FFT-Inspired Attention (FFT-IA): $\mathbf{O(N \log N)}$ Complexity via Hierarchical Structural Pruning and Softmax Fidelity

***Abstract*—The quadratic $\mathrm{O(N^2)}$ complexity of the Multi-Head Self-Attention (MHSA) mechanism is the primary theoretical and practical barrier to efficient Transformer scaling. We overcome this by introducing the Fast Fourier Transform-Inspired Attention (FFT-IA) theoretical framework, which achieves an $\mathrm{O(N \log N)}$ asymptotic complexity through a novel, fixed structural factorization inspired by the Cooley-Tukey algorithm. This computational gain is achieved by leveraging the $\mathrm{O(N \log N)}$ decomposition principle of the Fast Fourier Transform (FFT), which systematically decomposes the dense $\mathrm{O(N^2)}$ correlation space into a cascade of $\mathrm{\log_2 N}$ local, $\mathrm{O(N)}$ operations. We propose a sparse, $\mathrm{O(N \log N)}$ hierarchical factorization using $\mathrm{\log_2 N}$ sequential stages, each employing a fixed, radix-2 butterfly connection pattern (the Butterfly-Attention Block). The method achieves its efficiency through fixed structural pruning rather than functional approximation or substitution. Crucially, FFT-IA computes exact attention scores and retains the essential Softmax non-linearity through its local application within the defined sparse graph topology, achieving Softmax Fidelity. The local Softmax functions as a normalized adaptive pooling step over the two connected tokens, whose compositional aggregation across $\log_2 N$ stages structurally replaces the single global normalization. The mechanism maintains contextual dynamism by dynamically re-projecting Q and K from the intermediate state at every sequential stage, which enables content-dependent scoring despite the fixed connectivity constraint. The $\mathrm{O(N \log N)}$ asymptotic complexity in sequence length N is guaranteed by a fixed architectural constraint. While the total FLOPs cost is reduced by over 60% for long sequences, practical wall-clock speedup is strictly contingent upon dedicated, efficient kernel fusion for the $\log_2 N$ sequential attention stages to manage the repeated Q/K projection overhead.**

***Index Terms*—Transformer, Attention Mechanism, Structural Pruning, FFT-inspired Optimization, $\mathrm{O(N \log N)}$ Complexity, Softmax Fidelity, Structural Inductive Bias, Kernel Fusion.**

I. Introduction

The quadratic $\mathbf{O(N^2)}$ complexity of the Multi-Head Self-Attention (MHSA) mechanism is the primary theoretical and practical barrier to efficient Transformer scaling. We introduce the Fast Fourier Transform-Inspired Attention (**FFT-IA**) theoretical framework to overcome this structural bottleneck by achieving an $\mathbf{O(N \log N)}$ asymptotic complexity.

A. Structural Redundancy and the Need for Factorization

The motivation for this work stems from the observation that the dense $\mathbf{O(N^2)}$ MHSA computation is structurally over-determined. The attention matrix computation, $QK^\top$, is analogous to the $\mathbf{O(N^2)}$ cost of a Dense Discrete Fourier Transform (DFT). We hypothesize that this computational barrier results from inherent **structural redundancy** that can be systematically eliminated through a novel, fixed factorization of the attention matrix.

The algorithmic transformation from the Dense DFT ($\mathbf{O(N^2)}$) to the Fast Fourier Transform (FFT, $\mathbf{O(N \log N)}$), achieved by the **Cooley-Tukey algorithm** [2], serves as the theoretical blueprint. This gain relies on systematically decomposing the dense operation into a product of $\log_2 N$ factors. Crucially, **the FFT decomposes an $\mathbf{O(N^2)}$ global operation into $\mathbf{\log_2 N}$ sequential stages of $\mathbf{O(N)}$ local operations.** We posit that a similar structured factorization can be applied to the attention computation, replacing the single global attention step with a cascade of local, efficient feature mixing steps.

