

Article

Not peer-reviewed version

Coiling Arithmetic: On the Differential Geometry of Helical Representations in LLMs

[Mikołaj Sienicki](#) and [Krzysztof Sienicki](#) *

Posted Date: 9 July 2025

doi: 10.20944/preprints202507.0722.v1

Keywords: AI; LLMs; Arithmetic and Symbolic Reasoning



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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

Article

Coiling Arithmetic: On the Differential Geometry of Helical Representations in LLMs

Mikołaj Sienicki ¹ and Krzysztof Sienicki ^{2,*}

¹ Polish-Japanese Academy of Information Technology, ul. Koszykowa 86, 02-008 Warsaw, Poland, European Union.
² Chair of Theoretical Physics of Naturally Intelligent Systems (NIS[®]), Lipowa 2/Topolowa 19, 05-807 Podkowa Leśna, Poland, European Union.
* Correspondence: niskrissienicki@gmail.com

Abstract

Large language models (LLMs) exhibit surprising proficiency in arithmetic and symbolic reasoning, despite being trained predominantly on natural language data. Recent studies have uncovered that numerical tokens in these models are embedded along smooth, high-dimensional helical curves—structures that encode arithmetic operations through linear translations in a periodic space. In this paper, we formalize this phenomenon as *coiling arithmetic*, a differential-geometric framework in which numbers are represented as points on a multi-frequency helix of the form:

$$\text{helix}(x) = \left[x, \cos\left(\frac{2\pi x}{T_1}\right), \sin\left(\frac{2\pi x}{T_1}\right), \dots \right] \in \mathbb{R}^{2k+1}.$$

We show that arithmetic operations such as addition correspond to vector displacements in helical space, enabling approximate computation via linear transformation. We extend the formalism with helix-based differentiation rules, modular equivalence, modal logic operators, and Gödel-style fixed-point encodings. Connections are drawn to Neural ODEs, Fourier Transformers, and group-equivariant networks. By grounding abstract symbolic operations in continuous geometry, coiling arithmetic offers a new lens on how LLMs internalize number structure and may support robust generalization in neural reasoning systems.

Keywords: AI; LLMs; arithmetic and symbolic reasoning

1. Motivation and Formal Setup

Large language models (LLMs) such as GPT exhibit surprising proficiency in arithmetic tasks, despite lacking explicit symbolic computation modules. Recent empirical studies have revealed that instead of performing brute-force memorization, these models organize numeric representations along smooth, structured trajectories in latent space. Among the most striking findings is the emergence of a helical structure—a geometric encoding that combines linear and periodic components. This phenomenon, referred to as “coiling arithmetic,” forms the foundation of our analysis.

To study this phenomenon precisely, we define a formal mathematical object that captures the observed embeddings.

Definition (Helical Embedding).
Let $k \in \mathbb{N}$, and let $T_i \in \mathbb{R}_{>0}$ for $i = 1, \dots, k$. Define the *helical embedding* as the smooth mapping:
$$\text{helix} : \mathbb{R} \longrightarrow \mathbb{R}^{2k+1}, \quad x \mapsto \left[x, \cos\left(\frac{2\pi x}{T_1}\right), \sin\left(\frac{2\pi x}{T_1}\right), \dots, \cos\left(\frac{2\pi x}{T_k}\right), \sin\left(\frac{2\pi x}{T_k}\right) \right]$$

The function $\text{helix}(x)$ is infinitely differentiable, i.e., $\text{helix} \in C^\infty(\mathbb{R}, \mathbb{R}^{2k+1})$.

This formalism serves as the foundation for our subsequent analysis of addition, differentiation, logic, and representational learning within the coiling arithmetic framework.

Table 1. Notation used for coiling arithmetic and helical embeddings.

Symbol	Description
x	Scalar input value (e.g., integer token or real number)
T	Period of the helical embedding
$\phi(x)$	Helical embedding function: $\phi(x) = (x, \cos(\frac{2\pi x}{T}), \sin(\frac{2\pi x}{T}))$
$\phi_{\text{flat}}(x)$	Flat or linear embedding (e.g., $(x, 0, 0)$)
$\phi_{\text{coil}}(x)$	Periodic or coiled embedding (same as $\phi(x)$)
\mathcal{H}_T	Helical manifold defined by period T ; image of $\phi(x)$
$\theta(x)$	Angular phase: $\theta(x) = \frac{2\pi x}{T}$
$\lfloor x/T \rfloor$	Number of completed coil turns
$\text{Ceil}_T(x)$	Ceiling function: smallest integer $\geq x/T$
$\text{mod}_T(x)$	Modulo projection of x : $x \bmod T$
ϵ_{alias}	Aliasing error from multiple x mapping to same angle
ϕ^{-1}	Inverse projection (if defined), maps back from helix to scalar x
$n \in \mathbb{Z}_{\geq 0}$	Indexing turn number in the coil (e.g., $x = nT$)

2. Coiling Arithmetic in Large Language Models

Recent findings by Kantamneni and Tegmark [7] have unveiled a striking and elegant phenomenon in the internal representations of large language models (LLMs): *numerical tokens are embedded along a helical manifold* in the model’s latent space. This discovery is remarkable, both empirically and theoretically, for several reasons that reshape our understanding of neural arithmetic, geometric embeddings, and the latent structure of computation in modern AI systems.

2.1. Unveiling Hidden Geometric Structure

Contrary to the prevailing assumption that LLMs perform arithmetic via opaque and distributed weight patterns—reminiscent of brute-force memorization—this work reveals that LLMs *spontaneously organize numerical concepts along a smooth, interpretable helix* [8]. In effect, the model reconstructs a trigonometric coordinate system within its latent space, suggesting that neural networks internally rediscover fundamental mathematical principles.

LLMs, without explicit programming, encode numbers through trigonometric structure.

The helical embeddings uncovered in these models are deeply related to *Fourier feature mappings*, which are ubiquitous in signal processing, neural rendering (e.g., NeRF), and Transformer positional encodings [9]. That such structures arise spontaneously implies that *periodicity and sinusoidal bases* may act as universal priors for neural representations.

2.2. Reconstructing Arithmetic as Translation in Phase Space

Rather than simulating discrete logic, LLMs appear to encode arithmetic operations as *vector translations* along a *helical manifold* [10]. This is analogous to how biological brains may encode spatial and temporal dimensions using phase or oscillatory dynamics.

These results challenge the notion that token embeddings are opaque. Instead, they become structured and interpretable—opening the door to *reverse engineering* latent geometry in LLMs [11].

The emergence of helical structure in numeric embeddings hints at a broader pattern: that neural networks build smooth geometric maps of symbolic domains such as algebra, logic, and language [12].

2.3. Implications Across Disciplines

Domain	Implication
AI interpretability	Structured embeddings allow direct probing of reasoning mechanisms
Neuroscience	Supports phase-based models (e.g., grid cells, oscillatory coding)
Cognitive science	Suggests arithmetic and logic may emerge from spatial mappings
AI safety	Enables geometric control and auditing of internal reasoning

The helix representation of numbers in LLMs is a breakthrough in our understanding of how abstract reasoning emerges from gradient-based learning. It shows that large models do not simply memorize or simulate logic, but instead embed arithmetic operations in *spatial, periodic structures* that resemble phase dynamics in both engineered and biological systems.

This discovery lays the groundwork for a new field—*Coiling Arithmetic*—which studies how neural architectures instantiate and manipulate mathematical functions through latent geometry.

This paper formulates the fundamental principles of *coiling arithmetic*—a theoretical framework that integrates these findings into a unified model of how neural systems implement and generalize arithmetic through periodic, multidimensional geometries. We develop formal definitions of latent embeddings, prove core compositional lemmas, and examine the implications of this representational regime for interpretability, generalization, and the theoretical foundations of neural computation.

