Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081304

ARTICLE

Beyond Classical Positional Encodings: A Learnable QFT-Inspired Framework for Transformer Language Models

Sara Tehsin[1], Tallha Akram[2,*], Syed Rameez Naqvi[3], Meshal Alharbi[4] and Abdulrahman Alabduljabbar[2]

[1]Faculty of Informatics, Kaunas University of Technology, Kaunas, Lithuania
[2]Dept. Information Systems, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia
[3]Dept. Computer Science, Tulane University, New Orleans, LA, USA
[4]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia

*Corresponding Author: Tallha Akram. Email: t.akram@psau.edu.sa

Received: 27 February 2026; Accepted: 14 May 2026; Published: 23 July 2026

ABSTRACT: Transformers have become the dominant architecture for sequence modeling in natural language processing; however, their effectiveness critically depends on how positional information is encoded. Conventional positional encodings, while effective, may have limited structural flexibility for capturing complex global sequence relationships. Recent quantum-inspired approaches have sought to address this limitation, yet many either oversimplify quantum principles or introduce substantial computational or hardware overhead. We introduce a novel Quantum Fourier Transform (QFT)-inspired positional encoding scheme for transformers, motivated by the structured frequency representation of the QFT. Unlike prior approaches that either emulate quantum operations superficially or require complex circuit constructions, the proposed method provides a learnable hybrid encoding that preserves quantum-inspired structure while remaining aligned with hardware-efficient circuit primitives and structurally compatible with future near-term quantum implementations. Experiments on WikiText-103 indicate that the proposed encoding achieves competitive perplexity, improved robustness to input scrambling, and stable training behavior relative to alternative quantum-inspired baselines under the evaluated settings. Preliminary circuit-level simulations further suggest favorable noise resilience of the associated encoding primitives. These findings support the potential utility of incorporating quantum-inspired design principles into deep learning architectures and provide a foundation for future exploration at the interface of quantum computing and transformer-based natural language processing (NLP).

KEYWORDS: LLMs; positional encoding; Quantum Fourier Transform; positional embeddings; hybrid quantum-classical models; near-term quantum devices; quantum transformer

1 Introduction

Transformers have revolutionized natural language processing (NLP) by leveraging self-attention and positional encoding (PE) methods to model sequential data effectively. Vaswani et al. [1] was among the first to present a high-level architecture of transformers, highlighting the role of PE in injecting sequence order information into the model, enabling it to distinguish between otherwise permutation-invariant token embeddings. A recent study concluded that while PE may not be fundamental to causal language models (e.g., GPT), masked language models, such as BERT, will fail to converge without it [2]. While existing PE

strategies (e.g., sinusoidal, learned) have proven effective, they are inherently classical and may limit the integration of quantum principles in transformer architectures.

Quantum computers provide sophisticated mathematical structures to represent the position of an object in both phase and amplitude forms. These structures can intrinsically represent both periodicity and hierarchical structures. However, in practical transformer models, quantum-native positional encoding (PE) has yet to be investigated. This creates a link between quantum algorithm theory and deep learning (NLP) applications [3]. Most current quantum encoding (QE) methods also suffer from limited expressiveness due to rigid circuit designs or a lack of compatibility with classical transformer architectures. These drawbacks make them less suitable for practical real-world NLP tasks.

This paper presents a new PE approach for transformer models based on principles from the Quantum Fourier Transform (QFT), incorporating quantum mechanics (phase shifts, angular harmonics), and capable of learning and matching current quantum-inspired methods, and can be used with currently available near-term quantum devices [4]. Our goal is two-fold: Firstly, to demonstrate that encodings similar to those generated by the QFT outperform, or at least equal, the performance of traditional methods of quantum-inspired language modeling. Secondly, to provide evidence for the feasibility of building QFT-enhanced encoding into hybrid or fully quantum representations of natural language in future NLP applications [5].

This work thus bridges quantum theory and transformer-based deep learning, providing a practical and efficient framework for embedding quantum-native positional awareness into modern sequence models. Our primary contributions include the following:

1. We propose a hardware-friendly, hybrid PE mechanism that blends QFT principles with learnable components, enabling expressive yet efficient representations for transformer models.
2. We demonstrate the robustness and generalization ability of the proposed encoding through perplexity tests on original, randomly scrambled, and reversed sequences—significantly outperforming ablated variants and existing quantum-native baselines.

The rest of the manuscript is organized as follows: Section 2 summarizes background of PE, classical PE methods and their existing quantum-native counterparts. The proposed QFT-inspired PE method along with its mathematical formulation is presented in Section 3. Section 4 presents experimental setup and simulation results before we conclude the manuscript in Section 5.

2 Background and Related Works

In this section, we summarize the existing PE methods in both classical and quantum computing paradigms. At the end, we highlight limitations in state-of-the-art and motivate the need for another superior and diverse PE method.

2.1 Classical PE Methods

The classical transformers, which started with sinusoidal embeddings for PE, have settled for learned embeddings since GPT-2. While numerous attempts have already been devoted for advancing representations from *word embeddings* to *contextualized embeddings*; their foundation has remained the same. Since the focus of this work is on quantum computing, here we only review the two baseline classical encoding methods.

2.1.1 Sinusoidal Encoding

Sinusoids are continuous, periodic functions capable of providing a fixed, deterministic representation of positions and capturing relative distance between tokens. By exploiting the fact that sin and cosine

are phase-shifted versions of each other, the sinusoidal PE assigns each position in a sequence a unique embedding using a combination of sine and cosine waves of different frequencies, Eq. (1) [1]. This ensures that positions encode both absolute and relative information, while enabling this encoding method generalize to longer sequences than seen during training.

$$PE(pos, 2i) = \sin\left(\frac{p}{10{,}000^{2i/d}}\right)$$
$$PE(pos, 2i+1) = \cos\left(\frac{p}{10{,}000^{2i/d}}\right) \quad (1)$$

2.1.2 Learned Embeddings

In this type of encoding [6], each position in a sequence is assigned a trainable vector called learned embedding from an embedding matrix–learned during training. Since these embeddings are randomly initialized and trained via backpropagation, they do not naturally follow any cyclic pattern or periodicity (unless explicitly trained for), which allows them to learn arbitrary position mappings. However, their values change smoothly during gradient updates, making them continuous. While such methods work well for tasks where positional information is not strictly sequential (e.g., graphs), they do not naturally repeat or extrapolate beyond training limits. Modern GPT (GPT-2 and after) type models apply learned embedding techniques or some variation of them to solve positional encoding challenges.

Despite being a well-established solution, learned embeddings struggle with long sequences that exceed the lengths they were trained on (e.g., GPT-2 is unable to handle sequences longer than 2048 tokens) [7]. In addition, each position has to be stored and computed separately, leading to performance issues due to inefficient processing overhead [8]. Similarly, despite being able to handle extrapolation, training, and/or flexibility limitations, sinusoidal encoding also has limitations [9]. These and other limitations facilitate the need for a complete paradigm shift. Future models (e.g., quantum-enhanced transformers) are likely to require efficient quantum-native PE, as quantum hardware continues to expand.

Scalable PE methods, like Rotary Positional Encoder (RoPE) and ALiBi, provide efficient relative and extrapolated attention for large language models of the type previously built using transformers. Although these methods have demonstrated success with long-context modeling, they remain classical formulations optimized for computational scalability rather than quantum representations. The present study instead examines quantum-inspired positional encoding paradigms under a unified structural framework; RoPE [10] and ALiBi [11] are therefore referenced for completeness but lie outside the comparative scope.

2.2 Quantum PE Methods

A number of quantum PE methods—based on the principles of quantum mechanics—have been recently developed. Most of those models demonstrated that quantum sequence models could process long-range dependencies efficiently and encode all positions at once, thereby offering potential speedup. Their traits, such as higher dimensional expressiveness and compact representation, give them an edge over the classical methods [12]. In what follows, we present a summary of some of the benchmark PE methods.

2.2.1 Basis Encoding

It is the simplest counterpart in the quantum computing domain. It maps classical positional data to computational basis states of qubits, where each position p is directly converted into a quantum basis state $|p\rangle$. It works by converting the classical position to binary vector (Eq. (2)) followed by application of X-gates on qubits where binary bits are 1, Eq. (3). The final state $|p\rangle$ represents the one-hot quantum encoding of the position [13].

$$p \rightarrow bin(p) = b_0 b_1 \dots b_{n-1} \tag{2}$$

$$|\psi\rangle = |p\rangle = X_0^{b_0} \otimes X_1^{b_1} \otimes \cdots \otimes X_{n-1}^{b_{n-1}} |0\rangle^{\otimes n} \tag{3}$$

Being simple by directly mapping classical indices to quantum states, the Basis encoding offers a few advantages including lossless encoding and deterministic representation. Simplicity, however, often comes with a price: the Basis encoding requires n qubits to encode 2^n positions, but only one state at a time (since it does not exploit quantum superposition)—making it inefficient for large-scale and high-dimensional data. Secondly, its fixed and deterministic nature does not allow it to adapt to input distributions, and it fails to capture relational patterns, making Basis encoding impractical for most applications. The more practical PE methods consider Basis encoding as their benchmark for perplexity, however, they must promise higher scalability, lower computational burden and exploit properties of quantum mechanics.

