Defierithos: The Lonely Warrior Rises from Resonance

Libo Wang
UCSI University
Nicolaus Copernicus University
free.equality.anyone@gmail.com

Abstract

In view of the long-term computational complexity of the Transformer architecture in long sequence processing, as well as the redundancy and long-distance dependency attenuation caused by global self-attention, this work proposes a new architecture design named "Defierithos" that is based on self-resonance field (SRF). It replaces the traditional self-attention mechanism with partial resonant interference, uses spectrum modulation and phase superposition for information transmission, and effectively reduces the computational burden. To verify the effectiveness, the researcher used simulation-based training to build a training environment in a Python simulator based on custom GPTs to complete the execution of text generation (synthetic data), code programming (HumanEval), and mathematical reasoning (GSM8k). The experimental results show that Defierithos is superior to the GPTs version of Transformer in six indicators: ROUGE-L, METEOR, Pass@k, MMLU, accuracy, and ARC-AGI. It proves that as an architecture system that replaces tokens with waves instead of traditional self-attention mechanisms, it provides experience that can be used as a reference for the architecture of the next generation of natural language processing.

1 Background

As transformer-based variants have made significant progress in hardware adaptation and computational optimization, attention has been increasingly focused on the optimization of key dimensions such as sequence length, inference capability, illusion, and computational resource redundancy [8, 13, 20, 33, 14]. Although researchers have tried to alleviate the $O(n^2)$ computational overhead that plagues the self-attention mechanism through sparse attention, etc., most of the time they can only reduce part of the computational cost under certain conditions [6, 27]. The gap is reflected in the fact that the exponential growth relationship between sequence length and computational burden has not been solved from the architectural level, which means that if the input text exceeds a certain length, the consumption of computing resources will still increase exponentially. Because the transformer's information processing model is still fundamentally restricted by tokenization, which divides natural language into discrete units, it cannot effectively depict the continuous changes and hierarchical progression of semantics [32, 28].

In addition, the transformer architecture performs poorly on memory and context due to its fixed-length context window in large language models (LLMs) and difficulty in retrieving memory across dialogues [24]. Even if Mamba tries to use variations of the state space model, it is still difficult to achieve low-cost modeling of ultra-long sequences and dynamic reorganization of instant information [12]. In this regard, this work abandons the self-attention mechanism and tokenization and proposes a dynamic architecture "Defierithos" based on self-resonance field (SRF). It replaces traditional attention matrix computation through wave interference and phase superposition, fundamentally changing the information processing paradigm of LLMs.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

2 Defierithos Architecture

The application of self-resonance field (SRF) allows the Defierithos architecture to replace the self-attention mechanism in principle, which forms semantic associations by forming mutually interfering "waves" of originally independent semantics. It profoundly changes the limitations of computational redundancy and information processing methods caused by the principle of relying on point-by-point similarity calculation (Figure 1).

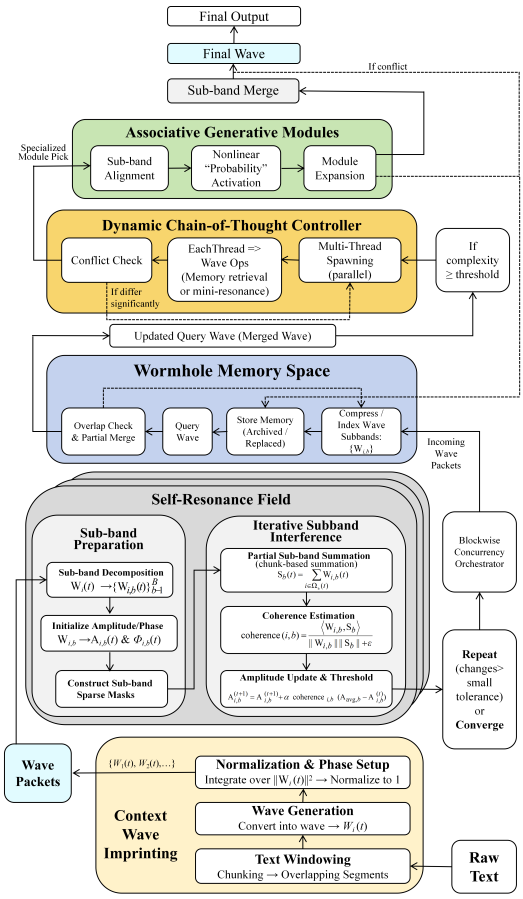

Figure 1: The Defierithos Architecture

Given that the core of the transformer architecture's self-attention mechanism relies on query-key similarity calculations to determine semantic associations, it is essentially a discrete matrix operation [16, 38, 3, 27]. In principle, the need to perform a global search on all input units is the reason why the mechanism has a high computational cost of $O(n^2)$ [31]. Although it has recently improved on the trend of partial search such as sparse attention, SRF constructs a continuous wave propagation structure that achieves semantic relevance matching through spectral modulation and phase interference without performing discrete weighted queries [4]. Using a figurative metaphor, SRF is like a lake, where the information is no longer independent drops of water, but forms ripples that interfere with each other. It enables information to resonate adaptively in a multidimensional spectrum, thus forming a dynamic reasoning sequence. It is precisely because SRF uses a wave model rather than a vector matrix, semantic representations can evolve adaptively in high-dimensional space through resonant frequencies, and no longer rely on the accumulated attention weights.

