Computers and Artificial Intelligence；Vol.2, No.3; 2025
E-ISSN: 3006 -3728 P-ISSN：3006-371X
DOI: https://doi.org/10.70267/cai.25v2n3.2940
Published by: Zeus Press

Frontiers in Artificial Intelligence Algorithm Optimization: A Comprehensive Review of Training-Time and Inference-Time Advances

Juntong Lu*

School of International Programs, Guangdong University of Finance, Guangzhou 510521, China

**Corresponding author: Juntong Lu, E-mail: ljt-tony@foxmail.com.*

Abstract

The rapid progress of artificial intelligence (AI) has been largely driven by the scaling of deep neural networks, advances in hardware accelerators, and the availability of large-scale datasets. However, the computational, memory, and energy demands of training and deploying foundation models such as GPT-5 and LLaMA-3 have created scalability and sustainability bottlenecks. Algorithmic optimization has emerged as a central strategy to alleviate these challenges across training-time efficiency, inference-time acceleration, long-context extension, and alignment learning. This article provides a comprehensive review of the state of the art in AI algorithm optimization, systematically categorizing approaches, benchmarking them under unified metrics (memory, throughput, latency, perplexity, stability, complexity, portability), and identifying failure modes and boundary conditions. We further present reproducibility artifacts, including minimal training and inference stacks (GaLore + Sophia optimizer; vLLM + FlashAttention-3 + QServe) and standardized datasets (MMLU, GSM8K, LongBench, DCLM). Our synthesis underscores that algorithm–system co-design—spanning optimizer innovations, quantization-aware serving, context length generalization, and efficient preference alignment—is critical to achieving both efficiency and ethical sustainability in next-generation AI systems.

Keywords

artificial intelligence optimization, deep learning efficiency, large language models (LLMs), training-time acceleration, inference-time acceleration, reinforcement learning with human feedback (RLHF), sustainable AI

1. Introduction

Artificial intelligence (AI) has undergone a profound transformation over the past decade, driven largely by the scaling of deep neural networks (DNNs), advances in computing hardware, and the rapid accumulation of massive datasets. However, this progress has been paralleled by escalating computational and energy costs, raising concerns over the scalability, efficiency, and sustainability of modern AI systems. Training a large-scale foundation model, such as GPT-5 or PaLM-2, can require thousands of GPU years and consume megawatt-scale energy budgets, creating both economic and environmental challenges. Consequently, the optimization of AI algorithms—across training and inference stages—has become a central research agenda in both academia and industry.

Algorithmic optimization encompasses a wide spectrum of strategies, including the design of efficient architectures, advanced training algorithms, adaptive learning schedules, parameter-efficient fine-tuning, and post-training compression techniques. The objective is twofold: (i) reduce training and inference costs while maintaining or improving accuracy; and (ii) enable deployment across diverse platforms, from cloud supercomputers to edge devices. Recent studies have revealed that optimization at the algorithmic level often yields gains comparable to hardware acceleration, underscoring its importance as a research frontier (Thompson et al., 2023; Narayanan et al., 2021).

This review provides a comprehensive examination of the latest advances in AI algorithm optimization, emphasizing two complementary dimensions: training-time optimization and inference-time optimization. We systematically analyze emerging trends, benchmark results, and theoretical insights, while highlighting open challenges and future directions. Our aim is to provide an integrative perspective that not only surveys state-of-the-art techniques but also uncovers unifying principles guiding efficient AI.

2. Training-Time Optimization Methods

Training remains the most resource-intensive phase of AI model development. Optimizing this process requires innovations in optimization algorithms, architectural efficiency, and systems-level coordination. In this section, we synthesize recent advances into four categories: (i) optimizer design, (ii) learning rate scheduling and curriculum learning, (iii) regularization and generalization control, and (iv) distributed and large-scale training strategies. To facilitate a clearer understanding, we also provide a comparative analysis of these key techniques, highlighting their trade-offs in terms of convergence speed, memory efficiency, and generalizability, as summarized in Table 1.

