

Module G: Infinite RoPE Context Scaling and Dynamic Vocabulary Updating

Part of the Eidos Unified Framework for Persistent, Dynamic, and Adaptive Multimodal
Intelligence

Contents

1 Abstract	2
2 Introduction and Motivation	2
3 Preliminaries and Notation	2
4 Formal Definitions and Mathematical Formulation	4
5 Algorithmic Description	5
6 Theoretical Analysis and Guarantees	6
7 Integration with the Overall Eidos Framework	7
8 Implementation Considerations	7
9 Conclusion	8
10 Module Summary	8

1 Abstract

This module rigorously defines the *Infinite RoPE Context Scaling and Dynamic Vocabulary Updating* component of the Eidos framework. It introduces Rotary Positional Embeddings (RoPE) as a mechanism to encode relative positional information in a manner that naturally extends to arbitrarily long (or even infinite) contexts. In addition, this module formalizes a dynamic vocabulary updating mechanism, which continuously integrates new multi-token sequences into the existing vocabulary. By merging infinite-context scaling with adaptive vocabulary expansion, the module provides the linguistic and structural backbone for long-range dependencies and continual learning. The framework is presented with full mathematical rigor, including formal definitions, algorithmic descriptions, and theoretical guarantees.

2 Introduction and Motivation

Modern language and multimodal models must handle sequences whose lengths are not fixed a priori. Traditional absolute positional encodings impose a hard upper bound on context length, limiting the model’s ability to process long sequences. Rotary Positional Embeddings (RoPE) overcome this limitation by encoding relative positional information through continuous, rotational transformations applied to query and key vectors in attention mechanisms. Simultaneously, as models process increasing amounts of data, the vocabulary must evolve to incorporate new, frequently occurring multi-token sequences. The dual objectives of infinite context scaling and dynamic vocabulary updating are critical for a truly adaptive and extensible system.

The goals of this module are to:

- (a) Define a mathematically rigorous formulation of RoPE that supports infinite context scaling.
- (b) Establish theoretical guarantees (e.g., relative positional invariance) for the RoPE mechanism.
- (c) Formally define a dynamic vocabulary updating function that integrates new multi-token sequences into the existing vocabulary without loss of consistency.
- (d) Detail the algorithmic procedures for applying RoPE and updating the vocabulary in real time.

3 Preliminaries and Notation

We introduce the following notation to ensure clarity:

- Let Σ denote the base alphabet (e.g., Unicode).
- $X_{\text{proc}} \in \mathcal{X}_{\text{proc}}$ is preprocessed input (from Module A).
- \mathcal{V} denotes the complete vocabulary, defined as:

$$\mathcal{V} = \mathcal{V}^{(0)} \cup \mathcal{V}^{(1)},$$

where $\mathcal{V}^{(0)}$ is the base vocabulary (natural language tokens, Unicode characters, programming symbols) and $\mathcal{V}^{(1)}$ is the set of learned multi-token sequences.

- Each token $t \in \mathcal{V}$ is represented as:

$$t = (u, \pi, \chi),$$

with:

- u as the underlying unit (string/symbol),
- $\pi \in \Pi \subseteq \mathbb{R}^{d_\pi}$ representing intrinsic properties,
- $\chi \in \mathbb{R}^{d_\chi}$ representing contextual statistics.
- A unique identifier mapping is defined as:

$$\eta : \mathcal{V} \rightarrow \mathbb{N},$$

so that $\text{ID}(t) = \eta(t)$.