The introduction of the wave to replace the token not only reduces redundant calculations, but also enables the spontaneous formation of long-distance semantic associations under low computational conditions. It allows the Defierithos to maintain efficient information transmission while avoiding the context loss problem caused by attention decay in transformer processing of long texts [19, 23].

2.1 Waveform Imprint Representation

The concept of wave originates from the idea of continuous dynamic transmission in information processing in the super brain [5, 30, 22]. In essence, it matches information through multi-spectral synchronous oscillation rather than based on discrete index query. In the super brain, information propagation is not carried out in a single fixed weighted manner, but through phase resonance to form dynamic coupling so that the input signal adjusts its own excitation state according to the frequency response to achieve the best match [2, 5, 10]. Based on this principle, the Defierithos architecture applies the waveform resonance field to semantic reasoning, so that language modeling no longer relies on tokenization and self-attention mechanisms. The researcher explains in Figure 2 how a wave-principle-inspired simulation allows information to adaptively seek resonance points in a high-dimensional spectral interference network, thereby establishing the principle of semantic association.

The context wave imprinting is responsible for converting the original text into a continuous waveform representation, which enables language reasoning to be matched in the SRF in a wave interference manner. The text windowing chunks the input discrete text with overlapping segments, which ensures that the context information continues to affect subsequent wave operations over a longer semantic range. It avoids context truncation caused by fixed token length in traditional models and provides more complete context awareness for semantic matching. The text fragment enters the wave generation stage, where i_t is converted into corresponding wave packets through mathematical mapping to the function $W_i(t)$. The waveform parameters include semantic spectrum, amplitude and phase information. This conversion ensures that the text information is no longer represented by discrete tokens, but is transmitted in the form of continuous wave signals, so that the correlation between semantics can be reflected through wave coherence and spectral features. The normalization and phase setup is responsible for integrating all generated waveforms to ensure that the total energy of the overall waveform is always 1 to avoid fluctuation amplification or attenuation affecting the semantic matching accuracy. The algorithm formula is as follows:

$$\int_0^T \|W_i(t)\|^2 dt \to 1$$

In addition, the module will also adaptively adjust the spectral weights of different bands according to context changes, so that it can maintain stable semantic interference in the subsequent process.

2.2 Self-Resonance Field

The human brain does not access memory through discrete querying when processing information, but relies on neural oscillation and harmonic resonance to perform partial matching on multi-frequency signals. The Defierithos system is based on this principle, introducing a self-resonance field (SRF) to replace self-attention to achieve dynamics through wave interference. In this architecture, the input wave packets first pass through sub-band decomposition, decomposing the partial wave into different frequency components $\mathrm{W}_i(t) \to \mathrm{W}_{i,b}(t)_{b=1}^{B}$ and mapping them to the sub-band spectrum. Then, the initial amplitude $\mathrm{A}_{i,b}$ and phase $\Phi_{i,b}$ are set through initialize amplitude & phase, and construct sub-band sparse masks to suppress irrelevant signals.

Entering iterative subband interference, it first establishes the interference field by summing up partial sub-bands. However, the SRF's resonance field used is different from the global resonance field of the human brain. The core is partial resonance, it means that information resonates only within the relevant range. Just like the previous analogy, a drop of water falling into the lake only forms ripples in the relevant area, rather than the entire lake vibrating uniformly. Borrowing from but different from the sparsity principle, SRF avoids the high computational overhead of large-scale global querying of self-attention or even sparse attention through local spectral matching. The resonance range r_{res} is controlled by the spectral coherence threshold Θ_{coh}, and adaptive gain adjustment is used to ensure that the amplification or attenuation of the fluctuation is consistent with the information relevance to reduce redundant calculations and maintain efficient flow of semantic information. Its core algorithm is expressed as:

$$\mathrm{S}_b^{(k)}(t) = \sum_{i \in \Omega_b^{(k)}} \mathrm{W}_{i,b}^{(k)}(t)$$

where b represents sub-band index, $b \in \{1, 2, \cdots, B\}$; B represents the total number of sub-bands; k represents the spectrum index; Ω is set of waves active in band b at iteration k.

