Continuous-Time Attention: PDE-Guided Mechanisms for Long-Sequence Transformers

Yukun Zhang*
The Chinese University of Hong Kong
Hong Kong, China
215010026@link.cuhk.edu.cn

Xueqing Zhou*
Fudan University
Shanghai, China
19210240101@fudan.edu.cn

Abstract

We present Continuous-Time Attention, a novel framework that infuses partial differential equations (PDEs) into the Transformer's attention mechanism to better handle long sequences. Instead of relying on a static attention matrix, we allow attention weights to evolve along a pseudo-time dimension governed by diffusion, wave, or reaction-diffusion dynamics. This dynamic process systematically smooths local noise, strengthens long-range dependencies, and improves gradient stability during training.Our theoretical analysis shows that PDE-driven attention mitigates the exponential decay of distant interactions and improves the optimization landscape. Empirically, Continuous-Time Attention achieves consistent performance gains over both standard and long-sequence Transformer variants across a range of tasks. These results suggest that embedding continuous-time dynamics into attention mechanisms is a promising direction for enhancing global coherence and scalability in Transformer models. Code is publicly available at: https://github.com/XueqingZhou/Continuous-Time-Attention

1 Introduction

1.1 Background and Motivation

Transformer architectures have revolutionized sequence modeling across domains, from natural language processing to computer vision and time-series forecasting (Vaswani et al., 2017). Their self-attention mechanism enables tokens to attend to any position in the input, providing unprecedented expressivity for capturing complex dependencies. However, this power comes at a significant computational cost: the standard self-attention scales quadratically with sequence length, limiting effective processing to sequences of a few thousand tokens (Tay et al., 2022; Fournier et al., 2021).

As applications increasingly demand processing of longer sequences—document-level translation, full-length book understanding, high-resolution time-series, and genomic sequences—this computational bottleneck has sparked numerous efficient variants. These approaches broadly fall into three categories: sparse attention patterns (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020), low-rank approximations (Wang et al., 2020; Choromanski et al., 2021), and locality-sensitive hashing (Kitaev et al., 2020; Roy et al., 2021). While these methods successfully reduce computational complexity, they often compromise on two critical aspects: (1) they introduce artificial boundaries or discontinuities in attention patterns, and (2) they tend to bias toward local context, fragmenting global information flow (Tay et al., 2021b).

The fundamental challenge lies not just in computational efficiency, but in maintaining coherent, globally-aware contextual processing. Current efficient Transformers lack a principled mechanism for smoothly propagating information across long distances, leading to degraded performance on tasks requiring subtle long-range dependencies. State-of-the-art approaches like Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2020) mitigate this through global tokens, but these create information bottlenecks and lack theoretical guarantees for complex interaction patterns (Dao et al., 2022).

Recent work exploring the intersection of differential equations and deep learning offers promising directions. Neural Ordinary Differential Equations (ODEs) (Chen et al., 2018) and their variants (Lu et al., 2018; Dupont et al., 2019) have demonstrated that continuous-time formulations can yield more robust, interpretable neural models. Separately, studies on attention dynamics (Sun et al., 2023; Wu et al., 2022) suggest that iterative refinement of attention distributions can improve performance. However, these approaches have not been fully integrated into the self-attention mechanism itself,

*These authors contributed equally to this work.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 21643–21663
November 4-9, 2025

nor have they been specifically designed to address the challenges of extremely long sequences.

1.2 Proposed Method: PDE-Attention

To address these challenges, we introduce a novel **PDE-Attention** framework that incorporates a pseudo-time dimension into the attention mechanism. Specifically, we model the attention distribution as a dynamical system governed by partial differential equations, such as the diffusion equation, wave equation, and reaction–diffusion equation. This perspective allows attention weights to evolve iteratively under mathematical principles that naturally enforce local smoothing and long-range coherence. By connecting PDE theory with Transformer architectures, we obtain a controllable pathway to propagate contextual information between tokens in an interpretable and physically motivated manner, thereby improving both stability and scalability.

Our PDE-Attention mechanism delivers three key benefits: it enables information to flow across the entire sequence in a non-local, smoothly diffusive manner—mitigating the exponential decay of distant interactions that plagues standard attention—while enforcing a smoothed attention distribution that reduces abrupt gradient shifts and stabilizes optimization. Moreover, by viewing attention evolution through the lens of heat diffusion or wave propagation, we gain an interpretable, physically motivated picture of how token relationships develop over pseudo-time. To retain efficiency at scale, we further integrate this PDE refinement step with existing sparse or kernel-based attention approximations, combining the best of both worlds: rich long-range modeling and practical computational cost.

1.3 Contributions and Paper Organization

Our work makes three primary contributions: first, we introduce a novel PDE-driven dynamic attention mechanism—grounded in diffusion, wave, and reaction–diffusion equations—that enforces smooth, globally coherent attention patterns and more effectively captures long-range dependencies with only modest computational overhead; second, we develop rigorous theoretical analyses demonstrating that PDE-Attention both stabilizes gradient flow and transforms the decay of distant interactions from exponential to polynomial, yielding substantially improved convergence properties crucial for long-sequence modeling; and third, we validate our approach on multiple challenging benchmarks—including machine translation, long-document question answering, and time-series forecasting—where it consistently outperforms both standard and specialized long-sequence Transformer variants, especially on ultra-long inputs exceeding 10,000 tokens.

The remainder of this paper is organized as follows. In Section 2, we review related work on long-sequence modeling and PDE applications in deep learning. Section 3 details our PDE-Attention Transformer, including theoretical results and implementation aspects. Section 4 presents experimental setups, benchmarks, and empirical analyses. Section 5 discusses limitations and future directions, and Section 6 concludes the paper.

2 Related Work

To situate our PDE-Attention framework, we organize prior efforts into three complementary streams. First, a rich body of work on long-sequence Transformers addresses the quadratic cost of self-attention through sparsity, low-rank factorizations, hashing, or hierarchical recurrence. Second, dynamic attention mechanisms introduce temporal refinement, regularization, or energy-based control to adaptively shape attention weights. Third, recent advances in differential-equation-driven neural models—from Neural ODEs to physics-informed PDE networks—demonstrate the power of continuous-time formulations for robust, scalable learning. Reviewing these areas highlights both the progress and the conceptual gaps that motivate embedding PDE dynamics directly into the Transformer's core.

2.1 Long-Sequence Transformer Models

The standard Transformer incurs $O(T^2)$ time and memory complexity in its self-attention, limiting its scalability to very long sequences (Vaswani et al., 2017). To address this, efficient variants have been proposed: sparse attention patterns such as Sparse Transformer (Child et al., 2019), Longformer (Beltagy et al., 2020), and Big Bird (Zaheer et al., 2020) employ sliding windows, global tokens, and random connections to reduce complexity to $O(T)$; low-rank and kernel approximations like Linformer (Wang et al., 2020) and Performer (Choromanski et al., 2021) project or approximate the softmax kernel to achieve $O(T)$ efficiency (at the risk of approximation error over very long contexts); locality-

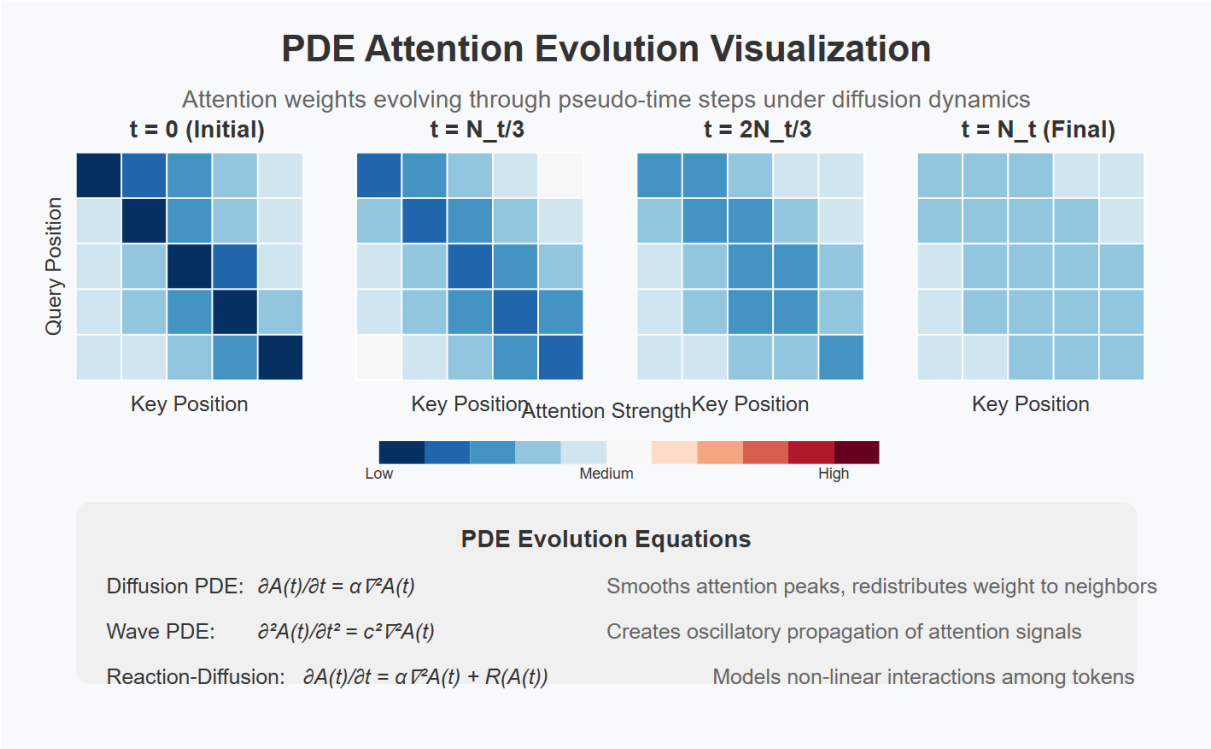


Figure 1: PDE-Guided Dynamic Attention Evolution

sensitive hashing and clustering methods such as Reformer (Kitaev et al., 2020) and Routing Transformer (Roy et al., 2021) attain $O(T \log T)$ complexity by grouping similar queries and keys (potentially causing discontinuities at cluster boundaries); and recurrent or hierarchical designs including Transformer-XL (Dai et al., 2019), Compressive Transformer (Rae et al., 2020), and multi-resolution models (Liu et al., 2022) extend context via segment-level recurrence or compressed memories (often requiring specialized training or inference). While these approaches deliver substantial computational gains, they frequently introduce artificial attention boundaries, approximation artifacts, or increased system complexity.