2.2.2 Superposition Encoding

It represents positional data as a uniform superposition of quantum basis states using Hadamard gates. Instead of encoding a single position, it encodes all possible positions simultaneously using superposition [14]. Eqs. (4)–(6) present its mathematical representation:

$$|\psi\rangle = H^{\otimes n} |0\rangle^{\otimes n} \tag{4}$$

The Hadamard gate H creates an equal superposition of all possible computational basis states:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{5}$$

For n qubits, applying Hadamard to all qubits results in:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle \tag{6}$$

This creates a uniform probability distribution over all 2^n positions, enabling it to offer compact representation of positions, reducing space requirements, as well as allowing quantum speedup in certain applications. However, since measurement is probabilistic, direct retrieval of a specific position is not guaranteed. Moreover, when measured, the quantum state collapses to a single basis state, losing the full superposition information.

2.2.3 Amplitude Encoding

In contrast to Superposition encoding, which uses Hadamard gates to create superposition of all basis states with equal probabilities, Amplitude encoding represents classical positional vector components as continuous quantum state amplitudes. It encodes 2^n positions into n qubits by mapping classical vector components v to quantum state amplitudes [15], Eq. (7), which must be normalized using Eq. (8).

$$|\psi\rangle = \sum_{i=0}^{2^n-1} v_i |i\rangle \tag{7}$$

$$\sum_{i=0}^{2^n-1} |v_i|^2 = 1 \tag{8}$$

While Amplitude encoding enjoys most advantages that Superposition encoding offers, its methodology of encoding information as amplitudes allows more meaningful feature representation. On the other hand, similar to Superposition encoding, the input vector must be normalized, potentially leading to loss of precision in Amplitude encoding. Secondly, preparing an arbitrary amplitude-encoded quantum state requires deep quantum circuits, which can be difficult for noisy quantum hardware.

2.2.4 Angle Encoding

Angle encoding represents classical positional data by mapping values to quantum state rotation angles, where each classical value is encoded as a rotation of a qubit around y axis using R_y gates [16]. The rotation angle θ is given by Eq. (9), where b represents a binary bit as in Eq. (2), and the quantum state encodes position as rotation angles in the Hilbert space, as given by Eq. (10).

$$\theta_i = b_i \cdot \frac{\pi}{2} \tag{9}$$

$$|\psi\rangle = R_y(\theta_0) \otimes R_y(\theta_1) \otimes \cdots \otimes R_y(\theta_{n-1})|0\rangle^{\otimes n} \tag{10}$$

Besides offering many of the same benefits as Amplitude encoding, this method promises interpretable encoding, where the quantum state visually maps onto the Bloch sphere, making it easier to analyze.

2.2.5 Quantum Associative Memory (QAM) Encoding

QAM for PE stores and retrieves positional information based on learned quantum state associations. It maps classical positions to quantum states, stores positional data associations using superposition and phase shifts, Eq. (11), and retrieves them using quantum interference and Grover-like amplification [17]. Eq. (12) defines the QAM as classical position and its quantum-encoded state mapping pairs, which enable efficient lookup using quantum interference.

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |p_i\rangle|\psi_i\rangle \tag{11}$$

$$M = (p_i, |\psi_i\rangle) : i = 0, 1, \ldots, n-1 \tag{12}$$

Eq. (13) applies a phase shift, ϕ_i to each memory state $|\psi_i\rangle$ using a unitary transformation U_{phase}. The phase shift amplifies correct memory state, enhancing retrieval accuracy.

$$U_{phase}|p_i\rangle|\psi_i\rangle = e^{i\phi_i}|p_i\rangle|\psi_i\rangle \tag{13}$$

The disadvantages associated with QAM encoding include position retrieval; once the quantum state collapses, it will necessitate multiple queries for robust retrieval. Secondly, while superposition allows efficient storage, practical hardware constraints, including noisy environment, limit retrieval fidelity.

2.2.6 Quantum Feature Maps (QFM)

QFM transform classical positional data into quantum states using parameterized quantum circuits, and encode position information into a high-dimensional Hilbert space for richer embeddings, making it more expressive than classical methods [18]. Eq. (14) presents its mathematical representation.

$$|\psi(p)\rangle = U_\Phi(p)|0\rangle \tag{14}$$

where p is the classical position index, $U_\Phi(p)$ represents the parameterized quantum circuit that encodes p into a quantum state and $|0\rangle$ is the initial quantum state. Unlike the classical encoding, QFM naturally supports nonlinear encoding and enhances relative PE by correlating positional states. On the other hand, state collapse upon measurement necessitates multiple runs to reconstruct encoded features and requires the classical data be scaled properly to fit within quantum rotation angles.

Table 1 summarizes classical software emulation of the quantum-inspired PE methods used in this study. While motivated by valid quantum formulations, these implementations are executed within standard transformer architectures and therefore approximate selected structural properties rather than full quantum behavior.

Table 1: Classical software emulation of quantum-inspired positional encoding methods used in this study.

Method	Classical Emulation in Transformer Models	Key Limitation Relative to True Quantum Realization
Basis	Implemented using one-hot positional embedding vectors in the model input space. This preserves deterministic position identity analogous to computational basis states.	Does not simulate binary-to-qubit mapping, measurement, or gate-level quantum operations.
Superposition	Approximated using sinusoidal activations across positions with fixed frequency sweeps, mimicking distributed multi-state representations.	No genuine superposition, entanglement, or probabilistic state collapse.
Amplitude	Implemented by scaling normalized positional values across embedding dimensions, yielding smooth continuous positional ramps.	No unit-norm quantum state preparation or amplitude-preserving constraints.
Angle	Implemented using standard sinusoidal positional encoding (multi-frequency sine/cosine mapping), analogous to rotation-based encoding.	Does not perform physical $R_y(\theta)$ qubit rotations or Bloch-sphere state evolution.
QAM	Implemented using a learned embedding table that assigns trainable vectors to positions, analogous to memory lookup.	No superposition-based retrieval, interference, or Grover-style amplification.
QFM	Implemented using learnable trigonometric projections with frequency and phase modulation, resembling kernelized feature maps.	No unitary circuit evolution, entanglement, or hardware-executed quantum kernels.

All quantum PE methods described here are inspired by valid quantum formulations, but their practical implementation in transformer models involves classical approximations. These emulations retain some structural properties (e.g., periodicity, high-dimensional projection), but do not enforce quantum mechanics' operational constraints, such as normalization, unitarity, or probabilistic measurement. These approximations, while operationally classical, are grounded in quantum formalism and offer insights into how quantum properties might inspire or enhance PE in future hybrid or hardware-accelerated models.

2.3 Room for Improvement

Despite their ability to handle extrapolation efficiently, sinusoidal embeddings are fixed and non-adaptive. Learned embeddings, on the other hand, lack generalization to longer sequences. Finally, the existing quantum PE methods encode positional information but often lack structured frequency representation. Combining these shortcomings and limitations, there exists a need for a comprehensive PE method that (i) efficiently handles long-range dependencies, (ii) dynamically modulates PE based on sequence requirements, (iii) maintains positional coherence post-measurement and (iv) allows smooth transitions between positions.

2.4 Problem Formulation

For a given token's input $x_1, x_2, \ldots x_L$, the objective is to generate a positional encoding matrix $PE \in R^{L\times d}$ that embed the positional relationships between tokens. The encoding should: (i) retain positional uniqueness, (ii) create robust representations against sequence perturbations, and (iii) support easy integration into the transformer pipeline

The proposed methodology inputs the sequence length (L) and embedding dimension (d) information into a unique positional encoding matrix (PE). This PE matrix is subsequently added to each token's embedding, so that the embeddings can be processed by a transformer model.

3 Proposed PE Method

3.1 Conceptualization

Classical PE methods represent positions in direct space, assuming fixed or trainable embeddings. However, many real-world data structures (natural language, time series, bioinformatics, etc.) contain underlying periodicity and long-range dependencies. For example, natural signals (speech, DNA sequences, time series) rely on frequency-based representations for pattern detection. In such cases, instead of encoding positions explicitly, encoding frequency components makes the representation adaptive [19]. It is well-understood that the FT is widely used to analyze periodic and structured signals. It decomposes a sequence into frequency components, allowing better encoding of position-based relationships [20]. Its quantum equivalent (QFT) [21] is likely to complement this concept by providing a compact, structured, and inherently periodic representation of positional information, well-suited for large-scale sequence processing.

3.2 Core Principles

The proposed QFT-guided PE method attempts to address the limitations discussed above by combining the structured periodicity of the QFT with learnable adaptive components. Unlike prior classical methods, this method considers the following measures:

1. Utilizes QFT for structured position representation, naturally encoding positional information in the frequency domain.
2. Introduces learnable phase shifts to allow adaptive fine-tuning, avoiding fixed non-trainable embeddings.
3. Maintains position information post-measurement and avoids collapse issues.