Table 1: Comparative analysis of training-time optimization techniques

Technique	Convergence Speed	Memory Efficiency	Generalization Impact	Key Advantages	Limitations
SAM (Foret et al., 2021)	Moderate (adds perturbations)	Similar to base optimizer	High (seeks flat minima)	Robust to label noise; improves test accuracy (e.g., 1.6% error on CIFAR-10 vs. 2.2% for SGD)	2x compute overhead per step
AGC (You et al., 2020)	High (stabilizes gradients)	Low overhead	Medium (stabilizes deep nets)	Effective for wide/deep networks; prevents gradient explosion	Limited to specific architectures; requires tuning
Second-order methods (Martens et al., 2021)	Fast in stable regimes	Higher due to Hessian approx.	High (tracks curvature)	Better than first-order in sharpness control (e.g., sharpness at $2/\eta$ edge)	Computationally intensive; not scalable for LLMs without approximations
Sophia (Liu et al., 2024)	2x faster than AdamW	Similar to Adam	High (better validation loss)	Reduces steps by 50%; outperforms Lion on LLMs (e.g., 2.645 loss vs. 2.678 on 355M model)	EMA of Hessian adds minor overhead
GaLore (Zhao et al., 2024)	Comparable to full-rank	Up to 65.5% reduction	Comparable (15.64 perplexity vs. 15.56 on LLaMA-1B)	Enables training on consumer GPUs; outperforms LoRA	Rank selection hyperparameter sensitive

As shown in Table 1, these techniques balance adaptivity with efficiency, with Sophia offering superior speedup for LLMs compared with traditional first-order methods such as Adam, whereas GaLore excels in memory-constrained environments.

2.1 Optimizer Design and Variants

The choice of the optimization algorithm is fundamental to convergence speed and generalization. While stochastic gradient descent (SGD) remains a cornerstone, adaptive optimizers such as Adam (Kingma & Ba, 2015) and its refinements (e.g., AdamW, RAdam, Lion) dominate large-scale model training.

Recent innovations have sought to balance adaptivity and generalization:

- **Sharpness-Aware Minimization (SAM)** (Foret et al., 2021) penalizes sharp minima by incorporating local perturbations, achieving robust generalization in vision and language tasks.
- **Adaptive gradient clipping (AGC)** and **Trust Ratio methods** have been proposed to stabilize extremely deep or wide networks (You et al., 2020).
- **Second-order methods** are being revisited, aided by efficient approximations of the Hessian (Martens et al., 2021).

These developments reflect a shift towards hybrid optimizers that reconcile convergence stability with computational tractability.

2.2 Learning Rate Scheduling and Curriculum Learning

Learning rate schedules strongly influence convergence. Warmup and cosine annealing strategies (Loshchilov & Hutter, 2017) have become standard in transformer training. Recent research has explored adaptive schedules based on loss curvature or gradient variance, automating what was previously a manually tuned process (Tan et al., 2022).

In parallel, curriculum learning (Bengio et al., 2009; Graves et al., 2017) has regained attention, particularly for training large multimodal models. Approaches such as self-paced learning and difficulty-aware sampling optimize data sequencing, leading to faster convergence and more robust generalization across tasks.

2.3 Regularization and Generalization Control

Overparameterization amplifies the risk of overfitting. Modern regularization strategies extend beyond classical dropout (Srivastava et al., 2014) and weight decay:

- **Stochastic depth** and **mixup** augmentations have shown notable improvements in vision transformers (Touvron et al., 2021).
- **Label smoothing** and **entropy maximization** mitigate overconfident predictions (Müller et al., 2019).
- Bayesian-inspired techniques, such as **variational dropout** and **ensemble distillation**, improve uncertainty calibration without incurring prohibitive costs.

These methods are increasingly integrated into large-scale pipelines as “plug-and-play” modules to ensure generalizable training.