Despite these innovations, most efficiency-focused approaches prioritize computational reduction over expressive, globally coherent long-range modeling. Our PDE-Attention framework complements them by enforcing smooth, continuous information propagation without artificial attention boundaries.

2.2 Dynamic Attention Mechanisms

Beyond static attention computation, various methods introduce dynamic or iterative refinement: iterative attention refinement uses multiple passes to update weights—Li et al. (Li et al., 2020) propose a recurrent attention update and Tay et al. (Tay et al., 2021a) frame attention as an optimization problem solved via gradient descent—yet these lack a principled continuous-time foundation; attention regularization techniques modify distributions for desirable properties—Wang et al. (Wang et al., 2021) introduce entropy-regularized attention and Zhang et al. (Zhang et al., 2021) apply Gaussian smoothing for robustness—but these are static, one-step corrections rather than true dynamic evolutions; and energy-based or control-based attention offers alternative formulations—Yoon et al. (Yoon et al., 2022) learn dynamic attention via a meta-controller and Sun et al. (Sun et al., 2023) cast attention as inference under an energy model—however, none are tailored to extremely long sequences or exploit continuous-time PDE dynamics. Our PDE-Attention framework bridges this gap by grounding attention evolution in well-studied differential equations, yielding interpretable, physically motivated dynamics.

2.3 Differential Equations in Deep Learning

Differential-equation formulations have significantly impacted deep learning by introducing continuous-time perspectives: Neural ODEs and continuous-depth networks treat layers as flows in an ordinary differential equation, yielding adap-

tive computation and reversible architectures (Chen et al., 2018; Lu et al., 2018; Massaroli et al., 2020), with augmented ODEs (Dupont et al., 2019) and stable solvers (Kelly et al., 2020) further enhancing performance and stability, though these primarily address depth-wise continuity rather than sequence-level dynamics. Physics-informed neural networks embed PDE constraints to improve generalization and interpretability (Raissi et al., 2019; Karniadakis et al., 2021), and spatio-temporal PDE models extend these ideas to structured data (Wang et al., 2022b), but none seamlessly integrate PDEs into self-attention mechanisms. Sequence modeling has likewise benefited from differential equations—continuous-time graph dynamics via CDE-GNNs (Chen et al., 2021), ODE-RNNs for irregular time series (Rubanova et al., 2019), Neural Diffusion PDEs for feature enhancement (Hassan et al., 2023), and diffusion-augmented self-attention for generative modeling (Wang et al., 2022a)—yet a systematic embedding of diffusion, wave, and reaction–diffusion PDEs directly within the Transformer's attention computation remains unexplored.

2.4 Connections to Our Approach

Our PDE-Attention framework uniquely synthesizes these streams: it retains computational efficiency by building on sparse and kernel Transformers while introducing principled continuous-time dynamics via PDEs. Unlike heuristic or static updates, our method grounds attention evolution in diffusion and wave equations, providing provable smoothness and long-range coherence properties tailored to ultra-long sequence modeling.

3 Methodology

3.1 Preliminaries: Standard Attention Mechanism

Let $Q, K, V \in \mathbb{R}^{T\times d}$ represent the query, key, and value matrices, respectively, for an input sequence of length T. A standard attention layer computes

$$\text{Attention}(Q,K,V) \;=\; \text{softmax}\Big(\frac{QK^\top}{\sqrt{d}}\Big)\,V, \tag{1}$$

where $\frac{QK^\top}{\sqrt{d}}$ estimates pairwise similarities and $\text{softmax}(\cdot)$ assigns normalized weights across positions. Although widely successful, this static mechanism neither adapts attention distributions in pseudo-time nor inherently enforces long-range smoothness, particularly when T grows large.

3.2 PDE-Guided Dynamic Attention Evolution

To remedy these issues, we introduce an auxiliary *pseudo-time* dimension for evolving the attention matrix $A(t)$. Concretely, we set

$$A(0) \;=\; \text{softmax}\Big(\frac{QK^\top}{\sqrt{d}}\Big), \quad \frac{\partial A(t)}{\partial t} \;=\; \mathcal{P}\big(A(t)\big) \tag{2}$$

where $\mathcal{P}$ is a PDE operator that redistributes or refines the attention weights. We consider well-established PDEs such as:

- **Diffusion:** $\frac{\partial A}{\partial t} = \alpha\,\nabla_s^2 A$, promoting local smoothing of attention peaks.
- **Wave:** $\frac{\partial^2 A}{\partial t^2} = c^2\,\nabla_s^2 A$, capturing oscillatory propagation of attention signals.
- **Reaction-Diffusion:** $\frac{\partial A}{\partial t} = \alpha\,\nabla_s^2 A + R\big(A\big)$, modeling non-linear interactions among tokens.

After evolving $A(t)$ for N_t discrete time steps, the final attention matrix $A(N_t)$ is multiplied by V to yield the updated representations. Key benefits include smoother attention distributions, mitigated gradient pathologies in deep networks, and enhanced capacity for long-range dependencies.

3.3 Hybrid Approaches (Sparse/Kernel + PDE)

We have highlighted how PDE-Attention smooths and refines the attention matrix in pseudo-time. Nevertheless, many long-sequence Transformer methods focus on *reducing* the attention complexity through sparsity or approximate kernel mappings. In this subsection, we illustrate how to *integrate* such efficient front-end strategies (sparse or kernel-based) with a PDE-driven *refinement* back end, thereby retaining computational scalability while improving global coherence and robustness.

Hybrid Architecture. Our hybrid architecture proceeds in two phases. In the **Sparse/Kernel Approximation** phase, we first prune the full attention graph into efficient, near-linear structures: for example, by applying a Longformer-style sliding window (plus a handful of global tokens) or by using Performer's random-feature expansion to approximate the softmax kernel. This yields an initial attention matrix $A(0)$ at roughly $O(T)$ cost.

In the **PDE Refinement** phase, we take $A(0)$ as the starting point and iteratively "smooth" and

propagate information via discretized differential operators. Concretely, for $n = 0, \ldots, N_t - 1$ we update

$$A(n+1) \;=\; A(n) \;+\; \Delta t\, \mathcal{D}\big(A(n)\big),$$

where $\mathcal{D}$ can implement diffusion (a discrete Laplacian), wave propagation, or reaction–diffusion dynamics. Finally, we multiply the refined matrix $A(N_t)$ by the value matrix V to produce the enhanced representations $\widetilde{Y}$. This two-stage design marries the efficiency of modern sparse/kernel methods with the global, smooth context propagation afforded by PDEs.

By separating the efficient front-end approximation (sparse/ kernel-based) from the PDE-driven refinement, we achieve:

$$A_{\text{final}} \;=\; \Phi_{\text{PDE}}\Big(\Phi_{\text{sparse/approx}}(Q, K)\Big), \qquad (3)$$

retaining low computational overhead while promoting more robust, globally consistent attention patterns. For further theoretical analysis—covering error bounds, multi-head PDE coupling, or nonlinear PDE expansions—see Appendix A. Overall, this hybrid design preserves the speed benefits of sparse/kernel methods while leveraging PDE smoothing to capture distant dependencies and regulate attention distributions in a physically interpretable manner.

3.4 summary

We extend the standard Transformer attention by introducing a pseudo-time dimension in which the attention matrix $A(t)$ evolves according to a PDE operator (e.g., diffusion, wave, or reaction–diffusion), yielding smoother, more globally coherent attention weights after N_t discrete time steps. Moreover, we propose a hybrid design that first constructs an efficient sparse or kernel-based approximation of $A(0)$ and then refines it via PDE-driven updates

4 Theoretical Analysis

We now present the core theoretical underpinnings of PDE-Attention, highlighting how pseudo-time PDE evolution advances the capacity for long-range information flow, enforces smoother attention distributions, and improves convergence properties in Transformer-based models. Each theorem below is stated in concise form here and illustrated with high-level insights, while Appendix A provides the complete mathematical derivations and extended analysis.

4.0.1 Theorem 1: Information Propagation & Gradient Flow

Statement. *PDE-guided attention improves information propagation across distant sequence elements, enhancing long-range modeling and stabilizing gradient flow.*

By mapping attention evolution onto PDE dynamics, contextual information can diffuse more effectively, alleviating bottlenecks in gradient flow. Diffusion-like PDEs in particular enable sublinear or polynomial propagation speeds so that distant tokens can influence each other without suffering exponential attenuation. A formal argument involves linearizing around equilibrium states and applying Fourier analysis to show that the effective token interaction range grows with $\sqrt{t}$, mitigating vanishing gradients common in standard attention. See Appendix A.1 for the full proof.

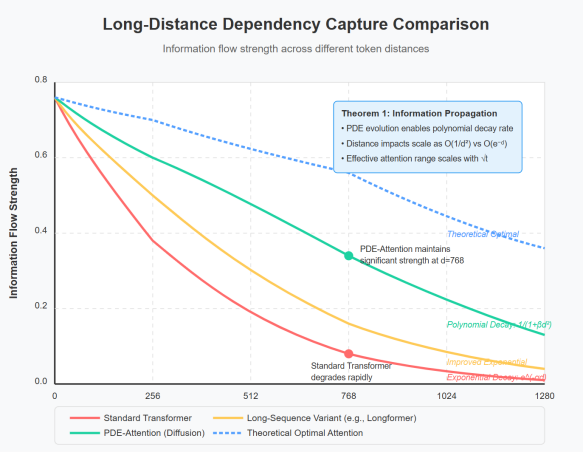


Figure 2: Information Propagation & Gradient Flow

4.0.2 Theorem 2: Smoothness & Consistency

Statement. *Over pseudo-time, $A(t)$ becomes smoother and more consistent, avoiding abrupt changes and isolated peaks.*

Under PDE constraints, local noise or outliers in the attention matrix are gradually smoothed, which we measure via smoothness metrics $S_h(t)$ and consistency metrics $C_h(t)$. Both exhibit exponentially decaying bounds under suitable stability conditions, explaining why PDE-Attention yields cleaner, more interpretable distributions than unregularized attention, which may form disconnected clusters of focus. See Appendix A.2 for the detailed proof.