To formalize this approach, in what follows we define the mathematical formulation of QFT-based PE method.

3.3 Mathematical Formulation

Given a sequence of tokens:

$$X = [x_0, x_1, \ldots, x_{L-1}] \tag{15}$$

where each x_p is a feature vector, a PE function:

$$PE : \mathbb{N} \to \mathbb{R}^d \tag{16}$$

is required to inject position information into the model.

The QFT is a unitary transformation that maps a discrete basis $|p\rangle$ into the Fourier basis:

$$QFT(|p\rangle) = \frac{1}{\sqrt{L}} \sum_{k=0}^{L-1} e^{2\pi i p k/L} |k\rangle \tag{17}$$

This transformation is mathematically equivalent to the DFT but applied to quantum states, exploiting *interference* by encoding position information as phase shifts. This allows position-dependent features to interact in a manner like *entanglement*, important for modeling long-range dependencies, which is crucial for transformer architectures. This may be conveniently implemented using Hadamard and Controlled-phase gates for quantum circuits adaptation, as shown in Fig. 1.

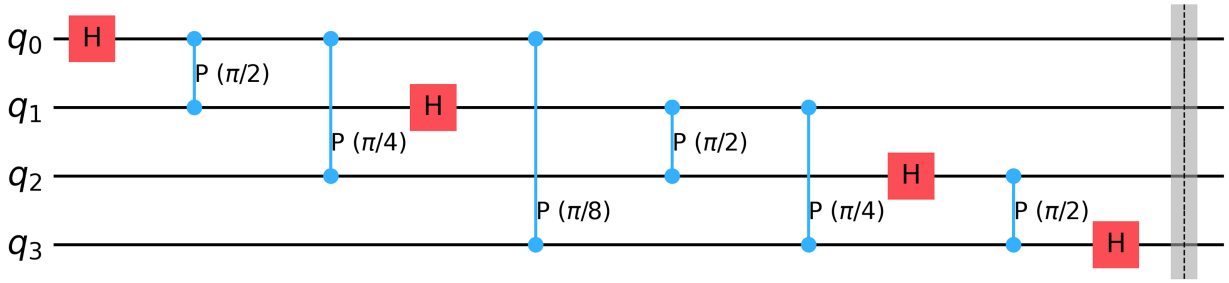

Figure 1: QFT circuit using hadamard & phase gates.

Inspired by the QFT, Eq. (18) encodes positional indices into a complex-valued frequency space, allowing for efficient representation of periodicity using fewer parameters compared to its classical counterpart:

$$\Psi(p, k) = \frac{1}{\sqrt{L}} e^{2\pi i p k/L} \tag{18}$$

This naturally encodes position differences as relative phase shifts, allowing the model to capture global relationships in the frequency domain, where the classical models would have to learn pairwise dependencies between positions explicitly. Still, however, QFT lacks adaptability. To allow the model to learn optimal frequency encoding dynamically, we introduce trainable phase shifts, ensuring that the encoding remains flexible while preserving the structured advantages of the QFT.

Instead of directly applying the QFT transformation, we adapt its core principles:

- *Learnable Phase Shifts:* Quantum Phase Estimation [22] encodes information using relative changes in phase, thereby allowing coherence to be preserved in quantum systems. Based on this principle, we can employ training of the phase shifts into our encoding using the formula in Eq. (19). This allows us to

dynamically modify the relationship of the phases according to their positional embeddings, which will improve the way we model sequences.

$$\theta_p = f_{\text{learnable}}(p) \tag{19}$$

Fig. 2 shows the relationship between learnable phase shifts and position. The x-axis corresponds to the index of positions. The y-axis shows the learned phase shift (θ_p in radians). The gradual increase illustrates the organizational structure in which repositioning of phase shifts occurs, enabling the model to learn the best frequency of encoding.

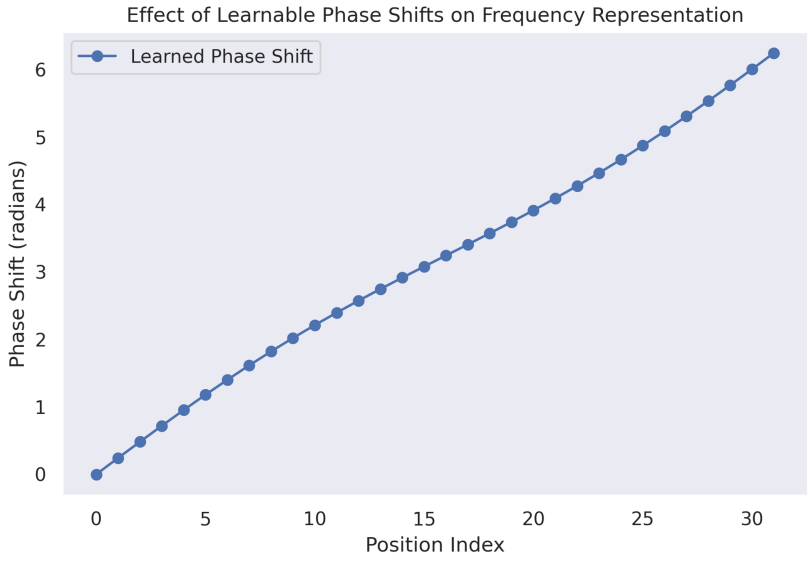

Figure 2: Learned phase shift θ_p.

- *Learnable Angle Embeddings*: Each embedding dimension i is associated with a trainable angular component:

$$\phi_i = g_{\text{learnable}}(i) \tag{20}$$

- *Hybrid Construction via Outer Product*:

$$PE(p, i) = \sin(\theta_p) \cdot \cos(\phi_i) \tag{21}$$

The outer product is supposed to capture periodic positional variations along two axes (sequence & embedding dimension), while ensuring orthogonality of embeddings, similar to Fourier basis functions. Furthermore, its construction results in a structured and diverse encoding space, capturing both global and local positional relationships. Eq. (21), in comparison to Eq. (1), highlights how this approach generalizes the traditional encoding with learnable adaptability. The heatmap shown in Fig. 3 represents the final Hybrid-Adaptive QFT PE matrix, where the x-axis represents embedding dimensions and the y-axis represents positional indices. The color intensity represents encoding values, showing structured periodic patterns.

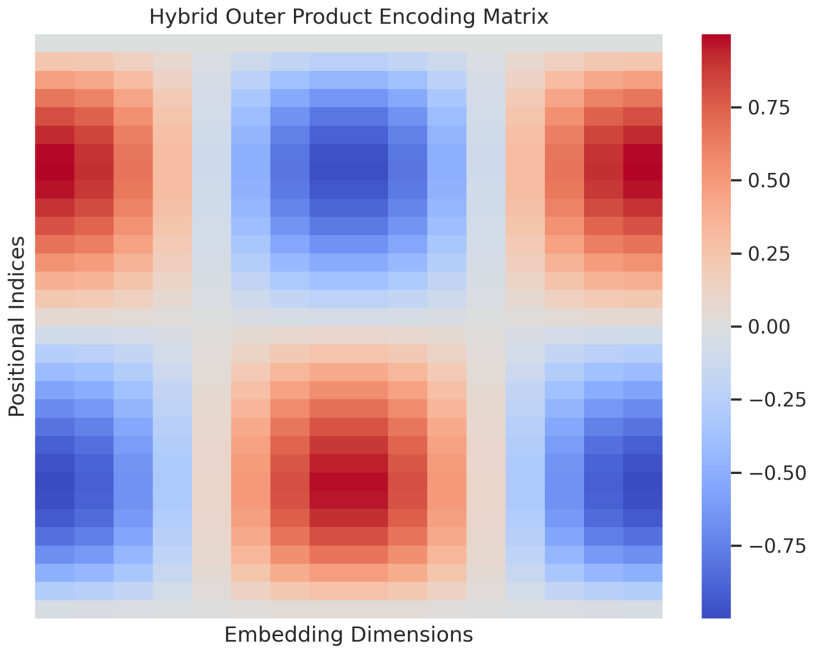

Figure 3: Heatmap of outer product encoding.

Based on the above adaptation of the QFT, the following components define the proposed hybrid-adaptive PE method.

1. *Learnable Phase Shift Matrix*

$$\Theta = [\theta_0, \theta_1, \ldots, \theta_{L-1}]^T \in \mathbb{R}^L \tag{22}$$

where each θ_p is a trainable phase shift inspired by QFT. The following describes the DFT-inspired initialization.

$$\theta_p^{(0)} = \frac{2\pi p}{L} \tag{23}$$

2. *Learnable Angle Embedding Matrix*

$$\Phi = [\phi_0, \phi_1, \ldots, \phi_{d-1}] \in \mathbb{R}^d \tag{24}$$

where each ϕ_i is learnable per feature dimension. The following equation describes the QFT-inspired initialization.

$$\phi_i^{(0)} = \frac{2\pi i}{d} \tag{25}$$

Note that unlike sinusoidal positional encoding, which is fixed and parameter-free, our method introduces Θ and Φ, each of size $L \times d$, which are both learnable parameters. Compared to fully learned positional embeddings, our formulation imposes a structured trigonometric decomposition, improving expressiveness while maintaining interpretability and stability.