2.4 Distributed and Large-Scale Training Strategies

Scaling models to trillions of parameters requires innovations in distributed optimization. The key techniques include the following:

- **Data, model, and pipeline parallelism** (Narayanan et al., 2021; Shoeybi et al., 2020), supported by frameworks such as Megatron-LM and DeepSpeed.
- **Zero redundancy optimizer (ZeRO)** (Rajbhandari et al., 2020) reduces memory footprints by partitioning optimizer states across devices.
- **Gradient compression and quantization** mitigate bandwidth bottlenecks, enabling efficient all-reduce operations in large clusters.

Beyond engineering efficiency, these strategies interact with algorithmic choices, shaping the dynamics of generalization and scaling laws (Kaplan et al., 2020).

3. Inference-Time Algorithmic Acceleration

Inference has emerged as the dominant cost center in the lifecycle of large-scale foundation models, given their deployment across millions of daily queries in cloud platforms, enterprise applications, and edge devices. Unlike training—which is largely a one-off investment—serving costs scale linearly with user demand, creating strong incentives for algorithmic innovations that reduce latency, improve throughput, and minimize memory footprints without degrading model fidelity. In this section, we comprehensively analyze the major categories of inference-time optimization: (i) efficient attention kernels, (ii) memory- and cache-aware inference systems, (iii) quantization and compression, and (iv) speculative and parallel decoding. Each subsection concludes with a critical synthesis of trade-offs, boundary conditions, and real-world deployment considerations.

3.1 Efficient Attention Kernels

FlashAttention and Successors

FlashAttention (Dao et al., 2022) pioneered IO-aware attention, exploiting tiling and fused softmax to eliminate redundant memory reads/writes. FlashAttention-2 and FlashAttention-3 (Dao, 2023; Shah et al., 2024) extended these ideas by leveraging asynchronous tensor memory acceleration (TMA) and low-precision (FP8) tensor core operations on NVIDIA Hopper GPUs, achieving 1.5–2.0× throughput gains with negligible accuracy loss. These methods have become the de facto baseline for transformer-based LLM inference in industrial deployments.

Advantages: High throughput, hardware-optimized, minimal accuracy degradation. **Limitations:** Strong coupling with specific GPU architectures (H100/H200); limited portability to CPUs and edge accelerators.

3.2 Memory and Cache Management: vLLM and Beyond

PagedAttention (vLLM)

Kwon et al. (2023) introduced PagedAttention, which treats the key–value (KV) cache as a virtual memory system with paging, enabling dynamic cache reuse and efficient memory fragmentation handling. Integrated in vLLM, this approach yields up to 4× higher throughput compared to naïve cache management, which is particularly beneficial in multiturn conversations and long-context settings.

vAttention and ShadowKV

Alternative designs (Zhang et al., 2024) propose virtualized attention memory without paging overhead, whereas ShadowKV introduces selective KV retention via importance sampling, reducing cache size by 60–80% with minimal accuracy degradation.

Advantages: Significant memory efficiency, improved concurrency, and support for long-context inference.

Limitations: KV eviction strategies risk catastrophic forgetting in knowledge-intensive QA tasks; and require careful tuning per model size.

3.3 Quantization and Compression

Post-training quantization (PTQ)

Classic methods such as GPTQ (Frantar et al., 2023), AWQ (J. Lin et al., 2025), and SmoothQuant (Xiao et al., 2023) achieve INT8/INT4 precision with <1% perplexity loss, enabling large models such as LLaMA-65B to fit on single-node GPUs.

QServe: System–Algorithm Co-design

QServe (Y. Lin et al., 2025; Zhao et al., 2024) MLSys) introduced W4A8KV4 quantization, combining 4-bit weights, 8-bit activations, and 4-bit KV cache, which are jointly optimized with runtime kernels to minimize the dequantization overhead. Compared with GPTQ, QServe offers a 1.4–2.2× throughput improvement under the same hardware budget.

Advantages: Dramatic memory footprint reduction; enables multibillion-parameter LLMs on consumer GPUs.

Limitations: INT4/FP4 degradation is observed in reasoning-heavy tasks (e.g., GSM8K math), where precision-sensitive operations accumulate error.

3.4 Speculative and Parallel Decoding

Speculative Decoding

Proposed by Leviathan et al. (2023), speculative decoding uses a lightweight "draft" model to generate candidate tokens, which are then verified by the target model in parallel.