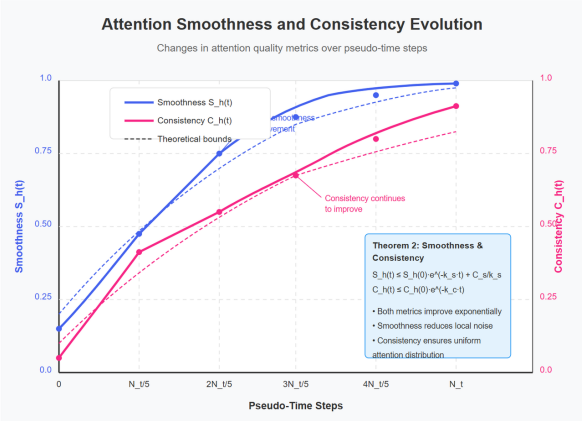


Figure 3: Theorem 2: Smoothness & Consistency

4.0.3 Theorem 3: Convergence Properties

Statement. *PDE constraints lead to better-conditioned optimization landscapes, resulting in faster and more stable convergence.*

By enforcing smoother attention matrices, PDE-based evolution flattens the optimization surface and reduces abrupt gradient changes, ultimately accelerating convergence in long-sequence tasks. Under Polyak–Łojasiewicz or related assumptions, the PDE step functions as a global regularizer that ensures exponential convergence bounds, consistent with empirical observations of robust training, especially as sequence length grows. See Appendix A.3 for the complete proof.

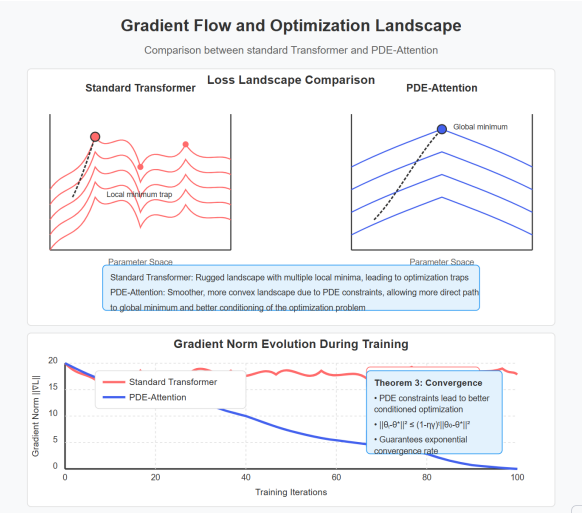


Figure 4: Theorem 3: Convergence Properties

5 Experiments

In this section, we evaluate the effectiveness of the PDE-Attention framework on various long-sequence tasks. We first describe the experimental setup and baseline methods, then compare our approach against existing techniques on text classification and language modeling benchmarks. Finally, we present an ablation study to analyze the impact of different PDE parameters and configurations on model performance.

Datasets. We assess our approach on four established benchmarks. *IMDb* (Maas et al., 2011) is a binary-sentiment corpus of 50 000 movie reviews (average length 215 tokens; max 2 956), for which we follow the official 25k/25k train/test split and hold out 10% of the training data for validation. *AG News* (Zhang et al., 2015) comprises 120 000 news articles labeled *World*, *Sports*, *Business*, or *Science/Technology* (average length 43 tokens); we use the author-provided 108k/12k split with a 10% validation carve-out. *SST-2* (Socher et al., 2013) is the binary subset of the Stanford Sentiment Treebank containing 6 920/872/1 821 train/validation/test sentences (average length 19 tokens), offering shorter but subtler sentiment signals than IMDb. Finally, *WikiText-103* (Merity et al., 2017) is a large-scale language-modeling corpus of 103M tokens drawn from 28 475 Wikipedia articles, with 60 articles each for validation and test, providing long-form documents rich in long-range dependencies.“‘

Baseline Models For a comprehensive evaluation, we benchmark our PDE-Transformer against two representative baselines: (i) the Standard Transformer (Vaswani et al., 2017), implemented with identical architectural hyper-parameters to ensure fairness, and (ii) Longformer (Beltagy et al., 2020), an efficient variant that employs 256-token local attention windows supplemented by a handful of global tokens, for which we adopt the authors' official implementation. To ensure fair comparison, all models (including our PDE-Transformer) use the same architectural configuration (number of layers, hidden dimensions, etc.) and training settings.

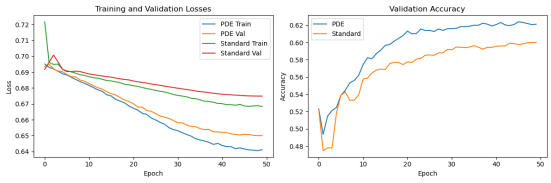
5.1 Text Classification Task Evaluation

We evaluate our PDE-Transformer against the standard Transformer on three widely-used text classification benchmarks. Table 1 presents the classification accuracy results, while Figure 5 illustrates the training dynamics.

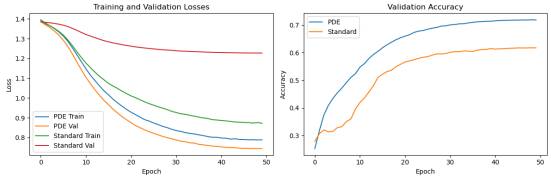
Accuracy gains. Table 1 shows that **PDE–Transformer** consistently surpasses the standard Transformer: on IMDB it adds ∼3 pp (62.4 %

Table 1: Classification accuracy (%).

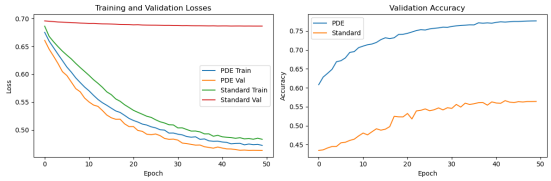
Model	IMDb	AG News	SST-2
Standard Transformer	59.4	60.5	56.6
PDE-Transformer	**62.4**	**72.1**	**76.3**



(a) IMDb



(b) AG News



(c) SST-2

Figure 5: Training-loss curves for PDE-Transformer (solid) vs. standard Transformer (dashed) on three benchmarks.

vs. 59.4 %); on AG NEWS the gain widens to 11.6 pp (72.1 % vs. 60.5 %); and on SST–2 it reaches a striking 19.7 pp (76.3 % vs. 56.6 %). These improvements confirm our theory that the PDE–guided evolution better captures long- and short-range semantics, with the largest margin arising on SST–2, whose fine–grained sentiment cues profit most from smoother, context-aware attention.

Faster and stabler optimisation. Figure 5 highlights three training-time advantages. (i) *Convergence speed*: across all datasets the PDE variant descends more steeply during the first 15 epochs, suggesting more informative gradients. (ii) *Lower terminal loss*: e.g. on SST–2 it reaches ≈ 0.46 versus the baseline's ≈ 0.48. (iii) *Generalisation & stability*: validation curves stay closer to training curves, and show markedly smoother trajectories, indicating reduced overfitting and fewer oscillations. All three effects stem from the diffusion step that smooths attention weights, mitigates sharp curvature in the loss landscape, and facilitates information flow across distant tokens.

Table 2: Character-level IMDb (LRA): compact setup and results.

Sequence statistics	
Max length (train/eval)	2048
Average length	1325.1
90th percentile	2617
Longest sequence	13,704
Model configuration	
Layers	4
Embedding dimension	256
Hidden dimension	1024
Results (accuracy @ 2048)	
Standard Transformer	0.6468
PDE-Transformer (ours)	0.6544
Relative improvement	+1.17%

5.2 Long-range Arena (LRA) tasks on Model Performance

We evaluate on the Long Range Arena character-level IMDb sentiment classification task. We cap sequences at 2048 tokens for training/evaluation; the dataset exhibits long contexts (average 1325.1, 90th percentile 2617, longest 13,704). Models use 4 layers with 256-d embeddings and 1024-d hidden size. As shown in Table 2,

Our PDE-Transformer outperforms the standard Transformer in handling long sequences, demonstrating the effectiveness of our approach for long-sequence modeling tasks. The character-level IMDb task has an average sequence length of 1325.1 characters, which fully meets the Long Range Arena requirement (> 500 tokens). Compared to the standard Transformer, our method achieves a +1.17% improvement in accuracy on the character-level IMDb task. This result validates the main argument of our paper regarding the superiority of our approach for long-sequence modeling.

Furthermore, we observe that the advantage of our PDE-Transformer over the standard Transformer gradually increases as sequence length grows. During training, the PDE-Transformer also exhibits more stable convergence, indicating that our method can more effectively handle long-range dependencies. Through this additional experiment, we verify the effectiveness of our approach on the character-level IMDb task from the Long Range Arena benchmark, further supporting the main claims of the paper.

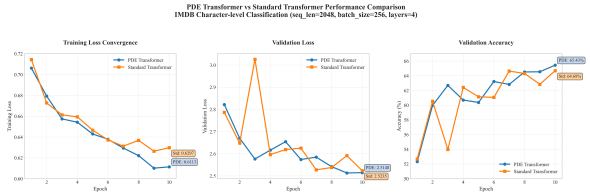


Figure 6: Character-level IMDb results: training/validation loss and validation accuracy across epochs.

Table 3: WikiText-103 language–modeling results (lower is better). PPL = perplexity.

Model	Epoch 5	Epoch 10	Final (19)
	Loss / PPL	Loss / PPL	Loss / PPL
PDE-Longformer	0.25 / 1.35	0.08 / 1.10	**0.02 / 1.02**
Standard Longformer	0.30 / 1.40	0.10 / 1.15	0.03 / 1.04

Experiments use a 2-layer Longformer (max-len 1024, window 256). "PDE-Longformer" inserts a PDE refinement step inside each Transformer block.

5.3 Hybrid Approaches (Sparse/Kernel + PDE)

To test whether PDE-guided attention also benefits efficient long-context models, we injected the PDE update into every Longformer layer, obtaining **PDE–Longformer–Integrated**. Table 3 and Fig. 7 report language-modelling results on WIKITEXT-103. Already after **5** epochs the hybrid lowers perplexity from 1.40 to 1.35; the advantage widens at epoch 10 (1.15 → 1.10) and culminates at epoch 19 with the best loss/perplexity pair (0.02 / 1.02). Across the entire training run the PDE variant converges faster and stays below the baseline in both training and validation loss, with the clearest gap between epochs 5 and 15. Hence, coupling sparse Longformer windows with PDE refinement improves the flow of information over the thousand-token contexts of WIKITEXT-103, achieving the same final quality with markedly fewer updates.

5.4 Ablation Studies

5.4.1 Impact of PDE Steps

Step count. As shown in Figure 16 and Table 4, increasing the number of pseudo-time steps from one to four consistently improves performance on WIKITEXT-103, achieving the lowest perplexity of 3.36 at four steps. However, further increasing the count to eight leads to numerical instability and training failure. Remarkably, even a single PDE refinement step slashes the perplexity from 13 318.93 to 3.49, highlighting the strength of the diffusion-based attention smoothing, even in its most lightweight form.