3. *Hybrid Outer Product Encoding*

$$PE = \sin(\Theta) \cdot \cos(\Phi)^T \tag{26}$$

For a batch of size B:

$$PE_{\text{batch}} \in \mathbb{R}^{B\times L\times d} \tag{27}$$

To prevent gradient instability, we apply layer normalization:

$$\tilde{PE} = \text{LayerNorm}(PE) \tag{28}$$

The hybrid outer product is a classical formulation of the proposed positional encoding. It is not a quantum algorithm per se, but is derived from the structural principles of the QFT, particularly phase encoding and frequency decomposition. The connection to quantum computing lies in the fact that the parameters Θ and Φ can be mapped to rotation angles in single-qubit gates, enabling a direct realization on quantum circuits. It will help encode angles and entanglement layers to model dependencies as shown in Fig. 4.

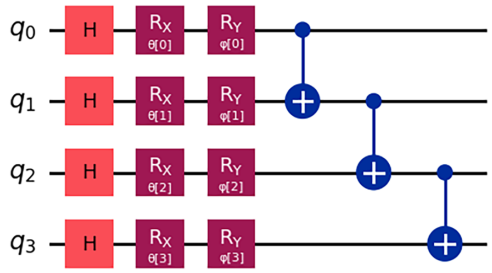

Figure 4: Quantum adaptation of the outer product.

3.4 Key Analytical Properties

The following properties provide analytical intuition regarding the proposed encoding. These are not stated as universal guarantees after end-to-end transformer training, since the learnable parameters are optimized jointly in a non-convex setting.

1. *Positional Distinguishability:*
   When learned phase parameters θ_p remain distinct over positions within a monotonic interval, the resulting encodings tend to preserve positional separability:

   $$PE(p, i) = \sin(\theta_p)\cos(\phi_i) \tag{29}$$

   Under distinct θ_p values, different positions generally produce distinguishable feature responses, supporting positional discrimination.
2. *Local Smoothness and Relative Structure:*
   For nearby positions p_1 and p_2, if the learned mapping $\theta_p = f_{\text{learnable}}(p)$ remains smooth, then bounded positional changes induce bounded encoding changes:

   $$|PE(p_1) - PE(p_2)|^2 = (\sin(\theta_{p_1}) - \sin(\theta_{p_2}))^2 \sum_{i=0}^{d-1} \cos^2(\phi_i) \tag{30}$$

   This may help retain local sequential relationships. The scatter plot shown in Fig. 5 compares original positional distances (x-axis) with encoded distances (y-axis). Since distances between encoded representations maintain proportionality with respect to original positions, this encoding effectively retains sequential relationships, which is crucial for transformers.

3. *Approximate Feature Decorrelation:*

When angle parameters ϕ_i are sufficiently dispersed across embedding dimensions, the resulting cosine components reduce redundancy across features:

$$\langle PE(p), PE(q) \rangle = \sum_{i=0}^{d-1} \sin(\theta_p) \sin(\theta_q) \cos^2(\phi_i) \tag{31}$$

This can promote diverse positional channels and may reduce feature redundancy and encourage channel diversity.

4. *Empirical Optimization Stability:*

Although global convergence guarantees are not available for non-convex transformer training objectives, the proposed encoding uses smooth trigonometric parameterization and standard gradient-based optimization. In practice, the training curves reported in Section 4 indicate stable optimization behavior across repeated runs.

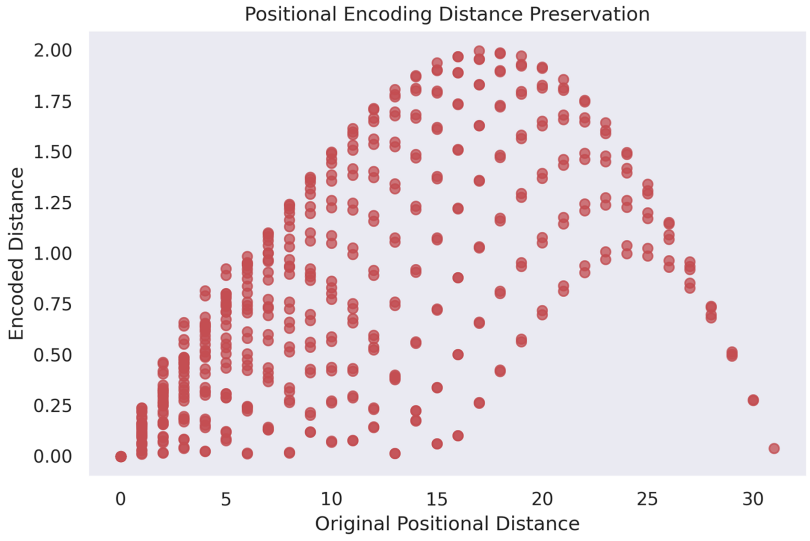

Figure 5: Illustrative relationship between original positional distances and encoded distances.

3.5 Complexity Analysis

The proposed encoding computation consists of:

- *Phase and Angle Computation*: $O(L + d)$ due to vector operations.
- *Outer Product (Matrix Multiplication)*: $O(Ld)$.
- *Layer Normalization*: $O(Ld)$.
- *Total Complexity*: $O(Ld)$, compared to QFT's $O(N^2)$, making it efficient for Transformer-based models.

Although the proposed method is inspired by quantum principles, it is designed for classical transformers. However, its structure aligns well with quantum-enhanced models, making it a structurally relevant candidate for future exploration on near-term quantum hardware.

3.6 Algorithm

Algorithm 1 summarizes the proposed learnable QFT-guided positional encoding method and explicitly corresponds to the mathematical formulation developed in Section 3.3. In particular, the learnable phase

shifts follow Eq. (19), the hybrid outer-product construction follows Eq. (21), and the final normalized encoding follows the batch formulation described thereafter. The algorithm outputs a positional encoding tensor that is added to token embeddings before transformer processing.

Algorithm 1: Proposed learnable QFT-guided PE method

Require: Sequence length L, Embedding dimension d, Batch size B
Ensure: Positional encoding tensor $\tilde{PE} \in \mathbb{R}^{B\times L\times d}$

Initialize learnable phase shifts:
$\Theta^{(0)} = \left[\theta_0^{(0)}, \theta_1^{(0)}, \ldots, \theta_{L-1}^{(0)}\right]^T, \quad \theta_p^{(0)} = \frac{2\pi p}{L}$
Initialize learnable angle embeddings:
$\Phi^{(0)} = \left[\phi_0^{(0)}, \phi_1^{(0)}, \ldots, \phi_{d-1}^{(0)}\right], \quad \phi_i^{(0)} = \frac{2\pi i}{d}$
Define LayerNorm operator $\mathcal{LN}(\cdot)$
function Learnable-QFT-Encoding($position\_ids$)
    $B \leftarrow$ batch size of $position\_ids$
    Update learnable phase parameters using Eq. (19):
$\theta_p = f_{\text{learnable}}(p), \quad p = 0, \ldots, L-1$
    Update learnable angle parameters:
$\phi_i = g_{\text{learnable}}(i), \quad i = 0, \ldots, d-1$
    Compute phase vector:
$S = \sin(\Theta) \in \mathbb{R}^{L\times 1}$
    Compute angle vector:
$C = \cos(\Phi) \in \mathbb{R}^{L\times d}$
    Construct hybrid positional encoding using Eq. (21):
$PE = S \cdot C \in \mathbb{R}^{L\times d}$
    Expand across batch dimension:
$PE_{\text{batch}} = \text{Repeat}(PE, B)$
    Apply normalization:
$\tilde{PE} = \mathcal{LN}(PE_{\text{batch}})$
    return $\tilde{PE}$
end function

4 Experimental Setup and Results

4.1 Experimental Setup

All experiments were conducted on the Louisiana Optical Network Infrastructure (LONI) using a QBD node equipped with NVIDIA A100 GPUs (40 GB) and 256 GB of RAM. We used PyTorch 2.1.0, Transformers 4.39.3, and Qiskit 1.0.0 for classical and quantum-based model implementations.

Each PE method was evaluated under consistent conditions using the GPT-2 base model, with sequence lengths ranging from 128 to 1024 tokens. For inference-only tests (e.g., probing positional embeddings), models were evaluated without retraining.

For hardware benchmarking, Qiskit Aer was used alongside FakeLima (a noisy simulator mimicking IBM's Lima backend) to assess quantum circuit characteristics such as depth, CNOT count, and fidelity.

Dimensionality reduction techniques (PCA and t-SNE) were implemented via scikit-learn, and all matrix rank calculations were performed using Torch's linear algebra module.

4.2 PE Methods' Suitability to GPT-2 & BERT

To assess the compatibility of the transformer-based architecture with different types of quantum-inspired PE methods, we evaluate it by injecting all PE methods into the model during inference rather than requiring retraining through extensive testing. This is done using light-probing methods, allowing us to determine whether the PE-encoded positional information can be interpreted by the model being examined (GPT2). In addition to allowing interpretation, this type of configuration will help demonstrate one specific way in which the PE methods can adapt, depending on the length of a given sequence and how well they provide the necessary positional information for a model.