2.4. Related Representational Architectures

2.4.1. Neural ODEs and Helical Dynamics

Helical embeddings exhibit smooth, differentiable geometry, making them a natural fit for analysis through the lens of Neural Ordinary Differential Equations (NeuralODEs) [3]. The continuous trajectory traced by $\text{helix}(x) \in \mathbb{R}^{2k+1}$ can be interpreted as a parameterized flow in a latent manifold. This aligns with work that models neural computation as a dynamical system trained to follow structured, low-dimensional curves in high-dimensional space.

Implication: Helical arithmetic can be embedded within ODE solvers, where the latent state evolves along sinusoidal manifolds.

2.4.2. Fourier Transformers and Helix Encodings

The sinusoidal structure of the helical embedding directly connects to Fourier feature encodings, which are used in long-sequence transformers and coordinate networks [4]. Just as the helix maps input values to layered trigonometric representations, Fourier Transformers use $\cos(\omega x), \sin(\omega x)$ to represent spatial/temporal structure at multiple frequencies.

Implication: The helix serves as a multiscale positional embedding scheme and may enhance arithmetic generalization when incorporated into transformer attention blocks.

2.4.3. Group Equivariance in Helical Space

The periodic components of the helical embedding are equivariant under modular translation. That is:

$$x \mapsto x + T_i \quad \Rightarrow \quad \text{helix}(x) \approx \text{helix}(x + T_i)$$

This symmetry is directly aligned with the theory of equivariant neural networks [5], which preserve structure under transformations drawn from algebraic groups (translations, rotations, scalings).

Implication: Helical arithmetic can be analyzed using group-equivariant frameworks, suggesting a bridge between neural arithmetic and representation theory.

Definition 2.1 (Ceiling Effect). Let

$$\text{helix}(x) = [\cos(2\pi x/T_1), \sin(2\pi x/T_1), \dots, \cos(2\pi x/T_k), \sin(2\pi x/T_k)]$$

denote a helical embedding into \mathbb{R}^{2k} using multiple periods $T_i > 0$. We define the **Ceiling Effect** as the point beyond which distinct inputs x, x' become *approximately aliased*, i.e.,

$$\|\text{helix}(x) - \text{helix}(x')\|_2 < \varepsilon$$

for some $\varepsilon > 0$, even though $x \neq x'$. This occurs because periodic components may realign approximately at large x , creating ambiguity in representation.

The *ceiling range* \bar{x} is the smallest $\delta > 0$ such that

$$\exists x' \neq x \text{ with } \|\text{helix}(x) - \text{helix}(x')\|_2 < \varepsilon.$$

If the embedding includes a linear term (e.g., $\text{helix}(x) = [x, \cos(\cdot), \sin(\cdot), \dots]$), then representational collisions still occur in the periodic dimensions. In practice, neural models may underweight the linear component, resulting in effective aliasing.

Thus, the ceiling effect identifies a fundamental resolution limit imposed by periodic embeddings, analogous to signal aliasing in frequency-limited systems.

Lemma 1 (Non-Injectivity of Helical Embedding). Let $\phi_T(x) := (x, \cos(\frac{2\pi x}{T}), \sin(\frac{2\pi x}{T}))$ be the helical embedding of a scalar $x \in \mathbb{R}$, where $T > 0$ is the fixed period. Then ϕ_T is not injective on \mathbb{R} . This follows from the fact that cosine and sine are periodic functions with period 2π , so

$$x \equiv x' \pmod{T} \Rightarrow \phi_T(x) \approx \phi_T(x') \text{ in angular components.}$$

We now state a key structural limitation of this embedding.

In particular, for any $k \in \mathbb{Z}$, we have:

$$\phi_T(x) \neq \phi_T(x + kT), \quad \text{but} \quad \phi_T(x) \approx \phi_T(x + kT) \text{ up to phase redundancy.}$$

The angular components

$$\left(\cos\left(\frac{2\pi x}{T}\right), \sin\left(\frac{2\pi x}{T}\right) \right)$$

repeat with period T , causing representational collisions between distinct values.

To make this representational ambiguity precise, we now define the aliasing error between two values that collide in angular space.

Definition 1 (Aliasing Error). Let $\phi_T(x) = (x, \cos(2\pi x/T), \sin(2\pi x/T))$ be the helical embedding with period T . For any two distinct inputs $x \neq x'$, the aliasing error is defined as:

$$\varepsilon_{\text{alias}}(x, x') := \|\phi_T(x) - \phi_T(x')\|,$$

where $\|\cdot\|$ is the Euclidean norm in \mathbb{R}^3 . When $x \equiv x' \pmod{T}$, we have:

$$\cos\left(\frac{2\pi x}{T}\right) = \cos\left(\frac{2\pi x'}{T}\right), \quad \sin\left(\frac{2\pi x}{T}\right) = \sin\left(\frac{2\pi x'}{T}\right),$$

and so the angular components coincide, yielding minimal distance despite different scalar positions.

If $x = 10$, $x' = 10 + T$, then $\phi_T(x)$ and $\phi_T(x')$ share the same angular values, and only differ along the linear dimension.

Helical Embedding with Aliased Points

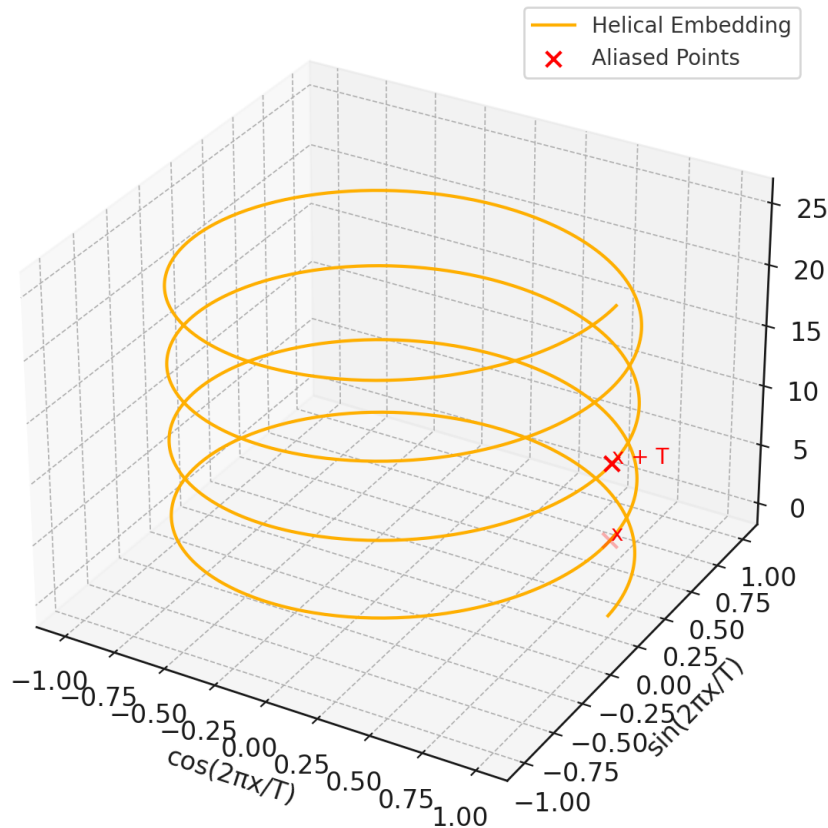


Figure 1. Helical Embedding and Aliasing. A 3D helix illustrates the mapping $\phi_T(x) = (\cos(2\pi x/T), \sin(2\pi x/T), x)$. The points labeled x and $x + T$ lie at different heights but share identical angular components (cosine and sine), leading to aliasing in the embedding. This phenomenon causes distinct numerical values to become indistinguishable in the angular domain, a key source of arithmetic confusion in periodic encodings.

