Human Factors in Artificial Intelligence and Social Computing, Vol. 203, 2026, 141–150
https://doi.org/10.54941/ahfe1008074

A Unified Taxonomy of Deep Learning Optimizers for Scalable and Efficient AI Systems

Carlos Villarreal, Jonathan Luzuriaga, Emilio Quinga, Nicolás Reinoso, Bryan Morales, and Diana Martinez-Mosquera

Department of Informatics and Computer Science, Escuela Politécnica Nacional, Quito, Ecuador

ABSTRACT

The rapid advancement of artificial intelligence (AI), particularly large language models (LLMs), has created a significant s ocio-technical divide. The immense computational resources required for AI training increasingly limit participation to a few well-funded entities, hindering the democratization of AI research and raising concerns about environmental sustainability. While optimization algorithms are critical to reducing these resource barriers, the current landscape is highly fragmented, offering limited practical guidance for practitioners in resource-constrained environments. To address this accessibility gap, we present a unified taxonomy of deep learning optimizers that systematically organizes methods by their order of information: zeroth, first, and second order, while integrating emerging, IO-aware and Flash attention paradigms. Instead of merely enumerating algorithms, our approach emphasizes cost-efficiency, memory usage, and hardware constraints as pivotal factors for equitable AI development. Our synthesis of the literature reveals that system-level considerations, particularly IO efficiency, are essential not just for computational p erformance, but for making large-scale AI accessible. We introduce a decision-oriented framework that translates theoretical insights into practical guidelines, establishing a structured foundation for broader communities to train and deploy human-centered AI systems sustainably and efficiently.

Keywords: Artificial intelligence, Computational efficiency, Deep learning, Efficient AI, Hardware/IO-aware, Optimization

INTRODUCTION

The rise of large language models (LLMs) and foundational Transformers has redefined the performance frontier in natural language processing (NLP), code synthesis, and multimodal reasoning. However, this progress carries a profound structural cost: training a competitive LLM now requires tens of thousands of Graphics Processing Unit (GPU) hours and hundreds of megawatts of electricity. This dynamic constitutes a growing socio-technical divide where the computational requirements of frontier AI research effectively exclude universities, independent laboratories, public institutions,

Received May 22, 2026; Revised May 31, 2026; Accepted June 15, 2026; Available online July 20, 2026

and practitioners from resource-constrained economies (Brown et al., 2020; Patterson et al., 2021). The resulting concentration of model-building capacity raises concerns not only about scientific reproducibility but also about the equitable governance of AI systems whose outputs increasingly shape critical human decisions.

From a Human Systems Integration (HSI) perspective, this computational barrier imposes a severe cognitive load on researchers and engineers. The choice of optimization algorithm directly governs training throughput, memory consumption, convergence stability, and the monetary and energetic cost of a training run. Despite this centrality, the optimizer landscape remains highly fragmented. Practitioners are forced to navigate classic stochastic methods, adaptive variants, quasi-Newton approximations, parameter-free optimizers, and hardware-aware system-level techniques. This frtagmentation severely increases decision fatigue and experimental friction, offering limited comparative guidance that simultaneously addresses computational cost, hardware constraints, and operational accessibility (Ruder, 2016; Schmidt et al., 2021).

The proliferation of emergent paradigms further complicates this navigation. Parameter-free optimizers like Prodigy (Mishchenko and Defazio, 2024) and Distance over Weighted Gradients (DoWG) (Khaled et al., 2024) eliminate the human burden of costly learning rate sweeps, directly reducing practitioner burnout. IO-aware attention kernels like FlashAttention-2 (Dao, 2024) and FlashMask (Wang et al., 2025) reframe the system-level bottleneck of Transformer training as an optimization problem. None of these developments are adequately captured by existing taxonomies anchored purely in the classic dichotomy of first- and second-order derivatives.

To address this gap, this paper presents a unified taxonomy of deep learning optimizers. Instead of merely cataloging algorithms by their mathematical properties, taxonomy functions as an ergonomic analytical instrument. Each category is evaluated through the lens of memory complexity, computational cost per step, hardware requirements, and practical trade-offs. This analysis culminates in an actionable, decision-oriented framework designed to reduce cognitive overhead, translating theoretical properties into conditional guidelines for practitioners operating under real-world resource constraints.

METHODOLOGY

This study adheres to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology (Page et al., 2021) to ensure transparency and rigor in literature selection (see Figure 1).