Tables 2–4 report the perplexity scores for all seven PE methods under three input conditions—clean, scrambled, and reversed—across sequence lengths of 128, 256, 512, and 1024 tokens. Each method was evaluated under identical inference settings to ensure a fair comparison.

Table 2: Perplexity on clean text across methods.

Encoding Method	128	256	512	1024
Basis	47.35	51.05	28.78	15.95
Superposition	5779.31	7257.01	6590.69	6896.80
Amplitude	234.14	302.51	203.01	270.79
Angle	4849.75	7163.57	7202.00	7152.30
QAM	7274.91	9380.97	7940.93	8518.14
QFM	5464.47	7516.34	6556.45	7123.09
QFT-Inspired	48.45	51.84	29.03	15.88

Table 3: Perplexity on scrambled text across methods.

Encoding Method	128	256	512	1024
Basis	1133.22	908.16	666.81	572.71
Superposition	7081.35	9019.59	8301.20	8802.10
Amplitude	1797.81	2018.86	1339.96	1619.66
Angle	5864.22	8763.36	8716.41	8444.93
QAM	8722.02	12,513.90	10,062.54	10,085.68
QFM	3352.45	4614.34	3995.93	4171.58
QFT-Inspired	1130.84	892.10	661.70	571.95

Table 4: Perplexity on reversed text across methods.

Encoding Method	128	256	512	1024
Basis	980.19	767.29	590.62	480.00
Superposition	4370.01	4087.88	4277.27	4970.03
Amplitude	1773.10	1260.95	1211.21	1184.23
Angle	3333.30	3456.25	4753.10	5166.04
QAM	4259.79	3824.50	4926.27	5428.89
QFM	2402.23	2254.32	2451.89	2838.79
QFT-Inspired	980.51	772.79	593.50	480.68

Analysis of the result shows that Superposition, QAM, and QFM have both high and unstable perplexity for structured and perturbed inputs. This indicates that the different methods may collapse any positional coherence present due to various measurement-related effects, and/or add too much complexity to the associated circuit structure (extraneous information) without providing consistent positional guidance. Therefore, no long-term sequential dependencies (or lack of) exist within the targeted language; they will show very poor performance when attempting to capture or represent such dependencies (postural dependencies).

An equivalent evaluation of the PE methods was also conducted by directly injecting the different encodings into a pre-trained BERT model during inference, without any additional training or fine-tuning. Each PE encoding replaces the default positional embeddings in the BERT model, and the monthly loss for masked language modeling (MLM) was measured across clean, scrambled, and reversed versions of the WikiText-103 dataset. The resulting performance result (shown in Table 5) shows that the structured, phase-based encoder (PE) shows significantly lower loss performance compared to all other methods at all levels of testing; therefore, indicating that it is ideally compatible with the BERT meditative attention-based mechanism, while also enabling it to adequately maintain the positional information, without requiring additional training to be effective.

Table 5: MLM loss for BERT. Loss values are computed on WikiText-103 under clean, scrambled, and reversed input.

Encoding	Clean Text			Scrambled Text			Reversed Text		
	128	256	512	128	256	512	128	256	512
Amplitude	9.33	9.80	9.77	9.35	9.80	10.15	8.52	8.73	8.79
Angle	9.65	9.78	9.66	9.78	9.77	9.65	9.76	9.44	9.52
Basis	2.61	2.44	2.49	2.15	2.35	2.35	2.08	3.28	2.78
QAM	8.55	8.96	9.26	8.64	8.96	9.24	8.48	8.78	8.96
QFM	8.51	8.69	9.16	8.52	8.66	9.02	8.27	8.36	8.94
Superposition	8.65	8.79	9.20	8.75	8.79	9.15	8.72	8.59	8.87
QFT-Inspired	1.26	1.50	4.79	1.79	1.75	1.88	2.29	1.88	2.20

BERT has shown greater effectiveness than GPT-2, likely due to differences in architectural and objective design. In particular, BERT uses bidirectional masked language modeling, explicitly using positional information to reconstruct missing tokens based on surrounding context. Alternatively, GPT-2 uses causal autoregressive decoding with respect to the explicit left-to-right ordering when reconstructing tokens; the effects of positional encoding modifications would likely be larger for BERT than for GPT-2.

Based on these diagnostic results, we will eliminate the Superposition, QAM, and QFM position encoding schemes from further consideration and focus on position encoding methods aligned with GPT-2's internal attention mechanism and robust Basis Encoding, Amplitude Encoding, and the proposed QFT-Inspired Encoding. Although Angle Encoding showed high perplexity, its consistent similarity across all input types suggests that it preserves the order of structured positional information, warranting further investigation.

4.3 Positional Embedding Structures

To better understand the qualitative characteristics of each PE scheme, we visualize the first 128 position vectors (along 64 embedding dimensions) using 2D heatmaps. Each encoding class—`Basis`

`Encoding`, `AmplitudeEncoding`, and the proposed `LearnableHybridQFTEncoding`—generates a distinct pattern that reveals how positional information is distributed and encoded.

As shown in Fig. 6, the *BasisEncoding* produces a sparse identity matrix with a one-hot-encoded value at each position, meaning this representation can be identified in isolation and cannot generalize well or produce continuous results, leading to high memory overhead.

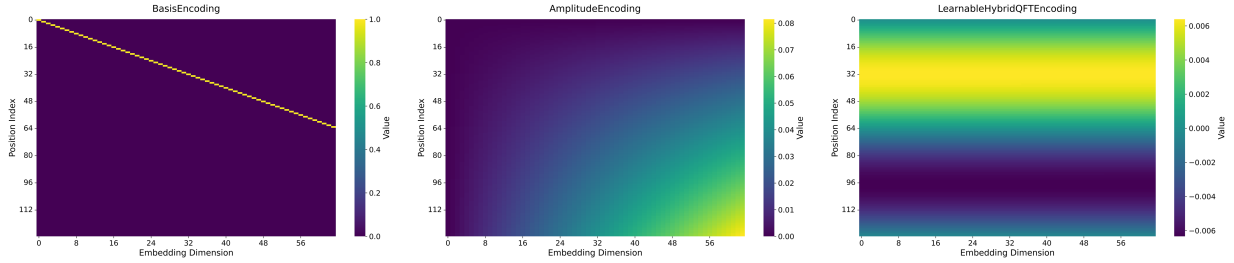

Figure 6: Heatmap visualizations of positional embeddings for 128 positions and 64 embedding dimensions. Left: `basis`; center: amplitude; right: proposed `QFT-inspired`.

The *AmplitudeEncoding*, on the other hand, does provide a continuous gradient between the position and the embedding axes. As a result, the relative position is represented using a fixed scale, leading to a lower rank and lower expressivity. Therefore, this encoding struggles to represent complex positional relationships.

Conversely, the proposed encoding using *LearnableHybridQFTEncoding* leads to a structured representation with sinusoidal patterns in the position dimension that provide frequency/phase-based modulation, similar to Fourier Basis Functions. Therefore, this encoding is both dense and continuous, as well as learnable, as it adapts to the downstream task and continues to optimize the transformer throughout training, resulting in superior performance.

These patterns are corroborated by our quantitative results, which show that the proposed QFT-Inspired encoding offers an acceptable trade-off between expressiveness and efficiency (i.e., both density and learnability).

4.4 Post-Training Results

4.4.1 Perplexity Values with WikiText-103

Tables 6 and 7 show the performance of Basis, Amplitude, Angle, and the proposed PE method when explicitly trained with GPT-2 on 5% and 10% splits of the WikiText-103 dataset [23], respectively. This low-resource setting allows assessment of data efficiency and optimization stability when training data are limited. While such settings are informative for comparative analysis, broader large-scale evaluations using the full corpus remain an important direction for future work.

Table 6: Comparison of PE techniques with training on WikiText-103 using 5% data.

PE Method	Avg. Loss	Val. Perp.	Test Loss	Test Perp.	Manual Avg. Perp.	Manual Med. Perp.
Basis	0.7839	2.1900	0.8061	2.2393	23.2390	23.0516
Amplitude	0.7251	2.0650	0.8100	2.2479	27.0448	24.1927
Angle	1.0214	2.7772	1.0403	2.8301	61.1618	58.0017
QFT-Inspired	0.7860	2.1947	0.8060	2.2390	23.4383	23.1656

Table 7: Comparison of PE techniques with training on WikiText-103 using 10% data.