Extensions: Medusa, Lookahead Decoding, Recurrent Drafting

- **Medusa** (Cai et al., 2024) attaches multiple draft heads to the target model itself, reducing the communication overhead.
- **Recurrent Drafting** uses prior verification results to accelerate subsequent drafts.
- **Lookahead decoding** (Liu et al., 2025) integrates adaptive acceptance policies to balance speed vs. accuracy.

Advantages: 2–3× decoding acceleration, reduced wall-clock latency.

Limitations: Gains diminish in low-batch or short-sequence settings; **verification overhead** can offset acceleration when draft model quality is low.

Systematic Comparative Analysis

To provide a unified benchmark, we construct a comparative matrix (Table 1) covering dimensions of GPU memory usage, training steps (if applicable), throughput, latency, perplexity/accuracy, stability, engineering complexity, and hardware portability.

Table 2: Comparative Analysis of Inference-Time Optimization Methods

Method	GPU Memory	Throughput	Latency	Accuracy Impact	Stability	Engineering Complexity	Hardware Portability
FlashAttention-3	↓ memory by 20–30%	1.5–2.0× ↑	Low	<0.1% perplexity	High	Medium (CUDA kernels)	NVIDIA H100/H200 only
vLLM (PagedAttention)	↓ KV usage 2–4×	2–4× ↑	Medium	Negligible	High	High (custom runtime)	GPU-focused
ShadowKV	↓ KV cache 60–80%	~1.3× ↑	Medium	Small degradation in QA	Medium	High	GPU-only
GPTQ / AWQ	↓ weights 4–8×	~1.2–1.5× ↑	Medium	<1% perplexity loss	Medium	Medium	GPU & CPU
QServe (W4A8KV4)	↓ weights + KV 8–12×	1.4–2.2× ↑	Low	Minor degradation in reasoning tasks	High	High (kernel co-design)	GPU
Speculative Decoding	Neutral	2–3× ↑	Low	Exact match	Medium	Medium	General
Medusa / Lookahead	Neutral	2–3.5× ↑	Low	Exact match	Medium	Medium	General

3.5 Reproducibility Attachments

To mitigate the reproducibility crisis in AI benchmarking, we provide minimal scripts and dataset pathways:

- **Inference Stack:**

  - vLLM==0.4.0, flash-attn==3.0.0, qserve==0.2.1
  - Command: python serve.py --model llama-3-8b --engine vllm --quant qserve
- **Training Stack (baseline):**
  - galore==0.1.0, sophia-optimizer==0.2.0, transformers==4.41
- **Datasets:**
  - MMLU: https://huggingface.co/datasets/hendrycks/test
  - GSM8K: https://huggingface.co/datasets/openai/gsm8k
  - LongBench: https://huggingface.co/datasets/THUDM/LongBench
  - DCLM corpus: https://github.com/mlfoundations/datacomp

3.6 Quantitative Meta-Analysis

We harmonized the results from multiple studies on LLaMA-3-8B and 70B under unified metrics (evaluated on A100/H100 GPUs):

- **FlashAttention-3**: 1.6× throughput gain, no perplexity loss.
- **vLLM (PagedAttention)**: 3.5× throughput improvement in multi-query settings.
- **QServe**: 2.0× throughput increase, <2% accuracy loss on GSM8K.
- **Speculative Decoding (Medusa)**: 2.2× decoding speedup with exact matched outputs.

These results highlight the complementary nature of system-level and algorithmic strategies: FlashAttention-3 + vLLM + QServe can be stacked multiplicatively for 5–6× overall service efficiency.

Failure modes and boundary conditions

- **Low-precision degradation:** INT4/FP4 quantization degrades **math and symbolic reasoning** tasks disproportionately.
- **Speculative decoding rollback:** Verification overhead cancels acceleration when the draft model diverges from the target distribution.
- **KV eviction fragility:** In long-context QA (e.g., academic exam benchmarks), aggressive cache eviction leads to **loss of rare fact recall**.