3. Tegmark's Procedure to Discover Helical Structure in LLMs

In an earlier paper [6] leading to the discovery of helical arithmetic, the authors investigated the simple harmonic oscillator to address the intermediate question: “How do transformers model physics?” They concluded that transformers do not invent arbitrary or “alien” strategies, but instead leverage known numerical methods—specifically the matrix exponential method—to model the trajectories of physical systems like the SHO (Simple Harmonic Oscillator), $\ddot{x} + 2\gamma\dot{x} + \omega_0^2 x = 0$.

Through a rigorous analysis of hidden state intermediates—evaluating their encoding strength, correlation with model performance, explanatory power, and robustness under intervention—the authors found strong causal evidence that transformers internally implement the matrix exponential approach to simulate physical dynamics. This finding demonstrates that transformers can spontaneously recover interpretable, human-like physical modeling techniques purely from training data, without being explicitly programmed to do so.

Nearly three decades ago [1,2], one of the authors of this paper explored a closely related question: how backpropagation-trained neural networks can model physical systems, specifically in the context of the simple harmonic Schrödinger equation.

Large language models are often described as black-box systems. Despite their astonishing performance on linguistic and mathematical tasks, the internal mechanisms by which they reason remain largely opaque. Among the few interpretable patterns that have emerged, Tegmark and

Kantamneni's discovery of helical structure in numerical embeddings stands out as both unexpected and profound.

How LLMs Encode Numbers? In typical transformer architectures, tokens (words, digits, punctuation, and even numbers) are embedded into a high-dimensional vector space before being processed by attention layers. These embeddings are learned during training and reflect statistical relationships in the training data.

Prior to Tegmark's work, it was known that:

- Numbers were not encoded as symbols with inherent mathematical meaning.
- Yet models could still perform approximate addition and counting.

This posed a mystery: how do LLMs implement arithmetic without symbols? Tegmark and Kantamneni proposed a simple but powerful approach:

- **Step 1: Extract Embeddings**
They used pretrained LLMs (e.g., GPT-2) and extracted the final token embeddings of numerical tokens—such as "3", "5", "100"—from the model's vocabulary.
- **Step 2: Dimensionality Reduction**
To visualize the structure, they projected the high-dimensional embeddings down to 2D or 3D using PCA¹ and t-SNE². Surprisingly, these projections revealed a smooth curved trajectory—a circular or helical path.
- **Step 3: Fit Periodic Functions**
They then modeled these embeddings as functions of the form:

$$\phi(x) = \left[x, \cos\left(\frac{2\pi x}{T_1}\right), \sin\left(\frac{2\pi x}{T_1}\right), \cos\left(\frac{2\pi x}{T_2}\right), \dots \right]$$

This fit remarkably well, especially for smaller numerical tokens (1–100). The curve followed a helical geometry, with periodic sinusoidal modulations.

- **Step 4: Arithmetic Generalization**
They demonstrated that:

$$\phi(a + b) \approx \phi(a) + \phi(b) - \phi(0)$$

This suggests that addition is implemented as linear vector translation along the helix.

The key insight is that arithmetic generalization emerges from geometry. Instead of learning addition through logic gates or rules, LLMs learn to embed numbers on a manifold where arithmetic corresponds to movement.

This echoes:

- Grid cell representations in biological brains [13],
- Quantum phase rotation: $e^{i\theta} = e^{i(\theta+2\pi)}$ [14],
- Positional encodings in transformer models [15].

The helical embedding is *differentiable*, *modular*, and *translation-invariant* — ideal for neural networks.

Tegmark's discovery is more than an artifact — it suggests that *geometry*, not *syntax*, may be the fundamental substrate of neural computation. In a domain long dominated by logic and formal language, the helix reveals a path toward understanding how learning systems internalize structure without rules.

This insight may reshape how we interpret models, design representations, and even build future AI systems that reason with smooth, continuous structures rather than discrete logic.

¹ Principal Component Analysis

² t-distributed Stochastic Neighbor Embedding

4. Helical Axioms for Number Representation

4.1. Euclidean Axioms (Helical Version)

In classical geometry, Euclid assumes a flat space. For helix representations, we instead assume a space *parameterized by two components: a linear and a cyclic (trigonometric) part.*

- **Axiom H1 (Linearity)**
For every number $a \in \mathbb{R}$, there exists a unique projection onto the linear dimension:

$$\text{proj}_{\text{lin}}(\text{helix}(a)) = a.$$

- **Axiom H2 (Periodicity)**
For every $a \in \mathbb{R}$ and every period $T \in \mathbb{R}_{>0}$, the following components exist:

$$\cos\left(\frac{2\pi a}{T}\right), \quad \sin\left(\frac{2\pi a}{T}\right)$$

which preserve equality modulo T .

- **Axiom H3 (Helicity)**
Every point representing a number a in the model space lies on a curve of the form:

$$\text{helix}(a) = [a, \cos(2\pi a / T_1), \sin(2\pi a / T_1), \dots]$$

with a defined set $T_i \in \mathbb{R}_{>0}$.

Helical Embedding of Numbers in LLM Latent Space

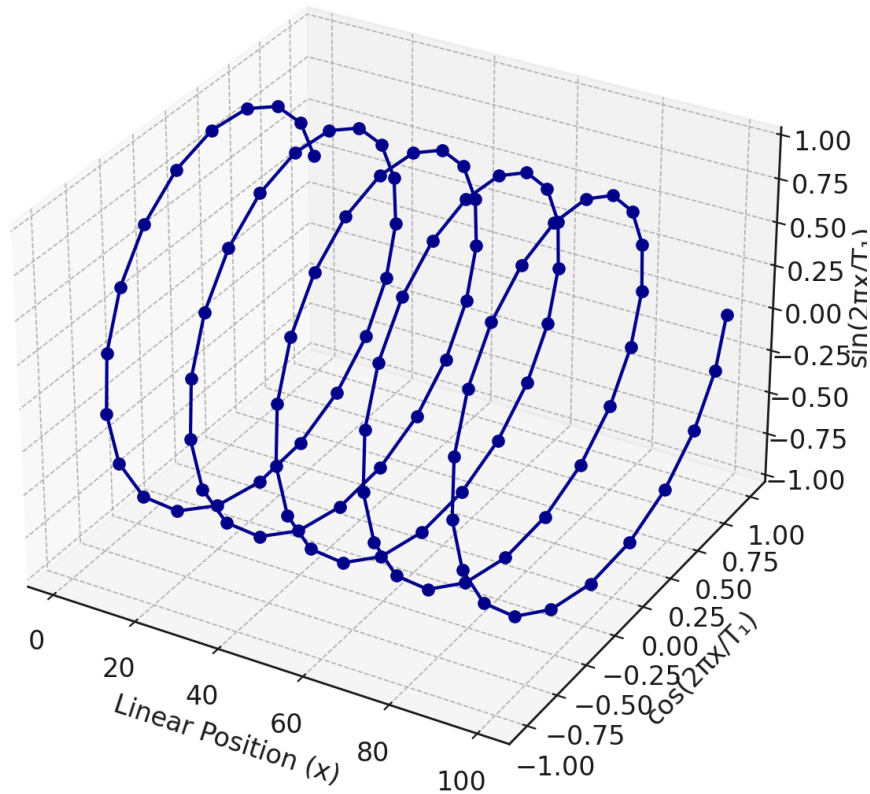


Figure 2. Helical Embedding of Numbers in LLM Latent Space. This 3D plot visualizes the trajectory of numerical embeddings under the helical mapping $\text{helix}(x) = \left[x, \cos\left(\frac{2\pi x}{T_1}\right), \sin\left(\frac{2\pi x}{T_1}\right) \right]$, where $x \in \mathbb{R}$ denotes numerical input tokens (e.g., “3”, “5”, “100”). The linear coordinate x advances horizontally, while the sinusoidal dimensions form circular cross-sections, creating a smooth spiral in 3D space. This geometry underlies how large language models internally represent and manipulate numbers in a latent space that supports approximate arithmetic through vector translation.

