



Review

Fractional Calculus for Large Language Models: A Survey of Potentials, Mechanisms, and Challenges

Wenwen Tu ¹, Xixing Liu ², Hexin Wang ², Xuhang Li ³ and Chunyang Han ^{4,5,*}

¹ Engineering Training Center, Kunming University of Science and Technology, Kunming 650500, China; 20240246@kust.edu.cn

² School of Artificial Intelligence Industry, Kunming University of Science and Technology, Kunming 650500, China; 202412202147@stu.kust.edu.cn (X.L.); 202412202143@stu.kust.edu.cn (H.W.)

³ School of Information, Yunnan University, Kunming 650500, China; xuhangli2024@163.com

⁴ Faculty of Transportation Engineering, Kunming University of Science and Technology, Kunming 650500, China

⁵ Yunnan Modern Logistics Engineering Research Center, Kunming University of Science and Technology, Kunming 650500, China

* Correspondence: sandiant@kust.edu.cn; Tel.: +86-156-7519-8196

Abstract

Large language models (LLMs) face critical bottlenecks like the “memory wall” and context degradation due to standard integer-order Markovian assumptions. This survey reviews integrating fractional calculus to explore potential solutions. Synthesizing the recent literature reveals a transitional landscape: while fractional mechanisms yield significant empirical gains in medium-scale architectures (e.g., up to 14.36% absolute accuracy improvements), explicit validation on autoregressive models exceeding one billion parameters remains largely unexplored. By introducing heavy-tailed power-law memory and non-local historical gradients, fractional operators help mitigate catastrophic forgetting and optimization instability. Furthermore, we propose dual-integration frameworks: a decoupled neuro-symbolic architecture for scientific computing, and an endogenous fractional evolution architecture for native long-context reasoning. While fractional dynamics offer mathematical pathways to alleviate LLM constraints, computational overhead and training stability challenges remain. Future research should prioritize selective fractionalization and data-driven scientific discovery.

Keywords: large language models; fractional calculus; long-context modeling; machine learning

Academic Editor: António Lopes

Received: 4 June 2026

Revised: 24 June 2026

Accepted: 30 June 2026

Published: 3 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In recent years, deep sequence models centered on Transformer architecture have driven rapid advances in natural language processing, multimodal learning, and scientific computing. With the emergence of ultra-large language models (Large Language Models, LLMs) [1], such as the OpenAI GPT series (GPT-3.5/GPT-4) [2,3], Google PaLM (PaLM 2) [4], and Gemini (Gemini 1.0/1.5 Pro) [5,6], the capability of self-attention mechanisms for long-range dependency modeling has been substantially enhanced. However, as model parameter scales continue to grow exponentially and context windows expand further, traditional deep learning frameworks have gradually exposed a series of fundamental

bottlenecks [7]. The quadratic spatiotemporal complexity of attention matrix computation has led to a severe “memory wall” problem, while extremely long-context scenarios exhibit pronounced “lost in the middle” phenomena and memory degradation [8–10]. At the same time, optimization instability in large-scale distributed training has become increasingly prominent.

Current mainstream deep learning systems are still fundamentally built upon classical integer-order calculus. Whether in gradient descent-based backpropagation or in the state update mechanisms of recurrent neural networks and state-space models, the underlying formulations implicitly or explicitly rely on local derivatives and first-order Markovian smoothness assumptions [11,12]. However, complex real-world dynamical systems often exhibit strong nonlocality, long-range dependency, and memory effects. For example, long-distance semantic entanglement across paragraphs in LLMs, anomalous diffusion phenomena in complex physical systems, and heavy-tailed distributions in time series are all difficult to accurately characterize using integer-order operators based on local Markovian assumptions. Such limitations in the underlying mathematical modeling framework may prevent current large-scale sequence models from fundamentally overcoming the memory limitations of long-context reasoning.

The theoretical motivation for introducing fractional calculus into large language models mainly arises from three aspects. As shown in Table 1, first, the power-law memory characteristics of fractional operators can alleviate the loss of long-distance information in traditional models, thereby improving the capability of LLMs in handling ultra-long contexts. Second, fractional optimization algorithms can exploit historical gradient information to mitigate gradient oscillation during large-scale network training, thus enhancing optimization stability [13,14]. Third, by constructing a decoupled architecture that combines LLMs with external fractional-order solvers, it becomes possible to compensate for the limitations of language models in numerical computation and enable them to handle complex physical evolution and scientific reasoning tasks [12].

Table 1. Current LLM limitations and theoretical motivations for fractional calculus.

Domain	Current Mechanism	Primary Bottlenecks	Fractional Approach	Key Advantage
Context Modeling	Self-attention matrices	Quadratic complexity “Memory wall” “Lost in the middle”	Power-law memory operators	Alleviates long-distance info loss in ultra-long contexts
Model Training	Standard gradient descent	Optimization instability in large-scale distributed training	Fractional optimization algorithms	Uses historical gradients to mitigate oscillation and enhance stability
Mathematical Foundation	Integer-order calculus; First-order Markovian assumption	Ignore strong nonlocality and memory effects Fails at anomalous diffusion	Non-local modeling framework	Accurately characterizes long-range dependency and heavy-tailed distributions
Scientific Computing	Pure language modeling	Limitations in numerical computation and complex physical evolution	Decoupled architecture with external solvers	Empowers LLMs to handle complex scientific reasoning tasks

At present, research at the intersection of fractional mathematics and LLMs remains in an early exploratory stage. Existing studies are still primarily concentrated on light-weight Transformer encoders, medium-scale neural network optimizers, scientific machine learning, and AI-based frameworks for solving fractional partial differential equations [15–17]. Large-scale validation on autoregressive language models with tens or hundreds of billions of parameters, such as GPT and LLaMA series models, remains relatively scarce. Despite fractional calculus demonstrating substantial empirical promise in small- and medium-scale networks, there remains a notable lack of reviews regarding its integration with large language models exceeding one billion parameters. This review explic-

itly addresses this gap by comprehensively analyzing the underlying mathematical mechanisms, evaluating current sequence modeling bottlenecks, and proposing concrete architectural pathways for future large-scale deployment.

To ensure the objectivity, rigor, and comprehensiveness of this review, we conducted a literature search across authoritative academic and preprint databases, including Web of Science, Scopus, IEEE Xplore, and arXiv. The search period primarily focused on publications from 2017 to 2026. The literature retrieval strategy adopted a combination of keywords covering both mathematical operator dimensions (e.g., “Fractional calculus,” “Fractional-order,” “Fractional derivative,” and “Power-law memory”) and modern deep learning architecture dimensions (e.g., “Large Language Models,” “Transformer,” “State-space models,” “Sequence modeling,” and “Deep learning optimization”). During the literature selection stage, our screening process was primarily centered on the intersection of fractional calculus and modern artificial intelligence. We specifically targeted studies encompassing core themes such as foundational mathematical mechanisms, fractional neural network backpropagation, attention architecture variants, nonlocal feature extraction, and data-driven scientific equation discovery (AI4Science). After the initial retrieval and deduplication of approximately 230 potentially relevant papers, we conducted thorough abstract screening and full-text evaluations.

The selection of the core intersecting literature was governed by a filtering protocol. While the core application studies were evaluated against these benchmarks, supplementary foundational papers, including classical mathematical definitions of fractional operators and seminal baseline language models, were purposefully retained to provide the necessary theoretical background. The comprehensive inclusion and exclusion boundaries, encompassing model scales and reproducibility standards, are systematically detailed in Table 2. Ultimately, these filtered core studies, combined with the essential background literature, constitute the 111 references analyzed in this survey.

Table 2. Inclusion and exclusion criteria for the core reviewed studies.

Criterion	Inclusion Criteria	Exclusion Criteria
Publication Period	Published between 2017 and 2026 for core fractional AI applications, with exemptions for fundamental mathematical theorems.	Published before 2017, unless utilized exclusively as foundational background definitions.
Research Domain	Integration of fractional calculus with deep learning, Transformers, KANs, scientific AI, or optimization landscapes.	Pure mathematical analysis without machine learning applications or explicit AI framework integration.
Model Scale and Type	Neural networks containing trainable learning components ranging from small-scale sequence models to large foundation models.	Traditional fractional discrete control systems lacking trainable parameters or adaptive neural architectures.
Reproducibility	Studies providing explicit mathematical formulations, algorithmic pseudo-code, quantitative benchmarks, or public repositories.	Purely theoretical propositions lacking numerical validation, algorithmic clarity, or reproducible methodology.

This paper investigates the integration of fractional calculus with large language models to improve long-range memory and scientific reasoning. Our core contributions are twofold.

First, we review recent advances in fractional artificial intelligence, encompassing optimization mechanisms, representation enhancement, spectral Transformers, Kolmogorov–Arnold Networks (KANs), and scientific equation discovery. Based on these mechanisms, we analyze their theoretical potential to alleviate critical sequence modeling bottlenecks, such as context memory degradation and optimization instability.

Second, we propose two conceptual integration frameworks: a decoupled neuro-symbolic architecture utilizing LLMs to drive external solvers, and an endogenous frac-

tional evolution architecture embedding nonlocal dynamics natively. We provide concrete guidance, algorithmic pseudocode, computational complexity analyses, and comprehensive evaluations of their respective hardware trade-offs and architectural challenges.

The remainder of this paper is structured as follows. Section 2 details the continuous and discrete mathematical foundations of fractional operators. Sections 3 and 4 review fractional deep learning mechanisms, empirical computational overheads, and their targeted applications to existing LLM bottlenecks. Section 5 explicitly details the proposed conceptual frameworks alongside concrete integration workflows, followed by concluding remarks and future research directions in Section 6.

2. Mathematical Foundations of Fractional Nonlocal Learning

2.1. Fractional Calculus and Nonlocal Operators

2.1.1. Fundamental Fractional Operators

Fractional calculus is a mathematical theory that studies derivatives and integrals of arbitrary order, extending the order of classical calculus to continuous real or complex spaces. Unlike integer-order derivatives, which depend only on local neighborhood information, fractional operators are defined through global or semi-global weighted integration over historical trajectories, thereby naturally possessing the capability to characterize nonlocal systems. In the analysis of continuous dynamical systems, three fundamental fractional operators are commonly adopted.

The Riemann–Liouville (RL) [18] fractional integral serves as the foundational operator of fractional calculus and is defined as:

$${}_{t_0}I_t^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^t (t - \tau)^{\alpha-1} f(\tau) d\tau \quad (1)$$

where $\Gamma(\cdot)$ denotes the Gamma function and t_0 represents the initial time. This definition indicates that the current system state depends on the entire historical trajectory, while the influence of past information decays according to the power-law kernel $(t - \tau)^{\alpha-1}$.

For initial value problems and physical system modeling, the Caputo fractional derivative is more commonly employed as Equation (2).

$${}_{t_0}^CD_t^\alpha f(t) = \frac{1}{\Gamma(n - \alpha)} \int_{t_0}^t (t - \tau)^{n-\alpha-1} f^{(n)}(\tau) d\tau \quad (2)$$

where n is the positive integer satisfying $n - 1 < \alpha \leq n$. The Caputo definition allows the use of standard integer-order initial conditions, and the derivative of a constant function remains zero, making it convenient for direct integration with existing dynamical system theories. $\alpha \in (n - 1, n]$ with $n \in \mathbb{N}^+$, $\Gamma(\cdot)$ is the Gamma function, t_0 is the initial time, and n is the smallest integer greater than α .

For numerical computation on discrete sequences, the Grünwald–Letnikov (GL) [19,20] definition provides the most direct implementation pathway:

$$D^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\lfloor (t-t_0)/h \rfloor} w_k^\alpha f(t - kh) \quad (3)$$

where $h > 0$ is the discretization step size, t_0 is the initial time, $\lfloor \cdot \rfloor$ denotes the floor function, and w_k^α are binomial coefficients defined recursively in Equation (3), where the binomial weights w_k^α satisfy the recursive formulation:

$$w_0^\alpha = 1, w_k^\alpha = \left(1 - \frac{\alpha + 1}{k}\right) w_{k-1}^\alpha \quad (4)$$

where $\alpha \in \mathbb{R}^+$ is the fractional order, $k \in \mathbb{N}^+$ is the summation index, and $w_0^\alpha = 1$. Through discretized weighted summation over historical states, the GL operator forms the underlying mathematical structure for nonlocal computation in sequence modeling.

The optimal selection of the fractional order α heavily depends on the specific dataset and the topology of the non-convex optimization landscape. In empirical practice, manual grid searching is frequently required, yet identifying the optimal scheduling for these hyperparameters remains profoundly difficult in neural network training [21].

To bypass manual tuning, recent advancements propose dynamic adaptive strategies. For example, the Learning to Optimize Caputo Fractional Gradient Descent (L2O-CFGD) algorithm utilizes meta-learning to automatically adjust hyperparameters, eliminating the need for manual fractional order selection [21]. Similarly, adaptive momentum algorithms based on Caputo fractional-order differences, such as Fractional Order Adam (FAdam), update parameters flexibly by continuously adjusting historical momentum information [22]. For a more comprehensive analysis regarding parameter selection risks and modern adaptive regularization methods, readers are directed to the detailed discussions presented in Section 5.5.5.

2.1.2. Discrete Caputo Fractional Difference

Besides the continuous and discrete Grünwald–Letnikov formulations, discrete Caputo fractional difference systems have received increasing attention. The primary mathematical advantage of the discrete Caputo formulation is its ability to evaluate and preserve standard initial conditions, providing a direct physical interpretation of the investigated system's initial state.

Discrete Caputo fractional models have been extensively investigated in complex nonlinear dynamics. Foundational theories define these discrete systems as equivalent to maps with falling factorial-law memory, allowing the construction of standard and logistic families of fractional difference maps [23,24]. Furthermore, numerical evaluations reveal high sensitivity to historical states, as avoiding transients strictly requires iteration numbers on the order of thousands, and every distinct initial condition corresponds to a completely different bifurcation diagram [25]. When expanded, these Caputo models systematically delay bifurcations and exhibit robust attractor coexistence and synchronous memory evolution [26–30].

In discrete neurodynamics, the discrete Caputo fractional Chialvo neuron model accurately reproduces electromagnetic induction memory. Structural mapping into memristive networks proves their capacity for complex state retention. For instance, integrating the Caputo difference can produce a highly dimensional fractional-order memristive ring network capable of generating hidden attractors, which is explicitly validated for secure remote sensing encryption tasks [31]. Extending Caputo operators into octonion frameworks further facilitates advanced synchronization [32]. This multi-stable modulation capability is consistently supported by extensive network coupling analyses [33–36].

Furthermore, comprehensive investigations into discrete Caputo fractional Rulkov neuron models prove that the discrete derivative allows the neuron to transition smoothly among resting, spiking, and chaotic bursting states depending entirely on the specific fractional order utilized [37,38]. Specific numerical simulations confirm that external interventions, such as a single control impulse, can transiently suppress chaos within the Caputo fractional Rulkov map [39]. This dynamic control is further corroborated by observations of extreme multi-stability and phase synchronization under discrete memristor coupling [40–42].

These specific dynamic state transitions confirm that discrete Caputo derivatives can effectively characterize non-local memory effects while maintaining physically meaningful initialization. Therefore, Caputo discrete fractional difference models represent an indispensable branch of discrete dynamics and must be considered when evaluating fractional mechanisms for recurrent memory modeling in sequential learning.

2.2. Fractional Memory and Long-Range Dependency

Traditional dynamical systems and sequence modeling theories are generally established upon the first-order Markov assumption, where the underlying states follow integer-order differential equations. In such systems, memory of historical information decays exponentially, meaning that the influence weights decrease in the form of $e^{-\lambda t}$ over time. Exponential decay causes systems to rapidly lose early-stage information after a finite number of time steps, fundamentally limiting their capability to capture long-range dependencies in complex systems.

In contrast, fractional-order systems break through the limitations of the Markovian assumption. According to classical theories of anomalous diffusion in complex physical systems, information propagation and state evolution in complex media often exhibit heavy-tailed distributions. Fractional calculus rigorously characterizes this phenomenon through the power-law memory kernel $(t - \tau)^{\alpha-1}$ embedded within fractional integrals. Compared with exponential functions, power-law functions possess significantly heavier tails, implying that historical information decays extremely slowly and that early-stage state variables can still make non-negligible statistical contributions to the current system evolution.

As visually contrasted in Figure 1, which plots memory retention weight against sequence distance, fractional-order systems replace the rapid exponential memory decay of traditional integer-order systems with a heavy-tailed power-law memory kernel. Rather than completely preventing information loss, this structural shift serves to significantly mitigate the abrupt truncation of early-stage information. This mathematically intrinsic long-tail effect provides a theoretical foundation for alleviating the forgetting problem in long-sequence processing. In tasks requiring the preservation of extended correlations, information aggregation mechanisms constructed utilizing fractional power-law kernels can theoretically reduce the inherent truncation vulnerabilities of integer-order systems, thereby facilitating more robust long-range memory retention.

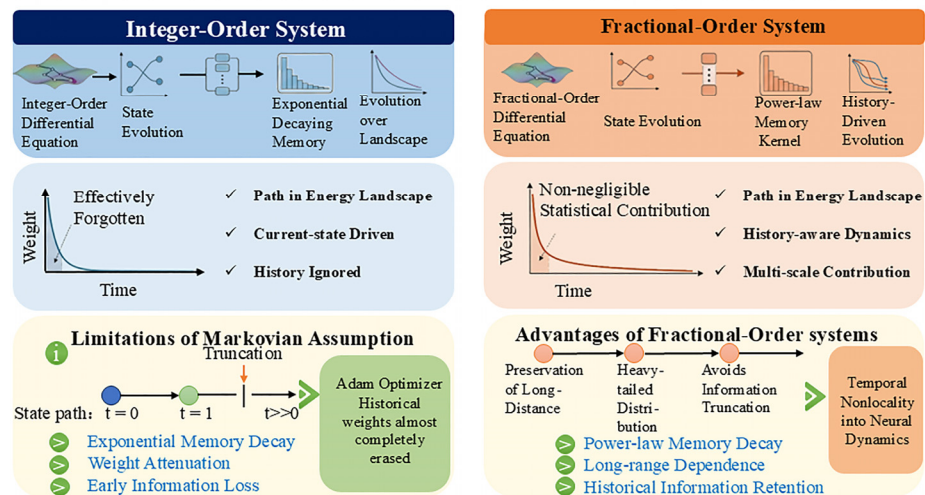


Figure 1. Contrast of memory profiles: Integer-order vs. Fractional-order systems.