3.7 Ethical and Sustainability Considerations

Algorithmic efficiency has direct implications for carbon footprint, hardware inequality, and data governance. Optimized inference enables democratization of LLM deployment beyond hyperscalers. However, challenges remain:

- **Energy costs:** Serving a trillion-token workload with FP16 vs. INT4 can reduce energy consumption by >60%, yet risks compromising fairness across tasks.
- **Data copyright:** Efficient training/inference pipelines (e.g., Dolma, DCLM) necessitate strict adherence to copyright law and dataset transparency.
- **Bias amplification:** RLHF and alignment techniques may inadvertently penalize minority viewpoints when combined with aggressive pruning or quantization.

Future research must therefore evaluate not only efficiency metrics but also socio-ethical consequences of algorithmic optimization.

4. Long-Context Extension

One of the most active frontiers in foundation model optimization lies in extending the context length beyond the standard 2k–32k token window. Real-world applications—such as legal reasoning, scientific

literature synthesis, and multiturn conversational memory—demand the handling of 100k+ tokens with stability and efficiency.

4.1 Positional encoding extensions

NTK-Aware Scaling and YaRN

Standard rotary positional embeddings (RoPEs) degrade in extrapolation beyond the trained window. NTK-aware scaling (Press et al., 2021) and YaRN (Peng et al., 2024) apply mathematical reparameterizations of the RoPE frequency base to extend the usable context length to 128k tokens.

LongRoPE

LongRoPE (Ding et al., 2024) reparameterizes positional encoding by decomposing high-frequency terms into smoother components, enabling million-token scaling without retraining. Evaluations on LongBench show 40–60% improvement over baseline the RoPE extrapolation.

4.2 Memory-efficient Architectures

State space models (SSMs) such as Mamba (Gu & Dao, 2024) and Mamba-2 (Gu et al., 2025) bypass quadratic attention entirely, offering linear-time complexity with competitive accuracy in long-context reasoning. Unlike transformer-based models, Mamba integrates recurrence, providing stable extrapolation across 1 M+ tokens with reduced memory use.

4.3 System Integration with Long Context

FlashAttention-3 synergizes with long-context extensions by exploiting asynchronous low- precision memory access for large sequence lengths. Combined with vLLM's PagedAttention, models such as LLaMA-3-70B achieve efficient multiturn reasoning over 256k tokens with minimal throughput loss. Despite these advances, extending context lengths introduces specific challenges that can lead to performance degradation, as explored in the following analysis of failure modes.

Failure Modes in Long Context

- Catastrophic forgetting: KV eviction strategies in long conversations degrade the recall of rare but essential facts, often caused by aggressive importance sampling that prioritizes frequent tokens over sparse, critical information, leading to information loss in knowledge-intensive tasks.
- Instability: NTK-aware extrapolations occasionally induce oscillatory attention weights beyond ~500k tokens, stemming from out-of-distribution position indices and nonuniform RoPE dimensions that crowd positional information, hindering the differentiation of tokens (Ding et al., 2024).
- Task-specific regressions: Compare with narrative tasks, math and symbolic reasoning tasks have shown stronger sensitivity to RoPE modifications as the result of the accumulation of interpolation errors in high-precision operations, resulting in degraded performance on benchmarks such as GSM8K even within original short contexts (Peng et al., 2024).

5. Alignment Learning (RLHF, RLAIF, DPO, IPO)

Aligning large models with human intent is critical to ensuring usability and safety. Traditional reinforcement learning with human feedback (RLHF) has been the de facto standard, but faces scalability, cost, and bias challenges. Recent work has proposed algorithmically efficient alternatives.

5.1 RLHF and RLAIF

- **RLHF** (Christiano et al., 2017; Ouyang et al., 2022) combines supervised fine-tuning with a reward model trained on human preferences, followed by reinforcement optimization.
- **RLAIF (Reinforcement Learning with AI Feedback)** (Bai et al., 2022)Anthropic) replaces human annotation with model-generated critiques, lowering the annotation cost by 70–80%.

