

Inference-Time Compute Allocation in LLMs: Dynamic Depth and Token Selection for Efficient Problem Solving

LIAO YUHAO, ZIJIAN ZENG

Institute of Computer Science & Digital Innovation,
UCSI University,
Kuala Lumpur,
MALAYSIA

Abstract: - Large language models (LLMs) deliver impressive performance across diverse natural language processing tasks, but their intensive computation and memory requirements severely restrict real-world deployment, especially on edge devices and in latency-sensitive cloud services. Static inference pipelines execute full layers and full tokens for all inputs, resulting in significant resource waste for simple and routine queries. Although dynamic inference methods have emerged to alleviate this issue, most focus only on either layer-level or token-level optimization, and few provide reliable mechanisms to guarantee bounded performance degradation. In this paper, we propose D²TS-LLM, a unified plug-and-play framework for Dynamic Depth and Token Selection that achieves on-demand computational resource allocation for LLM inference. The framework consists of a multi-dimensional lightweight perceptron for estimating input difficulty and token importance, a joint decision module for adaptive layer activation and token-aware computation reduction, and a lightweight performance monitor with dynamic fallback to ensure performance loss remains within a predefined safe threshold. Extensive experiments on representative LLMs (LLaMA-7B, LLaMA-13B, OPT-6.7B) and five common benchmarks (MMLU, GSM8K, SQuAD v2.0, CNN/Daily Mail, Alpaca) demonstrate that D²TS-LLM reduces computational cost by 42.3%–61.3% and accelerates inference by 38.2%–55.7% with performance loss controlled below 1%. Compared with representative state-of-the-art dynamic inference, quantization, and cache optimization methods, our approach achieves superior efficiency-performance trade-offs and strong cross-model and cross-dataset generalization.

Key-Words: - Large Language Models; Inference Optimization; Dynamic Computation Allocation; Dynamic Depth; Token Selection; Efficient Deployment; Performance-Aware Fallback; Artificial Intelligence Safety.

Received: April 16, 2026. Revised: May 8, 2026. Accepted: June 29, 2026. Published: October 23, 2026.

1 Introduction

Large language models (LLMs) have become foundational architectures in natural language processing, achieving breakthroughs in dialogue, reasoning, summarization, and code generation. Since the proposal of the transformer architecture [1], LLMs have evolved rapidly, with models like LLaMA [2] demonstrating strong performance while maintaining relative efficiency. However, their growing model size and transformer-based structure lead to extremely high computation and memory consumption during inference, which limits their practical deployment on edge devices and in real-time cloud systems—especially for long-context tasks where quadratic attention complexity and linear KV cache growth exacerbate resource constraints [3].

Traditional LLM inference follows a static pipeline: all transformer layers and all input tokens

are executed equally for every sample. This one-size-fits-all strategy ignores the huge variance in input difficulty and token importance, causing severe computational redundancy, especially for simple queries. For long-context scenarios, this redundancy becomes even more critical, as prefill-stage latency (Time-To-First-Token, TTFT) can dominate the overall inference time [3].

To improve efficiency, recent studies have explored dynamic inference, which can be divided into two categories: dynamic layer activation (early-exit or layer skipping based on input complexity) and dynamic token selection (pruning or reducing computation for unimportant tokens). Representative dynamic layer methods include D³ [4], Balcony [5], and SkipGPT [6], while typical dynamic token methods include SlimInfer [3], LazyLLM [7], and IG-Pruning [8]. Despite progress, these methods suffer from critical

limitations: they optimize only one dimension (layer or token), fail to exploit joint redundancy; token pruning often discards tokens directly, causing semantic loss; and few provide a lightweight mechanism to bound performance degradation. For example, SlimInfer [3] achieves significant TTFT acceleration via dynamic token pruning but lacks a unified strategy for layer-token joint optimization, while D³ [4] focuses on position-aware depth decay but does not address token-level redundancy.

To address these issues, we propose D²TS-LLM, a unified framework for Dynamic Depth and Token Selection. The core idea is on-demand resource allocation: simple inputs use fewer layers and fewer core tokens; complex inputs retain nearly full computation. This design aligns with recent findings on dynamic compute allocation, which highlight the importance of matching computational resources to task complexity [9].