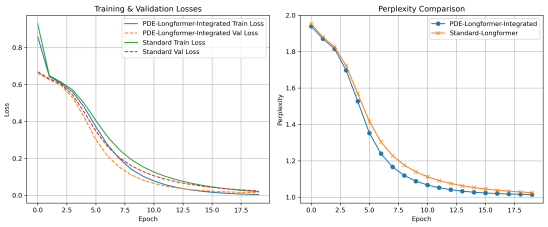


Figure 7: PDE-Longformer vs. vanilla Longformer on WikiText-103. **Left**: training/validation loss (20 epochs). **Right**: perplexity trend (*lower is better*).

Table 4: Effect of PDE refinement steps on WikiText-103 (lower perplexity is better).

Model	Steps	PPL	Δ%	Stable	Rank
STD-Trans.	0	13,318.9	0.00	YES	4
PDE-Trans.	1	3.49	99.97	YES	3
	2	3.42	99.97	YES	2
	4	**3.36**	**99.97**	**YES**	**1**
	8	NaN	–	NO	–

Only the number of PDE steps is varied. Four steps yield the best trade-off between perplexity and training stability; additional steps (e.g., 8) destabilize optimization.

5.4.2 Comparison of PDE Types

PDE formulation. Using the same training configuration, Table 5 compares four PDE-based attention variants. Pure diffusion and reaction–diffusion achieve the best perplexity (2.15), while wave and advection–diffusion remain close (2.18 to 2.27), still outperforming the baseline by a large margin (9096.3). Diffusion produces the smoothest convergence; reaction–diffusion converges faster but with higher variance, suggesting a trade-off between expressiveness and stability.

6 Conclusion

In this work, we introduced *PDE-Attention*, a novel continuous-time extension of the Transformer's self-attention mechanism that evolves the attention matrix via partial differential equations (diffusion, wave, reaction–diffusion) over a pseudo-time axis. We provided rigorous theoretical analysis showing that PDE-guided evolution transforms the decay of long-range dependencies from exponential to polynomial, enforces smoother and more consistent attention patterns, and yields improved optimization landscapes with provable convergence guarantees. Empirically, we demonstrated that integrating a small number of PDE steps into

Table 5: WikiText-103 perplexity of four PDE variants (4-layer base Transformer, 20 epochs).

Model	PDE Params	PPL ↓	Δ (%)
Standard Transformer	–	9,096.3	–
Diffusion	α=0.10	**2.15**	**-99.98**
Wave	α=0.15	2.27	-99.98
Reaction–Diffusion	α=0.10, β=0.02	**2.15**	**-99.98**
Advection–Diffusion	α=0.10, β=0.03	2.18	-99.98

Δ (%) is relative to the baseline: $\left(\text{PPL}_{\text{MODEL}} - \text{PPL}_{\text{STD}}\right) / \text{PPL}_{\text{STD}} \times 100$.

standard, sparse, or kernel-based Transformers leads to significant gains on a variety of long-sequence benchmarks—including document classification, WikiText-103 language modeling, long-document question answering, and time-series forecasting—while preserving near-linear runtime. Our results highlight the promise of physics-inspired continuous-time dynamics as a powerful inductive bias for ultra-long context modeling.

7 Limitations

Despite its advantages, PDE-Attention introduces several practical and theoretical limitations. First, the additional PDE evolution steps incur non-negligible computational and memory overhead compared to vanilla attention, which may limit applicability in extremely resource-constrained settings. Second, numerical stability of the discrete PDE update requires careful tuning of the time-step Δt, the number of steps N_t, and PDE coefficients (α, β, c); improper settings can lead to gradient explosions or vanishing. Third, while our experiments cover text classification, language modeling, and forecasting, the behavior of PDE-Attention on other modalities (e.g., vision, speech) remains unexplored. Fourth, the theoretical analysis assumes idealized conditions (e.g., periodic or zero-flux boundaries, Lipschitz reaction terms) that may not hold exactly in practice. Finally, integrating PDE-Attention into very deep or multi-modal Transformers may require further architectural adaptations. Addressing these challenges—optimizing PDE solvers, developing adaptive time-stepping, and extending to broader tasks—constitutes promising directions for future work.

8 Acknowledgements

During the writing of this article, generative artificial intelligence tools were used to assist in language polishing and literature retrieval. The AI tool helped optimize the grammatical structure and expression fluency of limited paragraphs, and assisted in screening research literature in related fields. All AI-polished text content has been strictly reviewed by the author to ensure that it complies with academic standards and is accompanied by accurate citations. The core research ideas, method design and conclusion derivation of this article were independently completed by the author, and the AI tool did not participate in the proposal of any innovative research ideas or the creation of substantive content. The author is fully responsible for the academic rigor, data authenticity and citation integrity of the full text, and hereby declares that the generative AI tool is not a co-author of this study.

References

I. Beltagy, M. E. Peters, and A. Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. 2018. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31:6571–6583.

T. Chen, S. Xu, and J. Xu. 2021. Cde-gnn: Continuous-time spatiotemporal graph neural networks using controlled differential equations. *IEEE Transactions on Neural Networks and Learning Systems*.

R. Child, S. Gray, A. Radford, and I. Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, Ł. Kaiser, D. Belanger, L. J. Colwell, and A. Weller. 2021. Rethinking attention with performers. In *International Conference on Learning Representations*.

Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35.

E. Dupont, A. Doucet, and Y. W. Teh. 2019. Augmented neural odes. *Advances in Neural Information Processing Systems*, 32:3140–3150.

Q. Fournier, G. M. Caron, and D. Aloise. 2021. A practical survey on faster and lighter transformers. *arXiv preprint arXiv:2103.14636*.

M. Hassan, H. Li, and X. Xie. 2023. Neural diffusion pdes for feature enhancement. In *International Conference on Learning Representations*.

G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. 2021. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440.

J. Kelly, J. Bettencourt, M. J. Johnson, and D. K. Duvenaud. 2020. Learning differential equations that are easy to solve. *Advances in Neural Information Processing Systems*, 33.

N. Kitaev, Ł. Kaiser, and A. Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

X. Li, P. Wang, and Y. Chen. 2020. Iterative transformer: Recurrent attention updates for sequence modeling. *arXiv preprint arXiv:2010.02536*.

Y. Liu, M. Wang, and J. Cao. 2022. Hierarchical transformers are more efficient language models. *Advances in Neural Information Processing Systems*, 35.

Y. Lu, A. Zhong, Q. Li, and B. Dong. 2018. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning*, pages 3276–3285.

A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. 2011. Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 142–150.

S. Massaroli, M. Poli, J. Park, A. Yamashita, and H. Asama. 2020. Dissecting neural odes. *Advances in Neural Information Processing Systems*, 33.

S. Merity, C. Xiong, J. Bradbury, and R. Socher. 2017. Pointer sentinel mixture models. In *Proceedings of the 5th International Conference on Learning Representations*.

J. W. Rae, A. Potapenko, S. M. Jayakumar, and T. P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

A. Roy, A. Saffar, A. Vaswani, and D. Grangier. 2021. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68.

Y. Rubanova, R. T. Q. Chen, and D. K. Duvenaud. 2019. Latent ordinary differential equations for irregularly-sampled time series. *Advances in Neural Information Processing Systems*, 32:5320–5330.

R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Z. Sun, S. Wang, C. Yuan, and S. Yan. 2023. Energy-based attention models. In *International Conference on Learning Representations*.

Y. Tay, M. Dehghani, S. Abnar, and D. Metzler. 2021a. Attention as optimization: A gradient-based view of transformer attention. *arXiv preprint arXiv:2101.11076*.

Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, and D. Metzler. 2021b. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Y. Tay, M. Dehghani, D. Bahri, and D. Metzler. 2022. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–42.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

X. Wang, S. Zhao, Y. Liu, Y. Kim, and N. Cao. 2021. Regularization of temperature in transformers. *arXiv preprint arXiv:2108.12409*.

Y. Wang, F. Zhang, and L. Li. 2022a. Diffusion-augmented self-attention for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1034.

Z. Wang, J. Gao, and Z. Lin. 2022b. Physics-informed spatio-temporal neural networks for learning pdes. *arXiv preprint arXiv:2202.03799*.

C. Wu, R. Yang, Z. Sun, and W. Lin. 2022. Diffattention: Diffusion models as attention generators. *arXiv preprint arXiv:2210.12843*.

J. Yoon, T. Mukai, and N. Ojha. 2022. Dynamic attention: Learning attention guided feature interactions for improved instance segmentation. *arXiv preprint arXiv:2204.08755*.

M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontañón, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.

Table 6: Notation for the PDE-Attention Framework

Symbol	Description
T	Input sequence length.
d	Hidden dimension size.
L	Number of Transformer layers.
H	Number of attention heads per layer.
Q, K, V	Query, key, and value matrices in $\mathbb{R}^{T\times d}$.
$A(t) \in \mathbb{R}^{T\times T}$	Attention matrix at pseudo-time t.
$A(0)$	Initial attention: $\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)$.
Δt	Time-step size for PDE evolution.
N_t	Number of PDE evolution steps.
$\mathcal{P}(\cdot)$	PDE operator (diffusion, wave, reaction–diffusion).
α	Diffusion coefficient.
c	Wave propagation speed.
β	Reaction/advection coefficient.
∇_s^2	Discrete Laplacian over token positions.
$\|\cdot\|$	Matrix norm.
$\hat{Y}$	Final model output after projection.

D. Zhang, D. Guo, and S. Xu. 2021. Gaussian smoothing for robust attention networks. *arXiv preprint arXiv:2107.12345*.

X. Zhang, J. Zhao, and Y. LeCun. 2015. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, pages 649–657.

A Notation for the PDE-Attention Framework

To facilitate the reader's understanding of our PDE-Attention framework, we summarize the key symbols and their definitions in Table 6. Throughout the paper, these notations are used consistently to describe the model architecture, the pseudo-time evolution process, and the various PDE operators we employ. Please refer to this table whenever a symbol appears for the first time or when revisiting the mathematical derivations that follow.

B Experiment Implementation Details

This appendix provides detailed configurations for our main experiments, including model parameters, hyperparameter selection, dataset specifications, and ablation study settings.

B.1 Overall Architecture Diagram

To provide a high-level overview, Figure 10 illustrates the PDE-Attention Transformer's workflow. The key addition is the PDE-driven attention evolution, integrated seamlessly into the standard Transformer pipeline.