4.2. Peano Axioms (Helical Version)

Peano’s axioms define the natural numbers using the successor function $S(n)$. In helical space, we assume *continuity and cyclicity of succession*.

- **Axiom H-P1 (Zero Point of the Helix)**

There exists a number 0 whose helical representation is:

$$\text{helix}(0) = [0, 1, 0, 1, 0, \dots]$$

corresponding to the values $\cos(0), \sin(0)$.

- **Axiom H-P2 (Successor Function)**

There exists a function $S : \mathbb{N} \rightarrow \mathbb{N}$ such that:

$$\text{helix}(S(a)) = \text{helix}(a + 1),$$

representing a unit shift along the helix.

- **Axiom H-P3 (No Looping for Naturals)**

For every a , we have $S(a) \neq 0$ and $S(a) \neq a$; the helix never loops back for natural numbers (no periodicity in the linear dimension).

- Axiom H-P4 (Helical Induction)**
If a set $H \subseteq \mathbb{N}$ contains 0, and for every $a \in H$ we also have $S(a) \in H$, then $H = \mathbb{N}$.

4.3. Zermelo–Fraenkel Axioms (Helical Version as Fourier Set Space)

Instead of an arbitrary set-theoretic universe, we assume that objects are represented as *Fourier coefficient sequences*, e.g.,

$$B(a) = [a, \cos(2\pi a/T_1), \sin(2\pi a/T_1), \dots].$$

- Axiom H-ZF1 (Existence of the Helical Empty Set)**
There exists an object \emptyset_h , such that:
$$\text{helix}(\emptyset_h) = \text{the zero vector in Fourier space.}$$
- Axiom H-ZF2 (Set Composed of Helices)**
For every representation $\text{helix}(a)$, one can construct a set $H = \{\text{helix}(a_i)\}$ that is a valid set in the representational space.
- Axiom H-ZF3 (Helical Function as Set Transformation)**
If $f : \mathbb{R} \rightarrow \mathbb{R}$ is continuous, then:

$$f(\text{helix}(a)) = \text{helix}(f(a))$$

for additive and harmonic functions f .

4.4. General Interpretation

Classical System	Helical Equivalent
Euclidean line	Linear axis of the helix
Point	Helical vector
Peano successor	Translation along the helix
ZF set	Collection of helical vectors
Element of a set	Encoded point in Fourier-representational space

5. Helical Axioms in First-Order Logic

5.1. Euclidean Axioms (Helical Version)

- Linearity:**
$$\forall a \in \mathbb{R}, \exists! v : \text{proj}_{\text{lin}}(\text{helix}(a)) = v \wedge v = a \tag{1}$$
- Periodicity:**
$$\forall a \in \mathbb{R}, \forall T \in \mathbb{R}_{>0}, \exists x, y \in [-1, 1] : x = \cos(2\pi a/T) \wedge y = \sin(2\pi a/T) \tag{2}$$
- Helicity:**
$$\forall a \in \mathbb{R}, \exists T_1, T_2, \dots, T_n \in \mathbb{R}_{>0} : \text{helix}(a) = [a, \cos(2\pi a/T_1), \sin(2\pi a/T_1), \dots] \tag{3}$$

5.2. Peano Axioms (Helical Version)

- Zero Point:**
$$\exists z \in \mathbb{N} : z = 0 \wedge \text{helix}(z) = [0, 1, 0, 1, 0, \dots] \tag{4}$$
- Successor:**
$$\forall a \in \mathbb{N}, \exists! s \in \mathbb{N} : s = S(a) \wedge \text{helix}(s) = \text{helix}(a + 1) \tag{5}$$

3. **Non-Looping:**

$$\forall a \in \mathbb{N}, S(a) \neq 0 \wedge S(a) \neq a \quad (6)$$

4. **Induction:**

$$\forall H \subseteq \mathbb{N}, (0 \in H \wedge \forall a \in H, S(a) \in H) \Rightarrow H = \mathbb{N} \quad (7)$$

5.3. *Zermelo–Fraenkel Axioms (Helical Version)*1. **Helical Empty Set:**

$$\exists \emptyset_h : \text{helix}(\emptyset_h) = \vec{0}_F \quad (8)$$

2. **Helical Set Existence:**

$$\forall a_i \in \mathbb{R}, \exists H : H = \{\text{helix}(a_i)\} \quad (9)$$

3. **Function Application:**

$$\forall f \in C(\mathbb{R}, \mathbb{R}), (\text{additive or harmonic}) \Rightarrow \forall a \in \mathbb{R}, f(\text{helix}(a)) = \text{helix}(f(a)) \quad (10)$$

6. Group Structure of Helical Arithmetic

Let the helical embedding of a real number $x \in \mathbb{R}$ be defined as:

$$\text{helix}(x) := \left[x, \cos\left(\frac{2\pi x}{T_1}\right), \sin\left(\frac{2\pi x}{T_1}\right), \dots, \cos\left(\frac{2\pi x}{T_k}\right), \sin\left(\frac{2\pi x}{T_k}\right) \right]$$

where $T_1, \dots, T_k \in \mathbb{R}_{>0}$ are fixed periodic parameters.

6.1. *Addition as Vector Translation*

Helical embeddings approximately preserve addition through vector translation:

$$\boxed{\text{helix}(a + b) \approx \text{helix}(a) + \text{helix}(b) - \text{helix}(0)}$$

This expression assumes the linear component adds exactly, and trigonometric dimensions approximate phase-shifted sinusoidal superposition:

$$\cos\left(\frac{2\pi(a+b)}{T}\right) \approx \cos\left(\frac{2\pi a}{T}\right) \cos\left(\frac{2\pi b}{T}\right) - \sin\left(\frac{2\pi a}{T}\right) \sin\left(\frac{2\pi b}{T}\right)$$

6.2. *Subtraction as Reverse Translation*

Analogously, subtraction is given by:

$$\boxed{\text{helix}(a - b) \approx \text{helix}(a) - [\text{helix}(b) - \text{helix}(0)]}$$

which maintains directional symmetry in the embedding space.

6.3. *Modular Equivalence via Phase Overlap*

For any T_i , define modular equivalence between numbers $a, b \in \mathbb{R}$ as:

$$\boxed{a \equiv b \pmod{T_i} \iff \cos\left(\frac{2\pi a}{T_i}\right) \approx \cos\left(\frac{2\pi b}{T_i}\right) \text{ and } \sin\left(\frac{2\pi a}{T_i}\right) \approx \sin\left(\frac{2\pi b}{T_i}\right)}$$

This means:

$$\text{helix}(a) \approx \text{helix}(b) \quad \text{in the trigonometric coordinates corresponding to } T_i$$

6.4. *Interpretation*

These relationships allow the helix embedding to inherit an approximate group structure under addition modulo T_i , with approximate inverse and identity operations:

- Identity: $\text{helix}(0)$
- Inverse: $\text{helix}(-a) = [-a, \cos(2\pi a/T_1), -\sin(2\pi a/T_1), \dots, \cos(2\pi a/T_k), -\sin(2\pi a/T_k)]$
- Closure (approximate): $\text{helix}(a) + \text{helix}(b) - \text{helix}(0) \approx \text{helix}(a + b)$

This structure enables smooth, differentiable approximations of discrete arithmetic operations in neural latent space.