- **RoPE Notation:**

- Let d_{att} be an even integer representing the dimensionality of the attention subspace.
- For a vector $v \in \mathbb{R}^{d_{\text{att}}}$, partition it into $\frac{d_{\text{att}}}{2}$ sub-vectors $v^{(j)} \in \mathbb{R}^2$ for $j = 1, \dots, \frac{d_{\text{att}}}{2}$.
- For each subspace j , let $\theta_j \in \mathbb{R}$ be the frequency parameter.
- Define the rotation angle at position i as:

$$\varphi_j(i) = i \theta_j.$$

- Define the 2D rotation matrix for subspace j at position i as:

$$R^{(j)}(i) = \begin{pmatrix} \cos(\varphi_j(i)) & -\sin(\varphi_j(i)) \\ \sin(\varphi_j(i)) & \cos(\varphi_j(i)) \end{pmatrix}.$$

- The block-diagonal rotation operator is then defined as:

$$R(i) = \text{diag}\left(R^{(1)}(i), R^{(2)}(i), \dots, R^{(d_{\text{att}}/2)}(i)\right).$$

- The RoPE transformation is given by:

$$\psi(i, v) = R(i) v.$$

- **Dynamic Vocabulary Updating:**

- Let $\mathcal{D}_{\text{learn}}$ denote the set of learning signals or new data from which additional multi-token sequences are extracted.
- Define the dynamic vocabulary update function:

$$\Delta_{\mathcal{V}} : \mathcal{V} \times \mathcal{D}_{\text{learn}} \rightarrow \mathcal{V}',$$

which expands the vocabulary to \mathcal{V}' such that $\mathcal{V} \subset \mathcal{V}'$.

4 Formal Definitions and Mathematical Formulation

Definition G.1 (RoPE Transformation)

Given a vector $v \in \mathbb{R}^{d_{\text{att}}}$ and a position $i \in \mathbb{N}$, the RoPE transformation is defined as:

$$\psi(i, v) = R(i) v,$$

where the block-diagonal rotation operator $R(i)$ is:

$$R(i) = \text{diag}\left(R^{(1)}(i), R^{(2)}(i), \dots, R^{(d_{\text{att}}/2)}(i)\right),$$

with each 2D rotation matrix

$$R^{(j)}(i) = \begin{pmatrix} \cos(i \theta_j) & -\sin(i \theta_j) \\ \sin(i \theta_j) & \cos(i \theta_j) \end{pmatrix}.$$

This transformation is applied elementwise to the query and key vectors in the attention mechanism, ensuring that the inner product between rotated queries and keys depends solely on their relative positions.

Theorem G.1 (Relative Invariance of the RoPE Dot Product)

Statement: For any two positions $i, j \in \mathbb{N}$ and any vectors $q, k \in \mathbb{R}^{d_{\text{att}}}$, let

$$q'_i = \psi(i, q), \quad k'_j = \psi(j, k).$$

Then, the dot product

$$q'_i{}^\top k'_j = q^\top R(i)^\top R(j) k$$

depends only on the relative position $j - i$; that is, there exists a function Φ such that:

$$q'_i{}^\top k'_j = \Phi(q, k; j - i).$$

Proof Sketch: Since each $R^{(j)}(i)$ is an orthogonal matrix, $R(i)^\top R(j)$ equals a block-diagonal matrix with blocks $R^{(j)}(j - i)$. Therefore, the dot product is computed as a sum of terms of the form:

$$q^{(j)\top} R^{(j)}(j - i) k^{(j)},$$

which depends solely on the difference $j - i$. \square

Definition G.2 (Infinite Context Scaling)

The RoPE transformation $\psi(i, v)$ is defined for all $i \in \mathbb{N}$. Thus, there is no fixed maximum context length:

$$\forall i \in \mathbb{N}, \quad \psi(i, v) \text{ is well-defined.}$$

This property enables the model to process sequences of arbitrarily long (or even infinite) length without modification to the positional encoding mechanism.

Definition G.3 (Dynamic Vocabulary Update)

Let \mathcal{V} be the current vocabulary and $\mathcal{D}_{\text{learn}}$ be a set of learning signals derived from new input data. The dynamic vocabulary update function is:

$$\Delta_{\mathcal{V}} : \mathcal{V} \times \mathcal{D}_{\text{learn}} \rightarrow \mathcal{V}',$$

such that:

- (i) For any $t \in \mathcal{V}$, $t \in \mathcal{V}'$ (preservation of the base vocabulary).
- (ii) For any novel multi-token sequence $\delta \in \mathcal{D}_{\text{learn}}$ that meets a predetermined frequency or confidence threshold, $\Delta_{\mathcal{V}}$ incorporates δ as a new token t_{δ} into \mathcal{V}' .
- (iii) There exists an extension of the unique identifier mapping:

$$\eta' : \mathcal{V}' \rightarrow \mathbb{N},$$

such that for all $t \in \mathcal{V}$, $\eta'(t) = \eta(t)$.