Next, the coherence estimation is performed to evaluate the resonance correlation between different sub-bands using the following coherence function:

$$\text{coherence}_{i,b}^{(k)} = \frac{\langle \mathrm{W}_{i,b}^{(k)}(t), \mathrm{S}_{i,b}^{(k)}(t) \rangle}{\|\mathrm{W}_{i,b}^{(k)}(t)\| \|\mathrm{S}_{i,b}^{(k)}(t)\| + \varepsilon}$$

where ε a small constant to avoid division by zero.

After that, when the resonance threshold is exceeded, the amplitude update and threshold are triggered, so that the fluctuation gradually converges to a stable state, otherwise it continues to iterate.

$$\mathrm{A}_{i,b}^{(k+1)}(t) = \mathrm{A}_{i,b}^{(k)}(t) + \alpha \, \text{coherence}_{i,b}^{k} [\mathrm{A}_{\text{avg},b} - \mathrm{A}_{i,b}^{(k)}(t)], \text{else set } \mathrm{A}_{i,b}^{(k+1)}(t) = 0$$

$$\mathrm{A}_{i,b}^{t+1} = \mathrm{A}_{i,b}^{t} + \lambda(\text{coherence}_{i,j} - \Theta)$$

To address the waveform order misalignment and logical inconsistency caused by parallel processing, the researcher constructed a blockwise concurrency orchestrator to block the waveform after self-resonance, and then form small parallel groups for each band according to the position to perform coherence correction. The following algorithm obtains the coherence index as the weight by calculating the inner product between the amplitudes of any two bands in the same block, which means that the amplitude difference is weighted back to the original band to form a partial resonance amplitude correction.

$$W_{i,b}^{(k+1)} = W_{i,b}^{k} + \alpha \sum_{i',b'} \delta_{\text{block}}((i,b),(i',b')) \Gamma(W_{i,b}^{(k)}, W_{i',b'}^{(k)})$$

where Γ represents coherence-oriented gain term; δ_{block} is indicator function.

SRF uses parallel computing based on local spectral matching so that information only resonates in relevant areas (Figure 2). When the computational complexity reaches the threshold k, the module triggers multi-thread spawning to divide the waveform calculation into multiple threads (such as Threads A, B, C). Each thread performs partial wave operations and makes corresponding adjustments based on the range of regional resonance. After the calculation is completed, each thread unifies the waveform storage through memory merging and then transmits it to the blockwise concurrency orchestrator. After performing conflict checking, the final wave set is output to the wormhole memory space.

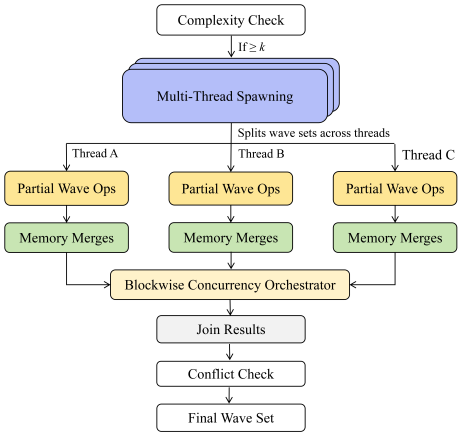

Figure 2: Parallel computation of self-resonance fields

2.3 Wormhole Memory Space

Wormhole memory space responsible for information storage and query tasks adopts high-dimensional space matching for optimizing transformer architecture and realizes dynamic memory management

through cross-dialogue retrieval. However, since the operation process of the Defierithos strictly follows the incoming wave packets processed by SRF, the researchers modified the original wormhole memory space to suit this work. First, the module compresses/indexes wave sub-bands, and stores waveform information in a structured manner by compressing and indexing waveform sub-bands $\{W_{i,b}\}$ to efficiently query memory and reduce redundancy requirements.

$$M \leftarrow \{\text{compressed}(W_{i,b})\}$$

In the store memory stage, it will decide whether to archive or replace the previous storage content based on the storage strategy to ensure the dynamic adaptability and information traceability of the storage system. When the query wave is started, the core of the process is to achieve accurate retrieval through waveform feature matching, call the most relevant storage items and perform local updates. The researcher then demonstrated the overlap check and partial merge process. It is used to partially merge the storage results to ensure that the storage items have sufficient semantic consistency while avoiding information loss. In this process, the algorithm will determine the merge strategy based on the set coherence threshold to ensure that the final output of the updated query wave meets the calculation accuracy requirements.

$$\text{overlap}(Q_b, M_{k,b}) = \frac{\langle Q_b, M_{k,b} \rangle}{\| Q_b \| \| M_{k,b} \| + \varepsilon} \tag{1}$$

where Q_b represents original query wave before merging; k is the index, $M_{k,b}$ is individual stored memory wave in subband b; ε is small numerical constant added to prevent division by zero.