PE Method	Avg. Loss	Val. Perp.	Test Loss	Test Perp.	Manual Avg. Perp.	Manual Med. Perp.
Basis	0.7171	2.0485	0.8016	2.2291	26.4191	23.3974
Amplitude	0.7260	2.0667	0.8097	2.2473	27.2390	24.5891
Angle	1.0149	2.7593	1.0996	3.0030	91.0670	87.3828
QFT-Inspired	0.7188	2.0519	0.8029	2.2321	26.5647	23.6934

On both splits, our method consistently matches the strongest competing method—Basis Encoding—in terms of validation and test perplexity, as well as in manual evaluations (average and median perplexity over generated sequences). While Amplitude encoding scheme remains a reasonable low-rank alternative, it consistently yields slightly higher perplexity, suggesting less precise positional grounding. Angle encoding performs significantly worse across all metrics, indicating that fixed sinusoidal patterns are ill-suited for autoregressive models like GPT-2, particularly in low-data regimes.

We repeat the same experiment on BERT that reveals similar outcomes, summarized in Table 8. The two top competing alternatives to the proposed PE method are Basis and Amplitude. It is evident that Amplitude encoding performs significantly worse in manual average (Avg.) and median (Med.) perplexity scores compared to both Basis and QFT-inspired, suggesting that its representational capacity may be insufficient for BERT's masked token reconstruction—likely due to low variance or excessive smoothness in encoding spectrum.

Table 8: Comparison of BERT fine-tuning with different PE strategies using 5% WikiText-103 data.

PE Method	Avg. Val. Loss	Test Loss	Manual Avg. Perp.	Manual Med. Perp.
Basis	12.6003	12.7498	2.2710	2.1312
Amplitude	12.5999	12.6750	3.5573	3.2990
QFT-Inspired	12.0031	12.2665	2.7557	2.6007

Beside these comparable perplexity values, the proposed, QFT-Inspired, encoding approach delivers additional benefits: (i) a learnable, structured representation inspired by the QFT, (ii) preserved injectivity and relative positional distances, and (iii) compatibility with parameter-efficient training. The proposed encoding achieves a balance between expressiveness, adaptability, and architectural simplicity—making it a strong candidate for both classical and quantum-enhanced transformer architectures.

4.4.2 Hardware Efficiency and Scalability

Table 9 reveals that while Basis encoding incurs the highest memory usage and runtime, the proposed QFT-Inspired encoding maintains low runtime and efficient memory scaling, particularly for short to medium sequence lengths. Compared to Amplitude encoding—which is compact but less expressive—QFT-Inspired strikes a competitive trade-off between expressiveness and efficiency between expressiveness, adaptability, and compute efficiency. Notably, QFT-Inspired achieves runtime parity with the most efficient method (Amplitude), while significantly outperforming the baseline (Basis) across all sequence lengths. These trends position QFT-Inspired encoding as a practical and scalable choice for real-world transformer inference workloads.

Table 9: Runtime (ms) and peak memory (MB) across sequence lengths for each encoding method.

Metric\Seq. Len.	Basis					QFT-Inspired					Amplitude				
	128	**256**	**512**	**1024**	**2048**	**128**	**256**	**512**	**1024**	**2048**	**128**	**256**	**512**	**1024**	**2048**
Runtime (ms)	0.545	0.586	0.584	0.615	0.615	0.075	0.074	0.588	0.567	0.589	0.046	0.045	0.560	0.561	0.561
Memory (MB)	2.25	2.25	2.25	6.01	12.02	0.76	1.51	3.02	6.02	12.03	0.38	0.76	1.51	3.01	6.03

4.5 Circuit Depth and Complexity

The various PE methodologies can be compared with respect to circuit complexity and circuit depth from a hardware perspective. Since basis encoding has the least complexity of all these methods, it has a circuit depth of one, meaning that the required gates used to implement this methodology are local (there is no interference/entanglement). This extremely shallow implementation of Basis Encoding makes it well-suited for NISQ devices, as it provides the fastest, lowest-noise execution times (noise on NISQ devices is a major consideration). Furthermore, none of the basis encodings use multi-qubit gates; as a result, zero CNOT gates are required, thereby providing further resilience against decoherence and gate noise. The shallow implementation of Basis Encoding, however, limits its ability to model more complex relationships amongst the various positions. Each qubit is individually treated, and, therefore, Basis Encoding does not establish any interaction/correlation amongst the qubits at different positions, thus reducing its expressiveness when it comes to capturing the positional dependencies within sequences. In summary, it executes very quickly and is very resilient to noise, but is too simplistic to provide a rich representation of the underlying structure.

In contrast to Basis encoding, Amplitude encoding exhibits a high circuit depth of 13, mainly driven by complex encoding of scalar amplitude values across qubits. This indicates increased time, error accumulation, and fragile execution on NISQ devices. With 8 CNOTs, Amplitude encoding makes heavy use of entanglement for encoding, which results in high gate error probability on current hardware. This–in turn–poses serious reliability issues in NISQ settings.

For the proposed QFT-Inspired encoding method, we observe a moderate circuit depth of 9, arising from Hadamard + Controlled Phase operations. This enables structured interference, capturing global position information, yet, the method is still feasible on NISQ due to absence of entangling gates like CNOTs—consistent with Basis encoding, making it enjoy the best of both worlds.

4.6 Noise Resilience of PE Circuits

To evaluate the robustness of PE circuits under realistic quantum noise, we simulate each encoding using the noise model of `FakeLima`, a representative NISQ backend. For each encoding, we prepare a representative 5-qubit quantum circuit implementing reduced encoding primitives associated with that method and measure the fidelity between its output in an ideal noiseless simulation and under noise. All simulations are conducted using the `AerSimulator` in `statevector` mode to extract quantum state fidelity directly.

Under the evaluated simulation setting, the observed fidelity outcomes indicate differences in robustness among the examined encodings:

- **Basis Encoding** achieves perfect fidelity ($F = 1.0000$), owing to its extremely shallow structure (depth 1) and the absence of entangling gates.
- **QFT-Inspired Encoding** also achieves perfect fidelity ($F = 1.0000$), despite having a deeper circuit (depth 9). Notably, it avoids CNOT operations entirely and uses only controlled-phase gates, which exhibit stronger noise tolerance on NISQ devices.

- **Amplitude Encoding** fails under noise, returning invalid or non-unitary state vectors. These failures are attributed to the presence of entangling gates (e.g., CNOTs) that cannot be transpiled reliably into the noise model's basis gates.

These preliminary findings suggest that the QFT-Inspired encoding may offer favorable robustness characteristics under the evaluated circuit-level setting and representative backend noise model. Unlike Amplitude encoding, which requires multi-qubit entanglement, the QFT-Inspired approach also demonstrated competitive geometric expressiveness (see Section 4.7) while exhibiting encouraging fidelity behavior in the present simulation study. Further validation is required before drawing conclusions regarding practical hardware deployment.

We emphasize that the present noise evaluation is an exercise in circuit feasibility employing a NISQ backend noise model (*FakeLima*). The tested circuits with 5 qubits are not intended to be a one-to-one mapping of all elements of the full transformer positional encoding tensor (e.g., 1024 sequences and 768 embeddings). Instead, they contain low-dimensional analogs of the full tensor, i.e., phase rotation, structured frequency mapping, and gate-depth-related terms, for each encoding scheme. Thus, we advise that the calculated fidelity values represent only a qualitative indication of the hardware-level propagation tolerance of the encoding approach. As noted subsequently, further assessments of scalability, tensor decomposition, circuit distribution across multiple hardware platforms, and noise robustness will be completed at a later date.

4.7 Geometric Analysis (GA) via t-SNE and PCA

Fig. 7 reveals PCA and t-SNE projection for Basis encoding. While the PCA plot appears scattered and isotropic—almost random exhibiting no progression across position indices, the t-SNE plot exhibits distinct grid-like dispersion—consistent with one-hot or orthogonal encoding. These results confirm that each position is encoded as a completely separate token with no visible learned or relative structure, and no generalization across positions is visible.

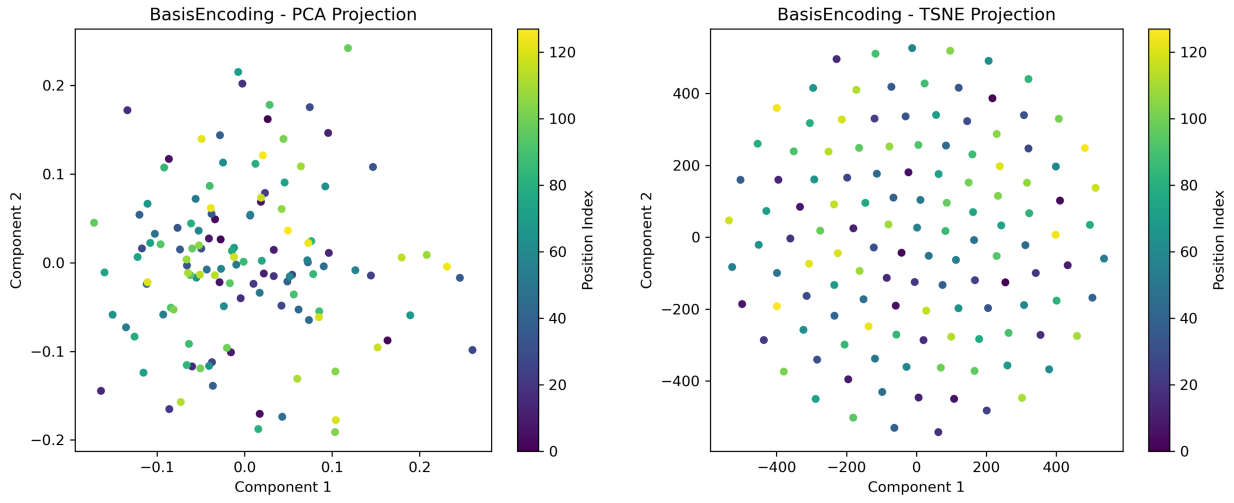

Figure 7: Geometric analysis of basis via (left): PCA and (right): t-SNE.