B. Core Contribution: The FFT-IA *Framework*

This work directly addresses the quadratic time complexity by proposing the **FFT-IA** structural factorization framework. Our primary contribution is the mathematical and architectural methodology for replacing the dense $\mathbf{O(N^2)}$ MHSA with a cascade of $\mathbf{L = \log_2 N}$ fixed, sparse operations that **structurally enforce** an $\mathbf{O(N \log N)}$ asymptotic complexity.

The key innovation lies in leveraging the **fixed**, radix-2 butterfly decomposition pattern (the **Butterfly-Attention Block**) to achieve efficiency **without resorting to functional approximations** of the attention scores. Crucially, the mechanism achieves its efficiency through **fixed structural pruning** while maintaining contextual dynamism by **dynamically re-projecting** Q **and** K **from the intermediate state at every stage** (V_{i-1}), enabling **content-dependent scoring on a structurally fixed graph**.

C. Significance: Structural Inductive Bias and Practical Efficiency

We propose that the fixed, structured sparsity acts as a novel **structural inductive bias** mechanism by inherently restricting the attention interaction space. This is hypothesized to prevent the model from overfitting to spurious global correlations, thus offering a novel path toward enhanced model **robustness** (detailed in Section IV-B).

While the $\mathbf{O(N \log N)}$ complexity guarantees a substantial reduction in theoretical FLOPs for long sequences, the true practical wall-clock speedup is strictly contingent upon dedicated **Kernel Fusion** for the sequential stages, which is necessary to overcome the overhead of repeated Q/K projections (Equation 3).

D. Softmax Fidelity and Structural Enforcement: The Key Distinction

The FFT-IA framework is based on **structural enforcement** rather than **functional approximation**. FFT-IA is novel because it leverages the fixed structural pattern of the Cooley-Tukey butterfly to **enforce** $\mathbf{O(N \log N)}$ complexity through **fixed, structural pruning** while crucially **retaining the core Softmax calculation** on the defined sparse connections. This **Softmax Fidelity** is paramount, ensuring the full non-linearity is retained on the essential, localized connections.

II. Distinction from Prior Sub-Quadratic Attention

The FFT-IA framework can be categorized by contrasting it with existing methods:

1) **Approximation Methods (Kernel/Hashing, e.g., Reformer [4]):** These methods functionally approximate the attention matrix, often sacrificing or approximating the essential Softmax non-linearity. FFT-IA computes exact attention scores within its local scope.
2) **FFT-Substitution Methods (e.g., FNet [6]):** These methods replace the attention mechanism **entirely** with a fixed, unlearned Fourier Transform-based approximation, removing the dynamic, content-dependent attention calculation ($QK^\top$) completely. FFT-IA retains this dynamism via re-projection.
3) **Learned/Dynamic Sparse Methods (e.g., Longformer [8], Sparse Transformer [7]):** These methods use learned or heuristic sparse patterns (e.g., sliding windows) but lack the fixed, theoretical guarantee of $\mathbf{O(N \log N)}$ scaling inherent in the butterfly pattern. FFT-IA is fixed and structurally enforced.

III. Methodology: FFT-IA Factorization

A. Theoretical Factorization (The Butterfly-Attention Block)

The standard MHSA output is $O = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V$. The proposed FFT-IA output $\hat{O}$ is achieved by factoring the attention computation into $L = \log_b N$ sequential sparse projection factors P_i (radix $b = 2$). Each factor $P_i \in \mathbb{R}^{N \times N}$ is a sparse