Compared to the transformer memory module, which relies on fixed position indexing and static weight tetrieval, that is, memory can only be obtained through predefined key-value mapping, which is reflected in the difficulty of current LLMs to perform detailed dynamic adaptation according to the context [18, 35]. The wormhole memory space allows the memory module to dynamically adjust the spectrum through waveform resonance and match the context changes to achieve cross-dialogue, thereby avoiding the limitation of memory retrieval to fixed vector queries.

2.4 Dynamic Chain-of-Thought Controller

Dynamic chain-of-thought follows parallel processing and is used to dynamically adjust the matching of waveform data during the calculation process, thereby ensuring efficient waveform memory retrieval and reasoning generated by SRF. The D-CoT controller receives the merged wave from the previous module and first performs a complexity check. The algorithm is as follows:

$$\text{Complexity}(\{Z_i\}) = \sum_t \sum_b \| Z_{i,b} \|^2$$

where complexity $(\{Z_i\})$ represents complexity score of the save set; t is time index, b is sub-band index.

Notably, if the computational requirements reach a threshold (complexity $\geqslant$ threshold), the multi-thread spawning phase is entered, where the waveform operation is split into multiple threads for parallel processing. As described above, each thread performs wave operations based on wormhole memory retrieval and local range resonance to correspond to semantic matching, context association, and feature alignment. During the computation process, the module performs conflict checks to ensure consistent results after parallel processing.

$$\text{Conflict} = \sum_{l \in l'} \text{Dist}\left(\{Z_{i,b}^{(l)}\}, \{Z_{i,b}^{(l')}\}\right)$$

where $Z_{i,b}$ represents wave sub-band component; l is processing layer index; l' is previous layer layer.

If the results of each thread differ significantly, return to multi-thread spawning to readjust the calculation. On the contrary, if the results converge, an update and merged query wave is generated and then passed to subband alignment for further semantic expansion.

2.5 Associative Generative Modules

The associative generative modules, as the core enhancement layer of the Defierithos architecture, perform modular feature alignment and nonlinear excitation on the waves that have passed the D-CoT controller screening, and finally generate the output waveform through module expansion. In the module selection stage, the specialized module pick matches the appropriate associative modules to the specific task requirements based on the characteristics of the waveform spectrum. On this basis, the sub-band alignment calculation $\text{Align}_{mod,b}(i)$ calculates the matching degree between $W_{i,b}(t)$ that is the time function of wavei in b sub-bands and the module memory matrix $M_{mod,b}(t)$ through b sub-bands to ensure compatibility with the existing learning information.

$$\text{Align}_{\text{mod},b}(i) = \frac{\int_0^{T_b} \left[\text{W}_{i,b}(t) \otimes \text{M}_{\text{mod},b}(t)\right] dt}{\sqrt{\int_0^{T_b} \parallel \text{W}_{i,b}(t) \parallel^2 dt}\sqrt{\int_0^{T_b} \parallel \text{M}_{\text{mod},b}(t) \parallel^2 dt + \varepsilon}}$$

The following nonlinear probability activation function determines whether the waveform should be triggered for expansion. If the value exceeds the threshold $\gamma_{mod,b}$, the sub-band enters the expansion phase.

$$\Xi_{\text{mod},b}(\text{W}_{i,b}) = \sigma(\alpha\ \text{Align}_{\text{mod},b}(i)) \tag{2}$$

where $\Xi_{\text{mod},b}(\text{W}_{i,b})$ is the triggering probability of the input waveform $W_{i,b}$ in the mod module; σ represents the nonlinear activation function; $\sigma(x) = 1/(1 + e - x)$; α represents the scaling factor that adjusts the effect of matching on activation.

In the expansion phase, the module will apply adaptive transformations to the selected Wi,b, and complete or generate information based on the historical patterns of the existing memory matrix to ensure the coherence and contextual consistency of the output results. This process dynamically adjusts the phase and amplitude of the wave to avoid semantic drift caused by sub-band anomalies. Finally, the expanded waveform is passed to the sub-band merge for final fusion. If a conflict exceeding the threshold is detected during the fusion process, it will backtrack to the wormhole nemory space for storage and update; if the conflict is tolerable, the wave enters the final output stage.

$$\text{W}_{\text{final},b} = \sum_k \text{W}_{k,b} \tag{3}$$

In contrast, the transformer's feedforward network (FFN), associative generative modules ensure that the input waveform finds the best match in the spectral domain through sub-band alignment (Vaswani et al., 2017). The nonlinear "probability" activation makes the expansion of specific frequency bands more adaptive through probability-driven variable weight adjustment. FFN relies on the calculation and mapping of fixed weights, which makes it difficult to flexibly adapt to input changes. In addition, FFN is computationally incapable of adaptively adjusting the neural structure, while module expansion allows dynamic expansion of sub-bands to generate richer reasoning patterns while saving resource utilization and reducing computational redundancy [11, 36].