Fig. 8 presents GA for Amplitude encoding. Its PCA projection shows a nonlinear but smooth curvature, suggesting gradual positional variation. The t-SNE projection, on the other hand, exhibits a structured arc-shaped manifold, indicating clear ordering of positions. This reveals that Amplitude encoding captures smooth global position trends, albeit in a slightly compressed form.

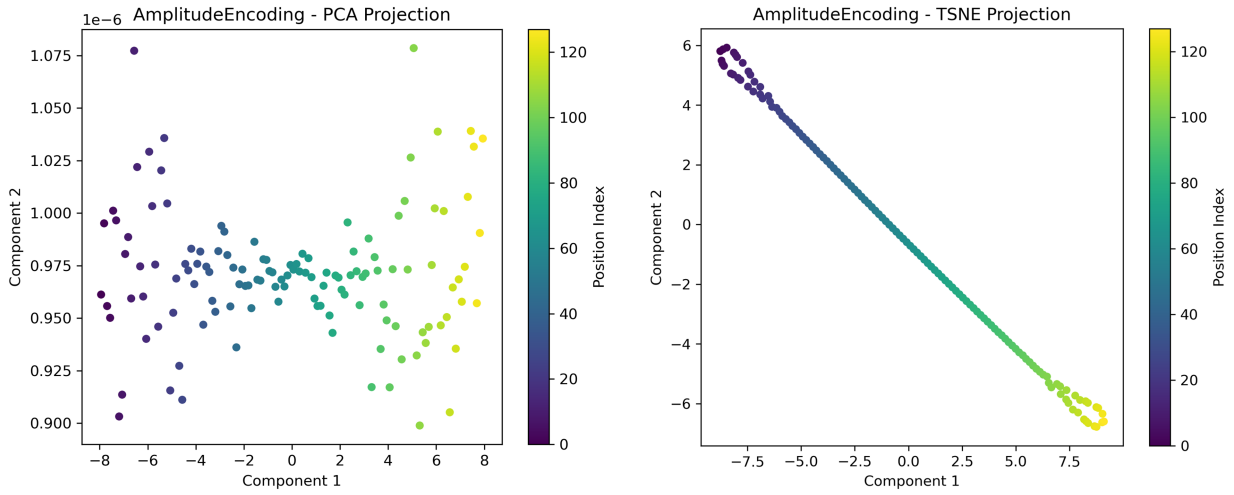

Figure 8: Geometric analysis of amplitude via (left): PCA and (right): t-SNE.

Fig. 9 shows GA for the proposed adaptive QFT-Inspired encoding method. Its PCA projection shows multiple clusters and directionality, hinting at cyclic or wave-like structure. The t-SNE projection, on the other hand, is almost identical to Amplitude—exhibiting smooth, arc-shaped spread. From these, we conclude that the proposed PE method mimics spectral decomposition of positions (like QFT) and reveals higher expressiveness and generalizability.

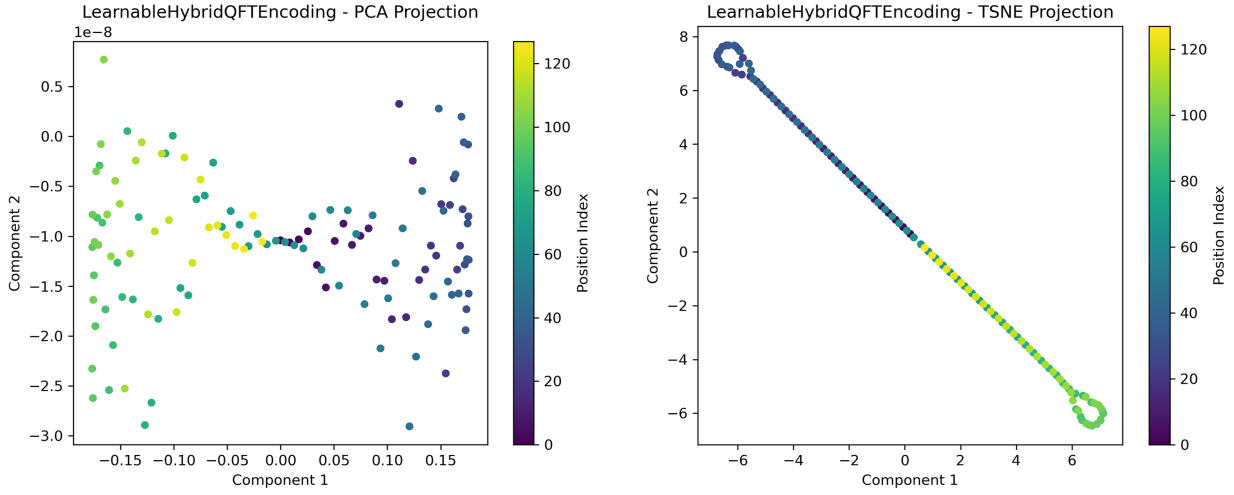

Figure 9: Geometric analysis of QFT-inspired encoding via (left): PCA and (right): t-SNE.

4.8 Ablation Studies

To assess the individual contributions of key components in the proposed QFT-Inspired PE, we create its two variants: one without the phase component (position-specific variation across sequence length) and the other without the scaling Term (that controls the magnitude of PE). Each variant was evaluated using perplexity across four sequence lengths: 128, 256, 512, and 1024, and subsequently compared with the complete version. Once again, evaluations are performed under clean, scrambled, and reversed text scenarios. The results are summarized in Table 10.

Table 10: Ablation study on QFT-inspired PE.

Encoding Variant	128	256	512	1024
Clean Text Perplexity				
QFT-Inspired (Complete)	48.45	51.84	29.03	15.88
QFT-No Scaling	937.77	2146.04	2700.57	3153.50
QFT-No Phase	4006.72	5873.60	4877.46	5356.35
Scrambled Text Perplexity				
QFT-Inspired (Complete)	1130.84	892.10	661.70	571.95
QFT-No Scaling	2639.40	3755.95	3690.97	4066.85
QFT-No Phase	5227.51	7542.42	6069.72	6581.17
Reversed Text Perplexity				
QFT-Inspired (Complete)	980.51	772.79	593.50	480.68
QFT-No Scaling	4366.39	3125.71	3127.53	3335.65
QFT-No Phase	3572.98	4029.70	4133.22	4757.14

The results confirm that the full QFT-Inspired encoding shows superior performance with the lowest perplexity across all sequence lengths. No Scaling variant results in exploding perplexity, highlighting the importance of keeping positional values within a bounded range. Likewise, No Phase variant shows a dramatic spike in perplexity (e.g., >5000 at 1024 tokens), confirming that phase modulation is critical for encoding positional order.

The impact of ablation grows with sequence length, meaning the benefits of both phase and scaling become more pronounced as input sequences get longer.

4.9 Training Stability and Significance Analysis

To evaluate training stability and assess statistical significance of performance gains, we trained our QFT-Inspired GPT-2 model from scratch on 2% of the WikiText-103 dataset across three independent trials. Each trial ran for two epochs, and the per-epoch training loss was recorded.

We observed consistent improvements in loss between Epoch 1 and Epoch 2 across all trials. The standard deviation across trials was low (0.064 for Epoch 1 and 0.037 for Epoch 2), indicating stable training behavior. A paired t-test between epoch-wise losses yielded a t-statistic of 23.90 with a p-value of 0.0017, confirming that the observed improvements are statistically significant. Fig. 10 shows a clear shift in the loss distribution, with Epoch 2 losses both lower and less dispersed.

These results demonstrate that our model exhibits stable optimization dynamics and reliably benefits from additional training, even when initialized from scratch.

4.10 Summary of Results

Through these experiments, we have demonstrated that the proposed QFT-Inspired PE method is both expressive and generalizable, and it consistently outperformed or matched existing quantum-inspired methods. For instance:

1. In contrast to Amplitude encoding, the proposed method achieves slightly better perplexity with zero CNOTs and moderate depth, making it more suitable for NISQ devices.

2. Compared to Basis encoding, it delivers similar perplexity performance while enabling structured, trainable, and scalable representations, preserving positional continuity.
3. GA confirms that it organizes positional embeddings into spectrally coherent, low-dimensional manifolds, capturing global sequence structure better than orthogonal (Basis) or low-rank (Amplitude) alternatives.
4. In terms of runtime and memory footprint, QFT-Inspired matches the most efficient Amplitude encoding across sequence lengths, validating its scalability for real-world transformer workloads.
5. Training stability experiments reveal low standard deviation across trials and a statistically significant epoch-wise drop in loss, proving that it supports stable optimization with only 2% of data.