2.3. Fractional Dynamics and Sequential Learning

From the perspective of continuous-time dynamics, sequence learning is essentially a state-space evolution process. Traditional local state updates correspond to first-order differential systems:

$$h'(t) = f(h(t), x(t)) \quad (5)$$

where $h(t)$ is the hidden state vector, $x(t)$ is the input signal at time t , and $h'(t)$ denotes the first-order derivative with respect to t . If this formulation is generalized into a fractional nonlocal continuous system:

$${}_{t_0}D_t^\alpha h(t) = f(h(t), x(t); \theta) \quad (6)$$

where θ denotes the learnable parameter vector of the network. Then the instantaneous rate of change in the hidden state $h(t)$ is no longer determined solely by the current input and state but instead becomes governed by the accumulated integration of the entire historical trajectory. The analytical solution of such fractional differential equations commonly involves the Mittag–Leffler function $E_\alpha(z)$. As a generalized exponential function, the Mittag–Leffler function enables a mathematically continuous transition between short-range exponential smoothing and long-range power-law memory [43].

After discretization, the resulting sequence update mechanism exhibits structural isomorphism with the global interaction mechanism in modern Transformer architectures. The self-attention mechanism dynamically assigns interaction weights through inner-product computations across global tokens and is essentially a discrete nonlocal integral operator. In contrast, fractional operators perform deterministic historical convolutions through rigorously defined power-law decay functions. From the perspective of mathematical mechanisms, fractional dynamics can therefore be interpreted as a physically grounded prior non-local attention mechanism.

As shown in Table 3, this structural isomorphism suggests that reconstructing the feature evolution of discrete sequences through fractional differential equations may provide a continuous dynamical theory for long-context modeling [44,45]. Furthermore, it also offers a mathematical interpretive framework for understanding the global aggregation mechanisms of existing large language models.

Table 3. Characteristics of fractional nonlocal learning.

Category	Concept/Operator	Primary Function	Core Mechanism	Key Advantage
Fundamental Operators	Riemann–Liouville	Characterizes continuous nonlocal systems.	Integration over entire historical trajectories.	Serves as the foundational operator for fractional calculus.
	Caputo Derivative	Solves initial value problems and physical models.	Ensure the derivative of a constant remains zero.	Allows seamless integration with existing dynamical theories.
	Grünwald–Letnikov	Handles discrete sequence computations.	Performs discretized weighted summation over past states.	Provides a direct implementation pathway for numerical tasks.
Memory Mechanisms	Fractional vs. Integer-Order	Addresses the inherent forgetting problem in sequences.	Replaces exponential decay with heavy-tailed power-law decay.	Breaks the Markov assumption to enable genuine long-range memory.
Sequence Learning	Fractional Continuous Dynamics	Reconstructs feature evolution in discrete sequences.	Governs state updates via accumulated historical integration.	Exhibits structural isomorphism to Transformer self-attention mechanisms.

3. Existing Fractional AI and Transformer Research: Mechanisms and Inspirations

To provide a structured overview of the current research landscape, Figure 2 illustrates the roadmap of key themes and sub-sections explored in fractional-order deep learning, including optimization, representation and token mixing.

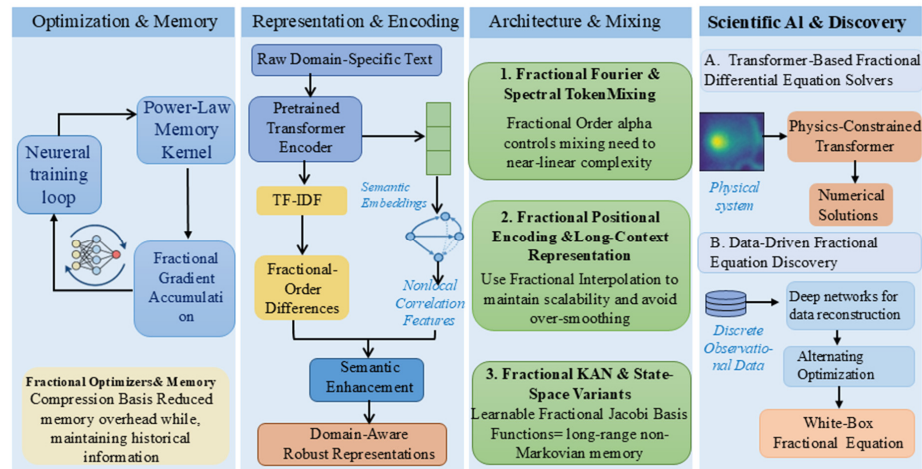


Figure 2. Roadmap of Section 3: Key themes and sub-sections in fractional-order deep learning.

3.1. Fractional Optimization and Long-Memory Learning

3.1.1. Fractional Gradient Dynamics

The optimization process of neural networks is essentially a gradient flow evolution over an energy landscape. Traditional deep sequence models are generally based on integer-order or first-order Markovian update equations, causing the weights of historical gradients to decay exponentially. As demonstrated by Khan [46] in a review, the historical gradient weights in the standard Adam optimizer are almost completely erased after approximately one hundred optimization steps. To overcome this locality limitation, El-Nabulsi and Anukool [43] proposed the fractional action Hamiltonian neural network framework. This architecture employs the power-law memory kernel of fractional derivatives, enabling parameter evolution to be driven by weighted historical trajectories and thereby introducing temporal nonlocality into neural dynamics. In terms of architectural design, Qiu et al. [47] extended recurrent tensor units into a fractional continuous domain and proposed the Fractional Tensor Recurrent Unit (fTRU), which preserves long-memory characteristics while maintaining the stability of time-series forecasting.

In discrete optimization, introducing fractional derivatives into gradient updates can be approximately expressed as a weighted accumulation of historical gradients:

$$w_{t+1} = w_t - \eta \sum_{k=0}^t \frac{\nabla L(w_k)}{(t-k+1)^\alpha} \quad (7)$$

where $\eta > 0$ is the learning rate and $\alpha > 0$ is the fractional order, α denotes the fractional order. This mechanism causes the current parameter to update to be smoothly constrained by the entire optimization trajectory. Building upon this principle, Olumide [48] demonstrated that fractional-order gradient descent introduces nonlocal memory into parameter evolution. Furthermore, Yang et al. [49] investigated stochastic fractional-order gradient descent algorithms and showed that power-law weighted accumulation can effectively improve convergence speed and robustness against noise.

To provide rigorous mathematical guarantees for such discrete evolution systems, Almatroud et al. [50] proved the existence, uniqueness, and stability of variable-order fractional discrete neural networks, thereby establishing a theoretical foundation for the dynamical evolution of complex neural weights.

To address the gradient vanishing and explosion problems that frequently occur in long-context modeling, fractional operators can effectively reconstruct the backward propagation path of gradients. Zhao et al. [51] proposed a fractional-order Lipschitz re-

current neural network combined with a frequency-domain gated attention mechanism, demonstrating that reconstructing update dynamics through fractional calculus can significantly alleviate gradient attenuation. Unlike the chaotic behaviors induced by local dynamic coupling in multi-region neural networks observed by Wang et al. [52], the introduction of fractional gradients effectively injects memory-based inertial regulation into complex evolution processes, improving convergence stability while preventing pathological oscillation.

In hybrid algorithm design, Zhang et al. [53] proposed an integer–fractional hybrid gradient descent algorithm, which introduces long-range memory while preserving physical interpretability and provides rigorous convergence analysis. In addition, Xie and Li [54] confirmed that when the temporal fractional-order parameter approaches a specific threshold, network training performance can be significantly improved. The appropriate utilization of fractional historical dependency therefore provides a new mathematical framework for mitigating optimization instability in large-scale models. To summarize the mathematical and engineering distinctions discussed above, Table 4 provides a multi-dimensional comparison between traditional integer-order optimizers and fractional gradient dynamics. Furthermore, it is necessary to clarify specific methodological capabilities that might have been ambiguously categorized in earlier contexts. For instance, Fractional-Order Gradient Descent (FOGD) could be explicitly recognized for its nonlocal properties; as a fundamentally fractional dynamic method, it inherently possesses a moderate global memory capability through its historical convolution kernel, which intrinsically distinguishes it from purely Markovian integer-order baselines.

Table 4. Multi-dimensional comparison between traditional optimizers and fractional gradient dynamics.

Evaluation Dimensions	Classical Integer-Order Optimizers	Fractional Gradient Dynamics (Fractional Optimizers)
Historical Gradient Memory Mechanism	Exponential Decay (Short-term)	Power-law Memory (Long-term)
Nonlocal Temporal Dependence	Local Markovian updates	Global trajectory accumulation
Robustness Against Gradient Vanishing/Explosion	Prone to attenuation in deep models	Alleviated via fractional backward propagation
Resistance to Pathological Oscillation (Noise)	Vulnerable to noise perturbations	Mitigated via memory-based inertial regulation
Computational and Memory Overhead	Low memory footprint	High memory burden (Requires storing historical trajectories)
Mathematical Convergence Guarantees	Well-established	Emerging proofs for discrete evolution
Physical Interpretability of Evolution	Black-box parameter drift	Explicit trajectory-driven Hamiltonian/Fractional action

To provide a balanced perspective, it is critical to acknowledge that the advantages of fractional gradient dynamics are highly application-dependent and possess severe limitations. The mechanisms outlined in Table 4 overstate the universal applicability of fractional systems if specific mathematical and computational constraints are ignored.

First, the global memory capability inherently alters the optimization landscape. Recent theoretical evaluations of fractional gradient descent explicitly warn that its convergence point is not the minimum point in the traditional sense due to the non-local property of the fractional-order differential [21]. Moreover, fractional dynamics are hyper-sensitive to numerical precision and initialization parameters. Numerical analysis of fractional difference systems reveals that distinct initial conditions produce entirely different bifurcation diagrams, severely complicating reproducibility [25].

Furthermore, extreme scaling scenarios expose critical vulnerabilities. When utilizing 64 accumulation steps, discrete fractional weighting heavily amplifies gradient noise and

causes catastrophic cancellation in standard FP32 arithmetic, leading to complete training collapse with test accuracy falling to exactly 1.00% [55]. Additionally, attempts to jointly optimize the fractional derivative order and the neural network parameters face profound training difficulties, requiring specialized loss gradient magnitude regularization to prevent divergence [56]. Therefore, fractional systems must be treated as highly sensitive, task-specific enhancements rather than universally superior replacements for classical optimization.

3.1.2. Fractional Gradient Accumulation

In distributed training of large-scale Transformer models, conventional gradient accumulation typically adopts arithmetic averaging across micro-batch gradients:

$$G_{avg} = \frac{1}{K} \sum_{k=1}^K g_k \quad (8)$$

where g_k denotes the gradient of the k -th micro-batch. In contrast, fractional gradient accumulation introduces a power-law weighted nonlocal aggregation mechanism, whose typical discrete formulation can be expressed as:

$$G_{facc} = \frac{1}{\Gamma(\alpha)} \sum_{k=1}^K w_{K-k} g_k \quad (9)$$

where the weights satisfy the proportional relationship: $w_m \propto (m+1)^{\alpha-1}$ for $m = K-k$, or more precisely $w_m = \frac{\Gamma(m+\alpha)}{\Gamma(\alpha)\Gamma(m+1)}$. When $\alpha = 1$, this mechanism strictly degenerates into conventional uniform accumulation. To explicitly show this architectural distinction, Figure 3 provides a mechanism comparison between conventional arithmetic gradient accumulation and fractional power-law weighted aggregation.

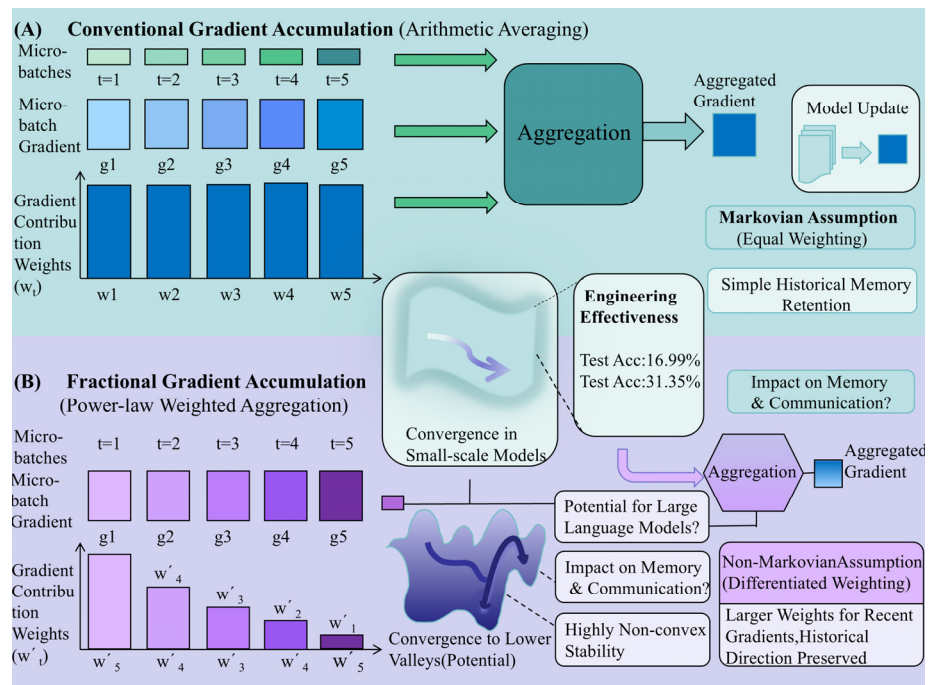


Figure 3. Mechanism comparison between conventional gradient accumulation (arithmetic averaging) and fractional gradient accumulation (power-law weighted aggregation).

The direct computation of weighting coefficients through falling factorials or the explicit ratio of Gamma functions introduces severe computational overhead. As established in the theoretical formulation of fractional difference equations, these operators are mathematically equivalent to falling factorial-law memory [23]. However, directly evaluating

these factorials for each microbatch at every training step is computationally prohibitive and risks catastrophic numerical overflow during large-scale deep learning training.

To resolve this limitation and improve optimization efficiency, advanced numerical approximation techniques are strictly required. Rather than evaluating massive Gamma functions directly, modern implementations leverage logarithmic transformations of binomial coefficients or efficient arbitrary-precision recursive formulations. Specifically, the discrete weighting coefficients w_k can be computed iteratively using the recursive mathematical relation $w_k = w_{k-1}(1 - (\alpha + 1)/k)$ with the initial condition $w_0 = 1$. This recursive formulation drops the computationally expensive factorial ratios entirely. Consequently, it reduces the computational complexity of weight generation to $O(1)$ per discrete step and preserves numerical stability. Similar recursive implementations must be applied to all subsequent discrete fractional convolutions, including the representations defined later in this section, to ensure practical feasibility in memory-constrained distributed systems.

By adjusting the fractional order α , this framework breaks the Markovian assumption of equal weighting among micro-batches and assigns differentiated historical forgetting or retention weights to gradients from different time steps.

Recent studies have validated the effectiveness of this mechanism in deep learning. Lee [55] proposed the FracGrad algorithm, which constructs a power-law weighting scheme based on the discretized Riemann–Liouville fractional integral. This mechanism assigns larger weights to recent micro-batches while preserving early gradient directions through gradually decaying historical contributions. Empirical experiments demonstrated substantial optimization gains on visual architecture. Specifically, on the mini-ImageNet + ResNet-18 benchmark, the test accuracy improved from 16.99% to 31.35% ($p \approx 2 \times 10^{-6}$). Theoretically, this framework exhibits potential for extension toward large language model pretraining.

Nevertheless, current empirical investigations remain largely confined to small- and medium-scale parameter regimes. For ultra-large language models containing tens or hundreds of billions of parameters, such as GPT or LLaMA [57] series models, it remains unclear whether fractional accumulation mechanisms may introduce additional memory-state overhead or inter-node communication latency within complex tensor-parallel and pipeline-parallel training frameworks. Moreover, whether such mechanisms can maintain convergence stability within highly nonconvex loss landscapes still requires large-scale engineering validation.

3.1.3. Fractional Optimizers and Memory Compression

Beyond gradient accumulation, several frontier studies have begun exploring the direct reconstruction of optimizer state update mechanisms using fractional-order dynamics. In conventional optimizers represented by Adam, the first-order momentum update is generally expressed as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (10)$$

where $m_t \in \mathbb{R}^d$ is the first-order moment estimate, $\beta_1 \in [0, 1)$ is the momentum decay coefficient, and $g_t = \nabla L(\theta_t)$ is the gradient at step t . From a mathematical perspective, this recursive formulation fundamentally represents an exponentially decaying memory mechanism, which rapidly overwrites and forgets early historical information. In contrast, fractional optimizers aim to replace exponential forgetting with power-law historical memory. Conceptually, the core formulation can be expressed as:

$$m_t^{(\alpha)} = \sum_{k=0}^t u_k^{(\alpha)} g_{t-k} \quad (11)$$

where the weighting coefficients follow the power-law decay: $u_k^{(\alpha)} \propto k^{-\alpha}$, $u_k^{(\alpha)} = \frac{\Gamma(k+\alpha)}{\Gamma(\alpha)\Gamma(k+1)}$, satisfying $u_k^{(\alpha)} \sim k^{\alpha-1}/\Gamma(\alpha)$. This nonlocal mechanism allows earlier gradients to continue exerting substantial influence on the current parameter update direction.

Based on this principle, El-Nabulsi and Anukool [43] proposed the fractional action Hamiltonian neural network framework, demonstrating that introducing fractional derivatives into neuronal and synaptic updates enables effective integration of historical sequence information through long-term memory kernels. This framework provides a mathematically rigorous approach for alleviating catastrophic forgetting and capturing long-range temporal correlations.

Similarly, Zhou et al. [58] proposed a fractional-order stochastic gradient descent method incorporating momentum and energy terms. Their experiments on complex non-convex problems demonstrated that fractional dynamics could achieve higher optimization accuracy compared with conventional integer-order optimization algorithms, thereby providing important theoretical insights for optimization challenges in large language models.