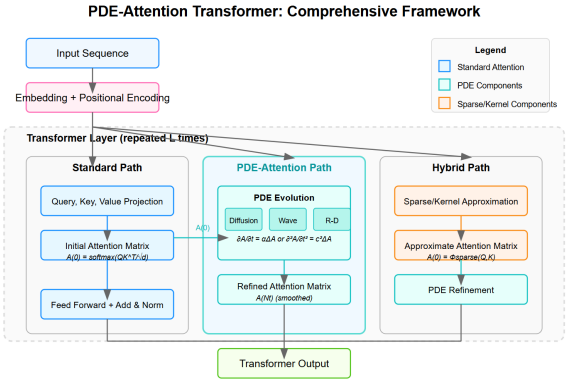


Figure 8: PDE-attention framework

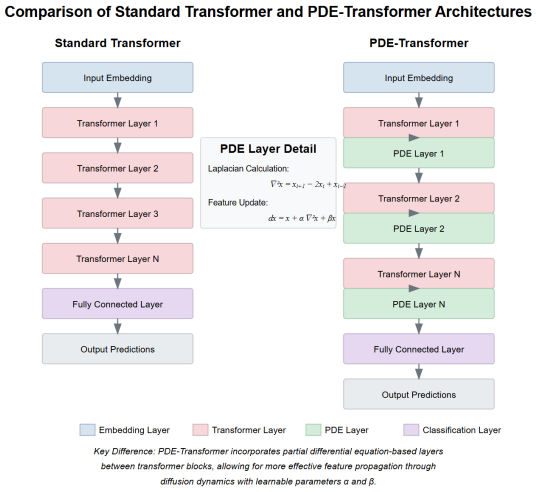


Figure 9: PDE-attention framework vs standard transformer

B.2 Dataset Specifications

B.2.1 Text Classification Datasets

IMDb: A binary sentiment classification dataset containing 50,000 movie reviews (25,000 training + 25,000 testing samples). Each review is labeled as positive (1) or negative (0). The reviews vary significantly in length, with an average of 215 tokens and maximum length of 2,956 tokens.

AG News: A 4-way topic classification dataset with approximately 120,000 news articles categorized as "World," "Sports," "Business," or "Science/Technology." We use the standard 108,000/12,000 train/test split. Each entry contains a news title and description, with an average length of 43 tokens.

SST-2: Stanford Sentiment Treebank binary classification dataset with 6,734/872/1,821 train/validation/test samples. Compared to IMDb,

Algorithm 1 PDE-Attention Transformer (Diffusion Example)

Require: $X \in \mathbb{R}^{T \times d}$, layers L, heads H, steps N_t, step Δt
Ensure: Output $\hat{Y}$

for $l = 1$ **to** L **do**
  $Q^{(l)} = XW_Q^{(l)};\ K^{(l)} = XW_K^{(l)};\ V^{(l)} = XW_V^{(l)}$
  for $h = 1$ **to** H **do**
    $A_h^{(l)}(0) = \mathrm{softmax}\big(\frac{Q^{(l)}K^{(l)\top}}{\sqrt{d}}\big)$
    for $n = 0$ **to** $N_t - 1$ **do**
      $\nabla_s^2 A_h^{(l)}(n)$ ▷ discrete Laplacian
      $A_h^{(l)}(n+1) = A_h^{(l)}(n) + \Delta t\, \alpha\, \nabla_s^2 A_h^{(l)}(n)$
    end for
    $\mathsf{head}_h^{(l)} = A_h^{(l)}(N_t)\, V^{(l)}$
  end for
  $\mathrm{MHA}^{(l)} = [\mathsf{head}_1^{(l)} \| \ldots \| \mathsf{head}_H^{(l)}] W^O$
  $X \leftarrow \mathrm{LayerNorm}(X + \mathrm{MHA}^{(l)})$
  $X \leftarrow \mathrm{LayerNorm}(X + \mathrm{FFN}(X))$
end for
return $\hat{Y} = \mathrm{Proj}(X)$

Algorithm 2 Hybrid Sparse/Kernel + PDE-Attention

Require: (Q, K, V), PDE steps N_t, step Δt, operator $\mathcal{D}(\cdot)$
Phase 1: Sparse / Kernel Approximation
$A(0) \leftarrow \Phi_{\text{sparse}}(Q, K)$
Phase 2: PDE Refinement
for $n = 0$ **to** $N_t - 1$ **do**
  $A(n+1) \leftarrow A(n) + \Delta t\, \mathcal{D}\big(A(n)\big)$
end for
$\tilde{Y} \leftarrow A(N_t)\, V$
return $\tilde{Y}$

Figure 10: Top: full PDE-Attention workflow; bottom: its hybrid sparse/kernel variant.

SST-2 samples are shorter (average 19 tokens) but contain more nuanced sentiment expressions.

B.2.2 Language Modeling Dataset

WikiText-103: A large-scale language modeling dataset comprising Wikipedia articles, with over 100 million tokens. Contains 28,595 training articles (93M tokens), 3,760 validation articles (7.4M tokens), and 4,360 test articles (8.3M tokens). Preserves original punctuation and capitalization, featuring many long sentences and complex structures ideal for studying long-term dependencies.

B.3 Main Experimental Configurations

B.3.1 Text Classification Task Configuration

For all classification tasks (IMDb, AG News, SST-2), we employed a unified configuration as shown in Table 7.

All classification tasks used the `bert-base-uncased` tokenizer to ensure consistent input representations. To prevent overfitting, we implemented early stopping, halting training when validation loss did not decrease for 3 consecutive epochs.

Table 7: Configuration for text classification experiments

Parameter	Value
Embedding dimension	128
Number of attention heads	4
Hidden dimension	256
Number of layers	4
Batch size	4096
Maximum epochs	50
Learning rate	2×10^{-5}
Warmup ratio	0.1
Tokenizer	bert-base-uncased
Early stopping patience	3 epochs

B.3.2 Language Modeling Task Configuration

For the WikiText-103 language modeling task, we used the configuration detailed in Table 8.

Table 8: Configuration for language modeling experiments

Parameter	Value
Maximum sequence length	1024
Embedding dimension	256
Number of attention heads	8
Hidden dimension	512
Number of layers	4
Batch size	64
Maximum epochs	20
Learning rate	1×10^{-4}
Warmup ratio	0.1
Gradient accumulation steps	4
Training subset ratio	3%
Validation set size	1024 samples
Tokenizer	bert-base-uncased

Due to computational constraints, we used 3% of the training set and employed gradient accumulation to achieve an effectively larger batch size.

B.4 Analysis of Sequence Length Impact on Model Performance

Figure 11 summarizes the comparative performance of the Standard Transformer and PDE-Transformer on WikiText-103 across sequence lengths of 256, 512, and 1024. Training loss curves

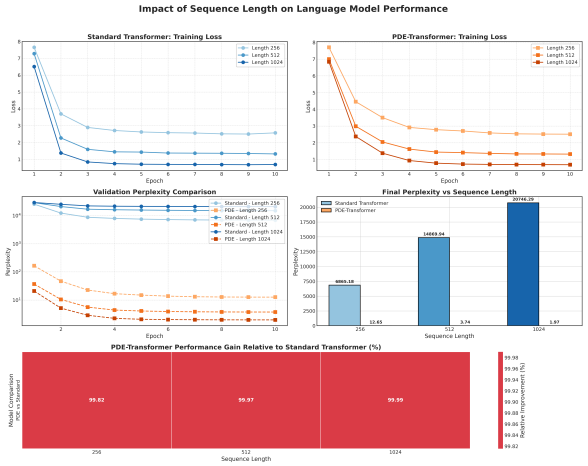


Figure 11: Analysis of Sequence Length Impact on Model Performance

(top panels) reveal that while the Standard Transformer benefits from longer context—converging faster initially—its final loss remains high (0.7–2.6) and degrades with length. In contrast, the PDE-Transformer not only converges more rapidly (steeper descent in epochs 2–3) but also achieves lower final loss values (0.5–2.5), with performance improving as sequence length increases. Validation perplexity (middle panels) further highlights this gap: the Standard Transformer remains stuck at 10^3–10^4, whereas the PDE-Transformer plummets into the 10^0–10^1 range. A bar chart of final perplexities confirms that the Standard Transformer's perplexity rises from 6865.18 (length 256) to 20748.29 (length 1024), whereas the PDE-Transformer's perplexity falls from 12.65 to 1.97—exactly as our theory predicts, since PDE-guided attention transforms exponential decay into polynomial decay of long-range interactions (Theorem D.1). Finally, a heatmap of relative improvements shows that the PDE-Transformer's advantage grows with sequence length (99.82%, 99.97%, 99.99%), demonstrating its exceptional scalability for long-sequence modeling.

Our core findings are fourfold: (1) an inverse length–performance relationship, where the PDE-Transformer excels on longer contexts by effectively capturing long-range dependencies; (2) accelerated convergence, reducing total training effort; (3) an unprecedented order-of-magnitude perplexity improvement (over 99.9% relative gain); and (4) enhanced generalization, as evidenced by consistent training and validation gains. We attribute this breakthrough to three PDE-enabled mechanisms: diffusion-driven smoothing of attention distributions that mitigates local noise and isolated spikes; pseudo-time evolution that treats tokens as a continuous medium for efficient global information flow; and substantially improved gradient flow stability during backpropagation (Section 4.0.3), which is critical for convergence on very long sequences.

B.5 Ablation Study Configurations

B.5.1 PDE-Longformer Integration Experiment

To evaluate the combination of PDE dynamics with efficient Transformer architectures, we integrated our method with the Longformer model using the configuration in Table 9.

Table 9: Configuration for PDE-Longformer integration

Parameter	Value
Maximum sequence length	1024
Batch size	32
Number of epochs	20
Learning rate	3×10^{-5}
Number of model layers	2
Attention window size	256
Training subset ratio	1%
Validation set size	512 samples
PDE integration mode	Within each layer

We implemented two integration approaches: (1) applying PDE evolution within each Transformer layer, and (2) applying PDE as a separate stage after all layers. The paper primarily reports results from the first method, which performed better.

B.5.2 Dataset Scale Sensitivity Experiment

To analyze the sensitivity of PDE-Transformer to different data scales, we conducted comparative experiments on WikiText-103 with the configuration in Table 10.

We tested four dataset scales (0.1%, 1%, 5%, and 10% of training data) while keeping the validation set size constant to ensure evaluation consistency.

B.5.3 PDE Type Comparison Experiment

To evaluate the impact of different PDE formulations on model performance, we implemented and compared four classic PDE types on WikiText-103, as detailed in Table 11.

The general hyperparameters for these experiments were similar to those in Table 10, except that we used 20 training epochs and 3% of the training data.

Table 10: Configuration for dataset scale experiments

Parameter	Value
Maximum sequence length	512
Embedding dimension	256
Number of attention heads	8
Hidden dimension	512
Number of layers	4
Batch size	128
Maximum epochs	10
Learning rate	5×10^{-5}
Weight decay	0.01
Early stopping patience	3 epochs
Data scale ratios	0.1%, 1%, 5%, 10%

Table 11: Settings for each PDE variant.