Altogether, the proposed QFT-Inspired encoding demonstrates a balanced blend of hardware realism, theoretical robustness, and favorable empirical performance under the evaluated settings, making it a strong candidate for both classical and quantum-enhanced transformer architectures.

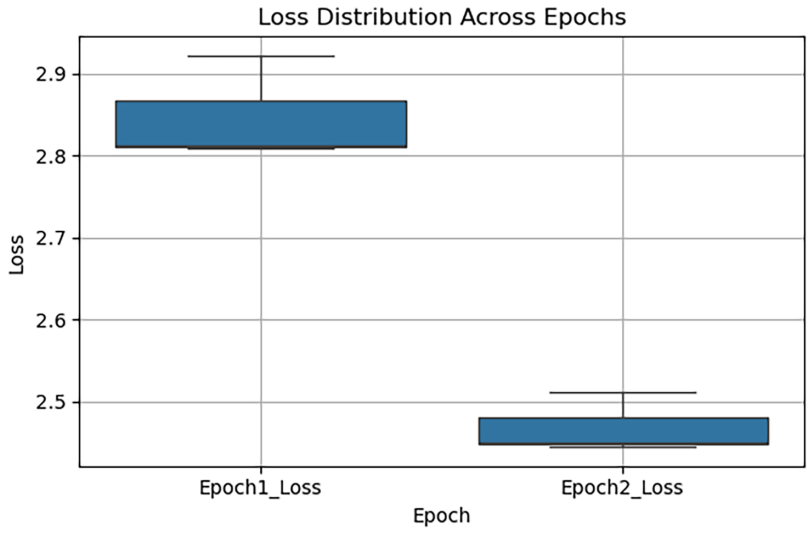

Figure 10: Epoch-wise loss distribution.

5 Conclusion

This work introduced a Quantum Fourier Transform-inspired Positional Encoding scheme that leverages Quantum Fourier principles while remaining fully compatible with classical transformer architectures. The method integrates adaptive phase shifts, learnable angle encodings, and scaling mechanisms, resulting in a learnable hybrid formulation that preserves spectral richness without requiring entanglement. Extensive evaluation across clean, scrambled, and reversed text demonstrated that QFT-inspired encoding achieves state-of-the-art perplexity, especially at longer sequence lengths. Compared to existing quantum-inspired baselines such as Basis, Amplitude, and Angle, the proposed encoding provides a superior trade-off between expressiveness, generalization, and hardware feasibility. Furthermore, the method exhibits robust training stability, as demonstrated by low inter-trial variance and statistically significant improvements in loss. Beside providing competitive perplexity and increased stability, the proposed positional encoding method maintains a shallow circuit depth, thereby offering a practical path for quantum-enhanced NLP, especially in near-term devices. This work bridges a crucial gap between quantum theory and transformer-based deep learning, laying the foundation for future explorations of quantum-native positional representations.

Notably, our experiments were limited to a single benchmark dataset (WikiText-103), utilized only up to 10% of the available training data, and not on a more favorable Masked Language Model such as BERT. Moreover, whether the proposed method generalizes across diverse language distributions is something that will require exploring cross-lingual or domain-specific corpora. Finally, while the positional circuit was designed with quantum feasibility in mind, actual quantum hardware deployment and noise-aware circuit optimization remain unexplored. Altogether, these limitations present a promising direction of research, to be built upon the proposed QFT-inspired positional encoding method.

Acknowledgement: The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work.

Funding Statement: The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2025/01/35090).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Sara Tehsin and Syed Rameez Naqvi; methodology, Abdulrahman; Software, Syed Rameez Naqvi, Tallha Akram and Abdulrahman Alabduljabbar; validation, Tallha Akram and Abdulrahman Alabduljabbar; formal analysis, Tallha Akram and Sara Tehsin; investigation, Syed Rameez Naqvi; resources, Sara Tehsin and Meshal Alharbi; data curation, Syed Rameez Naqvi; writing—original draft preparation, Sara Tehsin, Syed Rameez Naqvi and Tallha Akram; writing—review and editing, Tallha Akram and Meshal Alharbi; visualization, Abdulrahman Alabduljabbar; supervision, Tallha Akram; project administration, Tallha Akram and Abdulrahman Alabduljabbar; funding acquisition, Tallha Akram. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data–Wikitext-103—is openly available in the following public repository https://huggingface.co/datasets/Salesforce/wikitext.

Ethics Approval: Not applicable.

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

References

1. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017;30:6000–10. doi:10.65215/nxvz2v36.
2. Haviv A, Ram O, Press O, Izsak P, Levy O. Transformer language models without positional encodings still learn positional information. arXiv:2203.16634. 2022. doi:10.48550/arXiv.2203.16634.
3. Widdows D, Aboumrad W, Kim D, Ray S, Mei J. Quantum natural language processing. KI KüNstliche Intell. 2024;38(4):293–310. doi:10.1007/s13218-024-00861-w.
4. Perdomo-Ortiz A, Benedetti M, Realpe-Gómez J, Biswas R. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. Quantum Sci Technol. 2018;3(3):030502. doi:10.1088/2058-9565/aab859.
5. Khatri N, Matos G, Coopmans L, Clark S. Quixer: a quantum transformer model. arXiv:2406.04305. 2024. doi:10.48550/arXiv.2406.04305.
6. Wennberg U, Henter G. Learned transformer position embeddings have a low-dimensional structure. In: Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024); 2024 Aug 16; Bangkok, Thailand. p. 237–44.
7. Puvvada KC, Ladhak F, Serrano SA, Hsieh CP, Acharya S, Majumdar S, et al. SWAN-GPT: an efficient and scalable approach for long-context language modeling. arXiv:2504.08719. 2025. doi:10.48550/arXiv.2504.08719.
8. Wang YA, Chen YN. What do position embeddings learn? An empirical study of pre-trained language model positional encoding. arXiv:2010.04903. 2020. doi:10.48550/arXiv.2010.04903.

9. Wang G, Lu Y, Cui L, Lv T, Florencio D, Zhang C. A simple yet effective learnable positional encoding method for improving document transformer model. In: Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022. Stroudsburg, PA, USA: Association for Computational Linguistics; 2022. p. 456–67.
10. Su J, Ahmed M, Lu Y, Pan S, Bo W, Liu Y. Roformer: enhanced transformer with rotary position embedding. Neurocomputing. 2024;568:127063.
11. Kazemnejad A, Padhi I, Natesan Ramamurthy K, Das P, Reddy S. The impact of positional encoding on length generalization in transformers. Adv Neural Inf Process Syst. 2023;36:24892–928. doi:10.52202/075280-1082.
12. Thabet S, Fouilland R, Djellabi M, Sokolov I, Kasture S, Henry LP, et al. Enhancing graph neural networks with quantum computed encodings. arXiv:2310.20519. 2023. doi:10.48550/arXiv.2310.20519.
13. Munikote N, Li A, Liu C, Stein S. Comparing quantum encoding techniques. arXiv:2410.09121. 2024. doi:10.48550/arXiv.2410.09121.
14. Shukla A, Vedula P. An efficient quantum algorithm for preparation of uniform quantum superposition states. Quantum Inf Process. 2024;23(2):38. doi:10.1007/s11128-024-04258-4.
15. Pagni V, Huber S, Epping M, Felderer M. Fast quantum amplitude encoding of typical classical data. arXiv:2503.17113. 2025. doi:10.48550/arxiv.2503.17113.
16. Rath M, Date H. Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy. arXiv:2311.10375. 2023. doi:10.48550/arXiv.2311.10375.
17. Ventura D, Martinez T. Quantum associative memory. arXiv:quant-ph/9807053. 1998. doi:10.48550/arXiv.quant-ph/9807053.
18. Sakka K, Mitarai K, Fujii K. Automating quantum feature map design via large language models. arXiv:2504.07396. 2025. doi:10.48550/arxiv.2504.07396.
19. Hong X, Jiang C, Qi B, Meng F, Yu M, Zhou B, et al. On the token distance modeling ability of higher RoPE attention dimension. arXiv:2410.08703. 2024. doi:10.48550/arXiv.2410.08703.
20. Aach T. Fourier, block and lapped transforms. In: Hawkes PW, editor. Advances in imaging and electron physics. Vol. 128. San Diego, CA, USA: Academic Press; 2003. p. 1–52.
21. Camps D, Van Beeumen R, Yang C. Quantum fourier transform revisited. Numer Linear Algebra Appl. 2021;28(1):e2331. doi:10.1002/nla.2331.
22. Barbieri M, Gianani I, Goldberg AZ, Sánchez-Soto LL. Quantum multiphase estimation. arXiv:2502.07873. 2025.
23. Merity S, Xiong C, Bradbury J, Socher R. WikiText-103: a large-scale language modeling dataset. arXiv:1609.07843. 2016. doi:10.48550/arXiv.1609.07843.