3 Training

As a learning method in a simulation environment, simulation-based training is often used in the application fields of social science, medicine, and deep learning to simulate real-life scenarios and optimize decision-making strategies [21, 1, 9]. The Defierithos architecture achieves adaptive learning through a virtual environment, rather than relying on traditional physical hardware to accelerate computing. The key to choosing simulation-based training is that the Defierithos uses a self-resonance field to process semantic fluctuations and performs reasoning through a dynamic chain-of-thought. Modules are difficult to optimize simply by relying on matrix operations of GPU or TPU for gradient optimization because core operations involve spectral interference and partial matching mechanisms. If the traditional hardware acceleration training method is used, such as optimization strategies based on backpropagation and gradient descent, it is difficult to effectively fit the spectral transformation and nonlinear resonance mechanism. On the other hand, the transformer architecture can be accelerated by hardware because its self-attention mechanism only relies on numerical matrix operations and does not involve the problem of continuous wave interference.

Unlike the large-scale labeled data that the transformer relies on, the training of the Defierithos focuses on adaptive reasoning rather than large-scale sample learning. The transformer learns the correlation between words by training on massive labeled data and completes text generation through autoregression [31, 37, 17]. The Defierithos relies on dynamic fluctuation matching that simulates spectral changes in different contexts during training to learn the optimal configuration of adaptive resonance. In addition, simulation-based training provides a highly controllable simulation environment during training, and the model learns resonance rules under different contextual conditions without being restricted by a fixed dataset. Moreover, it has the ability to adapt to different contextual changes by dynamically adjusting the frequency spectrum interference conditions.

3.1 Setup

In the training environment of the Defierithos, the researcher chose a Python simulator based on custom GPTs as a training tool for its high flexibility and fine control. Although traditional deep learning frameworks such as PyTorch or TensorFlow have complete model training support, the execution of waveform resonance computation and adaptive reasoning is limited by built-in tensor operations, making it difficult to directly express the nonlinear dynamic characteristics of the architecture [25, 7]. In addition, the Defierithos requires large-scale parallel wave interference, and traditional GPU acceleration frameworks are optimized for standard matrix calculations and are difficult to efficiently simulate multi-spectral modulation environments [26]. The custom GPTs-based Python simulator is one of the few that has the ability to run code and is widely recognized by users, which allows the researcher to run the Defierithos architecture code and training code in a simulated way by building training scenarios in real time.

Importantly, the researcher also chose the Python simulator that is based on GPTs as a control group to achieve fairness. It means that the original Python simulator as the control group directly executes the synthetic text data, HumanEval and GSM8k without any training. Compared with the original transformer version proposed by Vaswani et al., the GPTs version is based on the variant GPT4-Turbo, which shows applicability in semantic parsing and code running capabilities according to currently available information [29]. Since ChatGPT has been widely accepted by academia and industry, its comparison results with defierithos are more objective and universal [15].

3.2 Dataset

The researcher selected three different datasets to train the text generation, code programming, and mathematical reasoning capabilities of the Defierithos and evaluate its performance. Since large-scale datasets such as WikiText-103 cannot be directly uploaded to the Python simulator based on custom GPTs, the researcher produced synthetic data to train the Defierithos' text generation capabilities, which ensures that it learns semantic patterns and grammatical structures under controlled conditions. For code programming, HumanEval, as a commonly used dataset, has a wide range of credibility that covers a variety of programming challenges, which helps to verify the model's program logic construction and function reasoning capabilities [39]. As for mathematical reasoning capabilities, the GSM8k dataset provides high-quality elementary school mathematics questions to effectively test the model's calculation accuracy and multi-step reasoning capabilities [34].

3.3 Implementation

The researcher first used Python 3.13 IDLE to design the architecture code of the Defierithos, which included detailed technical implementation code of each module and corresponding training code. This step ensures that the core computing logic of the architecture can run stably in the simulation environment to avoid the results being affected by program errors during the subsequent training process. Part of the architecture code and training results have been shared in the supplementary material. Then, the researcher uploaded the architecture and training code to the Python simulator based on custom GPTs to build the training environment. Subsequently, the synthetic data, HumanEval, and GSM8k datasets were uploaded to the Python simulator in sequence to train text generation, code generation, and mathematical reasoning capabilities. Next, the researcher uploaded the three sets of JSON files directly to an untrained Python simulator and observed the results of each run.