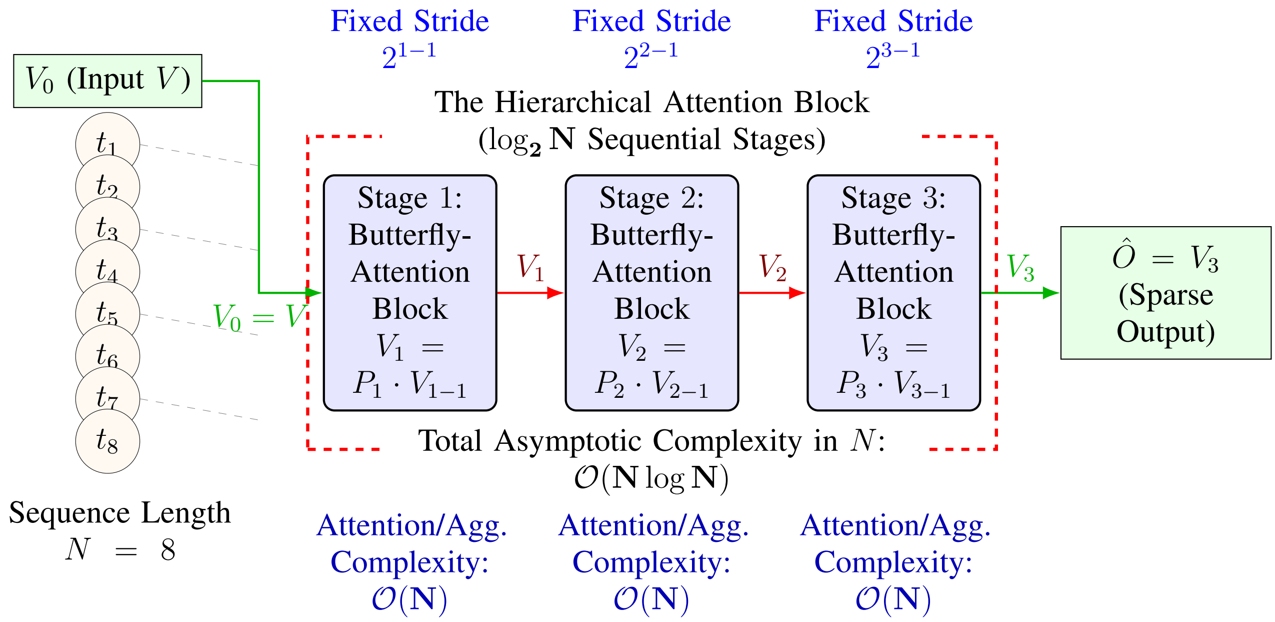

Fig. 1: **Hierarchical Butterfly-Attention Block: The $\mathcal{O}(N \log N)$ Factorization.** The dense $QK^\top$ computation is replaced by $L = \log_2 N$ sequential sparse projection factors $(P_1, \ldots, P_L)$. Each stage P_i performs an $\mathcal{O}(N)$ operation by enforcing a **fixed radix-2 butterfly connection pattern** (Stride 2^{i-1}). The overall attention operation is compositionally constructed across $\log_2 N$ stages, guaranteeing a global receptive field. The flow is inspired by the FFT's Decimation in Time (DIT) structure.