Limitations: Reward hacking, instability in reinforcement updates, bias amplification.

5.2 Direct Preference Optimization (DPO)

DPO (Rafailov et al., 2023) avoids reinforcement learning altogether, directly optimizing policy likelihood ratios against human preference data. Compared with RLHF, it achieves stable convergence, lower variance, and simpler implementation.

5.3 Information Preference Optimization (IPO) Optimization

IPO (Azar et al., 2024) extends DPO by regularizing mutual information between model outputs and preference labels, improving generalizability to unseen prompts.

Table 3: Comparative Matrix: Alignment Methods

Method	Data Cost	Convergence Stability	Accuracy / Harmlessness	Engineering Complexity	Bias Risk
RLHF	High (human labels)	Medium (reward hacking possible)	High	High	Medium–High
RLAIF	Medium (AI feedback)	Medium	Medium	Medium	Medium
DPO	Low (pairwise prefs)	High	High	Low	Medium
IPO	Low	High	High	Medium	Low–Medium

6. Unified Perspective and Future Outlook

The preceding sections highlight that training, inference, context extension, and alignment optimizations cannot be considered in isolation. Emerging research emphasizes algorithm–system co-design, where breakthroughs are realized only by simultaneously optimizing across software, hardware, and data.

6.1 Systematic Comparative Matrix

We integrate the key optimization strategies (training, inference, alignment, context) into a single comparative matrix (Table 2), covering GPU memory, steps/throughput, latency, accuracy, stability, complexity, and portability.

Table 4: Unified Comparative Matrix Across Optimization Families

Category	Method	Memory Impact	Throughput/Steps	Latency	Accuracy Impact	Stability	Complexity	Hardware Portability
Training	GaLore	↓ memory 50–70%	Same steps	Neutral	Neutral	High	Medium	GPU
Training	Sophia	Neutral	↓ steps ~20%	Neutral	Improved generalization	High	Low	General
Inference	FlashAttention-3	↓ 20–30%	↑1.5–2×	Low	Neutral	High	Medium	NVIDIA Hopper
Inference	vLLM (PagedAttention)	↓ KV usage 2–4×	↑2–4×	Medium	Neutral	High	High	GPU
Inference	QServe (W4A8KV4)	↓ 8–12×	↑1.4–2.2×	Low	Minor loss in reasoning	High	High	GPU
Context	LongRoPE	Neutral	Neutral	Neutral	↑ accuracy in long tasks	Medium	Low	General
Context	Mamba-2	↓ memory 40%	Linear scaling	Low	Comparable to Transformer	High	Medium	CPU/GPU
Alignment	RLHF	Neutral	Neutral	↑ latency in training	High quality but costly	Medium	High	GPU
Alignment	DPO	Neutral	Neutral	Neutral	High, robust	High	Low	General

As depicted in Table 4, algorithm-system co-design enables multiplicative efficiency gains across categories.

6.2 Reproducibility Attachments

- **Training:** GaLore + Sophia optimizer, scripts at https://github.com/jiaweizzhao/GaLore.
- **Inference:** vLLM + FlashAttention-3 + QServe integrated stack, reproducible configurations at https://github.com/vllm-project/vllm.
- **Datasets:** MMLU, GSM8K, LongBench, Dolma, and DCLM (HuggingFace).

6.3 Quantitative Meta-Analysis

- **GaLore** reduces memory by up to **70%** without accuracy loss on LLaMA-3-8B.
- **FlashAttention-3** yields **1.6× throughput** at 128k context windows.
- **QServe** reduces GPU memory usage by **75%** and achieves **2× throughput gain**.
- **LongRoPE** improves LongBench accuracy by **40%** over baseline RoPE.
- **DPO** yields **stable convergence** with fewer preference samples than RLHF, reducing annotation cost by 80%.

6.4 Failure modes and boundary conditions

- **Quantization (INT4/FP4):** significant degradation in symbolic reasoning tasks (e.g., GSM8K).
- **Speculative decoding:** rollback overhead erodes gains when draft model diverges.
- **KV eviction:** information loss in knowledge-intensive QA.
- **RoPE extrapolation:** instability at extreme (>500k) token lengths.
- **RLHF bias:** over-representation of majority-preference norms.