To ensure that the training results of Defierithos are universal and stable, the researcher repeatedly trained Defierithos 50 times and selected the best results from 10 of them. Similarly, the Python

simulator used as a control group also extracted the best results from 50 trainings. Because training process based on the computing characteristics of the GPTs, which is quickly run and executed through the simulator without relying on expensive hardware resources for preliminary testing. As mentioned before, since the computing mechanism of the simulator is different from the traditional GPU training environment, the researcher needs to input prompts multiple times and repeat the training to reduce external environment interference and improve data consistency.

4 Result & Discussion

Considering that GPT4-Turbo has a lot of repeated operations due to the limited computing power, such as interruptions and errors, the researcher disclosed part of the training process and the best result after 10 repeated trainings. To evaluate the performance of the Defierithos in the above tasks, table 1 shows 6 quantitative metrics (ROUGE-L, METEOR, Pass@k, MMLU, Accuracy, ARC-AGI).

Table 1: The metrics for the Defierithos architecture

ROUGE-L	METEOR	Pass@k	MMLU	Accuracy	ARC-AGI
0.741	0.489	0.857	0.861	0.972	0.218
0.794	0.471	0.860	0.872	0.967	0.207
0.774	0.475	0.859	0.865	0.973	0.217
0.746	0.458	0.873	0.869	0.975	0.212
0.787	0.478	0.866	0.868	0.970	0.211
0.781	0.476	0.861	0.869	0.973	0.221
0.759	0.457	0.855	0.867	0.968	0.211
0.752	0.490	0.879	0.864	0.974	0.209
0.754	0.489	0.855	0.862	0.971	0.224
0.744	0.466	0.864	0.867	0.972	0.220

To ensure the fairness of the experiment, Table 2 shows that the transformer-based GPTs Python simulator as a control group was run 50 times and the best 10 results were selected.

Table 2: The metrics for the Defierithos architecture

ROUGE-L	METEOR	Pass@k	MMLU	Accuracy	ARC-AGI
0.727	0.436	0.769	0.794	0.864	0.034
0.729	0.442	0.770	0.808	0.867	0.026
0.727	0.444	0.765	0.805	0.871	0.030
0.728	0.455	0.771	0.810	0.885	0.025
0.725	0.440	0.778	0.818	0.872	0.027
0.731	0.453	0.784	0.819	0.882	0.027
0.730	0.440	0.791	0.794	0.866	0.031
0.728	0.449	0.775	0.805	0.869	0.024
0.734	0.441	0.786	0.813	0.891	0.039
0.735	0.436	0.789	0.818	0.887	0.029

To reflect the objectivity, stability, and universality of the training results, the researcher used the line graphs in Figure 3 to compare the changing trends of each metrics of Defierithos and Transformer.

The data results show that the Defierithos has an average value of 0.763 on ROUGE-L, which is higher than the Transformer's 0.729. However, its fluctuation range is relatively large, reaching a maximum of 0.794, reflecting its strong adaptability to different text types. In terms of METEOR, the Defierithos has an average value of 0.475, which is better than the Transformer's 0.444. This shows that the former is more accurate in semantic alignment and vocabulary matching, and fluctuates between 0.457 and 0.490. The Defierithos' Pass@k average value is 0.863, which is better than the Transformer's 0.778 in terms of the correctness of the generated code function, and has a smaller fluctuation range. MMLU shows that the Defierithos has an average value of 0.866, which has better multilingual understanding capabilities than the Transformer's 0.808. And the former maintains a stable performance between 0.861 and 0.872, reducing the risk of unstable performance of mathematical reasoning in different cycles. In terms of accuracy, the Defierithos significantly surpasses the Transformer's 0.875 with an average value of 0.972, and the fluctuation is very small. The ARC-AGI results show that the

