputs are routed deeper. Beyond early exiting, this framework enables dynamic control over other efficiency mechanisms. The criticality signal can modulate the aggressiveness of token pruning and select different quantization levels, applying the most resource-intensive, high-precision configurations only when necessary.

To evaluate the efficiency of this framework, we measured latency, theoretical FLOPs, and empirical energy consumption in Joules using the `pynvml` library. This holistic approach ensures that computational expenditure is proportional to an input's conceptual demands.

Results

Our experiments demonstrate that the adaptive early-exit framework provides substantial computational savings with a negligible impact on model accuracy. The early-exit model achieved an accuracy of approximately **90.7%**, which is on par with the full 6-layer baseline model. However, it achieved this performance while reducing the average number of layers used from 6.0 down to a range of 1.04 to 3.15, depending on the confidence threshold. This directly translated to a significant reduction in real-world energy consumption. Our measurements show that the adaptive approach lowered the average energy per sample from a baseline of **0.26 Joules** to as low as **0.09 Joules**, a reduction of approximately **65%**, without sacrificing performance. Furthermore, the model's accuracy remained robust across all tested confidence thresholds, highlighting the reliability of the early-exit predictions.

Conclusion

In conclusion, this work demonstrates a successful proof-of-concept for a criticality-driven adaptive framework that can drastically improve computational efficiency. Future work will focus on implementing other planned efficiency techniques, such as **ToSA** and quantization, and applying this framework to larger, more complex models and tasks like **Llama-8B** on the **GSM8K** dataset.

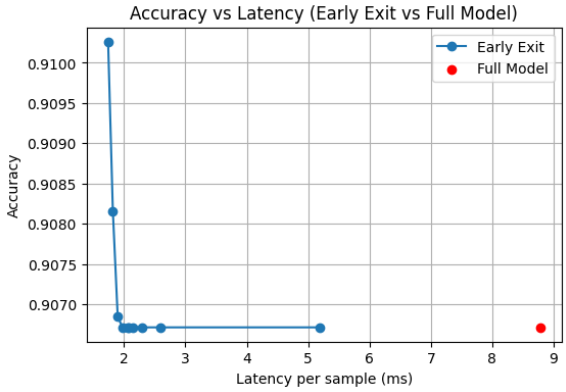

Figure 1: Accuracy vs Latency comparison between Early Exit and Full Model. The plot illustrates the trade-off between accuracy and latency, with Early Exit achieving higher accuracy at lower latency compared to the Full Model

References

Alizadeh, K.; Mirzadeh, I.; Shahrokhi, H.; Belenko, D.; Sun, F.; Cho, M.; Sekhavat, M. H.; Nabi, M.; and Farajtabar, M. 2024. Duo-llm: A framework for studying adaptive computation in large language models. *arXiv preprint arXiv:2410.10846*.

Kool, W.; McGuire, J. T.; Rosen, Z. B.; and Botvinick, M. M. 2010. Decision making and the avoidance of cognitive demand. *Journal of Experimental Psychology: General*, 139(4): 665–682.

Li, Z.; Dong, Q.; Ma, J.; Zhang, D.; and Sui, Z. 2025. Self-budgeter: Adaptive token allocation for efficient llm reasoning. *arXiv preprint arXiv:2505.11274*.

Liu, J.; et al. 2022. Deep Inference: Early Exiting for Efficient Transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Pope, A.; et al. 2023. Measuring the Carbon Intensity of AI in Cloud Environments. In *NeurIPS Workshop on Tackling Climate Change with Machine Learning*.

Schwartz, R.; Dodge, J.; Smith, N. A.; and Etzioni, O. 2020. Green AI. *Communications of the ACM*, 63(12): 54–63.

Shenhav, A.; Musslick, S.; Lieder, F.; Kool, W.; Griffiths, T. L.; Cohen, J. D.; and Botvinick, M. M. 2017. Toward a Rational and Mechanistic Account of Mental Effort. *Annual Review of Neuroscience*, 40: 99–124.

Teerapittayanon, S.; McDanel, B.; and Kung, H.-T. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, 2464–2469. IEEE.

Wang, Z.; et al. 2024. ToSA: Token Selective Attention for Efficient Vision Transformers. *arXiv preprint arXiv:2406.08816*.

Zhou, Y.; et al. 2025. TokenButler: Token Importance is Predictable. *arXiv preprint arXiv:2503.07518*.