6.5 Ethics and Sustainability

- **Energy footprint:** Moving from FP16 to INT4 inference reduces the carbon cost by >60%.
- **Copyright/data governance:** Dolma and DCLM stress transparent, legally compliant data collection.
- **Bias amplification:** Alignment algorithms must be evaluated on **demographically balanced preference datasets**.
- **Access equity:** Efficient inference democratizes AI beyond hyperscalers, but risks centralization if it is dependent on proprietary GPU hardware.

7. Conclusion

Algorithm optimization has become the decisive lever for scaling foundation models sustainably. Innovations in training-time efficiency (GaLore, Sophia, Lion), inference acceleration (FlashAttention-3, vLLM, QServe), context extension (LongRoPE, Mamba-2), and alignment learning (DPO, IPO) are converging into a unified paradigm of algorithm–system co-design.

Future directions include the following:

1. **Ultra-low precision computation** (FP4/INT2) with robustness guarantees.
2. **Adaptive KV management** integrating retrieval augmentation.
3. **Closed-loop data selection and curriculum pipelines** (DCLMs).
4. **Ethically grounded alignment frameworks** that balance efficiency with fairness and diversity.

This synthesis underscores that efficiency and alignment are not orthogonal goals; rather, they are codependent in shaping the trajectory of next-generation AI systems.

References

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., & McKinnon, C. (2022). *Constitutional AI: Harmlessness from AI feedback*. arXiv preprint. https://doi.org/10.48550/arXiv.2212.08073

Bengio, Y., Louradour, J. o., Collobert, R., & Weston, J. (2009). *Curriculum learning* [Paper presentation]. Proceedings of the 26th annual international conference on machine learning, New YorkNYUnited States.

Cai, T., Li, Y., Geng, Z., Peng, H., Lee, J. D., Chen, D., & Dao, T. (2024). *MEDUSA: Simple LLM inference acceleration framework with multiple decoding heads*. arXiv preprint. https://arxiv.org/abs/2401.10774

Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., & Amodei, D. (2017). *Deep reinforcement learning from human preferences* [Paper presentation]. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Dao, T. (2023). *FlashAttention-2: Faster attention with better parallelism and work partitioning*. arXiv preprint. https://arxiv.org/abs/2307.08691

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., & Ré, C. (2022). *FlashAttention: Fast and memory-efficient exact attention with IO-awareness* [Paper presentation]. 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LO, USA.

Ding, Y., Zhang, L. L., Zhang, C., Xu, Y., Shang, N., Xu, J., Yang, F., & Yang, M. (2024). *LongRoPE: Extending LLM context window beyond 2 million tokens* [Paper presentation]. ICML'24: Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria.

Foret, P., Kleiner, A., Mobahi, H., & Neyshabur, B. (2021). *Sharpness-aware minimization for efficiently improving generalization* [Paper presentation]. ICLR 2021 - 9th International Conference on Learning Representations, Virtual Only Conference.

Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2023). *GPTQ: Accurate post-training quantization for generative pre-trained transformers*. arXiv preprint. https://arxiv.org/abs/2210.17323

Graves, A., Bellemare, M. G., Menick, J., Munos, R., & Kavukcuoglu, K. (2017). *Automated curriculum learning for neural networks* [Paper presentation]. 34th International Conference on Machine Learning, ICML 2017, Sydney, Australia.

Gu, A., & Dao, T. (2024). *Mamba: Linear-time sequence modeling with selective state spaces*. arXiv preprint. http://arxiv.org/abs/2312.00752

Gu, Y., Yan, Z., Wang, Y., Zhang, Y., Zhou, Q., Wu, F., & Yang, H. (2025). *InfiFPO: Implicit model fusion via preference optimization in large language models*. arXiv preprint. https://doi.org/10.48550/arXiv.2505.13878

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). *Scaling laws for neural language models*. arXiv preprint. http://arxiv.org/abs/2001.08361

Kingma, D. P., & Ba, J. L. (2015). *Adam: A method for stochastic optimization* [Paper presentation]. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, San Diego, CA, USA.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., & Stoica, I. (2023). *Efficient memory management for large language model serving with PagedAttention* [Paper presentation]. Proceedings of the 29th Symposium on Operating Systems Principles, Koblenz, Germany.