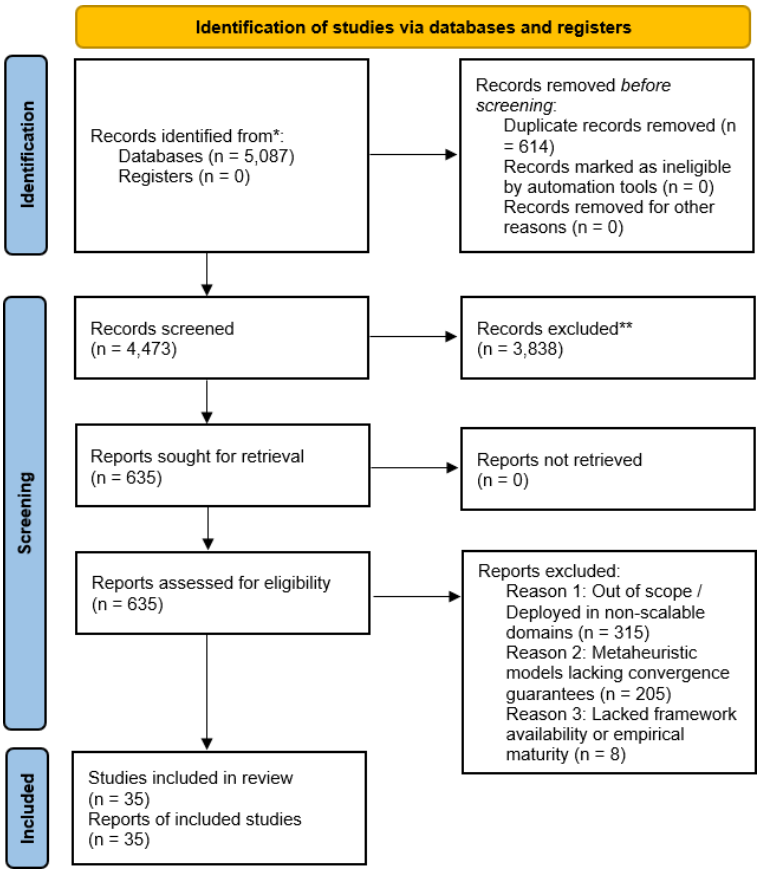

Figure 1: PRISMA 2020 flow diagram.

RESULTS

By grouping the optimizers, a unified taxonomy was established that organizes the algorithms into hierarchical levels defined by gradient information order and architectural operation level. This ordering reflects both algorithmic sophistication and practical accessibility: lower-order methods are computationally cheaper and widely available, whereas higher-order and system-level methods offer stronger convergence guarantees or throughput gains at the cost of implementation complexity.

The review yielded five distinct groups of deep learning optimizers:

(1) gradient-free or zeroth-order methods, (2) first-order gradient methods, (3) second order and quasi-Newton methods, (4) emergent paradigms (tuning-free and learned methods), and (5) system-level and IO-Aware optimization.

Gradient-Free or Zeroth-Order Methods (11 Algorithms)

Gradient-free methods optimize model parameters without relying on explicit derivative calculations, enabling complex architecture training under physical or black-box constraints. While their geometric spatial complexity, represented as $O(P \cdot p)$ where P is the population size and p the number

of model parameters, makes them unviable for pre-training massive deep neural networks (DNNs), they are indispensable for hardware-in-the-loop applications. Population heuristics, such as standard Genetic Algorithms (GA), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Particle Swarm Optimization (PSO), Dispersive Flies Optimization (DFO), and Species-based Genetic Algorithms (SP-GA), explore the loss space globally.

They avoid flat local minima but incur prohibitive memory costs by requiring multiple copies of model weights simultaneously. Conversely, specialized approaches target infrastructure limits. Decentralized Block Coordinate Descent (D-BCD) optimizes distributed on-device edge systems, mitigating gradient vanishing. Analytical algorithms like the Derivative-Free Loss Method (DFLM) and Theory-guided Deep Learning Forecasting (TgDLF) utilize Brownian walkers and ensemble logic for physical systems. Finally, Magneto-Optical Diffractive networks (MO-D2NN) enable physical optical learning, while quantum-inspired models like Adaptive Grover-driven Parallel Quantum Optimization (AG-PQO) and Quantum-Inspired Adaptive Superposition Optimization (QIASO) parallelize search operations. From an HSI perspective, these 11 algorithms expand the accessible ecosystem for engineers lacking traditional GPU access (Ye et al., 2022; Sajjad et al., 2025).