Although fractional operators theoretically possess superior global memory capabilities, retaining full historical integrals without truncation inevitably leads to uncontrollable computational cost and memory expansion. To address this engineering challenge, researchers have explored multiple memory compression and truncation approximation strategies. Herrera-Alcántara et al. [59] proposed the FAdamWav optimizer, which integrates wavelet transforms with fractional gradients. This method combines the gradient memory compression capability of wavelet transforms with the nonlocal modeling advantages of fractional gradients. Theoretical analysis and experimental results demonstrated that multilayer wavelet transforms can reduce gradient memory consumption by up to 87.5% while maintaining competitive optimization performance, thereby offering an innovative solution to the memory bottleneck in large-scale training.

Meanwhile, truncation approximation schemes based on the Grünwald–Letnikov discrete formulation have also demonstrated high engineering feasibility. As summarized by Khan [46], retaining only $M = 10$ historical gradient terms in practical applications is often sufficient to capture the essential fractional dynamical characteristics required for deep network training while effectively controlling per-step computational complexity and memory overhead.

In addition, empirical studies have shown that the final optimization performance is highly sensitive to the fractional order parameter α . To adapt to the complex geometric structures of nonconvex loss landscapes, recent frontier algorithms have gradually abandoned fixed fractional-order settings in favor of dynamic evaluation mechanisms. Khan [46] summarized that existing methods are already capable of dynamically adjusting fractional orders according to local curvature information or introducing meta-learning mechanisms to adaptively design fractional decay strategies, thereby providing rigorous constraints for dynamic optimization in high-dimensional parameter spaces. However, current experimental validations of fractional memory compression and dynamic optimization are still mainly concentrated on medium-scale vision models or general neural networks. Table 5 summarizes the fractional optimizer and memory compression strategies.

Table 5. Summary of fractional optimizers and memory compression strategies.

Category	Method/Framework	Core Mechanism	Memory Characteristic	Primary Advantage/Impact
Baseline Systems	Conventional Optimizers (e.g., Adam)	Reply on first-order momentum updates.	Exponential decay.	Limitation: Rapidly overwrites and forgets early historical information.

Theoretical Foundations	Core Fractional Optimizers	Replaces local derivatives with historical sequence integration.	Power-law historical memory.	Allows earlier gradients to maintain substantial influence on current parameter updates.
Advanced Architectures	Fractional Action Hamiltonian NN	Introduces fractional derivatives into neuronal and synaptic updates.	Long-term memory kernels.	Alleviates catastrophic forgetting and effectively captures long-range temporal correlations.
Optimization Algorithms	Fractional-Order SGD	Incorporates momentum and energy terms via fractional dynamics.	Global memory capabilities.	Achieves higher optimization accuracy on complex nonconvex problems compared to integer-order methods.
Memory Compression	FAdamWav Optimizer	Integrates wavelet transforms with fractional gradients.	Compressed gradient memory.	Reduces gradient memory consumption by up to 87.5% while maintaining competitive performance.
Memory Compression	GL Truncation Approximation	Retains only a finite number of historical gradient terms.	Truncated fractional dynamics.	Effectively controls per-step computational complexity and memory overhead.
Parameter Adaptation	Dynamic Order Evaluation	Adjusts fractional orders via local curvature or meta-learning.	Adaptive fractional decay.	Provides rigorous constraints for dynamic optimization, adapting to complex non-convex loss landscapes.

3.1.4. Limitations, Contradictory Findings, and Failure Modes

Although numerous studies have reported improved convergence characteristics using fractional-order mechanisms, these advantages are not universally consistent. Critical examinations of recent literature reveal that fractional methods can severely underperform, introduce computational bottlenecks, or even collapse under specific conditions.

Firstly, inappropriate tuning of the fractional order combined with extreme problem scales can trigger catastrophic training failures. It has been explicitly documented that in fractional gradient accumulation, when the accumulation steps reached extreme levels ($N = 64$) coupled with an aggressive recency bias ($\alpha = 0.3$), the optimization experienced catastrophic training collapse, with accuracy plummeting to random guessing levels (1.00%) [55]. This failure was primarily attributed to numerical precision degradation: the minuscule power-law weights assigned to early gradients suffered from catastrophic cancellation when summed with recent large-weighted gradients in standard FP32 arithmetic.

Secondly, empirical evidence indicates an inherent time-memory trade-off across fractional algorithms. While wavelet-based fractional optimizers theoretically compress gradient memory by up to 87.5%, practical deployments demonstrated that this memory reduction incurs a latency penalty, increasing execution time by 6.5% to 12.1% due to the discrete transformation operators [59]. Similarly, inside the network architecture, deploying trainable fractional Jacobi basis functions or rational Padé approximations introduces significant computational overhead, with empirical profiling indicating that these fractional activations can be two to three times slower to evaluate than standard ReLU operations [60,61].

Thirdly, fractional mechanisms exhibit profound dependency on the choice of the base optimizer and lack large-scale empirical validation. For instance, fractional gradient weighting severely underperformed when coupled with standard SGD (yielding a mere 12.78% accuracy), requiring adaptive optimizers like Adam to achieve convergence [55]. Furthermore, despite sophisticated theoretical guarantees, the empirical validation of advanced frameworks remains restricted to small-scale tasks. Evaluations range from MATLAB (R2020a, version 9.8.0) simulations on down-sampled MNIST digits [43] to 1.6 million parameter RNNs for time series [51], leaving their scalability to billion-parameter autoregressive language models unverified.

Therefore, fractional optimization should not be regarded as a universally superior replacement for integer-order methods, but rather a mathematically principled technique highly sensitive to numerical precision limits, parameter scaling, and computational throughput.

3.1.5. Quantitative Evaluation of Resource Overhead and Complexity

Although mitigating the “memory wall” and lowering peak training hardware requirements are frequently cited as primary motivations for introducing fractional calculus, a comprehensive, multi-dimensional quantification of actual resource footprints remains scarce. To evaluate the architectural efficiency systematically, Table 6 aggregates the explicit empirical hardware metrics, including parameter changes, memory utilization, and temporal latency, reported in representative fractional neural implementations relative to their classical integer-order baselines.

Table 6. Quantitative analysis of computational overhead and scale in representative fractional implementations.

Framework Paradigm	Maximum Evaluated Architecture	Parameter Increase (%)	Empirical Memory Footprint Changes	Empirical Latency/Training Time Changes
FAdamWav [59]	2-layer MLP (~50,000 parameters) on MNIST	0% (Zero-parameter overhead claimed)	−16.8% to −34.6% actual GPU memory reduction (from 28.8 MB baseline)	+6.5% to +12.1% increase in epoch execution duration
FracGrad [55]	ResNet-18 (~11 million parameters) on mini-ImageNet	0% (Zero-parameter overhead)	Minimal footprint increase (confined to N scalar weights)	<1% overhead (reported as statistically negligible)
FLRNN-FGA [51]	Time-Series Recurrent Network (1.60 million parameters)	0% (Internal weight restructuring)	Total training footprint documented explicitly at 0.49 GB	Fixed at 13.51 s per training epoch
fKAN [60]/rKAN [61]	CNN-based KANs on MNIST classification	Task-dependent small overhead	Not Quantitatively Reported	Trainable fractional activations are 2 to 3 times slower than standard ReLU
FPPformer [62]	Time-Series Spatial Transformer	Negligible structural addition	Footprint remains strictly comparable to iTransformer	Slightly longer training latency due to auxiliary GRU integration

As indicated in Table 6, a prominent engineering trade-off exists within the current literature: scalar-based fractional updates (e.g., FracGrad) preserve extreme computational throughput but offer restricted architectural modifications, whereas structural transformations (e.g., wavelet-compressed optimizers or rational KAN layers) provide profound non-local representation advantages at the cost of non-negligible latency overhead or prolonged training cycles. This empirical distribution highlights that extending fractional calculus to trillion-parameter scales requires solving the processing bottlenecks associated with non-local operations.

3.2. Fractional Representation Learning and Semantic Enhancement

Pretrained Transformer encoders demonstrate excellent performance in capturing local contextual information, yet they often struggle to fully exploit long-term statistical dependencies among domain-specific terminologies. To address this limitation, researchers have attempted to introduce fractional nonlocal statistical memory to enhance underlying semantic representations. Karim et al. [63] proposed a hybrid ensemble framework for engineering-domain sentiment analysis, in which the memory characteristics of fractional derivatives were utilized to compensate for the limitations of Transformers in modeling lexical interactions within specialized domains. Rather than reconstructing the internal Transformer architecture, this approach established an independent fractional lexical feature channel. Specifically, TF-IDF was first employed to obtain the basic feature weights, followed by the introduction of fractional-order differences based on the Grünwald–Letnikov definition to reconstruct the feature vectors. This transformation converts independent term statistics into nonlocal correlation features through weighted accumulation of historical information. Such a mechanism adjusts the weight distribution of textual fea-

tures and enhances the sensitivity of the model to subtle and nonlinear lexical interactions, thereby achieving superior performance compared with purely semantic embeddings in sentiment classification tasks.

In addition, nonlinear dynamical mechanisms provide a new perspective for improving the robustness of feature representations. Yan et al. [64] demonstrated in complex feature protection tasks that memristive chaotic mapping mechanisms based on the Rulkov neuron model can effectively improve feature robustness under dynamic environments. This finding suggests that introducing dynamical constraints into representation space optimization may serve as a potential pathway for enhancing the stability of semantic representations in large language models.

Overall, fractional mechanisms achieve semantic enhancement by characterizing long-term statistical relationships. Although current studies are primarily based on pre-trained encoders such as BERT, they reveal the complementary relationship between self-attention mechanisms for capturing deep semantic structures and fractional processing for modeling nonlocal correlations. This provides theoretical guidance for constructing highly robust and domain-aware representation layers in future large language models.

3.3. Fractional and Spectral Transformer Architectures

3.3.1. Fractional Fourier and Spectral Token Mixing

The computational complexity of the standard self-attention mechanism scales as $O(N^2)$, which severely limits the applicability of large language models in ultra-long-context scenarios. Achieving efficient token mixing through frequency-domain transformations has therefore become an important direction for overcoming this bottleneck. For example, Fourier Transform (FT)-based approaches can significantly reduce computational costs while preserving global interaction capability. Zhao et al. [51] utilized time-frequency transformations for frequency-domain sampling and incorporated gated mechanisms to filter redundant noise when constructing time-series models, thereby effectively improving long-sequence processing efficiency. This demonstrates that spectral mappings can preserve sensitivity to long-range nonlocal dependencies while substantially reducing computational overhead.

Within this context, the Fractional Fourier Transform (FrFT) has attracted increasing attention due to its continuous rotational characteristics in the time-frequency plane. The fractional order α and the rotation angle ϕ satisfy the relationship: $\alpha = \frac{2\phi}{\pi}$. When $\alpha = 1$, the formulation strictly degenerates into the standard Fourier Transform. The continuity of the fractional order grants the model the capability to dynamically balance representations between the time and frequency domains. Based on this principle, several studies have proposed Fractional Fourier Networks (FrFNets), which reconstruct Transformer encoders using discrete FrFT. Leveraging the nonlocal integral properties of FrFT, these architectures capture global nonlocal correlations through multi-angle projections and theoretically achieve near-linear complexity for long-context processing.

Furthermore, token mixing based on broader orthogonal bases in spectral spaces is becoming a major evolutionary direction. Qin et al. [65] proposed the Toeplitz Neural Network, which achieves near-linear sequence modeling through spectral decomposition of relative positional encoding. More recently, orthogonal transformations have been deeply integrated into novel dynamic mapping networks. Xu et al. [66] and Zhu and Connor [67] constructed Fourier-KAN architectures by replacing traditional spline functions with Fourier series. Meanwhile, Bozorgasl and Chen [68] and Seydi et al. [69] proposed deep nonlinear mapping architectures based on wavelet transforms. These developments suggest that introducing continuous orthogonal mappings such as fractional Fourier transforms

or wavelets into the token mixing stage may enable efficient low-computation modeling of multi-scale semantic frequency-domain correlations in ultra-long texts.

3.3.2. Fractional Positional Encoding and Long-Context Representation

In Transformers, standard sinusoidal positional encoding is generally formulated as: $PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right)$ and $PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{2i/d}}\right)$. However, in ultra-long-context and deep-network scenarios, token representations are highly susceptible to the over-smoothing problem, resulting in the loss of fine-grained discriminative capability. The mathematical advantages exhibited by fractional operators in balancing multi-scale features provide important insights for optimizing positional representations.

Yang et al. [70] demonstrated in optical flow research that dynamically adjusting the fractional order can achieve a balance between preserving global features and extracting local textures, thereby fundamentally alleviating over-smoothing. This mechanism suggests that introducing dynamically adjustable non-local operators into positional encoding may provide a key pathway for simultaneously preserving global semantics and local contextual details in long-context scenarios.

At the algorithmic level, constructing adaptive continuous positional representations through fractional interpolation has emerged as a frontier direction. Zhou et al. [62] proposed Fractional Positional Embedding (FPE) within the FPPformer architecture. By leveraging fractional calculus, FPE achieves scalable positional representations and overcomes the flexibility limitations of conventional trigonometric embeddings under diverse temporal patterns, thereby significantly improving multivariate time-series forecasting performance.

Furthermore, for insertion-based generation tasks, Zhang et al. [71] designed a continuous fractional positional encoding strategy that dynamically generates positional representations through fractional interpolation between neighboring token positions. This mechanism allows models to reuse historical representations, significantly reducing computational cost and decoding latency, thereby demonstrating the potential of fractional interpolation for continuous positional mapping and extrapolation.

3.3.3. Fractional KAN and State-Space Variants

Neural networks based on the Kolmogorov–Arnold representation theorem (KANs) replace traditional fixed activation functions with learnable one-dimensional splines or orthogonal polynomials along network edges, thereby exhibiting stronger function approximation capability and interpretability. Liu et al. [72] introduced the original KAN framework, which has rapidly expanded into time-series and dynamical system modeling applications. Dutta et al. [73] reviewed the rapid development of KAN-related methods. For example, Genet and Inzirillo [74] proposed the Temporal Kolmogorov–Arnold Network (TKAN), which successfully introduced more flexible basis function representations into sequential memory modeling.

To overcome the limitations of traditional basis functions in modeling non-Markovian long-range dependencies, Aghaei [60] proposed the Fractional Kolmogorov–Arnold Network (fKAN). The fKAN architecture replaces conventional splines with trainable fractional Jacobi basis functions $P_n^{(\alpha, \beta)}(x)$, thereby enabling continuous nonlocal representation of historical state variables and accurately capturing long-term dependencies.

In addition, Aghaei [61] proposed the rational rKAN architecture based on Padé approximation, which improves parameter efficiency while enhancing nonlinear fitting accuracy. Furthermore, Afzal Aghaei [75] developed the physics-informed KANtrol framework, directly applying KAN architectures to multidimensional fractional optimal control problems.

To address the challenge of dynamic stability in large-scale model training, Shafiya et al. [76] theoretically demonstrated through Linear Matrix Inequalities (LMI) that fractional calculus can provide stronger global stability guarantees for deep neural networks.

Based on these theoretical foundations, researchers have begun advancing architectural integration. Yang and Wang [77] proposed the Kolmogorov–Arnold Transformer (KAT), while Fares et al. [78] introduced the TFKAN architecture. Both frameworks replace the traditional feed-forward networks in Transformers with KAN layers based on continuous polynomial approximation. By integrating fractional nonlinear representation with KAN architectures, these methods provide a computationally efficient state-space variant for large-scale sequence models under rigorous parameter constraints while simultaneously introducing long-range memory capability.

Integrating these architectural advancements, Figure 4 illustrates a comprehensive four-stage dynamic state-space modeling framework that leverages fractional positional encoding (FPE) and fractional KANs (fKAN) to efficiently process multi-scale sequence inputs. Table 7 summarizes recent architectural advancements integrating fractional operators into token mixing, positional encoding, and state-space variants.

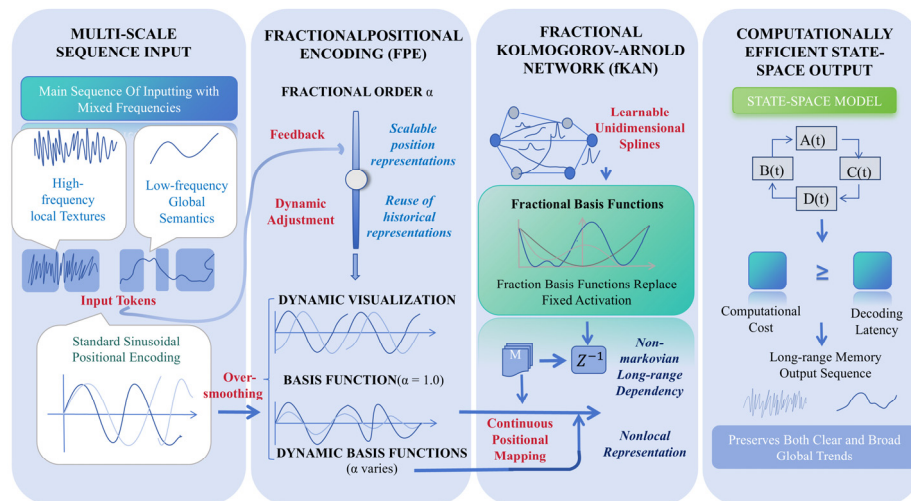


Figure 4. Four-stage dynamic state-space modeling using fractional positional encoding and Kolmogorov–Arnold Networks for long-context Transformers.

Table 7. Advanced fractional architectures for long-context modeling and state-space variants.

Domain	Framework/Technique	Core Mechanism	Key Limitation Addressed	Primary Advantage
Token Mixing and Spectral Methods	Standard Spectral Transform	Utilizes time-frequency transformations and gated sampling.	Standard self-attention quadratic complexity.	Significantly reduces computational costs while preserving global interaction capability.
	Fractional Fourier Networks (FrFNets)	Reconstructs Transformer encoders via continuous time-frequency rotation.	Rigid boundaries between time and frequency domain representations.	Dynamically balances domains and captures global correlations with near-linear complexity.
	Orthogonal Spectral Mappings	Integrates Toeplitz, Fourier series, or wavelets into dynamic mappings.	High-computation overhead in multi-scale semantic modeling.	Enables highly efficient, low-computation modeling of multi-scale frequency-domain correlations.
Positional Encoding (PE)	Dynamic Fractional Operators	Adjusts fractional order to balance feature extraction.	Token over-smoothing and loss of fine-grained discriminative capability.	Simultaneously preserves global semantics and local contextual details in deep networks.
	Fractional Positional Embedding (FPE)	Constructs adaptive, continuous positional representations.	Flexibility constraints of conventional trigonometric embeddings.	Achieves scalable representations and significantly improves multivariate forecasting performance.