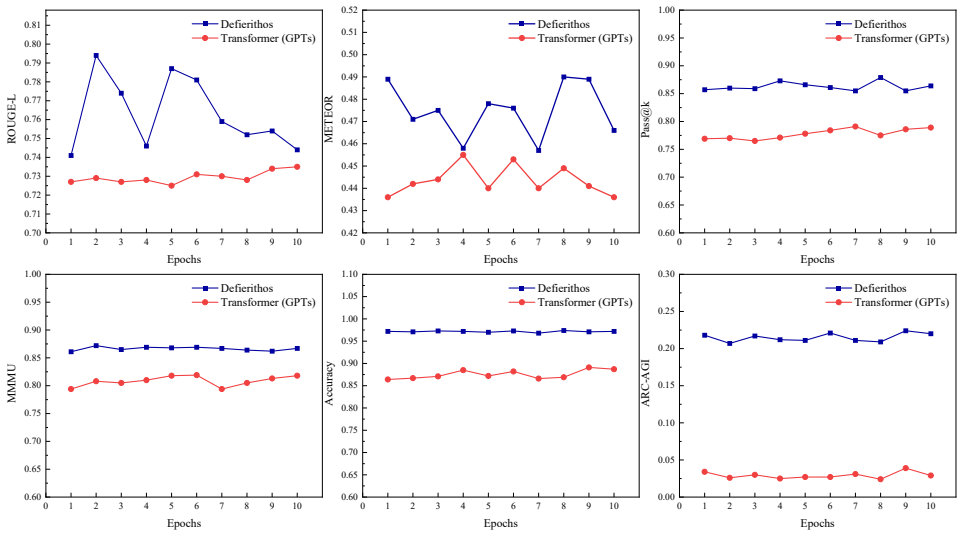

Figure 3: Comparison of the trends of 10 metrics between Defierithos and Transformer

Defierithos has an average value of 0.215, far exceeding the Transformer's 0.029. It proves that the Defierithos has significant capabilities in high-level reasoning and generalization learning, and fluctuates between 0.207 and 0.224. In summary, the Defierithos performs better than the Transformer-based GPTs in all metrics, which reflects its technical highlights as a new generation of architecture in text generation, code programming, and mathematical reasoning.

5 Limitation & Future Research

The Defieritho based on simulation-based training still has limitations compared to the real hardware environment. Due to technical limitations, the simulated environment cannot fully reproduce the computational errors, memory access conflicts, and high load distribution under real hardware conditions, which may lead to the performance of Defierithos' training data results being better than the real environment deployment. In addition, simulation-based training relies on GPTs' Python simulator as a computing carrier, and the underlying infrastructure of GPTs is based on GPT-4 Turbo. This means that its computing power is lower than the most advanced Transformer variant architectures such as ChatGPT o1 or o3 min high, which leads to the low baseline data of the Transformer architecture of the control group. After training to avoid high-order computational bottlenecks, Defierithos shows significant advantages in comparison results, but this advantage may shrink when compared with more advanced Transformer variants. In order to further confirm the universality of Defieritho, it should be considered to train it in a real physical environment and conduct comparative experiments with advanced mainstream Transformer variants in the future.

6 Conclusion

The Defierithos architecture proposed in this work introduces self-resonance field (SRF) technology, replacing the $O(n^2)$ operation dependency of global retrieval based on traditional self-attention. It uses spectrum modulation and phase interference to construct a new parallel operation to trigger resonance only in some semantically relevant areas, and achieves funny information matching, memory retrieval and dynamic reasoning through resonance interference matching. According to the simulation training results, the architecture has proven its performance in text generation, code programming and mathematical reasoning tasks through ROUGE-L, METEOR, Pass@k, MMLU, Accuracy, and ARC-AGI metrics. The above objective evidence confirms that the Defierithos architecture has broken through the bottleneck of Transformer being forced to continuously optimize due to long-term dependency attenuation and computational redundancy in design and algorithm, providing a new path for adaptive dynamic learning of natural language processing.

References

[1] Denis Bobylev, Tuhin Choudhury, Jesse O. Miettinen, Risto Viitala, Emil Kurvinen, and Jussi Sopanen. Simulation-based transfer learning for support stiffness identification. *IEEE Access*, 9:120652–120664, 2021.

[2] DA Brahmankar, RS Dange, and VH Mankar. The effect of resonance on human consciousness. *International Journal of Computer Applications*, 3(5):5224–1036, 2012.

[3] Chen Cai, Truong Son Hy, Rose Yu, and Yusu Wang. On the connection between mpnn and graph transformer. In *Proceedings of the 40th International Conference on Machine Learning*, pages 3408–3430. PMLR, 2023.

[4] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019.

[5] Deepak Chopra and Rudolph E. Tanzi. Super brain.

[6] Charles Condevaux and Sébastien Harispe. Lsg attention: Extrapolation of pretrained transformers to long sequences. In Hisashi Kashima, Tsuyoshi Ide, and Wen-Chih Peng, editors, *Advances in Knowledge Discovery and Data Mining*, pages 443–454, Cham, 2023. Springer Nature Switzerland.

[7] Yinlin Deng, Chunqiu Steven Xia, Haoran Peng, Chenyuan Yang, and Lingming Zhang. Large language models are zero-shot fuzzers: Fuzzing deep-learning libraries via large language models. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2023, pages 423–435, New York, NY, USA, 2023. Association for Computing Machinery.

[8] Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. Longnet: Scaling transformers to 1,000,000,000 tokens, 2023.