First-Order Gradient Methods (7 Algorithms)

First-order methods update parameters using only gradient information, imposing a linear computational cost per step. Standard stochastic baselines like Stochastic Gradient Descent with Momentum (SGD+M) and Nesterov Accelerated Gradient (NAG) are mathematically efficient but exact a high cognitive toll, requiring rigorous human-in-the-loop hyperparameter tuning. Adaptive methods resolve this ergonomic friction.

Adam and AdamW introduced dual exponential moving averages, allowing adaptive step sizes per parameter. Their widespread adoption is directly tied to their rapid early-stage convergence, which drastically simplifies the hyperparameter search process for the practitioner. Refined variants like AdaBelief and DiffGrad further stabilize gradient estimation to prevent overshooting in highly non-convex landscapes. Additionally, Generalized Fractional Gradient Descent (GLFGD) extends these principles using fractional derivatives to navigate complex topologies (Zhuang et al., 2020; Dubey et al., 2020; Abdulkadirov et al., 2024).

Second-Order and Quasi-Newton Methods (5 Algorithms)

The classic Newton's method is unfeasible in deep models due to the quadratic memory cost and cubic complexity of inverting the exact Hessian matrix (Anil et al., 2021; Tan & Lim, 2019). To overcome this bottleneck, modern paradigms capture curvature through approximations that avoid explicitly instantiating the matrix (Ren et al., 2021).

In practice, quasi-Newton approaches like Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) bypass direct computation using a compact gradient history. Alternatively, other methods factorize the structure:

Kronecker-Factored Approximate Curvature (K-FAC) approximates the Fisher matrix via layer-wise Kronecker products (Gomes et al., 2024), while Shampoo preconditions tensor axes independently to slash computational costs (Gupta et al., 2018).

When scaling LLMs, lightweight diagonal techniques emerge to minimize memory footprints. AdaHessian achieves this by combining Hutchinson's method with spatial averaging, whereas Sophia employs a stochastic approximation alongside coordinate clipping. This setup tames extreme curvature and accelerates pre-training, delivering highly competitive hardware efficiency (Kashyap, 2023; Liu et al., 2024).

Emerging Paradigms:Tuning-Free and Learned Methods (7 Algorithms)

Emergent optimizers address the AI socio-technical divide by eliminating costly manual hyperparameter sweeps that multiply energy consumption and exclude resource-constrained teams. Learning-Rate-Free (LR-Free) methods like Prodigy (Mishchenko & Defazio, 2024) and DoWG (Khaled et al., 2024) dynamically estimate step scales based on gradient history to achieve AdamW-level convergence, while Decayed Adaptation over Gradients (DAoG) stabilizes late-stage training via parameter-free decayed adaptation (Zhang et al., 2025).

Similarly, meta-adaptive methods like MADA automate hyperparameter search through hyper-gradient descent, shifting the heuristic burden from practitioners to the algorithm (Ozkara et al., 2024). Extending beyond heuristics, Geodesic Gradient Descent (GGD) provides a tuning-free update by intrinsically respecting the objective function's Riemannian manifold (Hu et al., 2026). Finally, Learned Optimizers such as PyLO (Janson et al., 2025) and Celo (Moudgil et al., 2025) deploy neural networks to optimize other architectures under strict compute diets, delivering hardware-efficient scalability without manual tuning.

System-Level and IO-Aware Optimization (5 Algorithms)

IO-aware methods reconceptualize the training bottleneck not as an arithmetic problem measured in Floating Point Operations per Second (FLOPs), but as a memory bandwidth challenge. FlashAttention-2 partitions attention computation into blocks residing in Static Random Access Memory (SRAM), eliminating the quadratic memory footprint historically written to High Bandwidth Memory (HBM). FlashMask extends this logic to sparse representations for complex fine-tuning, while Gated Linear Attention (GLA) adapts these kernels for hardware-efficient linear transformers (Dao, 2024; Wang et al., 2025; Yang et al., 2024).

Finally, pipeline scheduling frameworks like MEPipe and PipeMesh demonstrate that consumer-grade accelerators (e.g., NVIDIA RTX 4090) can achieve competitive Model FLOPs Utilization (MFU) when training massive models by overlapping computation and communication slices (Sun et al., 2025; Li et al., 2025).

Decision-Oriented Framework for Practitioners

The taxonomic analysis reveals that no single optimizer universally dominates all resource profiles. The practical challenge transcends selecting the theoretically superior method; it lies in identifying the optimization architecture whose cost-benefit profile resolves a binding operational constraint (e.g., Video Random Access Memory [VRAM] limits, hyperparameter tuning budget, or hardware architecture).