Fractional KAN Variants	Continuous Fractional Interpolation	Generates representations via interpolation between neighboring tokens.	High computational cost and decoding latency in insertion tasks.	Allows models to reuse historical representations, enabling efficient continuous positional mapping.
	Fractional KAN (fKAN)	Replaces conventional splines with trainable fractional basis functions.	Limitations of traditional basis functions in non-Markovian systems.	Enables continuous nonlocal representation to accurately capture long-term sequence dependencies.
	KAN-Transformer Integration (KAT/TFKAN)	Replaces traditional Transformer feed-forward networks with KAN layers.	Parameter inefficiency and training instability in large-scale models.	Provides computationally efficient state-space variants with rigorously proven global stability guarantees.

3.4. Fractional Scientific AI and Equation Discovery

3.4.1. Transformer-Based Fractional Differential Equation Solvers

In the field of scientific machine learning, deep neural networks have been widely applied to the equation-free discovery and numerical solution of complex non-Markovian physical systems. In recent years, for Fractional Partial Differential Equations (FPDEs) involving long-range historical dependencies, Firoozsalari et al. [79] proposed a novel neural architecture named deepFDENet, which effectively achieves numerical approximation of fractional operators under complex boundary conditions. To further overcome computational bottlenecks, Taheri et al. [80] introduced derivative operational matrices Into Fractional Physics-Informed Neural Networks (FPINNs), significantly accelerating convergence when handling nonlocal evolution equations. As highlighted by Coelho et al. [81] and Vellappandi and Lee [12], neural fractional differential equations and physics-informed fractional neural networks have emerged as state-of-the-art computational paradigms for modeling long-memory effects in complex systems. Furthermore, to address the reconstruction of dynamical systems, Koenig et al. [82] proposed the KAN-ODEs framework, which employs Kolmogorov-Arnold Networks to uncover hidden physical laws, thereby providing a new theoretical pathway for basis-function representations of nonlinear systems.

To address the limited convergence accuracy of conventional physics-informed neural networks when solving highly stiff fractional equations, Transformer-based architectures have demonstrated significant advantages in capturing nonlocal long-range dependencies through self-attention mechanisms. Li et al. [83] proposed an AI-enhanced physics-constrained Transformer framework specifically designed for solving fractional partial differential equations. The framework leverages the attention mechanism of Transformers to effectively capture long-range dependencies within data and combines convolutional neural networks for positional encoding, thereby mitigating the gradient vanishing and explosion problems commonly encountered by conventional sequence models when handling fractional operators. Numerical experiments demonstrated that the proposed method achieves high accuracy in solving one-dimensional, two-dimensional, and spatiotemporal fractional partial differential equations. In addition, to address the numerical challenges posed by stiff equations, Kang et al. [84] combined the L_1 approximation of the Caputo fractional derivative with a two-stage adaptive optimization strategy, further improving parameter stability and computational accuracy in modeling nonlinear diffusion processes.

When dealing with strongly nonlinear complex dynamical systems, the flexibility of fractional-order modeling has been increasingly integrated with the generalization capability of deep learning. Von Kluge et al. [85] proposed a physics-inspired hybrid modeling framework specifically designed for fractional chaotic systems. This study utilized deep learning techniques to directly approximate complex fractional evolution dynamics from observational data and, for the first time, extended the modeling capability to variable-order systems, thereby providing an accurate mathematical paradigm for describing

physical processes with time-varying memory characteristics. Meanwhile, to address time-series prediction in chaotic systems, Hassan et al. [86] optimized a deep nonlinear autoregressive exogenous neural network framework using the Levenberg–Marquardt algorithm and successfully achieved high-precision modeling of fractional-order Lorenz–Lü–Chen chaotic trajectories. These studies indicate that deep temporal architectures can overcome the computational bottlenecks of traditional numerical forecasting approaches when modeling highly sensitive fractional nonlinear systems, thereby providing a new data-driven pathway for understanding complex spatiotemporal dynamics.

To address the physically unreasonable topological oscillations that frequently arise when neural networks approximate nonlocal operators, recent studies have increasingly focused on the smoothness and stability of solutions. As emphasized by Partohaghighi et al. [87] in their investigation of roughness-informed machine learning based on Persistence-Sensitive Optimization (PSO) [88], incorporating dynamically adaptive topological roughness penalties on top of physics-constrained losses can preserve the global topological smoothness of solutions. By introducing persistence-sensitive optimization, researchers can effectively identify and eliminate short-lived topological features associated with noise, thereby enabling Transformer-based fractional physical solvers to avoid numerical artifacts caused by overfitting. Table 8 categorizes fractional solvers across their architectural features and multidimensional modeling capabilities.

Table 8. Multidimensional capability matrix of transformer-based and fractional differential equation Solvers.

Framework/Technique	FPDE Solving	Transformer/Attention	Chaotic Dynamics	Convergence and Stability	Topological Smoothness	Key Contribution
deepFDNet	✓	×	×	×	×	Approximates fractional operators efficiently under complex boundary conditions.
Modified FPINNs	✓	×	×	✓	×	Accelerates convergence for nonlocal evolution equations using operational matrices.
Neural FDEs and PiFNNs	✓	×	×	×	×	Models long-memory effects in complex non-Markovian physical systems.
KAN-ODEs	×	×	✓	×	×	Uncovers hidden physical laws through advanced basis-function representations.
Physics-Constrained Transformer	✓	✓	×	✓	×	Mitigates gradient vanishing/explosion when solving spatiotemporal FPDEs.
Adaptive L1-Optimization	✓	×	×	✓	×	Improves parameter stability for highly stiff fractional diffusion equations.
Physics-Inspired Hybrid Model	×	×	✓	×	×	Approximates variable-order, time-varying complex fractional evolution dynamics.
Deep NARX Network	×	×	✓	×	×	Achieves high-precision modeling and forecasting of fractional chaotic trajectories.
Roughness-Informed ML (PSO)	×	×	×	×	✓	Preserves global smoothness and eliminates noise-induced numerical artifacts.

Note: ✓ indicates that the framework or technique possesses the respective capability or feature; × indicates the absence of that capability or feature.

3.4.2. Data-Driven Fractional Equation Discovery

In the modeling of complex natural systems, conventional integer-order equations derived from first principles are often overly simplified and therefore incapable of effective

tively capturing non-Markovian memory effects. To overcome this limitation, data-driven fractional equation discovery has emerged as a frontier direction in scientific machine learning. Early studies primarily relied on sparse regression or physics-informed neural networks; however, sparse regression performs poorly in high-dimensional noisy environments, whereas physics-informed neural networks strongly depend on prior assumptions regarding equation structures. To directly recover explicit physical laws from discrete observational data, researchers have increasingly explored white-box algorithms capable of accurately characterizing nonlocal dynamics while maintaining strong interpretability. As demonstrated by the stepwise data-driven framework proposed by Xiangnan et al. [89] for complex physical systems, the method reconstructs data using deep neural networks and combines automatic differentiation with Gauss–Jacobi quadrature to approximate fractional derivatives. Its core mechanism lies in employing alternating optimization algorithms to simultaneously identify linear coefficients and nonlinear fractional orders, thereby achieving high-precision computation while effectively handling singularities.

The field of deep learning for scientific computing imposes extremely high numerical approximation requirements on data-driven reconstruction of fractional dynamics. To address the gradient deficiencies of conventional neural networks when modeling evolutionary systems with long-range memory decay, [90] designed a novel neural architecture with customized topological structures. By incorporating specific algebraic constraints and penalty terms, the proposed model achieved efficient numerical approximation of nonlocal partial differential equations. To further enhance the discovery capability of deep evolution operators in complex non-Markovian physical systems, Nehma and Tiwari [91] demonstrated the substantial potential of leveraging novel continuous-feature networks to improve deep Koopman operator identification. These studies indicate that integrating fractional calculus operators with physics-informed architectures can significantly improve the accuracy of AI solvers in reconstructing unknown nonlocal equations from large-scale discrete observational data.

In recent years, large language models and symbolic reasoning techniques have also been introduced into automatic equation discovery tasks. Under this paradigm, large language models primarily serve as generators of candidate symbolic expressions and symbolic suggestions, whereas the core solution process still relies on numerical optimization and sparse regression. As demonstrated by the LLM4ED framework proposed by Du et al. [92], the commonsense reasoning capability of language models can be utilized to constrain the search space of symbolic regression, thereby attempting to overcome interdisciplinary computational barriers in discovering complex fractional equations. Under this framework, the discovery process can be abstracted as an optimization problem over the parameters ξ and fractional orders α : $\partial_t u \approx \sum_i \xi_i L_i^{(\alpha_i)} u$, where $L_i^{(\alpha_i)}$ denotes the candidate operator library containing fractional operators. Although this paradigm indicates that large language models have begun to participate in fractional scientific computing as logical controllers, it should be rigorously emphasized that describing such systems as “large language models automatically discovering physical laws” remains somewhat overstated at the current stage.

Overall, data-driven fractional equation discovery is evolving from pure numerical fitting toward physically consistent symbolic discovery. The empirical study by Xiangnan et al. [89] demonstrated strong robustness on both synthetic anomalous diffusion datasets and frozen-soil creep experimental data, effectively identifying equation structures containing power-law memory effects. Future research should focus on further integrating the heuristic guidance capability of large language models with the precise numerical representation of fractional operators, thereby constructing generalized physical discovery platforms capable of handling large-scale, high-noise real-world observational data. At present, the mathematical foundations underlying this integration remain in an exploratory

tory stage, and its theoretical completeness still requires rigorous validation through more stringent physical experiments.

3.5. Fractal Geometry and Embedding Space Analysis

Understanding the geometric structure of high-dimensional embedding spaces in large pretrained models is a critical pathway toward interpreting model behavior. Existing studies suggest that the ability of large language models to perform long-context reasoning fundamentally arises from their capacity to capture the latent topological structures of natural language. Ribeiro et al. [93] demonstrated that human language structures exhibit strong fractal patterns at both macroscopic and microscopic levels. When such self-similar natural language sequences are mapped into the high-dimensional embedding spaces of large language models, their intrinsic fractal geometric characteristics become embedded within the semantic representation matrices of the models.

During the evolution of deep Transformer architectures, token embeddings gradually converge toward higher-level semantic representations. Lee [94] conducted an in-depth analysis of the geometric evolution of embedding spaces and found that the embedding similarity gradients across different network layers exhibit pronounced fractal self-similarity. This finding confirms from the perspective of fractal geometry that semantic alignment is a progressively refined hierarchical process displaying fractal-like structural characteristics. To quantitatively characterize this geometric complexity, researchers introduced the correlation dimension D_2 as a measurement metric. Lee [95] demonstrated through an analysis of GPT-2 embedding spaces that the correlation dimension remains highly stable across different lexical subspaces, revealing the existence of robust structural properties within internal language representations.

From the perspective of model weight distributions, high-dimensional embedding spaces and parameter manifolds often exhibit highly complex nonsmooth characteristics. Partohaghighi et al. [87] proposed a unified roughness-informed framework revealing the intrinsic mathematical relationship between the statistical roughness of deep neural network weights and fractal geometry. Based on random matrix theory, the empirical spectral density of weight matrices in overparameterized networks typically follows a pronounced power-law heavy-tailed distribution: $p(\lambda) \sim \lambda^{-\zeta_i}$, where ζ_i denotes the heavy-tail exponent. Such distributions endow models with strong representational capability while simultaneously introducing nonsmooth scale invariance. By incorporating a dual-scale effective dimension combining the Fisher information matrix and fractal box-counting dimensions, researchers can effectively quantify manifold roughness and provide a new geometric perspective for selecting large language model architectures, namely by constraining heavy-tail exponents to mitigate severe overfitting and training instability.

The stability of semantic representations can also be interpreted through attractor analysis in dynamical systems theory. Wang et al. [52] found through the analysis of attractor distributions in multi-region neural networks that complex multi-stability characteristics exist within such systems, and that the topological structure of attractors is strongly modulated by synaptic coupling parameters. This perspective suggests that the token encoding process can be regarded as a convergence process toward specific semantic attractors. Employing fractal geometric tools to analyze the dimensions of these latent attractors may therefore help evaluate the robustness of semantic representations under input perturbations. To characterize the nonlocal propagation of semantic features within embedding spaces, Xu et al. [96] proposed an ultraslow diffusion model based on the Caputo nonlocal structural derivative. The model suggests that semantic diffusion processes in embedding spaces are influenced not only by local tokens but also by structural constraints derived from global historical contexts, thereby providing a new physical-geometric interpretation for quantifying memory decay in long-context modeling. Table 9

summarizes recent analytical frameworks based on fractal geometry, manifold roughness, and dynamical systems.

Table 9. Fractal geometric frameworks and embedding space analysis in large language models.

Domain	Core Concept/Methodology	Primary Mechanism/Finding	Key Implication for LLMs
Natural Language Topology	Fractal Geometric Patterns	Human language structures exhibit strong macroscopic and microscopic self-similarity.	Inherent geometric characteristics naturally embed into the semantic representation matrices of models.
Embedding Space Evolution	Similarity Gradients	Token embeddings across different network layers exhibit pronounced fractal self-similarity.	Confirms that semantic alignment is a progressively refined, hierarchical process.
Structural Complexity	Correlation Dimension	Geometric measurement metrics remain highly stable across diverse lexical subspaces.	Reveals the existence of robust, complex structural properties within internal language representations.
Parameter Manifold Roughness	Heavy-Tailed Distributions and Dual-Scale Dimension	Weight matrices exhibit power-law distributions that introduce nonsmooth scale invariance.	Quantifies manifold roughness to guide architectural selection, mitigating severe overfitting and instability.
Semantic Stability	Attractor Analysis in Dynamical Systems	Token encoding operates as a convergence process toward specific semantic attractors.	Evaluates the robustness of semantic representations under input perturbations using fractal geometric tools.
Memory Decay and Propagation	Ultraslow Diffusion Model	Semantic diffusion is structurally constrained by global historical contexts via nonlocal derivatives.	Provides a novel physical-geometric interpretation for quantifying memory decay in long-context modeling.

3.6. Reproducibility and Open-Source Ecosystem

Assessing the engineering accessibility and reproducibility of fractional artificial intelligence algorithms is essential for understanding their readiness for large language model deployment. Our systematic review reveals a transitional landscape. While foundational theoretical works rarely provide accessible artifacts, the latest architecture demonstrates an accelerating trend toward open-source compliance. As summarized in Table 10, a growing number of recent high-quality studies have begun to release their official codebases and hardware configurations.

Table 10. Reproducibility and benchmarking profiles of the reviewed fractional AI architectures.

Specific Literature/Architecture	Code Availability	Primary Evaluation Benchmarks Utilized
FROND [11]	Yes (https://github.com/zknus/ICLR2024-FROND) (accessed on 15 June 2026)	Graph representation and long-term dependency tasks
DRAGON [97]	Yes (https://github.com/zknus/NeurIPS-2024-DRAGON) (accessed on 15 June 2026)	Complex graph dynamics
FADE [98]	Yes (https://github.com/cuiwjTech/NeurIPS2025_FADE) (accessed on 15 June 2026)	Attention mechanisms and fractional differential equations
FracGrad [55]	Partial (Distributed via PyPI package fracgrad; dataset available via Hugging Face https://huggingface.co/datasets/timm/mini-imagenet) (accessed on 15 June 2026)	mini-ImageNet, ResNet-18
FAdamWav [59]	Yes (Hosted and accessible at http://ia.azc.uam.mx/sourcecode) (accessed on 15 June 2026)	MNIST handwritten digit database
FLRNN-FGA [51]	No (The official repository URL is omitted in the manuscript)	Multi-domain time series (Electricity, ETT, Weather, Traffic, Exchange)
fKAN [60]	Yes (https://github.com/alirezaafzalaghaei/fKAN) (accessed on 15 June 2026)	Synthetic regression targets, Fashion-MNIST, IMDB

rKAN [61]	Yes (https://github.com/alirezaafzalaghahi/rKAN) (accessed on 15 June 2026)	Synthetic regression targets, Fashion-MNIST, IMDB
FPPformer [62]	Yes (https://github.com/jancely/FPPformer) (accessed on 15 June 2026)	IoT-driven multivariate time-series (Weather, PEMS08)

Cross-referencing the surveyed corpus indicates that while many early fractional AI papers do not provide public implementations, a growing cohort of recent top-tier works has released their code. However, evaluation suites remain heavily fragmented. Fractional algorithms are currently validated across isolated and domain-specific tasks ranging from time-series forecasting to small-scale vision classification, rendering cross-architectural algorithmic comparison impossible.

This absence of unified evaluation protocols underscores the urgent need for the research community to establish a standardized benchmarking framework. Developing a unified Fractional Large Language Model Evaluation Suite (Frac-LLM Suite), analogous to GLUE or HELM, would provide baseline distributed training environments and standardized long-context language tasks. Such a repository-driven suite is strictly necessary to facilitate reproducible and comparative engineering validation of fractional non-local dynamics on large-scale generative sequence models.

4. Large Language Models: Core Bottlenecks and Potential Opportunities for Fractional Mechanisms

The review presented in Section 3 demonstrates that fractional calculus has already shown preliminary theoretical potential for overcoming the limitations of traditional Markovian assumptions in foundational sequence modeling tasks, including optimization dynamics, long-range feature representation, and spectral transformers. However, to effectively transfer these underlying mechanisms into generative LLMs, it is first necessary to thoroughly analyze the architectural and computational bottlenecks of current LLM systems. Based on the current state of development, this section aims to summarize the major limitations of contemporary large language models and examine potential entry points for integrating fractional operators to enhance large-scale model architectures. Figure 5 outlines the comprehensive training paradigm of modern Large Language Models, setting the foundation for analyzing the optimization and memory bottlenecks that arise during these large-scale scaling phases.