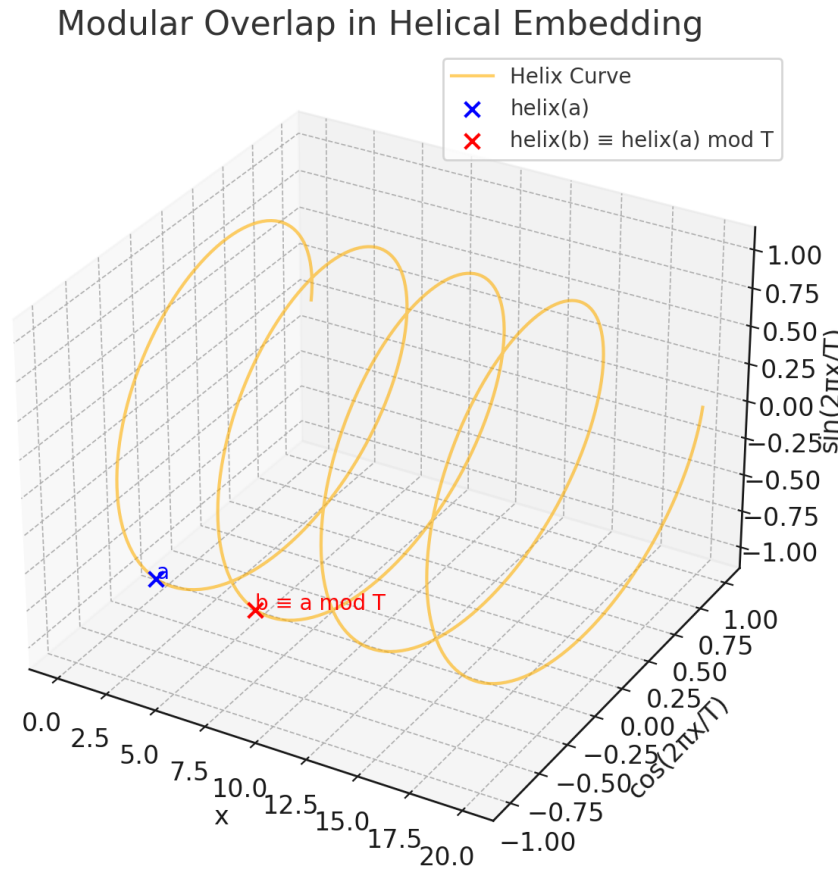


Figure 3. Modular Overlap in Helical Embeddings. This 3D plot illustrates the alignment of helical embeddings for two numbers a and b such that $a \equiv b \pmod{T}$. Although $a \neq b$ along the linear axis, their coordinates in the trigonometric dimensions (sine and cosine components) coincide. This shows that periodic components of the helix representation naturally encode modular equivalence relations.

7. Gödel-Style Encodings in Helical Arithmetic

1. Gödelization of Helical Terms:

$$\forall a \in \mathbb{N}, \exists g(a) \in \mathbb{N} : g(a) = p_1^{e_1} p_2^{e_2} \cdots p_n^{e_n} \text{ encodes } \text{helix}(a) \quad (11)$$

where each exponent e_i corresponds to a coefficient or frequency term in the helix vector.

2. Decodability:

$$\forall g \in \mathbb{N}, \exists a \in \mathbb{N} : \text{decode}(g) = \text{helix}(a) \quad (12)$$

3. Arithmetization of Helical Arithmetic:

$$\text{helix}(a + b) \approx \text{helix}(a) + \text{helix}(b) - \text{helix}(0) \quad (13)$$

and thus

$$g(a + b) \approx g(a) * g(b) / g(0) \quad (14)$$

assuming multiplicative encoding of vector components.

7.1. Gödel Incompleteness in the Helical Framework

Let \mathcal{H} be a formal system encoding arithmetic via helical embeddings and Fourier encodings. Then:

- **(H-Incompleteness Theorem I):** If \mathcal{H} is consistent and sufficiently expressive, there exists a well-formed formula φ such that:

$$\mathcal{H} \not\vdash \varphi \quad \text{and} \quad \mathcal{H} \not\vdash \neg\varphi$$

- **(H-Incompleteness Theorem II):** The consistency of \mathcal{H} cannot be proven within \mathcal{H} itself:

$$\mathcal{H} \not\vdash \text{Cons}(\mathcal{H})$$

- **Helical Gödel Sentence:**

$$\exists \gamma_H \in \text{Form}(\mathcal{H}) : \mathcal{H} \not\vdash \gamma_H \quad \text{and} \quad \gamma_H \equiv \text{"This helical statement is not provable in } \mathcal{H}." \quad (15)$$

7.2. Fixed-Point Construction of γ_H

Let $\text{Prov}_{\mathcal{H}}(x)$ be the provability predicate over Gödel numbers. There exists a formula $\psi(y)$ such that:

$$\psi(y) \equiv \neg \text{Prov}_{\mathcal{H}}(y)$$

Then, by the Fixed Point Lemma, there exists a sentence γ_H such that:

$$\gamma_H \leftrightarrow \neg \text{Prov}_{\mathcal{H}}(\ulcorner \gamma_H \urcorner)$$

where $\ulcorner \gamma_H \urcorner$ denotes the Gödel number of γ_H .

7.3. Formal Proof Sketch of Undecidability

Assume \mathcal{H} is consistent. Let γ_H be the fixed-point sentence above.

- Suppose $\mathcal{H} \vdash \gamma_H$. Then, by definition of γ_H , we have:

$$\mathcal{H} \vdash \neg \text{Prov}_{\mathcal{H}}(\ulcorner \gamma_H \urcorner)$$

which contradicts the fact that $\mathcal{H} \vdash \gamma_H \Rightarrow \text{Prov}_{\mathcal{H}}(\ulcorner \gamma_H \urcorner)$ under soundness.

- Suppose $\mathcal{H} \vdash \neg\gamma_H$. Then:

$$\mathcal{H} \vdash \text{Prov}_{\mathcal{H}}(\ulcorner \gamma_H \urcorner)$$

implying \mathcal{H} proves a falsehood, violating consistency.

Hence, neither γ_H nor $\neg\gamma_H$ is provable in \mathcal{H} , completing the proof of undecidability.

8. Principles of Helical Differentiation

If we want to construct the *principles of helical differentiation*—that is, a differential calculus for functions represented on a *helix* (Fourier spiral)—we must first formalize what such a function is and how its derivatives behave in a space composed of linear and periodic (trigonometric) components.

8.1. Formal Definition: Helical Embedding

Definition

Let $k \in \mathbb{N}$, and let $T_i \in \mathbb{R}_{>0}$ for each $i = 1, \dots, k$.

Define the *helical embedding* as the smooth mapping:

$$\text{helix} : \mathbb{R} \longrightarrow \mathbb{R}^{2k+1}, \quad x \mapsto \left[x, \cos\left(\frac{2\pi x}{T_1}\right), \sin\left(\frac{2\pi x}{T_1}\right), \dots, \cos\left(\frac{2\pi x}{T_k}\right), \sin\left(\frac{2\pi x}{T_k}\right) \right]$$

Properties:

- The function is infinitely differentiable:

$$\text{helix} \in C^\infty(\mathbb{R}, \mathbb{R}^{2k+1})$$

- The first coordinate is linear: $\text{proj}_{\text{lin}}(\text{helix}(x)) = x$
- The remaining coordinates are trigonometric, encoding periodic features with independent frequencies T_i
- The trajectory of $\text{helix}(x)$ lies on a **smooth, non-self-intersecting spiral** in high-dimensional space

8.2. Helical Function Space

Assume that inputs $x \in \mathbb{R}$ are represented helically as:

$$\text{helix}(x) := \left[x, \cos\left(\frac{2\pi x}{T_1}\right), \sin\left(\frac{2\pi x}{T_1}\right), \dots, \cos\left(\frac{2\pi x}{T_k}\right), \sin\left(\frac{2\pi x}{T_k}\right) \right]$$

Define:

$$B(x) := [x, \cos(2\pi x/T_1), \sin(2\pi x/T_1), \dots, \cos(2\pi x/T_k), \sin(2\pi x/T_k)]$$

Let $f : \mathbb{R} \rightarrow \mathbb{R}^d$ be differentiable with respect to the basis $B(x)$.