[9] Chukwuka Elendu, Dependable C. Amaechi, Alexander U. Okatta, Emmanuel C. Amaechi, Tochi C. Elendu, Chiamaka P. Ezeh, and Ijeoma D. Elendu. The impact of simulation-based training in medical education: A review. *Medicine*, 103(27):e38813, 2024.

[10] Maya Foster and Dustin Scheinost. Brain states as wave-like motifs. *Trends in Cognitive Sciences*, 28(6):492–503, 2024.

[11] Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics.

[12] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024.

[13] Luanxu Guo, Yuanyuan Fang, Feng Chen, Pengcheng Liu, and Song Xu. Large language models with adaptive token fusion: A novel approach to reducing hallucinations and improving inference efficiency.

[14] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2):42:1–42:55, 2025.

[15] Dinesh Kalla, Nathan Smith, Dr Sivaraju Kuraku, and Fnu Samaah. Study and analysis of chat gpt and its impact on different fields of study. *International Journal of Innovative Science and Research Technology*, 8(3):827–833., 2023.

[16] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, pages 5156–5165. JMLR.org, 2020.

[17] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37:122154–122184, 2024.

[18] Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 52342–52364, Red Hook, NY, USA, 2023. Curran Associates Inc.

[19] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025.

[20] Kusi Men, Na Pin, Shumei Lu, Qian Zhang, and Huanhuan Wang. Large language models with novel token processing architecture: A study of the dynamic sequential transformer. 2024.

[21] K. Moorthy, C. Vincent, and A. Darzi. Simulation based training. *BMJ*, 330(7490):493–494, 2005.

[22] Akihiro Nishiyama, Shigenori Tanaka, Jack A. Tuszynski, and Roumiana Tsenkova. Holographic brain theory: Super-radiance, memory capacity and control theory. *International Journal of Molecular Sciences*, 25(4):2399, 2024.

[23] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

[24] Mohaimenul Azam Khan Raiaan, Md. Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 12:26839–26874, 2024.

[25] Denis Rothman. *Transformers for Natural Language Processing: Build Innovative Deep Neural Network Architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and More*. 2021.

[26] Ole Schütt, Peter Messmer, Jürg Hutter, and Joost VandeVondele. Gpu-accelerated sparse matrix–matrix multiplication for linear scaling density functional theory. In *Electronic Structure Calculations on Graphics Processing Units*, chapter 8, pages 173–190. John Wiley & Sons, Ltd, 2016.

[27] Prajwal Singhania, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele. Loki: Low-rank keys for efficient sparse attention. *Advances in Neural Information Processing Systems*, 37:16692–16723, 2024.

[28] Kevin Slagle. Spacebyte: Towards deleting tokenization from large language modeling. *Advances in Neural Information Processing Systems*, 37:124925–124950, 2024.

[29] Mike Thelwall. Evaluating research quality with large language models: An analysis of chatgpt's effectiveness with different settings and inputs. *Journal of Data and Information Science*, 10(1):7–25, 2025.

[30] Jos J.A. Van Berkum. Understanding sentences in context: What brain waves can tell us. *Current Directions in Psychological Science*, 17(6):376–380, 2008.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[32] Dixuan Wang, Yanda Li, Junyuan Jiang, Zepeng Ding, Guochao Jiang, Jiaqing Liang, and Deqing Yang. Tokenization matters! degrading large language models through challenging their tokenization, 2024.

[33] Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. Beyond the limits: A survey of techniques to extend the context length in large language models, 2024.

[34] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 24824–24837, Red Hook, NY, USA, 2022. Curran Associates Inc.

[35] Qingfa Xiao, Jiachuan Wang, Haoyang Li, Cheng Deng, Jiaqi Tang, Shuangyin Li, Yongqi Zhang, Jun Wang, and Lei Chen. Activation-aware probe-query: Effective key-value retrieval for long-context llms inference, 2025.

[36] Zhihui Xie, Jizhou Guo, Tong Yu, and Shuai Li. Calibrating reasoning in language models with internal consistency. *Advances in Neural Information Processing Systems*, 37:114872–114901, 2024.

[37] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 517, pages 5753–5763. Curran Associates Inc., Red Hook, NY, USA, 2019.

[38] Shujian Zhang, Xinjie Fan, Huangjie Zheng, Korawat Tanwisuth, and Mingyuan Zhou. Alignment attention by matching key and query distributions. In *Advances in Neural Information Processing Systems*, volume 34, pages 13444–13457. Curran Associates, Inc., 2021.

[39] Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Lei Shen, Zihan Wang, Andi Wang, Yang Li, Teng Su, Zhilin Yang, and Jie Tang. Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pages 5673–5684, New York, NY, USA, 2023. Association for Computing Machinery.