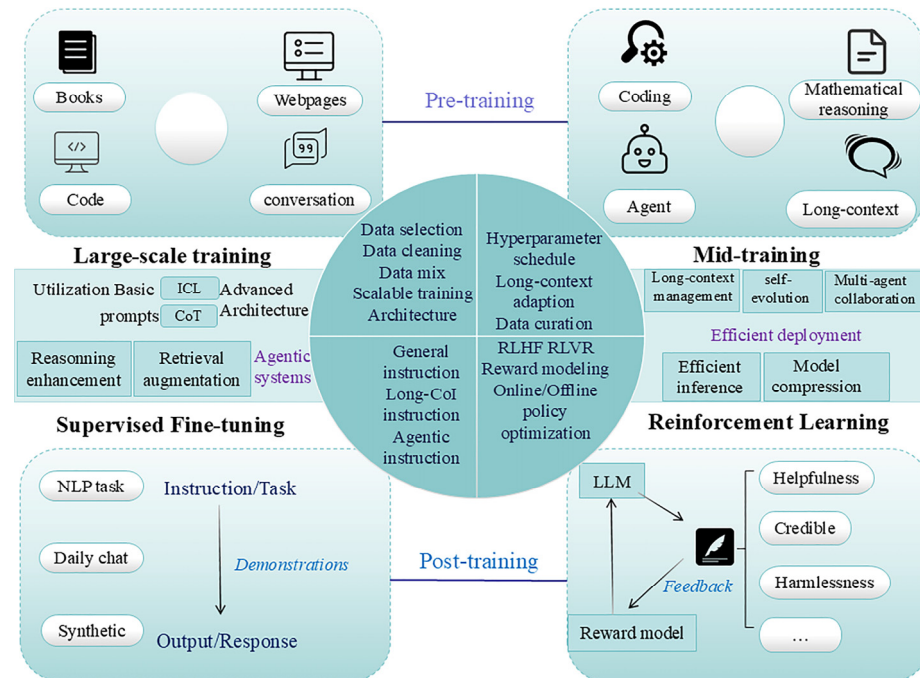


Figure 5. The comprehensive training paradigm of Large Language Models, encompassing pre-training, mid-training, and post-training phases.

4.1. Long-Context Memory and Nonlocal Dependency Challenges in LLMs

The self-attention mechanism in Transformer architectures exhibits $O(N^2)$ temporal and spatial complexity, which constitutes the core computational bottleneck limiting the expansion of context windows in large language models. Constrained by a fixed explicit working memory capacity, historical information beyond the context window is typically truncated, resulting in memory degradation, information loss, and “Ghost Context” hallucinations during long-term interactions, namely the erroneous activation of irrelevant historical information [99].

A central challenge in modern sequence modeling lies in accurately capturing non-local dependencies within data. As Karpinska et al. [100] demonstrated, even state-of-the-art models equipped with artificially expanded context windows primarily excel at surface-level retrieval tasks (e.g., “needle-in-the-haystack” tests), while persistently failing to maintain deep, non-local relational reasoning across extended narratives. Existing attention mechanisms in large language models lack explicit modeling of underlying dynamical processes, leading to rapid semantic decay when processing long sequences. Hassan et al. [86] demonstrated in their study of nonlinear chaotic systems that fractional derivative mechanisms can rigorously characterize complex long-range evolutionary dynamics. This suggests that introducing operators capable of perceiving non-local integral characteristics may provide a promising pathway for improving the robustness of long-text semantic understanding.

When processing ultra-long contexts, mainstream sequence models based on Markovian evolution paths suffer from catastrophic forgetting, where information decay follows an exponential pattern. Fractional dynamics provide a potential theoretical breakthrough for this limitation. El-Nabulsi and Anukool [43] demonstrated that the state deviation decay of fractional neural networks no longer follows an exponential form but is instead governed by the Mittag–Leffler function, exhibiting heavy-tailed power-law characteristics. Embedding such non-Markovian memory mechanisms into sequence architectures may enable models to maintain gradual decay of historical information over ex-

tended temporal spans, thereby fundamentally alleviating the breakdown of long-range contextual dependencies.

The computational bottlenecks of large language models and the non-local computational challenges of fractional dynamics are mathematically highly homologous. Fractional derivatives require global convolutional integration over historical trajectories. As noted by Xiangnan et al. [89] in their framework for fractional equation discovery, explicitly modeling long-range interactions among historical states introduces extremely high memory and computational overhead. This mechanism faces the same fundamental limitation as Transformer attention computation: how to preserve nonlocal correlations while avoiding the computational explosion induced by global historical integration. At present, the scalability and effectiveness of fractional mechanisms in ultra-large generative architectures still require validation through realistic distributed engineering benchmarks. To provide a structured overview of the theoretical intersections and engineering challenges between large language models and fractional dynamics, Table 11 categorizes the core bottlenecks of current sequence architectures along with their corresponding fractional-order solutions.

Table 11. Comparative analysis of long-context challenges in LLMs and fractional dynamics solutions.

Analysis Dimension	Baseline LLM Architecture	Core Limitation and Symptom	Fractional-Order Approach	Expected Theoretical Advantage
Context Expansion	Standard Transformer self-attention mechanism.	High spatiotemporal complexity. Fixed explicit memory capacity. Context truncation causes “Ghost Context” hallucinations.	Non-local integral operators.	Perceives global historical characteristics to fundamentally mitigate long-term memory degradation.
Reasoning Depth	Artificially expanded context windows.	Restricted to surface-level retrieval. Failure in deep, non-local relational reasoning.	Explicit modeling of evolutionary dynamics.	Captures complex long-range semantic interactions across extended narratives.
Memory Decay Pattern	Conventional Markovian evolution paths.	Catastrophic forgetting. Rapid semantic decay following an exponential pattern.	State updates governed by the Mittag-Leffler function.	Introduces heavy-tailed power-law characteristics to maintain gradient indicators over extended spans.
Computational Complexity	Global historical token aggregation.	Computational explosion induced by dense history tracking. Extreme memory and distributed training overhead.	Structured non-local integration with truncated kernels.	Mathematically bridges the homology of attention bottlenecks while optimizing engineering scalability.

4.2. Optimization Instability and Training Memory Constraints

Distributed optimization in large language models constitutes an extremely high-dimensional nonlinear control problem constrained by GPU memory limitations. Complex nonconvex loss landscapes exhibit significant geometric roughness, causing conventional first-order optimizers to become trapped in local minimum and undergo severe oscillations [101–104]. Furthermore, the evolution of heavy-tailed weight distributions during model training exhibits long-range dependence, thereby violating the Markovian smoothness assumptions underlying traditional integer-order optimization methods. Vellappandi and Lee [12] argued that introducing fractional calculus is mathematically inevitable for addressing optimization instability induced by power-law distributions. By applying fractional operators to perform nonlocal integral smoothing over historical trajectories, abrupt optimization jumps can be directly suppressed, thereby improving optimizer stability.

To address the phenomenon of “loss spikes” during training, fractional dynamics and robust control theory provide potential solutions. Wang et al. [52] pointed out that complex neural networks frequently exhibit chaotic behaviors highly sensitive to initial conditions. Von Kluge et al. [85] further demonstrated that combining fractional dynamics

with deep learning can construct control architectures with strong robustness and disturbance immunity. These findings suggest that introducing adaptive fractional-order feedback mechanisms into optimizer design could dynamically regulate the memory weights of historical gradients, thereby buffering severe optimization fluctuations. Combined with adaptive control theories for nonlinear system [105,106], extreme gradient noise may be treated as a system disturbance, while fractional robust feedback mechanisms could potentially reconstruct stable optimizers at a fundamental level.

Nevertheless, extending fractional optimization to large-scale models with tens or hundreds of billions of parameters faces severe computational and memory constraints. Parameter updates require discretized summation over historical gradients, which not only increases per-step computational cost but also necessitates the storage of long-term historical gradient tensors, thereby further exacerbating the “memory wall” bottleneck in distributed training [107]. To design scalable large-scale fractional optimizers, exploring memory-compressed continuous optimization schemes becomes essential. Nguyen et al. [108] proposed a novel optimization strategy based on Hamiltonian descent and demonstrated that dynamical-system-based continuous optimization methods can effectively reduce dependence on storing historical gradient tensors. This progress suggests that methods based on Hamiltonian dynamics and their fractional extensions may provide a new breakthrough for overcoming memory limitations in large-scale model training. Table 12 outlines the core challenges of large-scale optimization alongside theoretical solutions derived from fractional dynamics and control theory.

Table 12. Optimization instability and memory constraints in large-scale training: Challenges and fractional solutions.

Challenge Area	Core Bottleneck	Theoretical Solution	Key Mechanism	Expected Advantage
Optimization Instability	First-order optimizers suffer severe oscillations and local minima in rough nonconvex loss landscapes.	Fractional Calculus Integration	Applies nonlocal integral smoothing across historical trajectories.	Directly suppresses abrupt optimization jumps and enhances overall stability.
Weight Distribution	Heavy-tailed weight evolution violates traditional integer-order Markovian smoothness assumptions.	Non-Markovian Operators	Captures long-range dependence inherent in parameter evolution.	Mathematically resolves training instability induced by power-law distributions.
Training “Loss Spikes”	Complex networks exhibit chaotic behaviors and high sensitivity to initial conditions.	Adaptive Fractional Feedback	Dynamically regulates the memory weights of historical gradients.	Buffers severe fluctuations and builds robust disturbance immunity against extreme gradient noise.
Distributed Memory Wall	Discretized summation of gradients requires massive storage for long-term historical tensors.	Memory-Compressed Optimization	Reduces strict dependence on explicitly storing historical gradient tensors.	Overcomes severe GPU memory limitations when scaling to models with hundreds of billions of parameters.
Scalable Optimizer Design	High per-step computational cost limits the practical application of fractional updates.	Hamiltonian Descent Dynamics	Utilizes dynamical-system-based continuous optimization methods.	Provides a fundamentally scalable pathway for constructing large-scale fractional optimizers.

4.3. Limitations of Current Attention and Positional Encoding Mechanisms

Contemporary large language models rely heavily on the joint operation of self-attention mechanisms and positional encoding schemes; however, their intrinsic limitations under ultra-long sequences are becoming increasingly evident. Standard self-attention mechanisms allocate feature weights through inner-product operations:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (12)$$

Under extremely long-context scenarios, this mechanism suffers from attention attenuation and entropy collapse [109]. Due to the normalization property of the softmax function, the weights of meaningful historical information are diluted by many redundant background tokens, significantly weakening the model’s ability to capture critical long-

range semantics. Such discrete scalar aggregation lacks global long-range topological constraints, making it difficult to preserve feature fidelity over ultra-long spans [110].

Regarding positional information representation, Rotary Positional Encoding (RoPE) implicitly models relative positions through rotations in complex vector spaces. For feature vectors at positions m and n , the geometric transformation of their inner product satisfies:

$$q_m^T k_n = x_m^T R_{\theta, n-m} x_n \quad (13)$$

where $R_{\theta, n-m}$ denotes a parameterized orthogonal rotation matrix. Although RoPE demonstrates excellent distance-awareness capability within predefined context windows, it exhibits inherent limitations when extrapolated to unseen sequence lengths. Because the rotational frequencies are constructed from integer-order discrete basis functions, once the relative distance exceeds the training threshold, the feature inner products undergo uncontrollable periodic oscillations and rapid decay, causing the model to lose long-range positional discrimination capability almost instantaneously. This rigid frequency decay mechanism fundamentally limits the smooth extension of sequence models toward infinite contexts.

To overcome the attention attenuation and extrapolation bottlenecks, reconstructing underlying architectures using more generalized mathematical constraints has become an important evolutionary direction. Bauduin et al. [111] argued that embedding rigorous mathematical constraints into deep architectures can improve the generalization robustness of complex systems. Traditional integer-order rotations and discrete attention aggregation lack sufficient flexibility to model complex nonlocal dependencies. Introducing fractional calculus as a generalized architectural constraint therefore presents substantial potential. By utilizing fractional continuous operators to redesign positional encoding mechanisms, it may become possible to overcome the rigid limitations of fixed discrete frequencies and achieve smooth continuous extrapolation over ultra-long distances. Simultaneously, incorporating fractional integral constraints into attention matrix computation may replace uncontrollable feature oscillations with controllable power-law decay behavior.

4.4. Empirically Validated Results Versus Theoretical Expectations

Although fractional mechanisms have demonstrated promising performance in sequence modeling, it is essential to rigorously distinguish experimentally verified capabilities from theoretical expectations. Existing empirical evidence primarily originates from medium-scale neural networks, time-series forecasting models, and scientific optimization algorithms rather than large-scale autoregressive language models. Table 13 delineates this boundary by pairing each capability dimension with its current validation status and supporting literature.

Table 13. Current boundaries and supporting literature between empirically validated capabilities and theoretical expectations.

Capability Dimension	Theoretical Basis	Experimental Evidence
Optimization Stability	Strong	Yes (Validated on convolutional networks and ResNet-18 tasks)
Long-Memory Time Series	Strong	Yes (Validated via fractional positional representations and recurrent layers)
Memory Compression	Moderate	Partial (Validated via wavelet-compressed gradient states with latency trade-offs)
Scientific Reasoning	Moderate	Partial (Validated on specific continuous or discrete differential equation benchmarks)
Native Long-Context Text Reasoning	Strong	No (Currently remains a theoretical hypothesis for natural language text)
Mitigating “Lost-in-the-Middle”	Strong	No (Hypothesized based on non-local heavy-tailed mathematical profiles)

To support the qualitative classifications presented in Table 13, specific quantitative evidence from recent literature must be examined. Regarding optimization stability, explicit empirical validation has been achieved on convolutional frameworks. Specifically, when evaluating a ResNet-18 architecture with 11 million parameters on the mini-ImageNet database under a moderate gradient accumulation setup of $N = 16$ steps, injecting a fractional power-law reweighting scheme with $\alpha = 0.1$ increased the mean test accuracy from a baseline of 16.99% to 31.35% [55]. This performance gain represents a 14.36% absolute improvement and was substantiated by paired t -tests yielding a highly significant p -value of approximately 2×10^{-6} , accompanied by a 20.2% reduction in gradient norm variance [55].

Regarding long-memory time-series modeling, structural validation has been demonstrated across specialized sequence frameworks. The integration of fractional positional representations coupled with lag-aware attention operators yielded consistent performance gains across multivariate benchmark distributions including Weather and PEMS08 suites [62]. Furthermore, recurrent configurations utilizing fractional dynamics reported stable hidden state profiles on multi-domain forecasting data with a total training memory footprint documented at 0.49 GB [51].

However, the memory compression and scientific reasoning dimensions exhibit non-negligible engineering trade-offs that justify their categorization as only partially validated. While wavelet-based fractional optimization algorithms achieve significant gradient memory compression during parameter updates, empirical profiling showed that this reduction incurs a localized execution latency penalty, extending epoch training duration by 6.5% to 12.1% due to discrete transformation computation [59]. Similarly, replacing standard Multi-Layer Perceptrons with fractional or rational Kolmogorov-Arnold Network layers incorporates non-local approximation capabilities for differential equation solving, but evaluating these trainable fractional Jacobi basis functions or Padé polynomials is two to three times slower than executing classical ReLU activations [60,61].

Most critically, all current empirical successes are strictly bounded within small-scale or medium-scale architectures, leaving advanced capabilities such as native long-context text reasoning and the mitigation of the “lost-in-the-middle” effect completely unverified on large-scale autoregressive language models.

4.5. Comparison with Existing Long-Context Architectures

Long-context modeling has become a critical research frontier, leading to the development of several representative approaches including sparse attention, linear attention, state-space models, retrieval-augmented generation, and memory-compressed Transformers. Fractional mechanisms provide a fundamentally different perspective by introducing non-local memory dynamics and power-law decay characteristics. Table 14 summarizes the structural differences and representative implementations among these paradigms.

Table 14. Structural comparison between fractional mechanisms and long-context approaches.

Methodology	Core Memory Mechanism	Time Complexity	Long-Range Dependency	De-External Database Requirement	Representative Frameworks or Supporting Literature (Complete Title and Year)
Sparse Attention	Sparse token interaction patterns	$O(N\sqrt{N})$	Moderate	No	Longformer; BigBird
Linear Attention	Kernel feature approximation	$O(N)$	Moderate	No	Performer; Linear Transformer
State-Space Models	Hidden state exponential evolution	$O(N)$	Strong	No	Mamba; Structured State Spaces (S4)
RAG	Vectorized retrieval memory	$O(N) + O(\text{Retrieval})$	Strong	Yes	Standard Vector-Search Retrieval-Augmented Generation

Fractional Mechanisms	Heavy-tailed power-law memory	Task-dependent	Strong	No	[55,60–62]
-----------------------	-------------------------------	----------------	--------	----	------------

The comparative advantages of fractional operators over established long-context methodologies rest on distinct mathematical attributes. Unlike modern State-Space Models such as Mamba, which fundamentally rely on an exponential decay formulation (β^t) for hidden state evolution, fractional frameworks implement a discrete approximation of the Riemann–Liouville integral that naturally preserves a heavy-tailed power-law decay profile proportional to $t^{\alpha-1}$ [55]. Quantitative mapping shows that under an accumulation window of $N = 16$ steps with $\alpha = 0.1$, the oldest historical token or gradient index still retains a non-zero structural weight of approximately 0.025, maintaining an informative ratio of 11.4% relative to the most recent step, whereas an exponential scheme with a standard decay factor of $\beta = 0.9$ drops geometrically to $0.9^{16} \approx 0.185$, rapidly approaching total information truncation [55].

Furthermore, unlike Retrieval-Augmented Generation architectures that resolve long-context dependencies by introducing external vectorized databases and computationally expensive similarity search loops, fractional integration operators function as an intrinsic mathematical mechanism embedded directly within the forward feature propagation graph. Empirical implementations indicate that fractional positional encoding allows the network to adaptively shift its memory kernel decay across distinct token sequences, recording optimal configurations at $\alpha = 0.8$ for highly structured electricity patterns and shifting to $\alpha = 0.3$ for complex traffic datasets [62]. Rather than seeking to replace sparse or linear attention systems, fractional mechanisms provide standalone non-local memory properties that can be structurally combined with existing Transformer layers to establish hybrid, long-tail context processing models.