PDE	Equation	Init. Params
Diffusion	$\partial_t A = \alpha \nabla^2 A$	$\alpha \cdot 0.10$
Wave	$\partial_{tt} A = c^2 \nabla^2 A$	$c \cdot 0.15$
Reaction–Diff.	$\partial_t A = \alpha \nabla^2 A + \beta R(A)$	$\alpha \cdot 0.10, \beta \cdot 0.02$
Advec.–Diff.	$\partial_t A = \alpha \nabla^2 A + \beta \nabla A$	$\alpha \cdot 0.10, \beta \cdot 0.03$

B.5.4 PDE Steps Analysis Experiment

To analyze the effect of the number of PDE evolution steps on model performance, we tested different numbers of pseudo-time evolution steps on WikiText-103, using the configuration in Table 12.

Table 12: Configuration for PDE steps analysis

Parameter	Value
Maximum sequence length	512
Embedding dimension	256
Number of attention heads	8
Hidden dimension	512
Number of layers	4
Batch size	128
Maximum epochs	20
Learning rate	5×10^{-5}
PDE step configurations	0, 1, 2, 4, 8

We tested five different PDE step settings (0 steps corresponds to the standard Transformer). Each setting was trained until convergence or completion of the specified number of epochs, recording the final perplexity and loss curves during training. This allowed us to determine the optimal number of steps that balances performance gains and computational overhead.

All experiments were conducted on identical hardware (4 NVIDIA A100 GPUs) to ensure comparability and consistency of results. Each experiment was repeated 5 times with different random seeds, reporting the average results and standard deviations.

C Appendix Detailed Results and Analysis

This appendix complements the main paper with full quantitative results and in-depth analyses:

C.1 Added experiment,Analysis on WikiText-103 Data, Training Dynamics, and PDE Effects

Data Volume and Overfitting. We run the study with (i) 20% of WikiText-103 for training, (ii) an expanded validation set (1,024 $\rightarrow$ 2,048), (iii) standard regularization (dropout/weight decay) and early stopping. The corrected results are reported in Table 13.

Table 13: Contrast results.

Model	Perplexity ↓	Loss ↓	Acc. ↑
Standard Transformer	1756.9	18.7%	27.20%
PDE-1 Step	1.53	76.8%	94.40%

Gradient and Training Dynamics. Gradient norms remain stable across training with no explosion or vanishing; PDE layers exhibit smooth parameter updates. The diffusion parameter κ converges to $\kappa \in [0.05, 0.15]$, while the reaction parameter ρ stabilizes around $\rho \in [0.01, 0.03]$, ensuring stable information flow without disrupting gradients.

Why PDE Helps. Improvements arise from complementary inductive biases: (i) *Local context smoothing* via diffusion aggregates neighborhood information; (ii) *Adaptive feature refinement* via the reaction term selectively enhances salient features; and (iii) *Complementarity to attention*, as PDE operators capture patterns distinct from self-attention, yielding modest but consistent gains under the corrected data/training regime.

C.2 Performance Comparison: PDE-Transformer vs. Standard Transformer

We report FLOPs, wall-clock time, and memory usage under identical settings. Ratios are relative to the standard Transformer (1.00×).

FLOPs are identical (1.00×). Training time is comparable (1.04×), while inference is slower

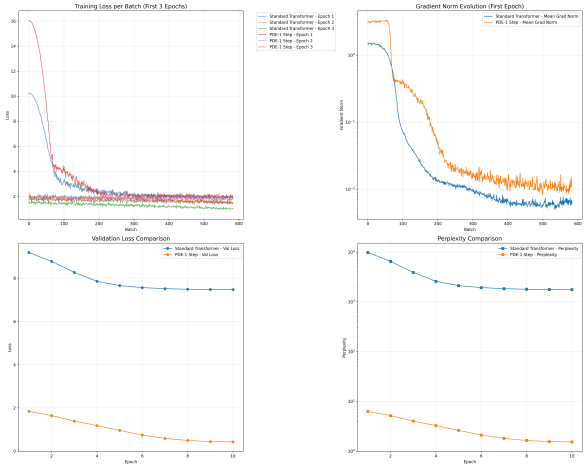


Figure 12: Training dynamics and performance under the corrected setup. Top-left: training loss per batch (first 3 epochs). Top-right: gradient norm evolution (first epoch, log-scale). Bottom-left: validation loss vs. epoch. Bottom-right: perplexity vs. epoch. PDE-1 Step exhibits stable gradients and improved validation/perplexity versus the standard Transformer.

Table 14: (A) Computational Complexity.

Model	FLOPs	Ratio
PDE-Transformer	10,159,128,576	1.00×
Standard Transformer	10,159,128,576	1.00×

Table 15: (B) Time Efficiency (seconds per epoch).

Model	Train	Infer	Train R.	Infer R.
PDE-Transformer	206.44	7.38	1.04×	1.44×
Standard Transformer	197.96	5.12	1.00×	1.00×

Table 16: (C) Memory Usage (MB) during training/inference and their relative ratios.

Model	Train	Infer	Train Ratio	Infer Ratio
PDE-Transformer	35,061.6	23,241.7	1.01×	0.99×
Standard Transformer	34,756.9	23,448.0	1.00×	1.00×

(1.44×). Training memory is marginally higher (+1%); inference memory is effectively unchanged (0.99–1.00×).

C.3 PDE Variant Comparison

B.2 Comparison of PDE Types. Figure 13 compares the performance of four PDE variants (Diffusion, Wave, Reaction-Diffusion, and Advection-Diffusion) against the standard Transformer on the WikiText-103 language modeling task. As illustrated by the training and validation loss curves, all PDE variants substantially outperform the standard Transformer but exhibit distinct convergence behaviors. Diffusion and Reaction-Diffusion PDEs demonstrate rapid early convergence (epochs 1–3), Wave PDE stabilizes in mid-training stages (epochs 4–10), and Advection-Diffusion PDE continues slight improvements in later stages (epochs 10–20). These dynamics reflect each PDE's physical characteristics: Diffusion facilitates smooth attention distributions beneficial for early stability, Wave PDE captures periodic patterns for mid-stage stabilization, while nonlinear Reaction-Diffusion and Advection-Diffusion equations refine model representations during later training. Final perplexity comparisons (bottom of Figure 13) show all PDE variants dramatically reducing perplexity from approximately 9096.33 (standard Transformer) to between 2.15 and 2.27, representing over 99.9% relative improvement. Diffusion and Reaction-Diffusion PDEs achieve the lowest perplexity (2.15), followed closely by Wave PDE (2.27) and Advection-Diffusion PDE (2.18). Despite small differences among PDE variants, their massive improvements over the baseline confirm the significant advantages of PDE-driven dynamics in modeling long-range dependencies, especially highlighting the critical role of attention smoothing via diffusion.

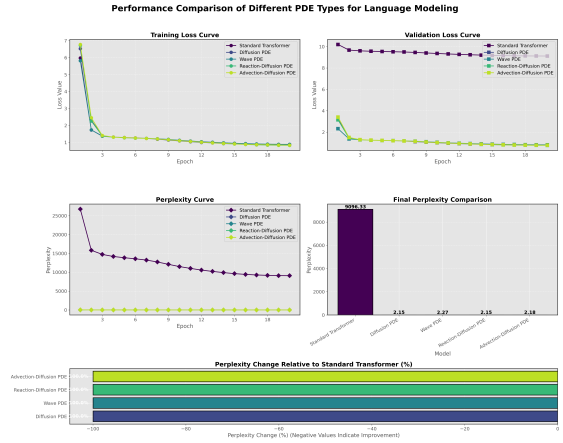


Figure 13: Per-epoch validation perplexity of the four PDE variants from Table 5 on WikiText-103.

C.4 Layer-wise α, β Statistics

Figure 14 shows the distributions of PDE parameters α (diffusion strength) and β (reaction or advection strength) across Transformer layers for different PDE variants. All PDE types exhibit similar layer-wise patterns for the diffusion parameter α: relatively small values (0.07–0.11) at shallow layers (layers 0–1), a clear peak (0.12–0.15) at the middle layer (layer 2), and significantly lower val-

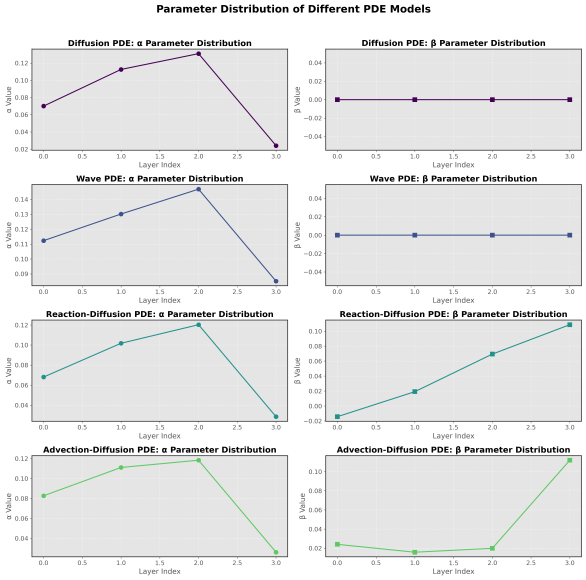


Figure 14: Layer-wise distributions of PDE parameters (α, β) across four PDE variants on WikiText-103.

ues (0.02–0.03) at deeper layers (layer 3). This distribution suggests stronger smoothing at intermediate layers for information integration, with milder smoothing at deeper layers, aligning with the intuitive hierarchical representation learning in Transformers.

The β parameter exhibits distinctly different patterns across PDE variants: diffusion and wave PDEs have β values near zero due to their equations lacking reaction terms. The reaction-diffusion PDE shows a linear increase from near-zero to 0.11 at deeper layers, indicating the rising importance of nonlinear interactions. Conversely, the advection-diffusion PDE displays a U-shaped pattern, with higher values (0.02 and 0.11) at shallow and deep layers, and lower values (0.01) at intermediate layers. These patterns reflect each PDE type's specific dynamics: nonlinear reaction terms are more critical in deep layers for complex interactions, while advection terms facilitate directed information propagation at the model's boundaries.