Our contributions are summarized as follows:

- 1) We propose a joint dynamic optimization framework that unifies depth adaptation and token-aware computation, achieving stronger efficiency than single-level methods (e.g., [3], [4], [6]).
- 2) We design a multi-dimensional lightweight perceptron to evaluate input difficulty and token importance without extra training, avoiding the computational overhead of complex routing networks [8].
- 3) We introduce a lightweight performance monitor and dynamic fallback mechanism to ensure performance loss is strictly constrained (e.g., $\leq 1\%$), addressing the reliability gap in existing dynamic methods [3], [7].
- 4) The framework is plug-and-play, compatible with Hugging Face Transformers and vLLM [10], and generalizes across LLM architectures, making it suitable for dynamic workloads that require balancing latency and throughput [11].

Extensive experiments show that D²TS-LLM significantly outperforms state-of-the-art methods in efficiency-performance trade-off, including dynamic token pruning [3], [7], dynamic depth methods [4], [5], and input-guided pruning [8], making it suitable for real-world LLM deployment.

2 Related Work

Related work on LLM inference efficiency mainly falls into three lines: dynamic token methods (e.g., SlimInfer, LazyLLM, IG-Pruning) that reduce prefill computation, dynamic depth methods (e.g., D³, Balcony, SkipGPT) that adaptively execute layers, and systems-level KV-cache optimization (e.g., vLLM) that improves memory management without directly reducing FLOPs. Existing dynamic methods typically optimize only one dimension (token or depth) and often lack a lightweight mechanism to bound quality degradation under deployment constraints. D²TS-LLM fills this gap by jointly selecting depth and token computation with a performance-aware fallback mechanism.

3 Methodology

3.1 Overall Framework

D²TS-LLM operates in a plug-and-play manner without modifying the original LLM weights or structure (consistent with the transformer-based LLM architecture proposed in [1]), consisting of four core components: a multi-dimensional lightweight perceptron, a joint decision module, a dynamic computation executor, and a performance monitor with dynamic fallback. The pipeline processes inputs sequentially: scoring input difficulty and token importance, computing dynamic allocation thresholds, performing adaptive computation, and ensuring performance safety through fallback mechanisms if needed. This design aligns with the principles of dynamic compute allocation [9], ensuring that resources are allocated based on actual task complexity rather than fixed heuristics.

3.2 Multi-dimensional Lightweight Perceptron

We design a lightweight perceptron with no extra training requirements to evaluate two key metrics: input difficulty and token importance. This avoids the overhead of complex routing networks [8].

3.2.1 Input Difficulty Scoring

The difficulty score d ($0 \leq d \leq 1$) is computed by fusing sequence length, syntactic complexity, numerical features, and keyword density, where a higher score indicates a more complex input. This aligns with observations that longer and syntactically complex inputs require more computational resources [3], [7].

3.2.2 Token Importance Scoring

Each token receives an importance score based on attention distribution (derived from the transformer attention mechanism [1]), positional information, and semantic saliency, identifying core tokens that require full computation. Unlike methods that directly discard low-importance tokens [3], [7], we retain non-core tokens with lightweight computation to preserve semantic integrity, addressing the semantic loss issue of existing token pruning methods.

3.3 Joint Dynamic Allocation Strategy

Based on the scores from the perceptron, we implement a joint dynamic allocation strategy for layers and tokens, unifying the strengths of dynamic depth [4], [5] and token selection [3], [7] methods.

3.3.1 Dynamic Depth Allocation

Based on difficulty score d , the active layer ratio is set as $t_l = d$, where only the top $L \times t_l$ layers (L is total model layers) are executed, and subsequent layers reuse the final hidden state of the last active layer to reduce redundancy. This position-aware depth adaptation is inspired by D^3 [4], which shows that later tokens require fewer layers, but we extend it to input-dependent difficulty scoring rather than fixed position-based decay.

3.3.2 Dynamic Token Computation

Instead of direct pruning [3], [7], we adopt a lightweight computation sharing strategy: core tokens perform full self-attention (consistent with [1]) and FFN computation, while non-core tokens only attend to core tokens and share FFN features. This preserves sequence integrity while reducing computation, similar to the block-wise sharing mechanism in IG-Pruning [8] but with token-level granularity.

3.4 Performance Monitoring and Dynamic Fallback

To guarantee bounded performance loss, we design a lightweight performance metric P_m based on prediction entropy (lower values indicate more stable performance) and key-token matching score (higher values indicate better task alignment). If P_m falls below a predefined threshold, the system triggers dynamic fallback: first increasing the number of core tokens, then increasing the number of active layers, and finally reverting to full inference if necessary. This mechanism addresses the performance instability issue of existing optimization methods [3], [7] and ensures reliability for dynamic workloads [11]. Unlike complex

fallback strategies that require extra training [5], our approach is lightweight and training-free, making it suitable for edge deployment.