5. Toward Fractional Large Language Models: Conceptual Frameworks and Potential Applications

5.1. Conceptual Architecture of Fractional Large Language Models

Based on the capability evolution and computational constraints of large language models, the conceptual framework of Fractional Large Language Models (FLLMs) can be divided into two primary pathways: the decoupled neuro-symbolic collaborative paradigm and the endogenous fractional evolution paradigm. As summarized in Table 15, these two architectures target different model scales and application objectives.

Table 15. Classification and comparison of conceptual architectures for fractional large language models.

Architecture Paradigm	Core Mechanism	Functional Role of LLMs	Functional Role of Fractional Operators	Development Goals and Application Scenarios
Decoupled neuro-symbolic collaborative architecture	External tool invocation and system decoupling	High-level cognitive orchestration (semantic parsing, constraint generation, parameter initialization)	Independent execution of nonlocal dynamical numerical solving and continuous optimization	Real-world engineering deployment (traffic scheduling, industrial optimization, anomalous diffusion prediction)
Endogenous fractional evolution architecture	Reconstruction of underlying information propagation mechanisms	Built-in fractional propagation mechanisms with native long-term memory capability	Embedded as the core evolution mechanism within neural feature propagation graphs	Fundamental theoretical exploration (ultra-long-context reasoning, cross-scale system modeling)

The decoupled architecture treats the large language model as a macroscopic cognitive agent whose underlying pretrained weights remain unchanged, while external fractional solvers handle continuous dynamical computations. Its essence is therefore a “LLM-driven scientific computing” system. In contrast, the endogenous architecture

seeks to introduce fractional dynamics directly into the fundamental information propagation mechanisms of neural networks, enabling the model to possess intrinsic nonlocal feature evolution capability. This paradigm represents a deeper reconstruction of cognitive architectures.

5.2. Decoupled Neuro-Symbolic Collaborative Architecture Based on Fractional Operators

Compared with directly modifying the underlying attention mechanisms of Transformers, constructing a decoupled architecture consisting of “large language models + external fractional solvers” is currently a more engineering-feasible pathway. Under this paradigm, the large language model does not directly intervene in the internal neural architecture; instead, it performs semantic bridging and cognitive planning through prompt engineering or tool calling. Physical evolution processes involving historical dependence and anomalous spatial diffusion are delegated to dedicated continuous dynamical modules. This mechanism effectively avoids the enormous computational cost of retraining large-scale models while fully exploiting the mathematical rigor of fractional systems in scientific reasoning. The collaborative division of labor in representative physical and operational scenarios is summarized in Table 16.

Table 16. Task allocation in the decoupled neuro-symbolic collaborative architecture for typical complex systems.

Task Scenario	LLMs (Cognitive Scheduling Agent)	External Fractional Solver (Continuous Dynamics Engine)	Core Mathematical Operators
Long-term personalized travel planning	Parse interactive demands and historical text, construct preference topology graphs, and initialize memory parameter	Perform long-term fatigue accumulation modeling and cross-regional nonlocal graph diffusion ranking for candidate points of interest.	Riemann–Liouville integrals, fractional graph Laplacian operators
Emergency material scheduling	Parse heterogeneous disaster reports and meteorological instructions, automatically generate disaster source terms and physical boundary constraints	Predict the spatiotemporal evolution of anomalous risk fields and execute trajectory planning minimizing cumulative exposure risk	Caputo time derivatives, fractional variational optimization

Taking long-term personalized travel planning as an example, conventional Markov Decision Processes (MDPs) are unable to characterize long-term fatigue accumulation. Within the decoupled architecture, after the large language model initializes constraints, the external solver takes over the dynamic optimization process and models the historical fatigue state $F(t)$ at time t using the Riemann–Liouville integral:

$$F(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} c(x(\tau)) d\tau \quad (14)$$

where $c(x(\tau))$ denotes the local behavioral cost, and $(t - \tau)^{\alpha-1}$ represents a heavy-tailed power-law memory kernel that prevents premature loss of historical information. $\alpha \in (0, 1]$ is the memory order and $c(x(\tau)) \geq 0$ is the local behavioral cost function at time τ . For POI ranking, a fractional graph diffusion equation is employed to overcome the limitations of local topology:

$${}_0^C D_t^\beta r(t) = -L^\gamma r(t) \quad (15)$$

where $r(t)$ denotes the dynamic score vector and L^γ represents the graph structure matrix containing fractional Laplacian operators. $\beta \in (0, 1]$ is the fractional order, $r(t)$ is the dynamic score vector, and $L^\gamma = (-\Delta)^{\gamma/2}$ is the fractional graph Laplacian operator with $\gamma > 0$.

In emergency scheduling tasks under extreme disaster scenarios, disaster propagation frequently exhibits heavy-tailed jump characteristics. The large language model is

responsible for generating the disaster source term, while the external solver performs spatiotemporal modeling of the risk field $R(x, t)$ based on anomalous diffusion dynamics:

$${}_0^C D_t^\alpha R(x, t) = K(-\Delta)^{\beta/2} R(x, t) + S(x, t) \quad (16)$$

where $\alpha_R \in (0, 1]$ is the temporal fractional order, $K > 0$ is the diffusion coefficient, $\beta_R \in (1, 2]$ is the spatial fractional order, and $S(x, t)$ is the source term. $S(x, t)$ is the spatiotemporal distribution function of the source term, typically generated by the large language model from parsing disaster reports. Subsequently, the system performs global spatiotemporal trajectory planning by solving a cost functional jointly optimizing energy consumption and cumulative risk:

$$J(x) = \int_0^T \left[\frac{1}{2} M \left({}_0^{RL} D_t^\gamma x(t) \right)^2 + \omega R(x(t), t) \right] dt \quad (17)$$

where $J(x)$ is the cost functional with respect to trajectory $x(t)$, $M > 0$ is the inertia coefficient, $\omega > 0$ is the risk weight coefficient, $\gamma \in (0, 1]$ is the fractional order, and ${}_0^{RL} D_t^\gamma$ denotes the Riemann-Liouville fractional derivative.

The essence of this framework is that the large language model functions solely as a semantic bridge between natural language and continuous mathematical systems, while never directly participating in numerical computation. Its engineering nature is fundamentally that of an LLM-driven external fractional scientific computing system.

The decoupled architecture treats fractional mechanisms as external modules that can be integrated into existing LLMs without modifying the original Transformer forward-pass structure. Such a modular design allows fractional optimization to be introduced incrementally while preserving compatibility with highly optimized distributed training frameworks (e.g., DeepSpeed or Megatron-LM).

To transition the decoupled framework from concept to practice, we provide specific implementation guidelines addressing overhead, parameter selection, and structural integration, grounded in empirical findings from recent literature:

(a) Expected Computational Overhead: Since the original Transformer blocks remain unmodified, the overhead is inherently minimized. Scalar-based fractional weighting (e.g., fractional gradient accumulation) introduces negligible latency ($< 1\%$ overhead) and requires minimal memory confined to N scalar values [55]. However, utilizing external fractional solvers or complex wavelet transforms may increase epoch latency by 6.5% to 12.1% due to auxiliary discrete transformations [59].

(b) Recommended Fractional Ranges and Safeguards: Optimal fractional orders in decoupled optimization vary based on the task, and manual grid search within $(0, 1)$ is currently the standard practice. Current literature suggests bounding the order $\alpha \in [0.1, 0.3]$ for gradient accumulation to balance recency bias without causing noise amplification [55]. For fractional Adam implementations, $\nu \approx 0.2$ is frequently recommended to outperform integer-order baselines [59]. Furthermore, practical deployments necessitate strict numerical safeguards; small perturbation constants (ϵ) or absolute value wrappers must be injected to prevent singularities when fractional orders interact with negative or near-zero synaptic weights.

(c) Architectural Integration Points: The most feasible integration point is within the optimization loop (post-backpropagation). Fractional operators act as a mathematical interceptor between the computed integer-order gradients and the parameter update step (e.g., executing fractional aggregation immediately after the standard gradient derivation phase and prior to the final weight update).

To illustrate the division of labor between the cognitive agent and the continuous dynamics engine, Algorithm 1 delineates the conceptual workflow of the decoupled ar-

chitecture. This workflow is intended to demonstrate the external tool-calling mechanism rather than serve as an executable script.

Algorithm 1: Conceptual Workflow of Decoupled Fractional-LLM Architecture

Input: Natural language task description (e.g., complex system scheduling)

Output: Final actionable decision or natural language solution

- 1: // Phase 1: LLM Cognitive Orchestration (Semantic Bridging)
 - 2: Parse the input task description using the Large Language Model to extract specific system parameters and boundary constraints.
 - 3: Determine the initial fractional order based on the parsed system dynamics and required memory length.
 - 4: // Phase 2: External Fractional Computation (Decoupled Execution)
 - 5: Delegate the parsed mathematical parameters and constraints to a dedicated external numerical solver.
 - 6: Execute fractional differentiation or integration (e.g., Riemann-Liouville evaluation) within the external solver to obtain precise numerical outcomes.
 - 7: // Phase 3: Result Interpretation
 - 8: Feed the computed numerical outcomes back into the Large Language Model's context window.
 - 9: Synthesize and output the final natural language response to interpret the numerical solution for the user.
-

5.3. Endogenous Fractional Evolution Architecture for Large Language Models

Unlike the decoupled collaborative paradigm, the endogenous fractional evolution architecture represents a more radical and theoretically exploratory direction. Instead of treating fractional operators as external tools, this paradigm reconstructs neural information propagation from its mathematical foundations, breaking the structural limitations imposed by local residual propagation and first-order Markovian recursion assumptions. Consequently, the intrinsic evolution of internal model states acquires long-term historical dependence and nonlocal dynamical propagation capability. The theoretical reconstruction of core LLM components under the endogenous architecture is summarized in Table 17.

Table 17. Comparison between conventional architectures and endogenous fractional evolution architectures.

Core Network Component	Conventional Integer-Order (Local Markovian) Mechanism	Endogenous Fractional (Nonlocal) Evolution Mechanism	Core Theoretical Advantages
State propagation	Layer-wise local updates relying on linear combinations of previous states	Cross-layer historical coupling through Grünwald–Letnikov historical convolution integrals	Mitigates long-range semantic decay in deep networks
Attention computation	Discrete inner-product aggregation with diluted long-range semantic weights	Power-law kernel modulation with fractional memory decay kernel W_α	Enhances long-term dependencies in cross-paragraph and long-chain reasoning
State-space dynamics	First-order differential evolution: $h'(t) = Ah(t) + Bx(t)$	Fractional differential evolution: ${}^C_0 D_s^\alpha h(s) = f(h(s), x(s))$	Improves continuous memory capacity under ultra-long contexts
Optimization dynamics	Local chain-rule recursion prone to oscillation and diffusion	Historical gradient memory modulated by global evolution trajectories	Enhance stability in deep nonconvex optimization

Conventional Transformers are essentially local interlayer recursive systems, where the hidden state update at layer l is represented as:

$$h_l = h_{l-1} + f(h_{l-1}, x) \quad (18)$$

where $h_l \in \mathbb{R}^{d_l}$ is the hidden state vector output by layer l , x is the input feature to the current layer (typically the output of the previous layer or the original embedding), and

$f(\cdot, \cdot)$ is the residual function (e.g., the output of multi-head attention). As network depth increases, long-range historical semantics gradually decay under local Markovian propagation. The endogenous architecture instead reinterprets state propagation as a continuous fractional dynamical system. If network depth is abstracted as a continuous variable s , the hidden-state evolution becomes:

$${}_0^C D_s^\alpha h(s) = f(h(s), x(s)) \quad (19)$$

where $s \in [0, L]$ is the continuous depth variable that discretizes the network layers, $h(s)$ is the continuous hidden state at depth s , and $x(s)$ is the continuous input at depth s . ${}_0^C D_s^\alpha$ denotes the Caputo fractional derivative and α is the memory order. This differential equation incorporates integral accumulation over the complete historical trajectory, transforming information propagation from “layer-wise local updating” into “cross-layer historical coupling.” In discrete deep neural networks, a Grünwald–Letnikov discretization can be employed to construct fractional residual propagation:

$$h_l = \sum_{k=1}^l w_k^\alpha h_{l-k} + f(h_{l-1}, x) \quad (20)$$

where w_k^α denotes generalized binomial coefficients corresponding to power-law non-local memory kernels. This structure enables the features of the current layer to be jointly influenced by all historical layer representations, thereby theoretically preserving semantic correlations over much longer temporal scales. As sequence length increases, long-range dependencies are easily weakened. The endogenous architecture introduces a fractional power-law memory kernel W^α to directly modulate the attention matrix:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} \odot W^\alpha\right) V \quad (21)$$

where $\odot$ denotes the Hadamard product (element-wise multiplication), $W^\alpha \in \mathbb{R}^{N \times N}$ is the power-law memory kernel matrix with entries $W_{ij}^\alpha = |i - j|^{-\alpha}$, and $\alpha > 0$ is the memory decay order. with entries $W_{ij}^\alpha = |i - j|^{-\alpha}$ for $i \neq j$, and $W_{ii}^\alpha = 1$. d is the dimension of the query and key vectors. When α is small, long-range historical information decays slowly in a heavy-tailed manner, significantly strengthening long-term dependencies in long-chain reasoning tasks.

Fractional dynamics can also be deeply integrated into State Space Models (SSMs). Conventional SSMs satisfy integer-order state evolution: $h'(t) = Ah(t) + Bx(t)$, which can be generalized into the fractional state-space form: ${}_0^C D_t^\alpha h(t) = Ah(t) + Bx(t)$. Compared with exponential decay, fractional systems preserve longer historical states through power-law memory mechanisms, thereby improving continuous memory capacity for long-context modeling. At the level of optimization dynamics, fractional mechanisms introduce historical gradient memory into parameter updates:

$$\theta_{t+1} = \theta_t - \eta \sum \gamma_k^\alpha \nabla L \quad (22)$$

where γ_k^α denotes the fractional gradient weighting kernel. Parameter updates are therefore continuously influenced by historical optimization trajectories, improving long-term training stability in deep networks.

Overall, the endogenous fractional evolution architecture attempts to fundamentally reconstruct information propagation and optimization mechanisms in deep neural networks from the perspective of underlying mathematics, enabling neural networks to evolve toward continuous dynamical systems with nonlocal propagation capability. However, this direction currently remains largely at the stage of theoretical exploration and small-scale experimentation. Stable training mechanisms, computational complexity control, and parallel acceleration strategies for ultra-large language models still lack systematic engineering validation.

Unlike the decoupled framework, the endogenous architecture embeds fractional dynamics directly into the LLM's internal computational graph. Key integration locations include positional representations and feed-forward networks (FFNs). These modifications enable non-Markovian memory effects and nonlocal dynamics to emerge intrinsically from the model's forward propagation.

Embedding fractional dynamics directly into the LLM's computational graph requires deeper structural modifications. The specific guidelines are as follows:

(a) Expected Computational Overhead: The endogenous framework introduces substantially higher computational complexity. For instance, rational or fractional basis functions in KAN variants require the computation of weighted polynomials, which empirical profiling indicates can be two to three times slower to evaluate than standard ReLU activations [60]. Memory consumption also increases due to the retention of nonlocal historical states.

(b) Recommended Fractional Ranges: For endogenous structural components like Fractional Positional Encoding (FPE), empirical studies report that the optimal order fluctuates significantly across data distributions (e.g., manually assigned as $\alpha = 0.8$ for electricity and $\alpha = 0.3$ for traffic data) [62]. To overcome the fragility of manual tuning, we strongly recommend adopting trainable fractional orders. By constraining the parameter $\gamma \in (0,1)$ via a Sigmoid wrapper, the network can adaptively learn the optimal fractional order via backpropagation [60].

(c) Architectural Integration Points: Key integration locations include the positional encoding initialization (modifying standard Rotary Positional Embeddings before the first attention layer) and the Feed-Forward Network (FFN) layers (replacing standard MLPs with trainable fractional activation functions or fractional KAN layers).

To clarify the integration methodology without loss of generality, Algorithm 2 illustrates a conceptual forward pass. This workflow is intended to illustrate the integration pathway and structural modifications rather than serve as a fully optimized deployment script.

Algorithm 2: Conceptual Forward Pass of Endogenous Fractional LLM

Input: Tokenized input sequence and a trainable fractional order parameter

Output: Probability distribution for the next token prediction

```

1:  // Initialization Phase
2:  Initialize the fractional positional encoding parameterized by the current trainable frac-
   tional order.
3:  Add the generated fractional positional encodings to the standard token embeddings to
   form the initial sequence representation.
4:  // Transformer Layer Iteration
5:  for each Transformer Layer do
6:      Apply standard layer normalization and multi-head self-attention to the representa-
   tions, retaining standard residual connections.
7:      Apply layer normalization to the attention output and process it through a feed-for-
   ward network equipped with an endogenous fractional-order activation function.
8:      Apply linear projection and residual connections to form the updated hidden state for
   the current layer.
9:  end for
10: // Output Generation
11: Apply a final linear transformation and SoftMax operation to the last layer's representations
   to generate the final token prediction probabilities.

```

5.4. Comparative Analysis of Fractional–LLM Architectures: Advantages, Limitations, and Applicable Scenarios

Although both architectures aim to enhance long-term historical modeling capability through fractional dynamics, they differ fundamentally in theoretical objectives, engineering implementation, and applicable scenarios. Table 18 compares the core characteristics of these two paradigms. Table 19 compares the core characteristics of traditional large language models and the two classes of fractional fusion architectures across multiple dimensions, including mathematical mechanisms, computational overhead, and physical consistency.

Table 18. Comparative analysis of fractional–LLM architectures.

Architecture Paradigm	Core Mechanism	Advantages	Limitations	Typical Applications
Decoupled neuro-symbolic collaborative architecture	External fractional solver + unchanged LLM	Low deployment cost; high interpretability; strong physical consistency	No intrinsic long-term memory; semantic mapping errors; solver dependence	Traffic scheduling, anomalous diffusion prediction, industrial optimization
Endogenous fractional evolution architecture	Internal fractional propagation and power-law memory kernels	Long-range historical coupling; stable long-context reasoning	Extremely high retraining and memory cost; limited stability theory	Ultra-long-context reasoning, persistent memory, multiscale system simulation

Table 19. Multi-dimensional comparison of core characteristics among traditional LLMs and fractional fusion architectures.