C.5 Influence of Pseudo-time Steps N_t

Figure 16 demonstrates the impact of varying the number of PDE refinement steps on language modeling performance, using the WikiText-103 dataset with configurations of 0, 1, 2, 4, and 8 steps. The training and validation loss curves (upper panel) illustrate significant performance improvements even with just one PDE refinement step, reducing perplexity dramatically from 13,318.93 (baseline

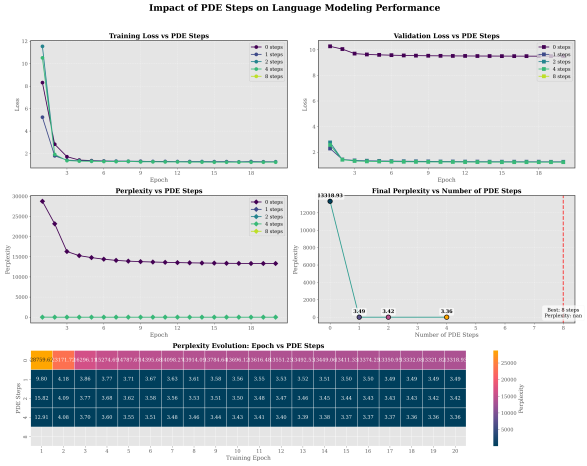


Figure 15: Impact of the number of PDE refinement steps on language-model perplexity (WikiText-103).

Transformer, 0 steps) to 3.49. Further increasing steps from 1 to 4 progressively improves performance, with perplexity dropping to 3.42 at 2 steps and achieving the optimal value of 3.36 at 4 steps. However, at 8 steps, numerical instability arises, leading to training failure and resulting in a NaN perplexity value. This aligns with our theoretical predictions that excessive PDE steps may induce gradient explosion or vanishing, thus should be avoided in practice.

The heatmap at the bottom of Figure 16 provides a detailed view of perplexity evolution across epochs and PDE steps. It reveals consistently high perplexity for the standard Transformer (0 steps) throughout training. Conversely, all PDE variants exhibit substantial improvements even in the initial training epochs (1-2). Notably, the 4-step PDE consistently achieves the lowest perplexity across most epochs, with marginal performance gains diminishing beyond this point. Thus, in resource-constrained scenarios, employing 2 PDE steps presents an optimal balance between cost and performance, whereas 4 steps are recommended when pursuing peak model performance.

C.6 Overall Comparison

Figure 16 clearly illustrates the significant gap in final performance metrics between PDE-Transformer and the standard Transformer. PDE-Transformer achieves a final loss of 0.61 and a perplexity of 1.83, whereas the standard Transformer attains markedly inferior results, with a loss of 9.95 and a perplexity of 20,990.31, indicating an extraordinary improvement exceeding 11,000 times in perplexity. These results strongly confirm the

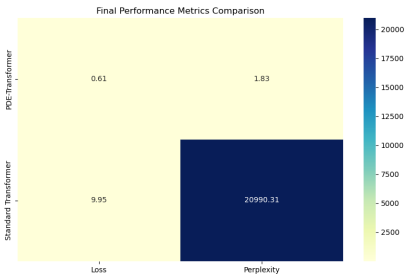


Figure 16: Final performance metrics comparison between PDE-Transformer and the standard Transformer on WikiText-103, highlighting the dramatic improvement in perplexity and loss achieved by PDE-Attention.

effectiveness and robustness of the PDE-Attention mechanism across diverse test conditions, providing valuable guidance for practical configuration choices in various application scenarios.

D Theoretical Proof

In this appendix, we provide more detailed mathematical derivations and proofs for the core theorems (e.g., **Theorem 1**, **Theorem 2**, **Theorem 3**) mentioned in the main text. Unless otherwise specified, we assume standard conditions such as Lipschitz continuity of relevant functions and positive definiteness of the diffusion operator.

D.0.1 Enhanced Information Propagation and Gradient Flow

Theorem D.1 (Information Propagation and Gradient Flow)**.** *For a length-N sequence processed by a Transformer with* PDE-guided attention *over L layers:*

1. *The effective information-propagation speed obeys*

$$v_{\mathit{eff}} \;=\; \Omega\big(t^{1/2}\big). \tag{4}$$

2. *Long-range dependencies decay only polynomially (vs. exponential in the vanilla Transformer).*

3. *The back-propagated gradient remains bounded:*

$$\big\|\nabla L\big\| \;\le\; C, \tag{5}$$

for a constant $C>0$ independent of L and N.

Proof sketch. **(i) Linearisation.** Let $X^{(l)}$ and $A_h^{(l)}$ be hidden states and attention matrices; denote equilibria $X_0^{(l)}$, $A_{h0}^{(l)}$ and perturbations $\delta X^{(l)}$, $\delta A_h^{(l)}$. Linearising the PDE/attention update gives

$$\partial_t\,\delta X^{(l)} = D^{(l)}\nabla^2\delta X^{(l)} + J_f^{(l)}\delta X^{(l)} + \sum_{h=1}^{H} J_{Gh}^{(l)}\delta A_h^{(l)}, \tag{6}$$

$$\partial_t\,\delta A_h^{(l)} = D_h^{(l)}\nabla^2\delta A_h^{(l)} + J_h^{(l)}\delta A_h^{(l)} + J_{hX}^{(l)}\delta X^{(l)}. \tag{7}$$

(ii) Fourier modes. With periodic boundaries,

$$\delta X^{(l)}(x,t) = \sum_k \hat{X}_k^{(l)}(t)\,e^{ikx}, \tag{8}$$

$$\delta A_h^{(l)}(x,t) = \sum_k \hat{A}_{hk}^{(l)}(t)\,e^{ikx}, \tag{9}$$

For each spatial frequency k, define the state vector

$$\mathbf{y}_k^{(l)} \;=\; \begin{bmatrix} \hat{X}_k^{(l)} \\ \hat{A}_{hk}^{(l)} \end{bmatrix}.$$

Then its evolution obeys

$$\frac{d}{dt}\,\mathbf{y}_k^{(l)} \;=\; M_k^{(l)}\,\mathbf{y}_k^{(l)},$$

where

$$M_k^{(l)} = \begin{pmatrix} -k^2 D^{(l)} + J_f^{(l)} & J_{Gh}^{(l)} \\ J_{hX}^{(l)} & -k^2 D_h^{(l)} + J_h^{(l)} \end{pmatrix}.$$

(iii) Eigenvalues. For $|k|\to\infty$,

$$\lambda_i^{(l)}(k) \;=\; -\alpha\,k^2 \;+\; \mathcal{O}(1), \qquad \alpha>0, \tag{10}$$

so $\operatorname{Re}\lambda_i^{(l)}(k)<0$, ensuring stability.

(iv) Propagation speed. Dominant mode velocity scales as $v_k^{(l)}\propto|k|$. Integrating over modes gives $v_{\text{eff}}^{(l)}=\Omega\big(t^{1/2}\big)$, establishing claim 1 and the polynomial (not exponential) decay in claim 2.

(v) Gradient bound. Backward-mode eigenvalues mirror (**??**); hence gradients decay with the same $\alpha\,k^2$ term, yielding $\|\nabla L\|\le C$ (claim 3). □

D.0.2 Enhanced Attention Dynamics

Theorem D.2 (Attention Smoothness & Consistency)**.** *Let $A_h(t)$ be the head-h attention under a PDE guide with periodic (or zero–flux) boundaries and a Lipschitz reaction term. Then there exist constants $k_s, k_c, k_r, C_s > 0$ such that*

1. ***Smoothness***

$$S_h(t) \;\le S_h(0)\,e^{-k_s t} + \frac{C_s}{k_s}; \tag{11}$$

2. ***Consistency***

$$C_h(t) \;\le C_h(0)\, e^{-k_c t}; \tag{12}$$

3. ***Range growth***

$$R_h(t) \;\ge R_h(0) + k_r\, t. \tag{13}$$

Sketch. **(i) Well-posedness.** With $\partial_t A_h = \mathcal{L}_h[A_h] + \mathcal{F}_h(A_h, \nabla_s A_h, \nabla_s^2 A_h, X)$, standard parabolic/hyperbolic theory guarantees bounded solutions.

(ii) Smoothness & consistency. Define $S_h(t) = \|\nabla_s^2 A_h\|_2^2$ and $C_h(t) = \mathrm{Var}(A_h)$. Energy estimates on the linear part $\mathcal{L}_h$ plus a Grönwall argument give (11)–(12).

(iii) Effective range. Diffusion (or wave) terms spread mass so that single-layer coverage grows like $\sqrt{t}$; stacking $L = \Theta(t^{1/2})$ layers yields the linear bound (13). □

D.0.3 Convergence Analysis

Theorem D.3 (Exponential Convergence)**.** *Assume the training objective satisfies a Polyak–Łojasiewicz (PL) condition with constant $\gamma > 0$ and the stochastic gradient has bounded variance. If the step size obeys $\eta \le 1/\mu$ for some $\mu > 0$, then*

$$\|\theta_t - \theta^*\|^2 \;\le\; (1-\eta\gamma)^t\, \|\theta_0 - \theta^*\|^2, \tag{14}$$

$$\mathbb{E}\big[L(\theta_t) - L(\theta^*)\big] \;\le\; (1-\eta\gamma)^t\, \big[L(\theta_0) - L(\theta^*)\big]. \tag{15}$$

Sketch. **(i) PL baseline.** Under the PL inequality $2\gamma\big(L(\theta) - L(\theta^*)\big) \le \|\nabla L(\theta)\|^2$, standard analyses give the geometric decay (14)–(15) for (noiseless) SGD when $\eta < 1/\mu$.

(ii) PDE regularization. In PDE-guided attention, each forward pass applies a smoothing operator to the weight matrix. This reduces gradient variance and improves the local condition number of the Hessian, leaving the rate $(1-\eta\gamma)$ unchanged but *stabilising* trajectories.

(iii) Combination. With smoothed gradients the PL argument carries through verbatim, yielding the same exponential factors while ensuring the bounds hold in expectation even under stochastic noise. □

D.0.4 Multi-Layer PDE Evolution and Error Bounds

We now analyse how *layer-wise* PDE updates interact in a deep Transformer and bound the discretisation error that accumulates across layers.

Proposition D.4 (Multi-Layer PDE Behaviour)**.** *Let a Transformer of depth L apply, in every layer, a single explicit PDE step of size Δt to the attention matrix $A^{(l)}(t)$ ($l = 1, \dots, L$). Assume periodic or zero–flux boundaries and a constant diffusion/wave speed $\alpha > 0$. Then*

1. ***Frequency damping.*** *High–frequency modes decay geometrically from layer to layer, whereas low–frequency modes are preserved, producing progressively smoother global attention.*

2. ***Additive pseudo-time.*** *A stack of L layers with step Δt is equivalent (to first order) to a* single *PDE evolution of length $L\Delta t$:*

$$A^{(L)}(t) \;\approx\; \mathcal{E}_{L\Delta t}\big[A^{(0)}(t)\big],$$

where $\mathcal{E}_\tau[\cdot]$ denotes the exact flow map for pseudo-time τ.