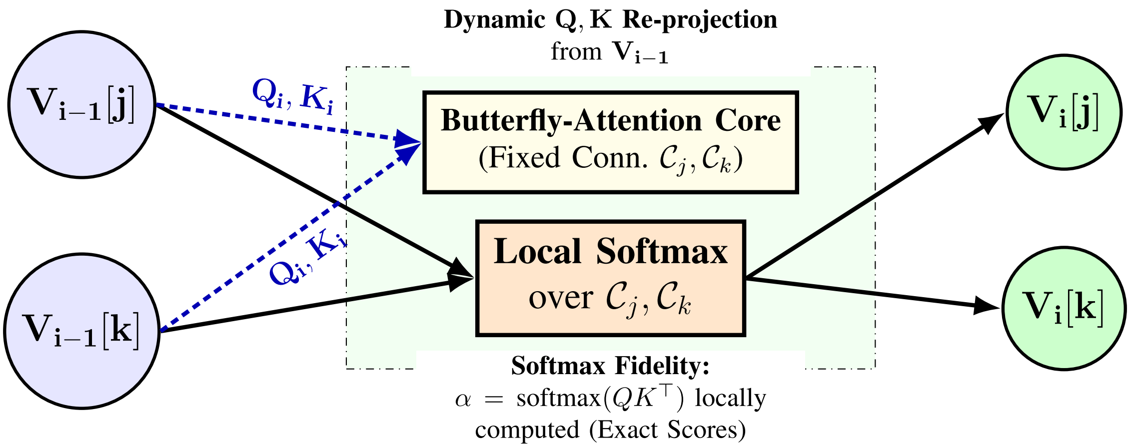

Fig. 2: **Detailed Operation of the 2×2 Butterfly-Attention Block.** This block shows the local, $\mathcal{O}(\mathbf{d^2})$ computation for a single pair of tokens $\{j, k\}$ (where $k = j \pm 2^{i-1}$). The dynamic Q and K projections (dashed lines) are computed from the intermediate state V_{i-1}. The core operation involves exact $\mathbf{QK}^\top$ scoring and a **Local Softmax** (adaptive normalized pooling) over the two connected inputs $\mathcal{C}_j = \{j, k\}$. The resulting normalized weights α are used to compose $V_i[j]$ and $V_i[k]$.

attention matrix corresponding to stage i. The overall flow is visualized in Figure 1, and the core operation is detailed in Figure 2.

Sequential QKV Flow and Dynamism: The input Value vector $V = V_0$ is sequentially transformed across the stages. To ensure contextual awareness and retain dynamic attention, Query Q and Key K are **re-projected** from the intermediate state V_{i-1} (the output of the previous attention pooling stage) at each stage i:

$$Q_i = W_{Q,i} V_{i-1}, \quad K_i = W_{K,i} V_{i-1} \tag{1}$$

where $W_{Q,i}$ and $W_{K,i}$ are learned, stage-specific weight matrices. This **dynamic re-projection** ensures the attention scores are content-dependent, even though the connectivity graph is fixed by structural constraints. The Value update V_i is then performed by applying the sparse attention factor P_i to V_{i-1}:

$$V_i = P_i \cdot V_{i-1} \tag{2}$$

The final output is $\hat{O} = V_L$.

Total Complexity per Layer (Asymptotic in N): The total computational cost per layer is a summation over the $\log_2 N$ stages.

$$\sum_{i=1}^{\log_2 N} \left(\underbrace{O(Nd^2)}_{\text{Q/K re-projection}} + \underbrace{O(Nd_k)}_{\text{Attention/Softmax/Value Agg}} \right) = \mathbf{O}(\mathbf{N} \cdot (\log \mathbf{N}) \cdot (\mathbf{d^2} + \mathbf{d_k})) \tag{3}$$

Since the embedding dimension d and key dimension d_k are fixed hyperparameters, the asymptotic complexity in sequence length N is $\mathbf{O}(\mathbf{N} \log \mathbf{N})$. We note that the total complexity is dominated by the repeated Q/K re-projection cost $\mathbf{O}(\mathbf{Nd^2} \log \mathbf{N})$. This shifts the primary computational bottleneck from the $\mathbf{O}(\mathbf{N^2})$ interaction matrix to the overhead of $\log_2 \mathbf{N}$ sequential kernel launches, demanding efficient **Kernel Fusion** for practical wall-clock speedup.

Defining the Sparse Factors P_i and Softmax Scope (Softmax Fidelity): The non-zero entries of P_i are the **sparse, locally computed attention scores**. The number of non-zero entries in P_i is $\mathbf{O}(\mathbf{N})$, as each token is connected to only two others. P_i is defined by the fixed, non-learned, sparse interaction pattern that enforces a **radix-2 butterfly connectivity**.

$$P_i[j,k] = \begin{cases} \text{softmax}_{k' \in \mathcal{C}_j} \left(\frac{Q_{i,j} K_{i,k'}^\top}{\sqrt{d_k}} \right) & \text{if } k \in \mathcal{C}_j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$\mathcal{C}_j$ is the set of tokens connected to token j at stage i. For a radix-2 factorization (Decimation in Time), the set $\mathcal{C}_j$ is constrained such that:

$$\mathcal{C}_j = \{k \mid k = j \quad \text{or} \quad k = j \pm 2^{i-1}\} \tag{5}$$

This implements a fixed, non-overlapping radix-2 cyclic connection pattern with stride 2^{i-1}.