Evaluation Dimensions	Standard LLMs	Decoupled FLLMs	Endogenous FLLMs
Breaks Local Markovian Evolutionary Limits	X	✓	✓
Native Cross-layer Long-term Memory	X	X	✓
Non-local Anomalous Diffusion Solving Capability	X	✓ (Relies on external operators)	✓ (Native support)
Avoids Long-context Soft Truncation and Hallucinations	X	✓	✓
Controllability of Memory and Computational Cost	✓	✓	X (Superlinear surge)
Zero-modification Deployment of Pre-trained Weights	✓	✓	X
Physical Interpretability for Dynamic Complex Systems	X	✓ (Explicit mathematical constraints)	X (Implicit in black-box parameters)
Continuous Positional Extrapolation and Spatiotemporal Interpolation	X	X	✓

Note: The symbol ✓ indicates that architecture possesses the core advantage or supports the feature in that dimension, while X indicates a significant limitation or lack of support.

From the perspective of engineering feasibility, the decoupled architecture possesses substantial practical deployment advantages. Since it does not require retraining foundation models, fractional computational modules can be integrated into existing ecosystems through prompt engineering, tool invocation, or lightweight fine-tuning. This enables low-cost deployment in complex tasks requiring continuous dynamical modeling, such as traffic scheduling and anomalous diffusion prediction. Furthermore, because fractional dynamics are executed by independent mathematical systems, their physical constraints and optimization objectives remain explicit and interpretable, satisfying the requirements of scientific AI for interpretability and physical consistency. However, its limitation lies in the fact that it fundamentally represents an external enhancement paradigm: the native architecture of the large language model itself remains unchanged and therefore does not possess intrinsic long-term memory capability. Moreover, semantic mapping errors inevitably arise during the translation from natural language into continuous mathematical parameters, and the framework heavily depends on domain-specific solvers, resulting in weak cross-task transferability.

In contrast, endogenous architecture attempts to reconstruct information propagation mechanisms from the foundational level. Fractional dynamics establish cross-layer historical coupling through power-law memory kernels, making this paradigm theoretically more suitable for ultra-long-context reasoning, persistent memory, and long-chain scientific argumentation tasks. In scenarios such as macroeconomic simulation or complex transportation systems, where system states are jointly influenced by short-term variations and long-term historical evolution, fractional dynamics naturally possess the ability to model such multiscale coupling behaviors. Nevertheless, this architecture faces severe theoretical and engineering challenges. Directly modifying propagation mechanisms requires retraining ultra-large models at enormous computational cost. Fractional derivatives rely on global historical integration, causing substantial increases in both forward/backward propagation complexity and memory consumption. In addition, theoretical understanding of gradient propagation behavior and convergence boundaries in ultra-deep fractional Transformers remains incomplete.

In summary, the decoupled architecture represents the current pragmatic engineering pathway for “using large language models to drive fractional scientific computing,” whereas the endogenous architecture represents a future-oriented theoretical direction aimed at “reconstructing neural cognitive mechanisms through fractional dynamics.”

5.5. Challenges and Future Research Directions for Fractional Large Language Models

Although fractional calculus provides a new paradigm for nonlocal modeling in large language models, current research remains at an early stage and still faces numerous challenges in mathematical unification, computational constraints, and training stability. Table 20 summarizes the major challenges that must be addressed.

Table 20. Core challenges and technical bottlenecks of fractional large language models.

Challenge Domain	Core Problem	Key Research Direction
Theoretical foundations	Lack of unified theory between local recursion and nonlocal fractional dynamics	Generalization and long-memory theoretical analysis
Computational overhead	Global historical integration causes high memory and computation cost	Efficient compression and low-complexity approximation
Optimization stability	Historical coupling increases gradient instability	Convergence and error propagation theory
Engineering implementation	Lack of large-scale training standards and operator selection principles	Adaptive fractional operators and scalable architectures

5.5.1. Standardized Ablation Protocols

The current literature bridging fractional calculus and artificial intelligence is frequently characterized by highly integrated, heterogeneous architectures where multiple structural innovations are introduced simultaneously. This compound design makes it difficult to isolate the exact empirical contribution of fractional operators from other auxiliary modules. For instance, the performance of these hybrid systems exhibits severe sensitivity to the choice of fractional orders, which currently vary widely and heuristically across different tasks—ranging from $\alpha \in [0.1, 0.3]$ for gradient accumulation [55] and $\alpha \in [0.3, 0.8]$ for positional embedding scaling [62], to an integration order of 1.8 for recurrent neural networks [51].

To decouple these confounded effects, future research must adopt standardized ablation protocols that systematically evaluate three distinct dimensions:

(a) Fractional Order Sensitivity: Evaluations should explicitly contrast fixed heuristic orders against baseline integer-order models ($\alpha = 1.0$ or 2.0) and dynamically learned adaptive orders to map the performance landscape across different layers.

(b) Truncation Memory Length (M): Because computing continuous historical kernels is computationally expensive, systematic ablation of the sliding memory window M is strictly required to quantify the exact trade-off between long-range dependency retention and hardware acceleration boundaries.

(c) Mathematical Formulation Discretization: Comparative baselines should implement identical structural networks under different discrete formalisms, such as the Grünwald–Letnikov expansion, Caputo fractional difference, and Riemann–Liouville approximations, to verify which corporate memory kernel suits autoregressive information propagation graphs.

5.5.2. Statistical Validation and Evidence Aggregation

A systemic methodological limitation observed across more than 80% of the surveyed experimental papers is the reliance on single-run training performance without reporting confidence intervals, standard deviations, or statistical significance metrics [51,59]. Such reporting practices make it difficult to definitively distinguish genuine non-local algorithmic advantages from favorable random weight initializations or dataset-specific optimization fluctuations.

To improve empirical reliability, future studies should enforce rigid statistical validation frameworks. Specifically, performance metrics need average over at least five independent random seeds, with results documented strictly in the format of “mean $\pm$ standard deviation.” A rare and exemplary exception in the current literature is found in the evaluation of fractional gradient accumulation, which validated its accuracy gains across five random seeds and substantiated the results via paired t -tests, yielding an explicit significance level of $p \approx 2 \times 10^{-6}$ [55]. Furthermore, given the highly sparse and heterogeneous nature of current empirical evidence spanning computer vision, time-series, and symbolic discovery, the community should implement formal meta-analysis techniques. Meta-analytical evidence synthesis provides a mathematically sound framework to aggregate sparse accuracy indicators across disjointed areas of the literature, facilitating more objective conclusions regarding the true scalability of fractional operators.

5.5.3. Task Adaptation and Selective Fractionalization

Based on these challenges, future research must clearly define the application boundaries of fractional mechanisms. Task-adaptation theory should explicitly identify that fractional dynamics are most suitable for systems characterized by long-term memory, anomalous diffusion, and multiscale coupling, thereby ensuring that increased computational cost can be translated into meaningful performance gains.

The most promising future architectural evolution direction is likely to be “selective fractionalization” rather than global reconstruction. In such hybrid architectures, fractional propagation mechanisms would be introduced only into specific modules, such as long-term memory layers or task-planning components, while efficient Transformer operators would still be retained for local language generation. This strategy is considerably more feasible from an engineering perspective.

5.5.4. Convergence with Scientific AI and Embodied Intelligence

Furthermore, the integration of fractional mechanisms with large language models is expected to deeply converge with scientific machine learning and embodied intelligence. In scientific AI, fractional large language models may serve as highly consistent reasoning frameworks for modeling anomalous physical evolution processes. In embodied intelligence, they may be utilized to model long-term environmental interactions and nonlocal behavioral accumulation, thereby enhancing the adaptability of intelligent agents in complex dynamic environments. Overall, the integration of fractional calculus with large lan-

guage models has the potential to drive AI systems beyond local short-range fitting paradigms toward architectures capable of modeling long-term historical dependence and cross-scale dynamical propagation.

5.5.5. Adaptive Fractional-Order Learning and Stability Risks

The selection of fractional-order parameters represents one of the most critical challenges in fractional artificial intelligence. Existing studies commonly employ fixed fractional orders ranging from 0.1 to 0.95 based on manual grid searches. However, optimal values are highly task dependent. For instance, the optimal order for fractional gradient accumulation was empirically bounded between 0.1 and 0.3 [55].

Inappropriate parameter choices pose severe risks to optimization stability and numerical consistency. Small fractional orders introduce intense recency bias and heavy skewness. As explicitly documented in recent optimization profiling, when extending the accumulation window to an extreme scale of $N = 64$ steps under a fractional order of $\alpha = 0.3$, the system experiences a catastrophic training collapse, where the final test accuracy drops severely to exactly 1.00%, which represents a complete optimization failure equivalent to random guessing for 100-class classification [55]. This structural collapse is characterized by a severely elevated training loss of 7.851 and an extreme gradient norm standard deviation of 675.802, demonstrating total numerical divergence [55].

The primary mathematical cause of this pathology is numerical precision degradation within standard deep learning hardware environments. Under single-precision floating-point arithmetic (FP32), which possesses approximately seven decimal digits of precision, assigning aggressive fractional orders across large sequences forces the smallest weights down to magnitudes between 10^{-4} and 10^{-5} [55]. When these tiny floating-point values are multiplied by gradient vectors and subsequently aggregated with large-weighted contemporary elements, catastrophic cancelation occurs, causing the mantissas of the historical parameters to be truncated and effectively rounded to zero [55].

To bypass the fragility of manual parameters and prevent numerical truncation, future architectures must transition toward adaptive fractional networks. Recent validation has demonstrated this potential by constructing trainable fractional Jacobi basis functions, allowing the deep model to dynamically optimize its non-local representation space via standard backpropagation without relying on static, hand-tuned constants [60]. Balancing this dynamic parametric flexibility with strict execution robustness remains an indispensable open engineering problem for deploying fractional mechanisms in ultra-large-scale autoregressive language models.

6. Conclusions

This survey has outlined the core mechanisms and potential applications of integrating fractional calculus into LLMs. The review indicates that incorporating fractional operators provides a mathematically viable pathway to help alleviate long-context memory decay and distributed optimization oscillations. Rather than a complete architectural overhaul, the immediate evolution of this interdisciplinary field could focus on hybrid strategies, such as decoupled frameworks and selective module fractionalization. While substantial engineering challenges regarding computational complexity and memory constraints remain to be resolved, exploring non-local power-law dynamics offers a promising alternative to traditional local Markovian frameworks, potentially providing new perspectives for cross-scale scientific reasoning.

Author Contributions: Conceptualization, W.T. and C.H.; methodology, W.T., C.H. and X.L. (Xixing Liu); validation, W.T., C.H. and H.W.; formal analysis, W.T., X.L. (Xixing Liu) and H.W.; investigation, W.T., C.H. and H.W.; resources, W.T. and C.H.; data curation, W.T., X.L. (Xixing Liu) and

H.W.; writing—original draft preparation, W.T. and C.H.; writing—review and editing, W.T., C.H., X.L. (Xixing Liu), H.W. and X.L. (Xuhang Li); visualization, W.T., H.W. and X.L. (Xuhang Li); supervision, W.T. and C.H.; project administration, W.T. and C.H.; funding acquisition, W.T. and C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Youth Talent Basic Research Special Project of the Scientific Research Fund of Yunnan Provincial Department of Education (Grant No. KKJJ202637032), the Natural Science Research Fund of Kunming University of Science and Technology (Grant No. KKZ3202537094), the Ministry of Education of Humanities and Social Science Research (Youth) Project (Grant No. 24YJC630065), and Yunnan Fundamental Research Projects (Grant No. 202501AU070110).

Data Availability Statement: No new data was created in this study. Data sharing is not applicable to this article as it is a review paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
AI4Science	Artificial Intelligence for Science
deepFDEnet	Deep Fractional Differential Equation Network
FDE	Fractional Differential Equation
FLLM	Fractional Large Language Model
FPDE	Fractional Partial Differential Equation
FPE	Fractional Positional Embedding
FPINN	Fractional Physics-Informed Neural Network
FracGrad	Fractional Gradient
FrFNets	Fractional Fourier Networks
FrFT	Fractional Fourier Transform
FT	Fourier Transform
fKAN	Fractional Kolmogorov–Arnold Network
fTRU	Fractional Tensor Recurrent Unit
GL	Grünwald–Letnikov
GPU	Graphics Processing Unit
KAN	Kolmogorov–Arnold Network
KAT	Kolmogorov–Arnold Transformer
LLM	Large Language Model
LMI	Linear Matrix Inequalities
MDP	Markov Decision Process
ML	Machine Learning
NARX	Nonlinear Autoregressive Exogenous
NN	Neural Network
ODE	Ordinary Differential Equation
PE	Positional Encoding
PiFNN	Physics-Informed Fractional Neural Network
POI	Point of Interest
PSO	Persistence-Sensitive Optimization
RL	Riemann–Liouville
RoPE	Rotary Positional Encoding
rKAN	Rational Kolmogorov–Arnold Network
SGD	Stochastic Gradient Descent
SSM	State Space Model
TF-IDF	Term Frequency-Inverse Document Frequency
TFKAN	Transformer based on Kolmogorov–Arnold Networks
TKAN	Temporal Kolmogorov–Arnold Network

References

1. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Dong, Z.; Hou, Y.; Zhang, B.; Min, Y.; Zhang, J.; Liu, P. A survey of large language models. *Front. Comput. Sci.* **2026**, *20*, 2012627.
2. Lee, J.-S.; Hsiang, J. Patent claim generation by fine-tuning OpenAI GPT-2. *World Pat. Inf.* **2020**, *62*, 101983.
3. Sanderson, K. GPT-4 is here: What scientists think. *Nature* **2023**, *615*, 773.
4. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **2023**, *24*, 1–113.
5. Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K. Gemini: A family of highly capable multimodal models. *arXiv* **2023**, arXiv:2312.11805.
6. Team, G.; Georgiev, P.; Lei, V.I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv* **2024**, arXiv:2403.05530.
7. Kasneci, E.; Seßler, K.; Küchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; Gasser, U.; Groh, G.; Günemann, S.; Hüllermeier, E. ChatGPT for good? On opportunities and challenges of large language models for education. *Learn. Individ. Differ.* **2023**, *103*, 102274.
8. Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 16344–16359.
9. Liu, N.F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P. Lost in the middle: How language models use long contexts. *Trans. Assoc. Comput. Linguist.* **2024**, *12*, 157–173.
10. Alman, J.; Yu, H. (Eds.) Fundamental limitations on subquadratic alternatives to Transformers. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025.
11. Kang, Q.; Zhao, K.; Ding, Q.; Ji, F.; Li, X.; Liang, W.; Song, Y.; Tay, W.P. Unleashing the potential of fractional calculus in graph neural networks with FROND. *arXiv* **2024**, arXiv:2404.17099.
12. Vellappandi, M.; Lee, S. Physics-informed neural fractional differential equations. *Appl. Math. Model.* **2025**, *145*, 116127.
13. Harjule, P.; Sharma, R.; Kumar, R. Fractional-order gradient approach for optimizing neural networks: A theoretical and empirical analysis. *Chaos Solitons Fractals* **2025**, *192*, 116009.
14. Elnady, S.M.; El-Beltagy, M.; Radwan, A.G.; Fouda, M.E. A comprehensive survey of fractional gradient descent methods and their convergence analysis. *Chaos Solitons Fractals* **2025**, *194*, 116154.
15. Joshi, M.; Bhosale, S.; Vyawahare, V.A. A survey of fractional calculus applications in artificial neural networks. *Artif. Intell. Rev.* **2023**, *56*, 13897–13950.
16. Fernandez, E.; Huilcapi, V.; Birs, I.; Cajo, R. The role of fractional calculus in modern optimization: A survey of algorithms, applications, and open challenges. *Mathematics* **2025**, *13*, 3172.
17. Karaca, Y.; Baleanu, D. Computational fractional-order calculus and classical calculus AI for comparative differentiability prediction analyses of complex-systems-grounded paradigm. In *Multi-Chaos, Fractal and Multi-Fractional Artificial Intelligence of Different Complex Systems*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 149–168.
18. Li, C.; Qian, D.; Chen, Y. On Riemann-Liouville and Caputo derivatives. *Discret. Dyn. Nat. Soc.* **2011**, *2011*, 562494.
19. Machado, J.T. The bouncing ball and the Grünwald-Letnikov definition of fractional derivative. *Fract. Calc. Appl. Anal.* **2021**, *24*, 1003–1014.
20. Wang, Z.; Huang, X.; Zhou, J. A numerical method for delayed fractional-order differential equations: Based on GL definition. *Appl. Math. Inf. Sci.* **2013**, *7*, 525–529.
21. Sobotka, J.; Šimánek, P.; Kordík, P. Enhancing Fractional Gradient Descent with Learned Optimizers. *arXiv* **2025**, arXiv:2510.18783.
22. Zhao, H.; Yang, H.; Chen, J.; Jiang, P.; Zeng, Z. Parameter training methods for convolutional neural networks with adaptive adjustment method based on Caputo fractional-order differences. *Chaos Solitons Fractals* **2025**, *199*, 116588.
23. Edelman, M. Fractional maps and fractional attractors. Part II: Fractional difference Caputo α -families of maps. *Discontinuity Nonlinearity Complex.* **2015**, *4*, 391–402.
24. Wu, G.-C.; Baleanu, D. Discrete fractional logistic map and its chaos. *Nonlinear Dyn.* **2014**, *75*, 283–287.
25. Danca, M.-F. Fractional order logistic map: Numerical approach. *Chaos Solitons Fractals* **2022**, *157*, 111851.
26. Wu, G.-C.; Baleanu, D. Chaos synchronization of the discrete fractional logistic map. *Signal Process.* **2014**, *102*, 96–99.
27. Ji, Y.; Lai, L.; Zhong, S.; Zhang, L. Bifurcation and chaos of a new discrete fractional-order logistic map. *Commun. Nonlinear Sci. Numer. Simul.* **2018**, *57*, 352–358.