Table 1 synthesizes the proposed taxonomy and translates it into a conditional framework. This matrix aligns all 35 evaluated algorithms with the constraint scenarios where they maximize efficiency, clearly exposing the technical complexity and the socio-technical trade-off the practitioner must assume.

Table 1: Taxonomic matrix and decision framework for optimizer selection.

Taxonomic Category	Representative Algorithms	Constraint Scenario/Optimal Use	Complexity & Critical Trade-Off
Zeroth Order (Population)	GA, CMA-ES, PSO, DFO, SP-GA	High-dimensional non-differentiable systems / Physical simulations.	***Memory*** $O(P \cdot p)$. Unviable for massive DNNs. High robustness to flat local minima at the expense of slow convergence.
Zeroth Order (Specialized)	D-BCD, DFLM, TgDLF, MO-D2NN, AG-PQO, QIASO	Decentralized devices (Edge), optical computing, and PDEs.	**Variable memory.** Enables learning on atypical hardware without backpropagation.
1st Order (Stochastic)	SGD+M, NAG	General training prioritizing final generalization.	**Memory** $O(p)$**.** Highly space-efficient but demands exhaustive and costly learning rate tuning.
1st Order (Adaptive)	Adam, AdamW, AdaBelief, DiffGrad, GLFGD	Standard baseline for NLP, Vision, and single GPU setups.	**Memory** $O(2p)$. Higher VRAM consumption, but fast/stable convergence with lower sensitivity to initial hyperparameters.
2nd Order (Quasi-Newton Methods)	L-BFGS	Deep networks with constraints where inverting the exact Hessian is unfeasible (Kashyap, 2023).	**Memory** $O(N)$ storing only $3 < m < 20$ previous iterations (Kashyap, 2023)

(Continued)

Table 1: Continued.

Taxonomic Category	Representative Algorithms	Constraint Scenario/Optimal Use	Complexity & Critical Trade-Off
2nd Order (Preconditioners)	K-FAC, Shampoo	Large-scale distributed training and large batches (large-batch) for tensor-structured models (Anil et al., 2021; Duvvuri et al., 2024; Gupta et al., 2018).	Shampoo requires $O(m^2 + n^2)$ in memory and $O(m^2 + n^2)$ in time (Duvvuri et al., 2024; Gupta et al., 2018). K-FAC imposes a time overhead of **1.89** × over SGD (Ueno et al., 2020).
2nd Order (Diagonal)	Sophia, AdaHessian	Efficient pre-training of LLMs and vision/NLP tasks under strict memory limits (Liu et al., 2024; Yao et al., 2021).	**Memory** $O(d)$**.** with a time overhead of **2 ×** in AdaHessian and almost none in Sophia (Liu et al., 2024; Yao et al., 2021).
Emerging (Tuning-Free)	Prodigy, DoWG, DAoG, MADA	Absence of hyperparameter search budget.	**Memory** $O(2p)$**.** Saves hundreds of GPU hours by auto-estimating step size. Slight computational overhead per iteration.
Emerging (Learned)	PyLO, Celo, GGD	Training under strict computational diets.	**Emergent scalability.** Transfers a heuristic burden to secondary networks or Riemannian geometry.
System & IO (Attention)	FlashAttention-2, FlashMask, GLA	LLMs with long contexts (>8K tokens) and limited memory.	**Memory** $O(N)$ ***(HBM).*** Eliminates quadratic write bottleneck. Indispensable for scaling; requires modern hardware (Ampere+).
System & IO (Pipeline)	MEPipe, PipeMesh	LLM training on consumer-grade accelerators (e.g., RTX 4090).	**Economic efficiency.** Overlaps communication/compute, surpassing the cost-benefit ratio of traditional enterprise clusters.

DISCUSSION AND IMPLICATIONS

The analysis demonstrates that selecting an optimization algorithm transcends pure technical computation to become a fundamental socio-technical decision. Historically, prioritizing theoretical convergence rates fostered a heavy reliance on high-performance multi-GPU infrastructures. This dramatically elevated the barriers to entry, inducing high cognitive and financial stress on researchers outside elite institutions.