4 Experimental Results and Analysis

4.1 Experimental Setup

We evaluate D²TS-LLM on three representative LLMs (LLaMA-7B, LLaMA-13B [2], OPT-6.7B) and five diverse benchmarks (MMLU for commonsense reasoning, GSM8K for mathematical problem solving, SQuAD v2.0 for reading comprehension, CNN/Daily Mail for text summarization, and Alpaca for general dialogue). We compare D²TS-LLM quantitatively with the representative baselines reported in Table 1. The broader set of reference methods used to position the experimental evaluation includes:

- Full inference baseline (FlashAttention2 kernel): FlashAttention2 [12]
- Dynamic token-selection methods: SlimInfer [3], LazyLLM [7], and TokenSelect [13]
- Dynamic depth methods: D^3 [4], Balcony [5], SkipGPT [6], and DiffSkip [14]
- Input-guided pruning: IG-Pruning [8]
- Hybrid optimization: ANIRA [9]
- KV-Cache optimization: vLLM [10]

The evaluation metrics include efficiency indicators (FRR for FLOPs reduction, ISR for inference speedup, VRR for memory reduction), and performance indicators (Avg.Score and Δ Score relative to the full baseline, as well as task-specific scores such as F1 and Rouge-L), and a balance indicator (EPR for efficiency-performance ratio). Experiments are conducted on NVIDIA RTX 4090 (24GB) for edge deployment simulation and NVIDIA A100 (80GB) for large-scale testing, covering dynamic workloads with varying batch sizes [11]. All benchmark metrics are reported on a 0–100 scale (percentage form) for consistency.

4.2 Core Experimental Results

4.2.1 Overall Performance

As summarized in Table 1, D²TS-LLM consistently improves the efficiency–quality frontier over single-dimensional baselines across ϵ . Under different ϵ budgets, D²TS-LLM achieves 42.3%–72.1% FLOPs reduction and $1.82\times$ – $3.55\times$ speedup ($bs=1$), with the average score decreasing from 58.6 to 56.2. Compared with SlimInfer, it delivers comparable speed while preserving a higher score ($2.47\times$ vs.

2.53 $\times$; 58.4 vs. 57.3). It also substantially outperforms the depth-only baseline D³ (58.7% FRR and 2.47 $\times$ vs. 42.3% FRR and 1.87 $\times$), validating joint depth–token optimization with performance-aware fallback.

Table 1. Core performance comparison across models and methods (LLaMA-7B, 32k context)

Method	Model	FRR (%)	ISR ($\times$)	VRR (%)	Δ Score (%)	Avg.Score (%)
FlashAttention2 [12]	LLaMA-7B	-	1	-	-	58.8
SlimInfer [3]	LLaMA-7B	55.2	2.53	48.9	2.55	57.3
D ³ [4]	LLaMA-7B	42.3	1.87	35.1	3.74	56.6
Balcony [5]	LLaMA-7B	45.7	1.59	38.2	3.40	56.8
IG-Pruning [8]	LLaMA-7B	48.2	1.48	40.5	4.76	56.0
D ² TS-LLM (Ours)	LLaMA-7B	58.7	2.47	52.3	0.68	58.4
D ² TS-LLM (Ours)	LLaMA-13B	61.3	2.49	56.6	-	61.2
D ² TS-LLM (Ours)	OPT-6.7B	43.9	1.81	36.4	-	45.4

Notes. FRR denotes FLOPs reduction rate; VRR denotes KV-memory reduction rate. ISR ($\times$) denotes end-to-end inference speedup over the FlashAttention-2 baseline under the same decoding configuration and output length. Avg.Score is the mean score over the five benchmarks. Δ Score (%) = (Avg.Score_{full} - Avg.Score) / Avg.Score_{full} $\times$ 100.

Source: created by the authors.

We observe consistent speedup trends under long-context settings, and the robustness under varying batch sizes is summarized in Table 2.