28. Wang, Y.; Liu, S.; Li, H. On fractional difference logistic maps: Dynamic analysis and synchronous control. *Nonlinear Dyn.* **2020**, *102*, 579–588.
29. Danca, M.-F.; Fečkan, M.; Kuznetsov, N.; Chen, G. Coupled discrete fractional-order logistic maps. *Mathematics* **2021**, *9*, 2204.
30. Smidtaite, R.; Kazlauskas, A.; Edelman, M.; Ragulskis, M. On the convergence rate of the caputo fractional difference logistic map of nilpotent matrices. *Fractal Fract.* **2026**, *10*, 40.
31. Zhang, C.; Wang, H. 9D fractional-order memristive neural network with hidden attractors for secure remote sensing image encryption. *Phys. Scr.* **2026**, *101*, 215206.
32. Ran, J.; Zhou, Y.; Abdeljawad, T.; Pu, H. Discrete fractional neural networks within the framework of octonions: A preliminary exploration. *J. Comput. Sci.* **2025**, *87*, 102586.
33. He, S.; Ma, M.; Wang, M.; Li, Z. Dynamics and information modulation in the Chialvo neuron networks with fractional-order discrete memristor coupling. In Proceedings of the 2023 IEEE International Conference on Memristive Computing and Applications (ICMCA), Jinan, China, 8–10 December 2023.
34. Shrama, T.R.; Gade, P.M. Fractional-order neuronal maps: Dynamics, control and stability analysis. *Pramana* **2024**, *98*, 53.
35. Dhakshinamoorthy, V.; Wu, G.-C.; Banerjee, S. *Chaotic Dynamics of Fractional Discrete Time Systems*; CRC Press: Boca Raton, FL, USA, 2024.
36. Yu, F.; Wang, X.; Guo, R.; Ying, Z.; He, Y.; Zou, Q. Discrete neuron models and memristive neural network mapping: A comprehensive review. *Chin. Phys. B* **2025**, *34*, 120501.
37. Brandibur, O.; Kaslik, E.; Mozyrska, D.; Wyrwas, M. A Rulkov neuronal model with Caputo fractional variable-order differences of convolution type. In *Dynamical Systems Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 227–235.
38. Brandibur, O.; Kaslik, E.; Mozyrska, D.; Wyrwas, M. Stability of systems of fractional-order difference equations and applications to a Rulkov-type neuronal model. In *Proceedings of the New Trends in Nonlinear Dynamics: Proceedings of the First International Nonlinear Dynamics Conference (NODYCON 2019)*; Springer: Cham, Switzerland, 2020; Volume III.
39. Telksniene, I.; Telksnys, T.; Smidtaite, R.; Uzdila, E.; Ragulskis, M. Transient suppression of chaos in a Caputo fractional Rulkov neuron model induced by a single control impulse. *Chaos Interdiscip. J. Nonlinear Sci.* **2026**, *36*, 053134.
40. Lu, Y.-M.; Wang, C.-H.; Deng, Q.-L.; Xu, C. The dynamics of a memristor-based Rulkov neuron with fractional-order difference. *Chin. Phys. B* **2022**, *31*, 060502.
41. Vivekanandhan, G.; Abdolmohammadi, H.R.; Natiq, H.; Rajagopal, K.; Jafari, S.; Namazi, H. Dynamic analysis of the discrete fractional-order Rulkov neuron map. *Math. Biosci. Eng.* **2023**, *20*, 4760–4781.
42. Ma, M.; Lu, Y.; Li, Z.; Sun, Y.; Wang, C. Multistability and phase synchronization of Rulkov neurons coupled with a locally active discrete memristor. *Fractal Fract.* **2023**, *7*, 82.
43. El-Nabulsi, R.A.; Anukool, W. Fractional action Hamiltonian neural networks: Long-term memory, sequential learning, and AI applications. *Neurocomputing* **2026**, *681*, 133389.
44. Vellappandi, M.; Lee, S. Reinforcement learning in continuous-time memory-dependent systems via neural fractional differential equations. *Neurocomputing* **2026**, *679*, 133216.
45. Shao, X.; Xu, L.; Zhang, S.; Sun, G.; Wu, C. Fractional-Order Dynamics Learning and Control via Data-Driven Approaches: Taking Soft Manipulator as an Example. *IEEE Trans. Neural Netw. Learn. Syst.* **2026**, 1–16.
46. Khan, A. Fractional Calculus in Modern Machine Learning Optimizers: A Systematic Review of Algorithms, Convergence, and Open Frontiers. Convergence, and Open Frontiers (29 April 2026), 2026. Available online: <https://ssrn.com/abstract=6680779> (accessed on 15 June 2026).
47. Qiu, H.; Li, C.; Weng, Y.; Sun, Z.; Zhao, Q. Fractional Tensor Recurrent Unit (fTRU): A Stable Forecasting Model With Long Memory. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *36*, 3598–3607.
48. Olumide, O.O. Fractional-order Gradient Descent for Enhancing Deep Learning Optimization. *Asian Res. J. Math.* **2025**, *21*, 40–50.
49. Yang, Y.; Mo, L.; Hu, Y.; Long, F. The improved stochastic fractional order gradient descent algorithm. *Fractal Fract.* **2023**, *7*, 631.
50. Almatroud, O.A.; Hioual, A.; Ouannas, A.; Sawalha, M.M.; Alshammari, S.; Alshammari, M. On variable-order fractional discrete neural networks: Existence, uniqueness and stability. *Fractal Fract.* **2023**, *7*, 118.
51. Zhao, C.; Ye, J.; Zhu, Z.; Huang, Y. FLRNN-FGA: Fractional-order Lipschitz recurrent neural network with frequency-domain gated attention mechanism for time series forecasting. *Fractal Fract.* **2024**, *8*, 433.
52. Wang, E.; Hu, B.; Guan, Z.-H. Chaotic dynamics and synchronization of multi-region neural network based on locally active memristor. *Chaos Solitons Fractals* **2025**, *197*, 116437.

53. Zhang, Y.; Xu, H.; Li, Y.; Lin, G.; Zhang, L.; Tao, C.; Wu, Y. An integer-fractional gradient algorithm for back propagation neural networks. *Algorithms* **2024**, *17*, 220.
54. Xie, J.; Li, S. Training Neural Networks by Time-Fractional Gradient Descent. *Axioms* **2022**, *11*, 507.
55. Lee, M. FracGrad: A Discretized Riemann–Liouville Fractional Integral Approach to Gradient Accumulation for Deep Learning. *Fractal Fract.* **2025**, *9*, 733.
56. Coelho, C.; Costa, M.F.P.; Niggemann, O.; Ferrás, L.L. Methodologies for Improved Optimisation of the Derivative Order and Neural Network Parameters in Neural FDE Models. *Fractal Fract.* **2025**, *9*, 471.
57. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
58. Zhou, X.; You, Z.; Sun, W.; Zhao, D.; Yan, S. Fractional-order stochastic gradient descent method with momentum and energy for deep neural networks. *Neural Netw.* **2025**, *181*, 106810.
59. Herrera-Alcántara, O.; Arellano-Balderas, S.; Rodríguez-Mondragón, S.; Reyes-Ortíz, J.A.; Navarro-Fuentes, J. FAdamWav: A Fractional Wavelet Gradient Optimizer for Neural Networks. *Fractal Fract.* **2026**, *10*, 149.
60. Aghaei, A.A. fkan: Fractional kolmogorov–arnold networks with trainable jacobi basis functions. *Neurocomputing* **2025**, *623*, 129414.
61. Aghaei, A.A. rkan: Rational kolmogorov–arnold networks. *arXiv* **2024**, arXiv:2406.14495.
62. Zhou, C.; Li, Y.; Ye, J.; Zhou, Y.; Zhou, X.; Huang, Y.; Tao, D.; Zhao, C. Fractional Position With Predictive Attention for Multivariate Time Series Forecasting. *IEEE Internet Things J.* **2025**, *12*, 26430–26443.
63. Karim, A.; Evi, T.; Lee, S. Hybrid Ensemble of Large Language Models and Fractional Derivative Features for Domain-Specific Engineering Sentiment Analysis. *Appl. Sci.* **2026**, *16*, 4266.
64. Yan, S.; Lu, R.; Zhang, J. Synchronization control of memristive chaotic maps based on Rulkov neuron models and their application in traffic image protection. *Chaos Solitons Fractals* **2025**, *201*, 117332.
65. Qin, Z.; Han, X.; Sun, W.; He, B.; Li, D.; Li, D.; Dai, Y.; Kong, L.; Zhong, Y. Toeplitz neural network for sequence modeling. *arXiv* **2023**, arXiv:2305.04749.
66. Xu, J.; Chen, Z.; Li, J.; Yang, S.; Wang, W.; Hu, X.; Ngai, E. Enhancing graph collaborative filtering with FourierKAN feature transformation. In Proceedings of the 34th ACM International Conference on Information and Knowledge Management, Seoul, Republic of Korea, 10–14 November 2025.
67. Zhu, B.; Connor, C. SugarNet: Personalized blood glucose forecast with a Fourier Kolmogorov–Arnold network. *Expert Syst. Appl.* **2025**, *280*, 127476.
68. Bozorgasl, Z.; Chen, H. Wav-KAN: Wavelet Kolmogorov–Arnold Networks. *arXiv* **2024**, arXiv:2405.12832.
69. Seydi, S.T.; Bozorgasl, Z.; Chen, H. Unveiling the power of wavelets: A wavelet-based kolmogorov–arnold network for hyperspectral image classification. *arXiv* **2024**, arXiv:2406.07869.
70. Yang, Q.; Wang, Y.; Liu, L.; Zhang, X. Adaptive fractional-order multi-scale optimization TV-L1 optical flow algorithm. *Fractal Fract.* **2024**, *8*, 179.
71. Zhang, Z.; Zhang, Y.; Dolan, W.B. (Eds.) Towards More Efficient Insertion Transformer with Fractional Positional Encoding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*; Association for Computational Linguistics: Dubrovnik, Croatia, 2023.
72. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljagic, M.; Hou, T.; Tegmark, M. KAN: Kolmogorov–arnold networks. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025; pp. 70367–70413.
73. Dutta, A.; Maheswari, B.; Punitha, N.; Raj, A.S.A.; Banu, S.S.; Balamurugan, M. The first two months of Kolmogorov–Arnold Networks (KANs): A survey of the state-of-the-art. *Arch. Comput. Methods Eng.* **2026**, *33*, 1017–1028.
74. Genet, R.; Inzirillo, H. A Temporal Kolmogorov–Arnold Transformer for Time Series Forecasting. *arXiv* **2024**, arXiv:2406.02486.
75. Afzal Aghaei, A. KANtrol: A physics-informed Kolmogorov–Arnold network framework for solving multi-dimensional and fractional optimal control problems. *Eng. Comput.* **2025**, *41*, 3257–3275.
76. Shafiya, M.; Nagamani, G.; Dafik, D. Extended dissipative performance of fractional-order neural networks via LMI approach. *J. Anal.* **2024**, *32*, 2329–2347.
77. Yang, X.; Wang, X. (Eds.) Kolmogorov–arnold transformer. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025.
78. Fares, I.A.; Abd Elaziz, M.; Aseeri, A.O.; Zied, H.S.; Abdellatif, A.G. TFKAN: Transformer based on Kolmogorov–Arnold networks for intrusion detection in IoT environment. *Egypt. Inform. J.* **2025**, *30*, 100666.

79. Firoozsalari, A.N.; Mazraeh, H.D.; Aghaei, A.A.; Parand, K. deepfdenet: A novel neural network architecture for solving fractional differential equations. *arXiv* **2023**, arXiv:2309.07684.
80. Taheri, T.; Aghaei, A.A.; Parand, K. Accelerating fractional pinns using operational matrices of derivative. *arXiv* **2024**, arXiv:2401.14081.
81. Coelho, C.; Costa, M.F.P.; Ferrás, L.L. Neural fractional differential equations. *Appl. Math. Model.* **2025**, *144*, 116060.
82. Koenig, B.C.; Kim, S.; Deng, S. KAN-ODEs: Kolmogorov–Arnold network ordinary differential equations for learning dynamical systems and hidden physics. *Comput. Methods Appl. Mech. Eng.* **2024**, *432*, 117397.
83. Li, X.; Li, Q.; Zhao, W. Ai-Powered Physics-Constrained Transformer Framework: A New Approach to Solving Fractional Partial Differential Equations. *Fractals* **2026**, *34*, 2550110-642.
84. Kang, X.; Li, Y.; Li, Y.; Hu, J.; Zheng, K. Solving the Fractional Allen–Cahn Equation and the Fractional Cahn–Hilliard Equation with the Fractional Physics-Informed Neural Networks. *Fractal Fract.* **2025**, *9*, 773.
85. Von Kluge, P.N.; Sengha, G.; Doumia, B. AI-augmented fractional-order modeling and control of chaotic systems: A physics-informed neural approach. *J. Comput. Appl. Math.* **2026**, *483*, 117335.
86. Hassan, S.A.; Raja, M.J.A.A.; Chang, C.-Y.; Shu, C.-M.; Shoaib, M.; Kiani, A.K.; Raja, M.A.Z. Nonlinear chaotic Lorenz–Lü–Chen fractional order dynamics: A novel machine learning expedition with deep autoregressive exogenous neural networks. *Chaos Solitons Fractals* **2024**, *189*, 115620.
87. Partohaghighi, M.; Marcia, R.F.; West, B.J.; Chen, Y. Roughness-informed machine learning—a call for fractal and fractional calculi. *J. Inf. Intell.* **2025**, *3*, 463–480.
88. Nigmatov, A.; Krishnapriyan, A.S.; Sanderson, N.; Morozov, D. Topological regularization via persistence-sensitive optimization. *arXiv* **2020**, arXiv:2011.05290.
89. Xiangnan, Y.; Hao, X.; Zhiping, M.; Hongguang, S.; Yong, Z.; Zhibo, C.; Dongxiao, Z.; Yuntian, C. A data-driven framework for discovering fractional differential equations in complex systems. *Nonlinear Dyn.* **2025**, *113*, 24557–24577.
90. Dana Mazraeh, H.; Nosrati Firoozsalari, A.; Afzal Aghaei, A.; Parand, K. A Novel Neural Network Architecture for Solving Fractional Differential Equations. *Comput. Methods Differ. Equ.* **2025**, in Press. <https://doi.org/10.22034/cmde.2025.63059.2803>.
91. Nehma, G.; Tiwari, M. Leveraging kans for enhanced deep koopman operator discovery. *arXiv* **2024**, arXiv:2406.02875.
92. Du, M.; Chen, Y.; Wang, Z.; Nie, L.; Zhang, D. Llm4ed: Large language models for automatic equation discovery. *arXiv* **2024**, arXiv:2405.07761.
93. Ribeiro, L.C.; Bernardes, A.T.; Mello, H. On the fractal patterns of language structures. *PLoS ONE* **2023**, *18*, e0285630.
94. Lee, M. Fractal analysis of GPT-2 token embedding spaces: Stability and evolution of correlation dimension. *Fractal Fract.* **2024**, *8*, 603.
95. Lee, M. Fractal self-similarity in semantic convergence: Gradient of embedding similarity across transformer layers. *Fractal Fract.* **2024**, *8*, 552.
96. Xu, W.; Liu, H.; Liang, Y.; Zhao, S. The Caputo Nonlocal Structural Derivative Ultraslow Diffusion Model of Language Change and the Microscopic Mechanism. *Fractal Fract.* **2024**, *8*, 66.
97. Zhao, K.; Li, X.; Kang, Q.; Ji, F.; Ding, Q.; Zhao, Y.; Liang, W.; Tay, W.P. Distributed-order fractional graph operating network. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 103442–103475.
98. Kang, Q.; Cui, W.; Li, X.; Ma, Y.; Fu, X.; Tay, W.P.; Li, Y.; Zha, Z.-J. Neural Fractional Attention Differential Equations. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 33503–33536.
99. Liu, X.; Tang, Z.; Dong, P.; Li, Z.; Li, B.; Hu, X.; Chu, X. Chunkkv: Semantic-preserving kv cache compression for efficient long-context llm inference. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 28728–28778.
100. Karpinska, M.; Thai, K.; Lo, K.; Goyal, T.; Iyyer, M. One thousand and one pairs: A “novel” challenge for long-context language models. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–14 November 2024.
101. Li, H.; Xu, Z.; Taylor, G.; Studer, C.; Goldstein, T. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, Montreal, Canada, 3–8 December 2018; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
102. Wu, K.; Jian, X.; Du, R.; Chen, J.; Zhou, X. Roughness index for loss landscapes of neural network models of partial differential equations. In Proceedings of the 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 15–18 December 2023.

103. Van der Wal, O.; Lesci, P.; Müller-Eberstein, M.; Saphra, N.; Schoelkopf, H.; Zuidema, W.; Biderman, S.R. Polypythias: Stability and outliers across fifty language model pre-training runs. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025.
104. Chung, W.; Hong, J.; An, N.M.; Thorne, J.; Yun, S.-Y. Stable Language Model Pre-training by Reducing Embedding Variability. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–14 November 2024.
105. Garza-Alonso, A.; Basin, M.; Rodriguez-Ramirez, P. C. Predefined-time backstepping stabilization of autonomous nonlinear systems. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 2020–2022.
106. Orlov, Y.; Kairuz, R.I.V. Autonomous output feedback stabilization with prescribed settling-time bound. *IEEE Trans. Autom. Control* **2022**, *68*, 2452–2459.
107. Rajbhandari, S.; Rasley, J.; Ruwase, O.; He, Y. Zero: Memory optimizations toward training trillion parameter models. In Proceedings of the SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, Atlanta, GA, USA, 9–19 November 2020.
108. Nguyen, S.; Chen, L.; Liu, B.; Liu, Q. Memory-efficient optimization with factorized hamiltonian descent. *arXiv* **2024**, arXiv:2406.09958.
109. Wang, S.; Liu, F.; Liu, B. Escaping the gradient vanishing: Periodic alternatives of softmax in attention mechanism. *IEEE Access* **2021**, *9*, 168749–168759.
110. Mudarisov, T.; Burtsev, M.; Petrova, T. Limitations of Normalization in Attention. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 98312–98335.
111. Bauduin, V.; Cuomo, S.; Di Cola, V.S. Constraint satisfaction approach in structuring neural network architectures. *J. Comput. Appl. Math.* **2026**, *476*, 117140.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.