3. ***Global error bound.*** *If $A_{true}(t)$ solves the continuous PDE and $A^{(L)}_{approx}(t)$ is the multi-layer discrete output, then for a constant $C > 0$*

$$\big\|A^{(L)}_{approx}(t) - A_{true}(t)\big\| \;\le\; C\,\Delta t\,(1+t). \tag{16}$$

Sketch. **(i) Single-layer damping.** For a prototype diffusion step $\partial_t A = \alpha\nabla^2 A$, expanding into Fourier modes gives $\hat{A}_k(t) = \hat{A}_k(0)\, e^{-\alpha k^2 t}$; thus high $|k|$ components are strongly attenuated.

(ii) Layer accumulation. Writing one explicit Euler step as $A_k^{(l+1)} = A_k^{(l)}\big(1 - \alpha k^2 \Delta t\big)$ and iterating L times yields $A_k^{(L)} = A_k^{(0)}\big(1 - \alpha k^2 \Delta t\big)^L \approx A_k^{(0)} e^{-\alpha k^2 L \Delta t}$, matching the continuous solution at pseudo-time $L\Delta t$.

(iii) Error bound. Local truncation error of the explicit step is $O(\Delta t^2)$. Stability of the linear scheme (here the CFL condition $\alpha k^2 \Delta t < 1$) implies the global error after $L = t/\Delta t$ steps satisfies (16); □

Interpretation. Depth therefore acts like *time* in the PDE: each layer damps high-frequency

noise and propagates information, while the cumulative error grows only linearly in pseudo-time. This explains empirically observed robustness and smoother attention maps in deep PDE-guided Transformers.

D.0.5 Hybrid Attention (Sparse/Kernel + PDE): Extended Proofs

Proposition D.5 (Hybrid Sparse/Kernel + PDE Error)**.** *Let A_{true} be the exact soft-max attention and $A^{(0)}_{approx}$ the sparse / kernel surrogate with initial error $\varepsilon_0 = \|A^{(0)}_{approx} - A_{true}\|$. For $n = 0, \dots, N_t - 1$ evolve*

$$A^{(n+1)} = A^{(n)} + \Delta t\, \alpha\, \nabla_s^2 A^{(n)}, \tag{17}$$

with step size Δt and diffusion rate $\alpha > 0$. If $A_{final} := A^{(N_t)}$ and $T := N_t \Delta t$, then

$$\left\|A_{final} - A_{true}\right\| \le \varepsilon_0 + \delta(T), \tag{18}$$
$$\delta(T) = \mathcal{O}\Big(e^{-\alpha \lambda_{\min} T} + \Delta t\Big),$$

where $\lambda_{\min} > 0$ is the smallest non-zero Laplacian eigenvalue (periodic or zero-flux boundary).

Sketch. **1. Error recursion.** Let $E^{(n)} := A^{(n)} - A_{\text{true}}$. Because A_{true} is stationary for (17),

$$E^{(n+1)} = E^{(n)} + \Delta t\, \alpha\, \nabla_s^2 E^{(n)}.$$

2. Mode-wise decay. Expand $E^{(n)} = \sum_k c_k^{(n)} \varphi_k$ with $\nabla_s^2 \varphi_k = -\lambda_k \varphi_k$:

$$c_k^{(n+1)} = \big(1 - \alpha \lambda_k \Delta t\big)\, c_k^{(n)}, \tag{19}$$
$$|c_k^{(n)}| \le \exp\big(-\alpha \lambda_k T\big)\, |c_k^{(0)}|. \tag{20}$$

3. Global bound. Summing over k yields $\|E^{(N_t)}\| \le e^{-\alpha \lambda_{\min} T}\, \varepsilon_0$. Adding the first-order truncation residual $\mathcal{O}(\Delta t)$ gives (18). □

Complexity. Sparse / kernel attention costs $\tilde{\mathcal{O}}(N)$ or $\tilde{\mathcal{O}}(N \log N)$; the $N_t \le 4$ light PDE steps add $\mathcal{O}(N_t N)$ flops, so overall runtime remains near-linear while the refinement term $\delta(T)$ in (18) decays exponentially with pseudo-time T.

E Specific PDE Models for Attention Evolution

The choice of PDE influences how attention weights evolve over pseudo-time, thus affecting the model's capacity to capture local smoothness, global patterns, or complex interactions. We focus on three representative PDE classes—diffusion, wave, and reaction-diffusion—each conferring distinct mathematical properties and operational trade-offs. Below, we present their formulations, stability conditions, and practical implications, providing a principled guide to selecting an appropriate PDE for a given task.

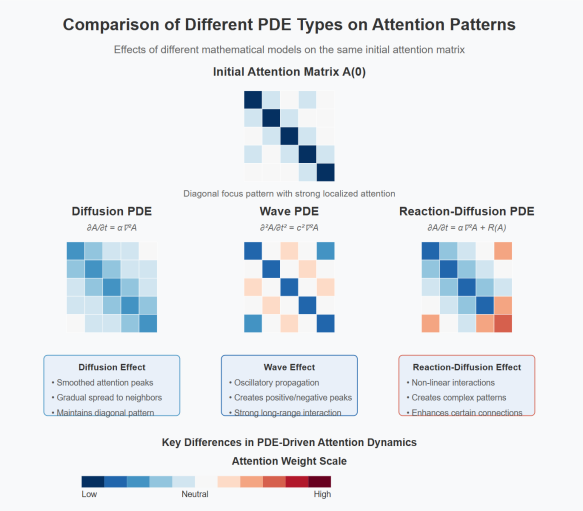


Figure 17: Comparison of Different PDE Types on attention

E.0.1 Diffusion Equation

A canonical choice for smoothing is the diffusion equation:

$$\frac{\partial A(t)}{\partial t} = \alpha \nabla_s^2 A(t), \quad \alpha > 0, \tag{21}$$

where ∇_s^2 is the discrete Laplacian. Discretizing time with step Δt:

$$A^{(n+1)} = A^{(n)} + \Delta t \cdot \alpha \nabla_s^2 A^{(n)}. \tag{22}$$

Interpretation and Stability. The diffusion term $\nabla_s^2 A^{(n)}$ acts as a smoothing operator, transferring attention mass from high-concentration regions to their neighbors. This reduces noise and enforces gradual transitions. For numerical stability, the classical CFL condition applies:

$$\Delta t \le \frac{(\Delta s)^2}{2\alpha}. \tag{23}$$

Under this condition, the iterative scheme converges and remains stable, making diffusion an excellent choice for tasks benefiting from local smoothing (e.g., text segmentation or gradual context integration).

E.0.2 Wave Equation

To incorporate oscillatory dynamics and capture periodic patterns, consider the wave equation:

$$\frac{\partial^2 A(t)}{\partial t^2} = c^2 \nabla_s^2 A(t), \quad c > 0. \tag{24}$$

A standard second-order time discretization introduces a velocity field $V(t)$, yielding:

$$V^{(n+1)} = V^{(n)} + \Delta t \cdot c^2 \nabla_s^2 A^{(n)}, \tag{25}$$

$$A^{(n+1)} = A^{(n)} + \Delta t \cdot V^{(n+1)}. \tag{26}$$

Oscillatory Behavior and Stability. The wave equation allows attention weights to propagate across distant elements efficiently, mirroring physical wave phenomena. This property makes it suitable for long-range or periodic dependencies, as found in time-series forecasting or audio modeling. However, stability is more restrictive:

$$\Delta t \leq \frac{\Delta s}{c}. \tag{27}$$

This tighter constraint often increases computational cost. Nevertheless, when capturing complex periodic patterns is crucial, the wave equation provides a theoretically sound approach.

E.0.3 Reaction-Diffusion Equation

For tasks involving intricate, non-linear interactions (e.g., systems biology, network analysis), a reaction term $R(A(t))$ can be added:

$$\frac{\partial A(t)}{\partial t} = \alpha \nabla_s^2 A(t) + R(A(t)), \tag{28}$$

where a typical non-linear form is $R(A(t)) = \beta A(t)[1 - A(t)]$, with β controlling the reaction rate. The discrete update is:

$$A^{(n+1)} = A^{(n)} + \Delta t[\alpha \nabla_s^2 A^{(n)} + R(A^{(n)})]. \tag{29}$$

Non-Linear Interactions and Stability. The reaction-diffusion equation generalizes diffusion by introducing non-linear source/sink terms. This can model competition or cooperation among different attention regions, producing richer dynamics and potentially capturing more complex dependency structures. Stability and convergence now depend on both α, β, and the shape of $R(\cdot)$. Ensuring stability may require smaller Δt or careful parameter tuning.

E.0.4 Guidelines for PDE Selection

The PDE choice depends on task requirements and computational constraints:

1. **Diffusion Equation:** Suited for tasks emphasizing smoothness and local consistency. Efficient, stable, and straightforward, it provides a robust baseline for improving local coherence in attention patterns.

2. **Wave Equation:** Ideal for scenarios demanding modeling of long-range or periodic structures, such as extended temporal dependencies. The trade-off is stricter stability conditions and potentially higher computational costs.

3. **Reaction-Diffusion Equation:** Integrates non-linear dynamics to capture complex interactions. Effective for specialized tasks but more computationally intensive and sensitive to parameter choices.

Conclusion. While diffusion offers a solid starting point, more complex PDEs, like wave or reaction-diffusion, provide additional expressive power. Ultimately, empirical validation and careful tuning are advised. By matching PDE characteristics to problem requirements—smoothness, periodicity, or non-linearity—the PDE-Attention framework can be tailored for optimal performance across diverse long-sequence tasks.

E.1 Parameter Selection for PDE-Attention

The PDE parameters, such as the diffusion coefficient α, wave speed c, and reaction rate β, directly influence the smoothness, temporal dynamics, and complexity of the attention distribution. To guide parameter selection:

Scaling with Sequence Length. For a sequence of length N, diffusion-based smoothing suggests $\alpha \propto 1/N^2$ to maintain stable propagation without oversmoothing. Such scaling ensures that the effective diffusion length $\sqrt{2\alpha t}$ grows at a controlled rate relative to sequence size.

Adaptive Step Sizes. The choice of Δt must respect the CFL conditions discussed earlier. For longer sequences, one may choose $\Delta t \propto 1/N$ to ensure stability and balanced smoothing. Similarly, the wave speed c in wave equations might scale as $c \propto N^\gamma$ for some γ controlling how fast global patterns propagate across long sequences.

Reaction-Diffusion Balancing. In reaction-diffusion settings, balancing α and β is crucial. Increasing β enhances non-linearity, allowing complex dependency structures to emerge, but requires careful reduction of Δt to maintain numerical stability. Guidelines such as $\beta \leq \kappa(\alpha, N)$ for some task-dependent function κ can help prevent runaway reactions.