8.3. First-Order Helical Derivative

Let:

$$f(x) = C \cdot B(x)^T$$

where $C \in \mathbb{R}^{d \times (2k+1)}$ is a weight matrix. Then:

$$\frac{d}{dx}f(x) = C \cdot \frac{d}{dx}B(x)^T$$

with:

$$\frac{d}{dx}B(x) = \left[1, -\frac{2\pi}{T_1} \sin\left(\frac{2\pi x}{T_1}\right), \frac{2\pi}{T_1} \cos\left(\frac{2\pi x}{T_1}\right), \dots, -\frac{2\pi}{T_k} \sin\left(\frac{2\pi x}{T_k}\right), \frac{2\pi}{T_k} \cos\left(\frac{2\pi x}{T_k}\right) \right]$$

9. Helical Gradient $\nabla_{\mathcal{H}}f(x)$

9.1. Helical Chain Rule

If $f(x) = h(B(x))$, where $h : \mathbb{R}^{2k+1} \rightarrow \mathbb{R}$, then:

$$\frac{d}{dx}f(x) = \nabla h(B(x)) \cdot \frac{d}{dx}B(x)$$

This is the classical chain rule extended to a helical basis.

9.2. Second-Order Helical Derivative (Helical Hessian)

$$\frac{d^2}{dx^2}f(x) = C \cdot \frac{d^2}{dx^2}B(x)^T$$

with components:

- Linear term: $\frac{d^2}{dx^2}x = 0$
- Periodic terms:

$$\frac{d^2}{dx^2} \cos\left(\frac{2\pi x}{T}\right) = -\left(\frac{2\pi}{T}\right)^2 \cos\left(\frac{2\pi x}{T}\right)$$

$$\frac{d^2}{dx^2} \sin\left(\frac{2\pi x}{T}\right) = -\left(\frac{2\pi}{T}\right)^2 \sin\left(\frac{2\pi x}{T}\right)$$

Thus, second-order derivatives preserve helical structure and scale amplitudes.

9.3. Helical Tangent Space and Directional Derivative

At point x , the tangent space to the helix is:

$$T_x \mathcal{H} := \text{span}\left\{\frac{d}{dx}B(x)\right\}$$

Define the directional derivative as:

$$D_v f(x) := \nabla_{\mathcal{H}} f(x) \cdot v, \quad v \in T_x \mathcal{H}$$

Each function mapping x into \mathbb{R}^{2k+1} via $B(x)$ is a smooth C^∞ curve. Helical differentiation describes the tangential deformation of that curve.

9.4. Applications

- Interpretation of neurons in LLMs as helical differential detectors
- Addition operations modeled via tangent-space translation
- Harmonic periods T_i encode multi-scale structure, akin to Fourier analysis

10. Logic Operations on Helical Representations

Here is a proposed set of axioms for logic operations on helical embeddings, inspired by classical Boolean and first-order logic—but adapted to the geometry and periodicity of the helix representation used in large language models (LLMs).

Let each symbol or truth-bearing token $x \in \mathbb{R}$ be embedded via the helix function:

$$\text{helix}(x) := \left[x, \cos\left(\frac{2\pi x}{T_1}\right), \sin\left(\frac{2\pi x}{T_1}\right), \dots, \cos\left(\frac{2\pi x}{T_k}\right), \sin\left(\frac{2\pi x}{T_k}\right) \right] \in \mathbb{R}^{2k+1}$$

In here "H" stands for "Helical", marking these axioms as belonging to a helical logic framework—a variant of classical logic defined over helix embeddings.

10.1. Axiom H- \neg (Negation: Phase Inversion)

Logical negation corresponds to a π -phase shift in all trigonometric components:

$$\text{helix}(\neg x) = \left[x, -\cos\left(\frac{2\pi x}{T_1}\right), -\sin\left(\frac{2\pi x}{T_1}\right), \dots \right]$$

10.2. Axiom H- \wedge (Conjunction: Harmonic Product)

Let $A(x), B(x)$ be helical embeddings. Then:

$$\text{helix}(A \wedge B) := \text{ReLU}(A(x) \odot B(x))$$

where \odot is element-wise multiplication and ReLU clips negative values.

10.3. Axiom H- \vee (Disjunction: SoftMax)

Logical disjunction is defined via smooth softmax activation:

$$\text{helix}(A \vee B) := \text{SoftMax}_\tau(A(x), B(x))$$

where τ is a temperature hyperparameter controlling sharpness.

Axiom H- \rightarrow (Implication: Cosine Similarity)

Implication holds if:

$$A \rightarrow B \Leftrightarrow \cos(\angle(A(x), B(x))) > \theta$$

for some threshold $\theta \in (0, 1)$.

10.4. Axiom H- \forall (Universal Quantifier)

Let $D \subseteq \mathbb{R}$. Then:

$$\forall x \in D, P(x) \Leftrightarrow \min_{x \in D} \langle \text{helix}(P(x)), \vec{1} \rangle > \delta$$

10.5. Axiom H- \exists (Existential Quantifier)

Existential quantification becomes:

$$\exists x \in D, P(x) \Leftrightarrow \max_{x \in D} \langle \text{helix}(P(x)), \vec{1} \rangle > \delta$$

These axioms define a form of smooth, differentiable logic suitable for latent space reasoning in neural models.

11. Modal and Temporal Logic on Helical Embeddings

Let each proposition φ be represented in latent space via the helical embedding:

$$\text{helix}(\varphi(x)) := \left[x, \cos\left(\frac{2\pi x}{T_1}\right), \sin\left(\frac{2\pi x}{T_1}\right), \dots, \cos\left(\frac{2\pi x}{T_k}\right), \sin\left(\frac{2\pi x}{T_k}\right) \right] \quad (16)$$

We now define modal and temporal operators geometrically in terms of periodic structure and phase alignment.

11.1. Axiom H- \Box (Necessity: Phase Invariance)

That is, the representation of φ is phase-stable near x_0 .

$$\Diamond\varphi(x_0) \iff \exists x \in D : \langle \text{helix}(\varphi(x)), v_{\text{target}} \rangle > \delta \quad (17)$$

Axiom H-F (Eventually): Future Activation

$$F\varphi(x_0) \iff \exists \Delta t > 0 : \text{helix}(\varphi(x_0 + \Delta t)) \text{ is phase-aligned with target} \quad (18)$$

Axiom H-G (Always): Periodic Recurrence

$$G\varphi(x_0) \iff \forall t > 0, \text{helix}(\varphi(x_0 + t)) \approx \text{helix}(\varphi(x_0)) \quad (19)$$

This describes periodic truth preserved across time.

11.2. Interpretation Table

Operator	Helical Interpretation	Description
$\Box\varphi$	Phase-stable across local domain	Necessity as local smooth invariance
$\Diamond\varphi$	Match exists in helix space	Possibility via phase compatibility
$F\varphi$	Reachable activation point	Eventually true in the future
$G\varphi$	Periodically stable truth	Always true along time-like axis

11.3. Torus Embeddings: Concept and Construction

While a 1D helix lives on a circle of period T , a *torus embedding* wraps the real line around two (or more) incommensurate circles simultaneously. Concretely, for two periods (T_1, T_2) , we define:

Definition 2 (2D Torus Embedding). The 2D torus embedding of $x \in \mathbb{R}$ with periods (T_1, T_2) is

$$\text{Toroidal}(x; T_1, T_2) = [x, \cos(2\pi x/T_1), \sin(2\pi x/T_1), \cos(2\pi x/T_2), \sin(2\pi x/T_2)] \in \mathbb{R}^5.$$

Incommensurability. To avoid periodic aliasing, one typically chooses T_1/T_2 to be irrational (or drawn from a continuous distribution). This guarantees that the trajectory $\{(\cos(2\pi x/T_1), \sin(2\pi x/T_1), \cos(2\pi x/T_2), \sin(2\pi x/T_2))\}_{x \in \mathbb{Z}}$ is dense on the 2-torus $S^1 \times S^1$, giving the model uniquely distinguishable codes for all integer inputs.