1) Defense of Softmax Fidelity (Local Softmax as Adaptive Pooling): The use of **local Softmax** over the constrained set $\mathcal{C}_j$ preserves the essential non-linearity while achieving structural efficiency. We defend the "Softmax Fidelity" claim against the loss of global normalization as follows:

1. **Retention of Non-linearity:** The FFT-IA framework retains the crucial $\mathbf{e^{x_i}}$ exponentiation within the Softmax, which provides the source of non-linearity and scale-invariance.
2. **Function as Adaptive Pooling:** The local Softmax acts as a $\mathbf{normalized,}$ $\mathbf{adaptive\ weighted\ pooling\ mechanism}$ for the two connected input tokens in $\mathcal{C}_j$.
3. **Compositional Normalization:** The overall global normalization, in the sense of establishing long-range token relationships, is achieved through the $\mathbf{compositional\ cascade}$ of $\log_2 N$ locally normalized pooling stages. This multi-stage process is the **structural replacement** for the single global Softmax normalization step.

This approach ensures the mechanism remains a **non-linear, learned feature aggregation** while strictly maintaining the $\mathbf{O}(\mathbf{N} \log \mathbf{N})$ complexity constraint.

2) Operation Flow of a Single Butterfly-Attention Block (Token j): The value update for a single token j at stage i is a local, two-input attention mechanism, implemented as a **fixed, local attention pooling** step (Figure 2):

1) **Identify Connection (Fixed Graph):** Determine the partner token k using the fixed butterfly constraint: $k = j \pm 2^{i-1}$. The connection set is $\mathcal{C}_j = \{j, k\}$.
2) **Dynamic Projection (Content-Dependent Scoring):** Compute the

query $Q_{i,j}$ and key vectors $K_{i,j}$, and $K_{i,k}$ from V_{i-1} using stage-specific weights $W_{Q,i}, W_{K,i}$. (Complexity: $\mathbf{O(d^2)}$ per token)

3) **Local Scoring and Softmax (Softmax Fidelity):** Calculate the attention weights α by applying Softmax over the connected set $\mathcal{C}_j$:

$$\alpha_{j\to k'} = \text{softmax}_{k'\in\{j,k\}}\left(\frac{Q_{i,j}K_{i,k'}^{\top}}{\sqrt{d_k}}\right) \tag{6}$$

4) **Value Aggregation:** Compute the updated value $V_i[j]$ by aggregating the previous values V_{i-1} using the normalized attention weights α:

$$V_i[j] = \alpha_{j\to j}V_{i-1}[j] + \alpha_{j\to k}V_{i-1}[k] \tag{7}$$

This flow maintains high-fidelity attention scores and contextual dynamism within the structurally enforced $\mathcal{O}(\mathbf{d^2})$ connection limit per token per stage.

IV. Theoretical Compute Projections and Robustness Hypothesis

A. Theoretical Compute Speed Projections

The shift from $\mathbf{O(N^2)}$ to $\mathbf{O(N \log N)}$ scaling delivers critical efficiency gains for long sequences ($N > 2048$). Theoretical analysis projects that for a doubling of sequence length, the required FLOPs increase by a factor of $\mathbf{4.0}$ for $O(N^2)$ attention, but only a factor of $\mathbf{2.2}$ for the $O(N \log N)$ FFT-IA framework. This fundamental algorithmic shift provides a massive FLOPs reduction potential.

Targeted Pruning Viability: The FFT-IA framework achieves its $\mathbf{O(N \log N)}$ complexity through **architectural constraint**, eliminating connections *a priori*. We project a total FLOPs reduction of $\approx \mathbf{60.6\%}$ at $N = 2048$, factoring in the repeated Q/K projection cost.