This mechanism allows the vocabulary to grow dynamically, incorporating additional tokens that capture frequently occurring multi-token patterns.

5 Algorithmic Description

The following pseudocode describes the application of the RoPE transformation to token representations and the dynamic vocabulary update process.

Algorithm 1 Infinite Context Scaling with RoPE and Dynamic Vocabulary Updating

- 1: **Input:** Sequence of tokens (t_1, \dots, t_n) with base embeddings $\{E_B(t_i)\}$; frequency parameters $\{\theta_j\}_{j=1}^{d_{\text{att}}/2}$; new learning signals $\mathcal{D}_{\text{learn}}$; current vocabulary \mathcal{V}
- 2: **for** each position $i = 1, \dots, n$ **do**
- 3: **for** each subspace $j = 1, \dots, d_{\text{att}}/2$ **do**
- 4: Compute rotation angle: $\varphi_j(i) \leftarrow i \cdot \theta_j$
- 5: Form rotation matrix:

$$R^{(j)}(i) \leftarrow \begin{pmatrix} \cos(\varphi_j(i)) & -\sin(\varphi_j(i)) \\ \sin(\varphi_j(i)) & \cos(\varphi_j(i)) \end{pmatrix}$$

- 6: **end for**
- 7: Construct block-diagonal rotation:

$$R(i) \leftarrow \text{diag}\left(R^{(1)}(i), \dots, R^{(d_{\text{att}}/2)}(i)\right)$$

- 8: Apply to token's query/key (if applicable): $q'_i \leftarrow \psi(i, q_i)$, $k'_i \leftarrow \psi(i, k_i)$
 - 9: **end for**
 - 10: **Dynamic Vocabulary Update:**
 - 11: **for** each candidate multi-token sequence $\delta \in \mathcal{D}_{\text{learn}}$ **do**
 - 12: **if** δ satisfies the frequency/confidence threshold **then**
 - 13: Add δ as a new token t_δ to \mathcal{V}'
 - 14: Extend identifier mapping: set $\eta'(t_\delta) \leftarrow$ new unique ID
 - 15: **end if**
 - 16: **end for**
 - 17: **Return:** Updated vocabulary \mathcal{V}' and rotated vectors $\{q'_i, k'_i\}$
-

6 Theoretical Analysis and Guarantees

Theorem G.1 (Relative Invariance of RoPE)

Statement: For any two positions $i, j \in \mathbb{N}$ and any vectors $q, k \in \mathbb{R}^{d_{\text{att}}}$, the dot product of the RoPE-transformed vectors satisfies:

$$q'_i{}^\top k'_j = \Phi(q, k; j - i),$$

i.e., it depends solely on the relative position $j - i$.

Proof Sketch: Since each subspace rotation $R^{(j)}(i)$ is orthogonal, we have

$$R(i)^\top R(j) = \text{diag}\left(R^{(1)}(j - i), \dots, R^{(d_{\text{att}}/2)}(j - i)\right).$$

Thus, the dot product becomes a function of the differences $j - i$ only. \square

Proposition G.2 (Infinite Context Capability)

Because the RoPE transformation $\psi(i, v)$ is defined for all $i \in \mathbb{N}$, the mechanism imposes no fixed limit on the sequence length. Therefore, the model can handle arbitrarily long contexts, limited only by computational resources.