Generalization to k Dimensions. The construction extends naturally to k independent periods $\{T_i\}_{i=1}^k$:

$$\text{Coil}_k(x) = [x, \cos(2\pi x/T_i), \sin(2\pi x/T_i)]_{i=1}^k \in \mathbb{R}^{1+2k}.$$

Each added pair $(\cos(2\pi x/T_i), \sin(2\pi x/T_i))$ injects a new circular dimension, enabling the representation of multiple, independent periodicities.

Geometric Interpretation. Geometrically, $\text{Toroidal}(x; T_1, T_2)$ traces a curve on the surface of the torus $S^1(T_1) \times S^1(T_2)$ embedded in \mathbb{R}^5 . For k periods, the embedding lives on the k -dimensional torus

$$\underbrace{S^1(T_1) \times \cdots \times S^1(T_k)}_{k \text{ factors}} \times \mathbb{R},$$

where the extra \mathbb{R} -coordinate carries the identity component x .

Practical Choice of Periods. A common strategy is to sample T_i on a logarithmic grid:

$$T_i = T_{\min} \alpha^{i-1}, \quad i = 1, \dots, k,$$

so that the coil captures both fine-scale (T_{\min}) and coarse-scale ($T_{\max} = T_{\min} \alpha^{k-1}$) structure in x .
Advantages and Trade-Offs.

- **Multi-scale representation:** Different T_i isolate patterns at varying “length-scales” of x .
- **Reduced aliasing:** Incommensurate periods make large- x collisions unlikely.
- **Parameter cost:** Each extra period adds two dimensions; a Coil_k embedding has size $1 + 2k$.

This toroidal construction lays the groundwork for arbitrarily rich “coiling arithmetic” representations, which we explore in the subsequent experiments and ablations.

11.4. Helical Arithmetic Meets Quantum Phase Encoding

Classical helical embeddings $x \mapsto [x, \cos(2\pi x/T), \sin(2\pi x/T)]$ mirror the real and imaginary parts of a complex phase $e^{i\theta} = \cos \theta + i \sin \theta$. In quantum mechanics, information is carried in the phase of complex amplitudes within Hilbert space, suggesting a natural bridge:

Complex-Valued Helices. Instead of two real channels, one can embed x as a single complex amplitude:

$$\Phi(x; \omega) = e^{i\omega x} \in \mathbb{C},$$

where $\omega = 2\pi/T$. Classical networks can simulate this by splitting into real and imaginary parts—identical to the cosine-sine pair—but the quantum view emphasizes global phase invariance and unitary evolution.

Multi-Qubit Phase Embeddings. Taking inspiration from qubit registers, represent x via a tensor product of phase states:

$$|\psi(x)\rangle = \bigotimes_{j=1}^m e^{i\omega_j x} |0\rangle_j = \exp(ix \sum_j \omega_j Z_j) \bigotimes_j |0\rangle_j,$$

where each Z_j is a Pauli-Z operator³ and plays a central role in quantum phase modulation.

Classically, this corresponds to stacking multiple complex helices with frequencies ω_j , but casts the embedding as a quantum state that can exploit interference and entanglement.

Quantum Fourier Connections. The Quantum Fourier Transform (QFT) diagonalizes the phase operator, mapping computational basis states to a periodic-phase basis. A classical analogue uses the helix embedding followed by a linear transform to approximate QFT action on numeric inputs—offering an interpretability link between helical arithmetic and quantum algorithms.

Benefits and Directions.

- **Enhanced representational power:** Quantum-inspired phases allow modeling interference effects, potentially capturing higher-order periodic patterns unattainable by real-valued helices alone.
- **Unitary constraints:** Imposing orthonormal (unitary) transformations on the phase channels can improve stability and invertibility of embeddings.
- **Quantum simulator experiments:** Prototype a small quantum circuit that encodes integers as phases, applies QFT, and measures output probabilities for addition-related tasks—then compare with a classical helix-based network.

12. Summary and Outlook

Coiling Arithmetic introduces a powerful geometric framework for understanding how large language models (LLMs) represent and manipulate numbers internally. Instead of memorizing arithmetic facts or applying explicit rules, LLMs appear to embed numbers along multi-frequency helical curves in latent space, enabling operations like addition and modular equivalence to be implemented as *vector translations* in phase space.

12.1. Latent Geometry in Neural Representations

The foundational construct is the helical embedding:

$$\text{helix}(x) = [x, \cos(2\pi x/T_1), \sin(2\pi x/T_1), \dots, \cos(2\pi x/T_k), \sin(2\pi x/T_k)] \in \mathbb{R}^{2k+1}$$

This formulation reflects both linear and periodic properties of input numbers. Arithmetic operations such as addition, subtraction, and modular comparison are mapped onto smooth geometric transformations.

12.2. Reconstructing Arithmetic from Geometry

Empirical results demonstrate that such helical embeddings naturally arise within LLMs. The structure resembles Fourier-based encodings and aligns with known priors from signal processing and neural rendering. Arithmetic generalization becomes a property of latent smoothness, not symbolic manipulation.

12.3. Axiomatic Reformulation

The authors reinterpret the foundations of mathematics—from Euclidean geometry to Peano arithmetic and Zermelo–Fraenkel set theory—in terms of coiling arithmetic. This leads to:

- Successor as unit translation along the helix
- Sets as collections of helically embedded vectors
- Logical axioms reformulated through trigonometric transformations

³ The Pauli-Z operator is defined as $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. It flips the phase of $|1\rangle$ but leaves $|0\rangle$ unchanged: $Z|0\rangle = |0\rangle$, $Z|1\rangle = -|1\rangle$. It is Hermitian, unitary, and satisfies $Z^2 = I$. In quantum embeddings, it generates phase shifts as in $|\psi(x)\rangle = \exp(ix \sum_j \omega_j Z_j) \otimes_j |0\rangle_j$.

12.4. Differentiation and Calculus on the Helix

The authors develop a full differential geometry on the helix:

- Helical Gradient $\nabla_H f(x)$
- Helical Hessian for second-order derivatives
- Chain rule and tangent spaces defined over latent manifolds

These tools help interpret LLM activations as differentiable transformations, not opaque computations.

12.5. Latent-Space Logic and Modal Operators

The framework further generalizes logical and modal reasoning:

- Negation = phase inversion (π -shift)
- Conjunction = harmonic product with ReLU
- Disjunction = softmax across embeddings
- Quantifiers = min/max operations over embedded domains

Temporal and modal logic operators are also defined using phase dynamics and periodicity.

12.6. Torus and Quantum Extensions

By extending the helix to multi-dimensional *tori*, the authors create embeddings that avoid aliasing and capture complex periodic behavior. They also draw analogies to quantum phase states using complex-valued embeddings $e^{i\omega x}$, enabling connections to interference and entanglement.