However, the integration of parameter-free (Tuning-Free) optimizers with IO-aware kernels represents a paradigm shift toward cognitive ergonomics and equitable access. As evidenced in the proposed framework, simultaneously reducing the friction of empirical step-size searching and the memory bandwidth bottleneck allows consumer-grade hardware to compete efficiently with industrial clusters. Furthermore, this matrix approach directly contributes to "Green AI" principles: selecting the appropriate algorithm based on the binding hardware constraint prevents massive waste of computational cycles in suboptimal configurations. By reducing trial-and-error runs, practitioners decrease both their cognitive decision fatigue and the overarching carbon footprint of deep learning training pipelines. This alignment of algorithmic design with human workflow realities fosters a more inclusive, sustainable, and operationally efficient research ecosystem.

CONCLUSION

This study introduced a unified taxonomy and decision framework for deep learning optimizers, advancing beyond classic classifications anchored solely in derivative order. Through a PRISMA-guided systematic review of 5,087 records, 35 representative algorithms were evaluated across spatial complexity, computational cost, and technical requirements.

The primary contribution, materialized in the Taxonomic Decision Matrix, translates these theoretical properties into actionable guidelines. We conclude that current limitations in large-scale LLM training are not strictly arithmetic but relate to memory management and heuristic tuning burdens. Adopting emergent approaches and system-level architectures is not merely a technical optimization strategy, but a socio-technical imperative. By aligning system capabilities with human workflow constraints, these tools democratize access to AI development, mitigate the concentration of technological capabilities, and promote a truly equitable and sustainable scientific ecosystem.

ACKNOWLEDGMENT

The authors would like to acknowledge the Vicerrectorado de Docencia of Escuela Politécnica Nacional for their support in this research.

REFERENCES

Abdulkadirov, R., Lyakhov, P. and Nagornov, N. (2024) 'Improving the accuracy of neural network pattern recognition by fractional gradient descent', IEEE Access, 12, pp. 1–14.

Anil, R., Gupta, V., Koren, T., Regan, K. and Singer, Y. (2021) 'Scalable second order optimization for deep learning', arXiv preprint arXiv:2002.09018.

Asseman, A., Antoine, N. and Ozcan, A. S. (2021) 'Accelerating deep neuroevolution on distributed FPGAs for reinforcement learning problems', ACM Journal on Emerging Technologies in Computing Systems (JETC), 17(2), pp. 1–17.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020) 'Language models are few-shot learners', Advances in Neural Information Processing Systems, 33, pp. 1877–1901.

Chen, Y. and Zhang, D. (2021) 'Theory-guided deep-learning for electrical load forecasting (TgDLF) via ensemble long short-term memory', Advances in Applied Energy, 1, 100004.

Dao, T. (2024) 'FlashAttention-2: Faster attention with better parallelism and work partitioning', In Proceedings of the 12th International Conference on Learning Representations (ICLR).

Davtyan, A., Molybog, I. and Madani, K. (2022) 'KOALA: A Kalman Optimization Algorithm with Loss Adaptivity', In Proceedings of the AAAI Conference on Artificial Intelligence.

Dubey, S. R., Chakraborty, S., Roy, S. K., Mukherjee, S., Singh, S. K. and Chaudhuri, B. B. (2020) 'diffGrad: An optimization method for convolutional neural networks', IEEE Transactions on Neural Networks and Learning Systems, 31(11), pp. 4500–4511.

Gomes, D. M., Zhang, Y., Belilovsky, E., Wolf, G. and Hosseini, M. S. (2025) 'AdaFisher: Adaptive second order optimization via fisher information', arXiv preprint arXiv:2405.16397.

Gupta, V., Koren, T. and Singer, Y. (2018) 'Shampoo: Preconditioned stochastic tensor optimization', In Proceedings of the 35th International Conference on Machine Learning (ICML), pp. 1842–1850.

Hu, L., Li, G., Wang, W., Zhang, X. and Xiang, Y. (2026) 'Geodesic Gradient Descent: A Generic and Learning-rate-free Optimizer on Objective Function-induced Manifolds', arXiv preprint arXiv:2603.06651.

Ivgi, M., Carmon, Y. and Hinder, O. (2023) 'DoG is SGD's Best Friend: A Parameter-Free Dynamic Step Size Schedule', In Proceedings of the International Conference on Machine Learning (ICML), pp. 14465–14499.

Janson, P., Thérien, B., Anthony, Q., Huang, X., Moudgil, A. and Belilovsky, E. (2025) 'PyLO: Towards Accessible Learned Optimizers in PyTorch', arXiv preprint arXiv:2506.10315.

Kashyap, R. (2023) 'A survey of deep learning optimizers—First and second order methods', arXiv preprint arXiv:2211.15596.