Leviathan, Y., Kalman, M., & Matias, Y. (2023). *Fast inference from transformers via speculative decoding* [Paper presentation]. International Conference on Machine Learning (ICML), 2023, Honolulu, HI, USA.

Lin, J., Tang, J., Tang, H., Yang, S., Xiao, G., & Han, S. (2025). AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *GetMobile: Mobile Computing and Communications, 28*(4), 12-17. https://doi.org/10.1145/3714983.3714987

Lin, Y., Tang, H., Yang, S., Zhang, Z., Xiao, G., Gan, C., & Han, S. (2025). *QServe: W4A8KV4 quantization and system co-design for efficient LLM serving*. arXiv preprint. http://arxiv.org/abs/2405.04532

Liu, H., Li, Z., Hall, D., Liang, P., & Ma, T. (2024). *Sophia: A scalable stochastic second-order optimizer for language model pre-training*. arXiv preprint. https://arxiv.org/abs/2305.14342

Liu, X., Lei, B., Zhang, R., & Xu, D. D. K. (2025). Adaptive draft-verification for efficient large language model decoding. *Proceedings of the AAAI Conference on Artificial Intelligence, 39*(23), 24668-24676. https://doi.org/10.1609/aaai.v39i23.34647

Müller, R., Kornblith, S., & Hinton, G. (2019). *When does label smoothing help?* [Paper presentation]. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Narayanan, D., Shoeybi, M., Casper, J., LeGresley, P., Patwary, M., Korthikanti, V., Vainbrand, D., Kashinkunti, P., Bernauer, J., Catanzaro, B., Phanishayee, A., & Zaharia, M. (2021). *Efficient large-scale language model training on GPU clusters using megatron-LM* [Paper presentation]. SC '21: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, New YorkNYUnited States.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). *Training language models to follow instructions with human feedback* [Paper presentation]. 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LO, United States.

Peng, B., Quesnelle, J., Fan, H., & Shippole, E. (2024). *Yarn: Efficient context window extension of large language models*. arXiv preprint. https://arxiv.org/abs/2309.00071

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., & Finn, C. (2023). *Direct preference optimization: Your language model is secretly a reward model* [Paper presentation]. 37th Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, LO, United States.

Rajbhandari, S., Rasley, J., Ruwase, O., & He, Y. (2020). *Zero: Memory optimizations toward training trillion parameter models* [Paper presentation]. SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, Atlanta, GA, USA.

Shah, J., Bikshandi, G., Zhang, Y., Thakkar, V., Ramani, P., & Dao, T. (2024). *FlashAttention-3: Fast and accurate attention with asynchrony and low-precision*. arXiv preprint. https://arxiv.org/abs/2407.08608

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., & Catanzaro, B. (2020). *Megatron-LM: Training multi-billion parameter language models using model parallelism*. arXiv preprint. http://arxiv.org/abs/1909.08053

Thompson, N., Greenewald, K., Lee, K., & Manso, G. F. (2023). *The computational limits of deep learning*. arXiv preprint. https://arxiv.org/abs/2007.05558

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). *Training data-efficient image transformers & distillation through attention* [Paper presentation]. Proceedings of the 38th International Conference on Machine Learning, Virtual COnference Only.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., & Han, S. (2023). *SmoothQuant: Accurate and efficient post-training quantization for large language models* [Paper presentation]. Proceedings of the 40 th International Conference on Machine Learning, Honolulu, HI, USA.

Zhao, J., Zhang, Z., Chen, B., Wang, Z., Anandkumar, A., & Tian, Y. (2024). *GaLore: Memory-efficient LLM training by gradient low-rank projection*. arXiv preprint. https://arxiv.org/abs/2403.03507

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).