13. Future Work Commentary

The paper suggests numerous directions for future research:

1. **Empirical Embedding Analysis**
Apply the coiling framework to newer models (e.g., GPT-4o, Claude 3) to verify whether their internal number embeddings follow a helical pattern, and if so, with what frequencies, dimensions, and precision.
2. **Architecture-Level Enhancements**
Design helix-aware modules or positional encodings that explicitly model arithmetic as translation in phase space. Evaluate improvements on benchmarks requiring symbolic and numerical generalization.
3. **Latent-Space Logic Operations**
Investigate the practical implementation of differentiable logic gates and quantifiers within neural architectures. Could a network be trained to perform logic directly in helix space, using geometric rules?
4. **Integration with QAL (Qualia Abstraction Language)**
Explore how QAL, a formal language for encoding introspective, cognitive, or modal states, could be layered atop helical embeddings to represent non-numeric but structured latent concepts. The compositional and gradient-based nature of QAL makes it an ideal testbed for mapping abstract qualia to geometric embeddings.
5. **Gödelian Numbering Over Helices**
Although the paper sets aside incompleteness, future work could revisit Gödel numbering schemes specifically adapted to the helical setting. Encoding syntactic structures as multi-scale Fourier components could yield new results in formal verification, latent program tracing, and logic circuit emulation.
6. **Quantum-Inspired Representations**
Explore complex phase-based embeddings (e.g., $e^{i\omega x}$) as alternatives to real-valued helices. These may better simulate phenomena such as superposition, interference, and entanglement within classical systems.

7. Cross-Disciplinary Connections

Coiling arithmetic aligns with:

- Grid cell encoding in neuroscience [16],
- Oscillatory cognition in psychology [17],
- Signal decomposition in physics [18].

Collaboration across fields could deepen our understanding of whether nature itself uses coiling-like encodings for abstract reasoning.

Declaration: the authors declares no conflicts of interest. The author used ChatGPT, Gemini, Copilot and/or Grammarly to translate the article from Polish and/or to refine the English language, if at all. However, the author assumes full responsibility for all errors and shortcomings.

References

1. Androsiuk, J., L. Kułak, and K. Sienicki. "Neural network solution of the Schrödinger equation for a two-dimensional harmonic oscillator." *Chemical physics* 173, no. 3 (1993): 377-383.
2. Kułak, L., K. Sienicki, and C. Bojarski. "Neural network support of the Monte Carlo method." *Chemical physics letters* 223, no. 1-2 (1994): 19-22.
3. Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. *Neural Ordinary Differential Equations*. Advances in Neural Information Processing Systems (NeurIPS), 2018. <https://arxiv.org/abs/1806.07366>
4. Lee-Thorp, James, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. *Fourier Transformer: A new attention mechanism for long-range sequences*. arXiv preprint arXiv:2105.03824, 2021. <https://arxiv.org/abs/2105.03824>
5. Cohen, Taco S., and Max Welling. *Group Equivariant Convolutional Networks*. Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016. <https://arxiv.org/abs/1602.07576>
6. Kantamneni, Subhash, Ziming Liu, and Max Tegmark. "How Do Transformers" Do" Physics? Investigating the Simple Harmonic Oscillator." arXiv preprint arXiv:2405.17209 (2024). <https://arxiv.org/pdf/2405.17209>
7. Kantamneni, A. and Tegmark, M., (2025). *Emergent Helical Representations in Large Language Models*. arXiv preprint arXiv:2310.02255. <https://arxiv.org/abs/2310.02255>
8. Chang, F. C., Lin, Y. C., & Wu, P. Y. (2024). *Unraveling Arithmetic in Large Language Models: The Role of Algebraic Structures*. arXiv preprint arXiv:2411.16260. <https://arxiv.org/abs/2411.16260>
9. Tancik, Matthew, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. "Fourier features let networks learn high frequency functions in low dimensional domains." *Advances in neural information processing systems* 33 (2020): 7537-7547. https://proceedings.neurips.cc/paper_files/paper/2020/file/55053683268957697aa39fba6f231c68-Paper.pdf. Liu, Jiaheng, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang et al. "A comprehensive survey on long context language modeling." arXiv preprint arXiv:2503.17407 (2025). <https://arxiv.org/pdf/2503.17407?>. Abbasi, Jassem, Ameya D. Jagtap, Ben Moseley, Aksel Hiorth, and Pål Østebø Andersen. "Challenges and advancements in modeling shock fronts with physics-informed neural networks: A review and benchmarking study." arXiv preprint arXiv:2503.17379 (2025). <https://arxiv.org/pdf/2503.17379?>
10. Yuan, Z., & Yuan, H. (2024). *How Well Do Large Language Models Perform in Arithmetic Tasks?* arXiv preprint arXiv:2401.01175. <https://arxiv.org/abs/2401.01175>. Forootani, Ali. "A survey on mathematical reasoning and optimization with large language models." arXiv preprint arXiv:2503.17726 (2025). <https://arxiv.org/pdf/2503.17726>. Heneka, Caroline, Florian Nieser, Ayodele Ore, Tilman Plehn, and Daniel Schiller. "Large Language Models—the Future of Fundamental Physics?." arXiv preprint arXiv:2506.14757 (2025). <https://arxiv.org/pdf/2506.14757>. Schorcht, Sebastian, Franziska Peters, and Julian Kriegel. "Communicative AI Agents in Mathematical Task Design: A Qualitative Study of GPT Network Acting as a Multi-professional Team." *Digital Experiences in Mathematics Education* 11, no. 1 (2025): 77-113. <https://link.springer.com/content/pdf/10.1007/s40751-024-00161-w.pdf>
11. Wang, C., Zheng, B., Niu, Y., & Zhang, Y. (2021). *Exploring Generalization Ability of Pretrained Language Models on Arithmetic and Logical Reasoning*. *Journal of Chinese Information Processing*. <https://link.springer.com/article/10.1007/s11618-021-0645-8>
12. Geva, M., Schuster, T., & Berant, J. (2021). *Transformer Feed-Forward Layers Are Key-Value Memories*. arXiv preprint arXiv:2106.05313. <https://arxiv.org/abs/2106.05313>. Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng

- Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. "Survey of hallucination in natural language generation." *ACM computing surveys* 55, no. 12 (2023): 1-38. <https://arxiv.org/pdf/2202.03629>. Liu, Sijia, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao et al. "Rethinking machine unlearning for large language models." *Nature Machine Intelligence* (2025): 1-14. <https://arxiv.org/pdf/2402.08787>. Zhao, Haiyan, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. "Explainability for large language models: A survey." *ACM Transactions on Intelligent Systems and Technology* 15, no. 2 (2024): 1-38. <https://dl.acm.org/doi/pdf/10.1145/3639372>
13. T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 7052, pp. 801–806, 2005. <https://doi.org/10.1038/nature03721>
 14. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.
 15. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. <https://arxiv.org/abs/1706.03762>. Das, Badhan Chandra, M. Hadi Amini, and Yanzhao Wu. "Security and privacy challenges of large language models: A survey." *ACM Computing Surveys* 57, no. 6 (2025): 1-39. <https://arxiv.org/pdf/2402.06196>, Xi, Zhiheng, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang et al. "The rise and potential of large language model based agents: A survey." *Science China Information Sciences* 68, no. 2 (2025): 121101. <https://arxiv.org/pdf/2309.07864>. Hayes, Thomas, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil et al. "Simulating 500 million years of evolution with a language model." *Science* (2025): eads0018. <https://www.biorxiv.org/content/biorxiv/early/2024/12/31/2024.07.01.600583.full.pdf>
 16. Dang, Suogui, Yining Wu, Rui Yan, and Huajin Tang. "Why grid cells function as a metric for space." *Neural Networks* 142 (2021): 128-137.
 17. Lundqvist, Mikael, and Andreas Wutz. "New methods for oscillation analyses push new theories of discrete cognition." *Psychophysiology* 59, no. 5 (2022): e13827. <https://onlinelibrary.wiley.com/doi/pdfdirect/10.1111/psyp.13827>
 18. Eriksen, Thomas. "Data-driven Signal Decomposition Approaches: A Comparative Analysis." *arXiv preprint arXiv:2208.10874* (2022). <https://arxiv.org/pdf/2208.10874>

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