Khaled, A., Mishchenko, K. and Jin, C. (2024) 'DoWG Unleashed: An Efficient Universal Parameter-Free Gradient Descent Method', In Advances in Neural Information Processing Systems (NeurIPS).

Li, F., Zhao, S., Qing, Y., Jiang, J., Chen, X. and Cui, H. (2025) 'PipeMesh: Achieving memory-efficient computation-communication overlap for training large language models', IEEE Transactions on Parallel and Distributed Systems, 36(5).

Liu, H., Li, Z., Hall, D., Liang, P. and Ma, T. (2024) 'Sophia: A scalable stochastic second-order optimizer for language model pre-training', arXiv preprint arXiv:2305.14342.

Loshchilov, I. and Hutter, F. (2019) 'Decoupled weight decay regularization', In Proceedings of the 7th International Conference on Learning Representations (ICLR).

Mishchenko, K. and Defazio, A. (2024) 'Prodigy: An Expeditiously Adaptive Parameter-Free Learner', In Proceedings of the International Conference on Machine Learning (ICML).

Moudgil, A., Knyazev, B., Lajoie, G. and Belilovsky, E. (2025) 'Celo: Training Versatile Learned Optimizers on a Compute Diet', In Proceedings of the International Conference on Learning Representations (ICLR).

Ozkara, K., Karakus, C., Raman, P., Hong, M., Sabach, S., Kveton, B. and Cevher, V. (2024) 'MADA: Meta-Adaptive Optimizers through hyper-gradient Descent', arXiv preprint arXiv:2401.08893.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... & Moher, D. (2021) 'The PRISMA 2020 statement: an updated guideline for reporting systematic reviews', BMJ, 372, n71.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., ... & Dean, J. (2021) 'Carbon emissions and Large Neural Network Training', arXiv preprint arXiv:2104.10350.

Ren, Y., Bahamou, A. and Goldfarb, D. (2022) 'Kronecker-factored quasi-newton methods for deep learning', arXiv preprint arXiv:2102.06737.

Ruder, S. (2016) 'An overview of gradient descent optimization algorithms', arXiv preprint arXiv:1609.04747.

Sajjad, H., Alshanbari, M., Almazah, M. M. A., Louati, H. and Rauf, S. (2025) 'Adaptive Grover-driven optimization for quantum-inspired deep learning: A gradient-free training framework', AIMS Mathematics, 10, pp. 26568–26592.

Sajjad, I. and AL Sobhi, M. M. (2026) 'The quantum-inspired adaptive superposition optimization for neural network training', AIMS Mathematics, 11(1), pp. 243–271.

Schmidt, R. M., Schneider, F. and Hennig, P. (2021) 'Descending through a crowded valley—benchmarking deep learning optimizers', In Proceedings of the International Conference on Machine Learning (ICML), pp. 9367–9376.

Sun, Z., Chen, S., Wang, Y., Sha, J., Feng, G. and Chen, W. (2025) 'MEPipe: Democratizing LLM training with memory-efficient slice-level pipeline scheduling', In Proceedings of EuroSys 2025.

Tan, H. H. and Lim, K. H. (2019) 'Review of second-order optimization techniques in artificial neural networks backpropagation', IOP Conference Series: Materials Science and Engineering, 495(1), 012003.

Wang, G., Zeng, J., Xiao, X., Wu, S., Yang, J., Zheng, L., Chen, Z., Bian, J., Yu, D. and Wang, H. (2025) 'FlashMask: Efficient and rich mask extension of FlashAttention', In Proceedings of the 13th International Conference on Learning Representations (ICLR).

Yang, S., Wang, B., Shen, Y., Panda, R. and Kim, Y. (2024) 'Gated linear attention transformers with hardware-efficient training', Proceedings of Machine Learning Research.

Ye, G., Yin, H., Chen, T., Xu, M., Nguyen, Q. V. H. and Song, J. (2022) 'Personalized On-Device E-Health Analytics with Decentralized Block Coordinate Descent', IEEE Journal of Biomedical and Health Informatics, 26, pp. 2778–2786.

Zhang, Y., Zhao, D., Li, H. and Pan, C. (2025) 'DAoG: decayed adaptation over gradients for parameter-free step size control', Artificial Intelligence Review, 58(11), pp. 1–37.

Zhuang, J., Tang, T., Ding, Y., Tatikonda, S. C., Dvornek, N., Papademetris, X. and Duncan, J. (2020) 'AdaBelief Optimizer: Adapting stepsizes by the belief in observed gradients', In Advances in Neural Information Processing Systems (NeurIPS).