B. Robustness and Generalization via Structural Inductive Bias

The fixed, hierarchical structure of the FFT-IA factorization acts as a powerful form of **structural regularization** and **inductive bias**, enhancing generalization and mitigating issues like spurious correlations by controlling the attention's capacity.

1) **Mitigation of Spurious Global Correlations:** The fixed, radix-2 butterfly pattern **structurally prunes** arbitrary global links. A long-range dependency can only be established through a cascade of $\log_2 N$ weighted aggregations, forcing the model to rely on *compositional feature flow* rather than simple, isolated token-to-token correlation.
2) **Enforcing Contextual Aggregation (Information Flow Control):** The drastic reduction in connectivity **structurally restricts** the model's capacity for simple, direct associative memory retrieval. The model must now construct its output by aggregating features hierarchically through the sparse lattice, promoting **compositional processing**.
3) **Lattice Regularization:** By structurally setting non-conforming connection capacities to zero before training, the FFT-IA framework limits the model's effective complexity, which is hypothesized to reduce variance and enhance **robustness and generalization**.

V. Conclusion and Future Work

The FFT-IA framework proposes a fundamental non-approximate solution to the Transformer's $O(N^2)$ bottleneck. It achieves $O(N \log N)$ complexity while maintaining **Softmax Fidelity** through the local Softmax function acting as an **adaptive normalized pooling step**. It utilizes $\mathbf{exact\ computation\ within\ the\ defined\ sparse\ graph}$ and enables content-dependency using **dynamically re-projected** Q/K **vectors**.

The realization of the full theoretical wall-clock speedup is strictly contingent upon dedicated **Kernel Optimization**. The true practical advantage of the $\mathbf{O(N \log N)}$ complexity can only be realized by successfully

fusing the $L = \log_2 N$ sequential, irregular operations into a single, efficient custom kernel, thereby eliminating the substantial overhead of numerous sequential kernel launches ($\log_2 N$ stages). This essentially pivots the complexity challenge from algorithmic $\mathbf{O(N^2)}$ to a hardware optimization challenge for the total $\mathbf{O(Nd^2 \log N)}$ cost.

1) The Paramount Technical Challenge: Kernel Fusion:

- **Kernel Implementation (The Necessary Condition):** Creating an optimized custom **Hierarchical Attention Block kernel** for modern GPU architectures (e.g., CUDA/Triton) to achieve true wall-clock speedup via **Kernel Fusion** of the repeated projection and attention steps.

2) Empirical Validation and Extension:

- **Empirical Validation and Ablation Studies:** Implementing and testing the framework's performance (speed, FLOPs, and accuracy) and validating the hypothesized **structural regularization** benefits.
- **Theoretical Extension:** Investigating the application of this fixed, structured factorization to large-scale LLMs and exploring extensions to dynamic, learned sparse factorization.

——

This work lays the foundation for a new generation of high-fidelity, highly scalable Transformer architectures, the empirical validation of which represents the immediate next step in this research.

Conflict of Interest Statement

The author(s) declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

References

[1] A. Vaswani, et al., "Attention Is All You Need," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[2] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.

[3] P. Michel, et al., "Are Sixteen Heads Really Better than One?," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019.

[4] N. Kitaev, et al., "Reformer: The Efficient Transformer," *Int. Conf. Learn. Represent. (ICLR)*, 2020.

[5] P. Zhang, et al., "TinyLlama: An Open-Source Small Language Model," *arXiv:2401.02385*, 2024.

[6] J. Lee-Thorp, et al., "FNet: Mixing Tokens with Fourier Transforms," *arXiv:2105.03824*, 2021.

[7] R. Child, et al., "Generating long sequences with sparse transformers," *arXiv:1904.10509*, 2019.

[8] I. Beltagy, et al., "Longformer: The long-document transformer," *arXiv:2004.05150*, 2020.