4.2.2 Ablation and SOTA Comparison

Figure 1 reports an ablation study on the core components (joint depth–token decision, multi-dimensional scoring, and performance-aware fallback), showing that removing any component consistently weakens the efficiency–quality trade-off. Compared with SOTA baselines:

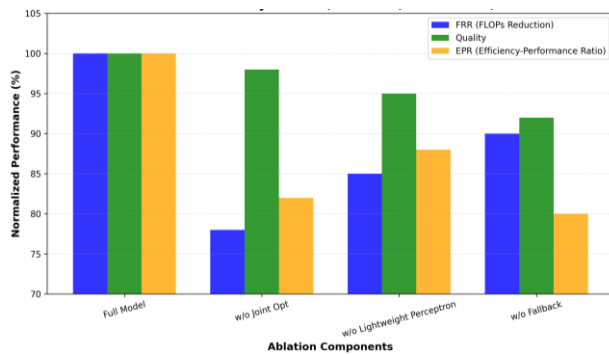


Fig. 1: Ablation study results

Source: created by the authors

Across the five benchmarks, D²TS-LLM consistently preserves accuracy close to the full baseline while improving efficiency, and its averaged score remains higher than representative token-only and depth-only baselines under the same 32k-context setting.

Compared with representative baselines, D²TS-LLM achieves a consistently better efficiency–quality trade-off. Under comparable score degradation constraints (Δ Score within a sub-1% range), it yields higher VRR/FRR and a higher average benchmark score. In addition, compared with hybrid approaches such as ANIRA, D²TS-LLM attains similar efficiency with substantially lower training overhead, since it does not rely on complex supervision.

As shown in Figure 2, D²TS-LLM lies in the favorable region of the Δ Score–FRR trade-off compared with the representative baselines. For example, on LLaMA-7B with 32k context length, D²TS-LLM achieves 58.7% FRR with Δ Score $\approx$ 0.68%, while SlimInfer achieves 55.2% FRR with Δ Score $\approx$ 2.55%, and D³ achieves 42.3% FRR with Δ Score $\approx$ 3.74%.

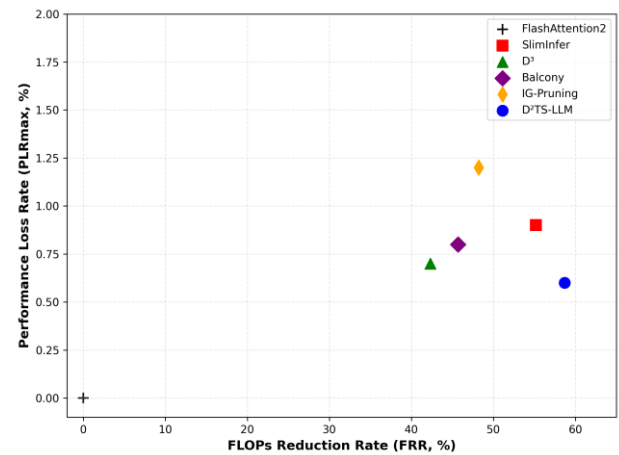


Fig. 2: Trade-off between Δ Score and FRR

Source: created by the authors

4.2.3 Robustness and Generalization

The framework demonstrates strong robustness and generalization:

- It performs stably under different performance constraints ($\epsilon=0.5\%/1\%/2\%/5\%$), adapting to dynamic workloads [11] by adjusting fallback thresholds.
- It generalizes well to unseen datasets (WikiQA, MathQA), outperforming ANIRA [9] by 7%–9% in score, as the multi-dimensional perceptron captures generalizable difficulty patterns.

- It supports various batch sizes (1/4/8) in cloud scenarios, maintaining efficiency gains across workload variations [11].
- Parameters calibrated on LLaMA-7B [2] can be directly transferred to LLaMA-13B [2] and OPT-6.7B with minor performance drop (<1%), showing strong cross-model generalization similar to Balcony [5].

Table 2. Robustness test under different performance constraints and batch sizes

Performance Constraint (ϵ)	FRR (%)	Avg. Score (%)	Batch Size =1	Batch Size= 4	Batch Size = 8
$\epsilon = 0.5\%$	42.3	58.6	$1.82\times$	$2.15\times$	$2.37\times$
$\epsilon = 1.0\%$	58.7	58.4	$2.47\times$	$2.89\times$	$3.12\times$
$\epsilon = 2.0\%$	65.3	57.5	$2.98\times$	$3.42\times$	$3.65\times$
$\epsilon = 5.0\%$	72.1	56.2	$3.55\times$	$3.98\times$	$4.22\times$