Proposition G.3 (Extensibility of Dynamic Vocabulary)

The dynamic vocabulary update function $\Delta_{\mathcal{V}}$ guarantees that:

- (i) The base vocabulary is preserved: $\forall t \in \mathcal{V}, t \in \mathcal{V}'$.
- (ii) New tokens are integrated seamlessly with unique identifiers maintained via an extension η' .

Thus, the vocabulary can be extended without disrupting existing mappings, ensuring modularity and consistency.

7 Integration with the Overall Eidos Framework

Module G, which provides infinite context scaling via RoPE and dynamic vocabulary updating, is fundamental to ensuring that the Eidos framework can process long-range dependencies and continually adapt its linguistic representation. Its integration points include:

- **Contextual Embedding (Module E):** The RoPE-transformed vectors are used as inputs to the deep contextual encoder, enhancing the attention mechanism with relative positional information.
- **Dynamic Vocabulary (Module D):** The dynamic update function $\Delta_{\mathcal{V}}$ augments the static vocabulary with new multi-token sequences, which are then used in tokenization and embedding.
- **Downstream Processing:** The infinite-context capability ensures that knowledge graphs (Module F) and deep model architectures (Module H) receive enriched, context-aware representations.

8 Implementation Considerations

- **Computational Efficiency:** The block-diagonal structure of $R(i)$ allows the RoPE transformation to be computed efficiently even for large i . Vectorized implementations in modern deep learning frameworks (e.g., PyTorch or TensorFlow) can further accelerate this process.
- **Frequency Parameter Selection:** The choice of θ_j is critical for ensuring that the rotations do not degenerate over long sequences. Typically, a decaying schedule (e.g., $\theta_j = 1/10000^{\frac{2(j-1)}{d_{\text{att}}}}$) is used.
- **Dynamic Vocabulary Update Strategy:** Implementing $\Delta_{\mathcal{V}}$ may involve unsupervised segmentation techniques (e.g., Byte-Pair Encoding or SentencePiece) and thresholding based on token frequency and contextual significance.
- **Storage and Retrieval:** New tokens must be indexed and stored efficiently. A database or hash table structure that maps token strings to unique identifiers is recommended.
- **Training and Fine-Tuning:** Both the RoPE mechanism and the dynamic vocabulary update may be trained jointly with the rest of the model, requiring careful balancing of learning rates and update frequencies.

9 Conclusion

In this module, we have rigorously defined the mechanisms for infinite context scaling using Rotary Positional Embeddings (RoPE) and the dynamic updating of the vocabulary. Key contributions include:

- A formal definition of the RoPE transformation $\psi(i, v) = R(i)v$ that guarantees relative positional invariance and enables processing of arbitrarily long sequences.
- Theoretical guarantees that the RoPE-transformed dot product depends solely on the relative positions of tokens.
- A dynamic vocabulary update function Δ_V that extends the vocabulary with newly learned multi-token sequences while preserving the unique identifiers of existing tokens.
- An algorithmic framework that integrates these components efficiently and robustly, ensuring seamless operation within the overall Eidos framework.

This module provides the critical ability for the system to handle long-range dependencies and to adapt its linguistic representation in response to continuous data, thereby forming a vital component of the Eidos architecture.

10 Module Summary

Completed:

- Module A: Input Processing.
- Module B: Universal Communication & Data Handling Interface and Coordination.
- Module C: Universal Streaming/Handling/Loading/Indexing Module.
- Module D: Multidimensional Vocabulary and Tokenization System.
- Module E: Contextual NLU/NLP Embedding and Multidimensional Tokenization.
- Module F: Deep Knowledge Graphs System (Base and Personal).
- Module G: Infinite RoPE Context Scaling and Dynamic Vocabulary Updating.

Remaining Modules:

- Module H: Core Model Architectures (RWKV and Transformer Modules, Mixture-of-Experts Style).
- Module I: Titans Memory Architecture (Multi-Layer Memory Module).
- Module J: Recursive Adaptive Dynamic Idempotent Feedback and State-Based Runtime Learning and Inference.
- Module K: Universal Training System.
- Module L: Final Decoding and Multimodal Output.