Source: created by the authors

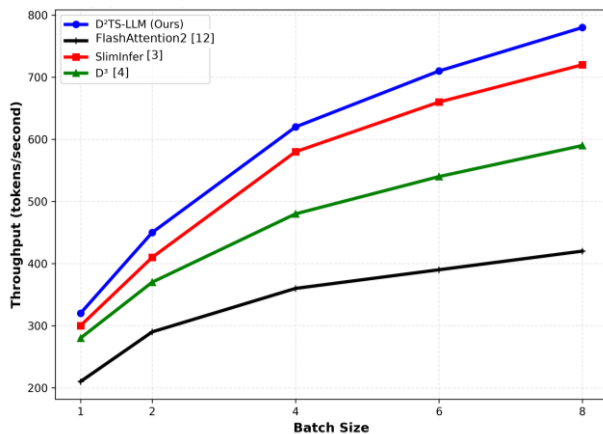


Fig. 3: Throughput comparison across batch sizes
Source: created by the authors.

As shown in Figure 3, D²TS-LLM maintains consistent throughput advantages across different batch sizes compared with the representative baselines. Case analysis on simple, medium, and complex math problems further validates the on-demand resource allocation strategy, which is more adaptive than fixed-ratio optimization methods [3], [7] and position-based depth decay [4].

5 Discussion and Conclusion

5.1 Discussion

The effectiveness of D²TS-LLM stems from three key factors: the synergistic effect of joint depth and token optimization (addressing the limitations of single-dimensional methods [3], [4], [7]), the score of multi-dimensional lightweight scoring (avoiding

the overhead of [8]), and the reliability of performance-aware fallback for real-world safety (filling the gap in [3], [8]). Compared with existing works, our framework offers several advantages:

- It unifies dynamic depth and token selection, achieving higher efficiency than single-dimensional methods [3], [4], [6], [7].
- It requires no extra training or complex optimization, reducing deployment cost compared with [5], [15], [16].
- It supports dynamic workloads [11] and cross-model generalization [5], making it suitable for diverse deployment scenarios.

Despite its strengths, the framework has limitations: it currently supports only autoregressive decoding (consistent with most mainstream LLMs [2]), and the token sharing strategy can be further refined. Future work will extend to non-autoregressive decoding and integrate quantization [17] and KV-Cache optimization [10] for hybrid acceleration, similar to the direction proposed in [15], [16]. Broader impacts include providing a practical solution for efficient and safe LLM deployment on edge and cloud platforms [5], [11], and inspiring further research on dynamic computation allocation for multi-modal models [9].

5.2 Conclusion

We proposed D²TS-LLM, a unified dynamic inference framework that jointly optimizes model depth and token computation for LLMs based on the transformer architecture [1]. With a multi-dimensional lightweight perceptron and performance-aware fallback mechanism, our method achieves substantial efficiency improvements while strictly bounding performance loss. Extensive experiments on representative models [2] and datasets demonstrate its effectiveness and superiority over state-of-the-art approaches [3], [4], [5], [6], [7], [8], [9], [11]. D²TS-LLM is plug-and-play, generalizable, and suitable for real-world LLM deployment scenarios, including dynamic workloads [11] and resource-constrained edge devices [5].

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and

responsibility for the accuracy, integrity, and final content of the publication.

References:

- [1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. Attention Is All You Need. *Advances in Neural Information Processing Systems* 30 (NeurIPS 2017), Long Beach, CA, USA, 2017, pp. 5998-6008, [Online]. <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html> (Accessed Date: January 31, 2026).
- [2] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*, 2023. DOI: 10.48550/arXiv.2302.13971.
- [3] Long, L., Yang, R., Huang, Y., Hui, D., Zhou, A., Yang, J. SlimInfer: Accelerating Long-Context LLM Inference via Dynamic Token Pruning. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol.40, No.38, AAAI-26, Singapore, 2026, pp. 32284-32292. DOI: 10.1609/aaai.v40i38.40502.
- [4] Fan, S., Fang, X., Xing, X., Han, P., Shang, S., Wang, Y. Position-Aware Depth Decay Decoding (D³): Boosting Large Language Model Inference Efficiency. *Findings of the Association for Computational Linguistics: ACL 2025*, Vienna, Austria, 2025, pp. 2990-3001. DOI: 10.18653/v1/2025.findings-acl.154.
- [5] Jamialahmadi, B., Kavehzadeh, P., Rezagholizadeh, M., Farinneya, P., Rajabzadeh, H., Jafari, A., Chen, B., Tahaei, M.S. Balcony: A Lightweight Approach to Dynamic Inference of Generative Language Models. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, 2025, pp. 24853-24867. DOI: 10.18653/v1/2025.emnlp-main.1263.
- [6] Zhao, A., Ye, F., Fan, Y., Tong, J., Xiong, J., Fei, Z., Su, H., Shen, X. SkipGPT: Each Token is One of a Kind. *Proceedings of the 42nd International Conference on Machine Learning*, PMLR Vol.267, Vancouver, BC, Canada, 2025, pp. 77844-77863, [Online]. <https://proceedings.mlr.press/v267/zhao25ab.html> (Accessed Date: January 31, 2026).
- [7] Fu, Q., Cho, M., Merth, T., Mehta, S., Rastegari, M., Najibi, M. LazyLLM: Dynamic Token Pruning for Efficient Long Context LLM Inference. *ES-FoMo II: 2nd Workshop on Efficient Systems for Foundation Models at ICML 2024*, Vienna, Austria, 2024. *arXiv:2407.14057*. DOI: 10.48550/arXiv.2407.14057.
- [8] Qiao, K., Zhang, S., Feng, Y. IG-Pruning: Input-Guided Block Pruning for Large Language Models. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, 2025, pp. 10618-10629. DOI: 10.18653/v1/2025.emnlp-main.537.
- [9] Moosa, I.M., Lohit, S., Wang, Y., Chatterjee, M., Yin, W. Understanding Dynamic Compute Allocation in Recurrent Transformers. *Proceedings of the 43rd International Conference on Machine Learning (ICML 2026)*, Seoul, South Korea, 2026. *arXiv:2602.08864*. DOI: 10.48550/arXiv.2602.08864.
- [10] Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J.E., Zhang, H., Stoica, I. Efficient Memory Management for Large Language Model Serving with PagedAttention. *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP 2023)*, Koblenz, Germany, 2023, pp. 611-626. DOI: 10.1145/3600006.3613165.
- [11] Hidayetoglu, M., Qiao, A., Wyatt, M., Rasley, J., He, Y., Rajbhandari, S. Shift Parallelism: Low-Latency, High-Throughput LLM Inference for Dynamic Workloads. *Proceedings of the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS 2026)*, Pittsburgh, PA, USA, 2026, pp. 1749-1763. DOI: 10.1145/3779212.3790219.
- [12] Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *Advances in Neural Information Processing Systems* 35 (NeurIPS 2022), New Orleans, LA, USA, 2022, pp. 16344-16359. DOI: 10.52202/068431-1189.
- [13] Wu, W., Pan, Z., Fu, K., Wang, C., Chen, L., Bai, Y., Wang, T., Wang, Z., Xiong, H. TokenSelect: Efficient Long-Context Inference and Length Extrapolation for LLMs via Dynamic Token-Level KV Cache Selection. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, Association for Computational Linguistics, 2025, pp. 21264-

21281. DOI: 10.18653/v1/2025.emnlp-main.1079.
- [14] Luo, X., Wang, W., Yan, X. DiffSkip: Differential Layer Skipping in Large Language Models. Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria, Association for Computational Linguistics, 2025, pp. 7221-7231. DOI: 10.18653/v1/2025.findings-acl.377.
- [15] Chen, Z., Li, D., Zhang, M., Hu, B., Zhang, M. Dynamic Long Context Reasoning over Compressed Memory via End-to-End Reinforcement Learning. Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), San Diego, California, USA, 2026, pp. 8064-8083. DOI: 10.18653/v1/2026.acl-long.365.
- [16] Tan, W., Li, J., Ju, J., Luo, Z., Song, R., Luan, J. Think Silently, Think Fast: Dynamic Latent Compression of LLM Reasoning Chains. Advances in Neural Information Processing Systems 38 (NeurIPS 2025), San Diego, CA, USA / Mexico City, Mexico, 2025, pp. 5220-5242. DOI: 10.52202/085713-0164.
- [17] Frantar, E., Ashkboos, S., Hoefler, T., Alistarh, D. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. The Eleventh International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda, 2023. DOI: 10.48550/arXiv.2210.17323.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Conceptualization, methodology, experiments, analysis, and writing—original draft: all authors. Writing—review & editing: all authors.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US