Eidos: A Unified Framework for Persistent, Dynamic, and Adaptive Multimodal Intelligence

—

Abstract

We introduce *Eidos*—an avant-garde, unified framework that seamlessly integrates raw input processing; universal communication, data handling, and streaming infrastructures; a multidimensional vocabulary and tokenization schema; dual-layer (base and adaptive) embeddings supporting both natural language understanding (NLU) and natural language processing (NLP); hierarchically organized knowledge graphs (base, personal, and unified); infinite context scaling via Rotary Positional Embeddings (RoPE) coupled with dynamic vocabulary refinement; a hybrid mixture-of-experts core model uniting Transformer, RWKV, and related modules; a multilayer "Titans" memory architecture (encompassing short-term, working, long-term, and personal memory); recursive adaptive idempotent feedback for continuous runtime learning; and an all-encompassing universal training system. The framework decodes outputs (text by default, with scalable pathways for additional modalities) in a fully modular, scalable, and extensible manner. Every symbol, process, function, and interface is meticulously defined using distinct notation and precise algorithmic pseudocode, constituting a robust blueprint for empirical implementation, testing, and iterative refinement.

Contents

1 Notational Conventions and Distinct Symbol Sets

Fundamental Axioms

The Eidos framework operates under three core axioms:

(**A1**) **Persistent Adaptivity**: $\forall t \, \exists \, \Delta\theta_t \in \Theta_{\text{adapt}}$ such that

$$f_{\theta_{t+1}} = f_{\theta_t} \oplus \Delta\theta_t.$$

(**A2**) **Idempotent Recursion**:

$$\mu^{\mathrm{R}} \circ \mu^{\mathrm{R}} \equiv \mu^{\mathrm{R}}.$$

(**A3**) **Unified Multimodality**:

$$\mathcal{O}_{\text{mod}} = \bigoplus_{m \in \mathcal{M}} \delta_m(\mathcal{Y}).$$

1.1 Raw Input and Preprocessing

- **Raw Input:** $X_{\text{raw}} \in \Sigma^*$, where Σ denotes the base alphabet (e.g., Unicode).
- **Preprocessed Input:** $X_{\text{proc}} \in \mathcal{X}_{\text{proc}}$.
- **Preprocessing Operator:** $\mathcal{P}_{\text{in}} : \Sigma^* \to \mathcal{X}_{\text{proc}}$.

1.2 Vocabulary and Tokens

- **Base Vocabulary:** $\mathcal{V}^{(0)}$ (e.g., English words, Unicode characters, programming symbols).
- **Learned Tokens:** $\mathcal{V}^{(1)}$ (multi-token sequences).
- **Complete Vocabulary:** $\mathcal{V} = \mathcal{V}^{(0)} \cup \mathcal{V}^{(1)}$.
- **Token:** $t \in \mathcal{V}$.
- **Unique Identifier Mapping:** $\eta : \mathcal{V} \to \mathbb{N}$, with $\mathrm{ID}(t) = \eta(t)$.
- **Token Structure:** Each token is represented as

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

  where

  - u: the underlying unit (string),
  - $\pi \in \Pi \subseteq \mathbb{R}^{d_\pi}$: intrinsic properties,
  - $\chi \in \mathbb{R}^{d_\chi}$: contextual statistics.

1.3 Enhanced Core Definitions

In addition to the standard representations, Eidos incorporates advanced mathematical constructs to capture quantum-like behavior and complex dynamics:

Quantum Token Representation

Each token exists in a superposition state:

$$|t\rangle = \alpha\,|u\rangle \otimes \beta\,|\pi\rangle \otimes \gamma\,|\chi\rangle \quad \text{where } |\alpha|^2 + |\beta|^2 + |\gamma|^2 = 1,$$

with measurement operators $\mathcal{M}_{\text{ctx}} = \{\Pi_{\text{base}}, \Pi_{\text{pers}}\}$.

Holomorphic Knowledge Embedding

Embeddings evolve via Cauchy–Riemann dynamics:

$$\frac{\partial E}{\partial \bar{z}} = 0 \quad \text{with } z = x + i\xi \in \mathbb{C}^{d_E \times d_{\text{pers}}},$$

ensuring analyticity in the joint embedding space.

Noncommutative Memory Operators

Memory operations form a C*-algebra:

$$[\mathcal{M}_i, \mathcal{M}_j] = i\epsilon_{ijk}\mathcal{M}_k \quad \text{with } \mathcal{M} \in \mathfrak{su}(d_{\mathcal{M}}),$$

capturing the inherent noncommutativity in dynamic memory updates.

1.4 Embeddings

- **Base Embedding:** $E_{\text{B}} : \mathcal{V} \to \mathbb{R}^{d_E}$.
- **Contextual Embedding:** $E_{\text{C}} : (\mathbb{R}^{d_E})^n \to (\mathbb{R}^{d_C})^n$, for a sequence of length n.
- **Fusion Operator:** $g : \mathbb{R}^{d_E} \times \mathbb{R}^{d_C} \to \mathbb{R}^{d_F}$.
- **Final Token Representation:** $\mathbf{z}_i = E_{\text{F}}(t_i, \xi) \in \mathbb{R}^{d_F}$, where ξ denotes contextual or personalized parameters.

1.5 Knowledge Graphs

- **Base Knowledge Graph (BKG):** $\mathcal{G}_{\text{BKG}} = (\mathcal{N}_{\text{BKG}}, \mathcal{E}_{\text{BKG}})$, with nodes defined by $E_{\text{B}}(t)$.
- **Personal Knowledge Graph (PKG):** $\mathcal{G}_{\text{PKG}} = (\mathcal{N}_{\text{PKG}}, \mathcal{E}_{\text{PKG}})$, derived from personalized embeddings $E_{\text{sup}}(t, \xi)$.
- **Graph Fusion Operator:** $\oplus_{\mathcal{K}} : \mathcal{G}_{\text{BKG}} \times \mathcal{G}_{\text{PKG}} \to \mathcal{G}_{\text{Unified}}$.
- **Unified Knowledge Graph:** $\mathcal{G}_{\text{Unified}} = \mathcal{G}_{\text{BKG}} \cup \mathcal{G}_{\text{PKG}}$.

Non-Euclidean Knowledge Fusion

Beyond simple set-union, knowledge integration is modeled in non-Euclidean spaces:

$$\mathcal{G}_{\text{Unified}} = \int_{\mathcal{M}_{\text{geom}}} \exp_{\mathcal{G}_{\text{BKG}}}\Big(t\,\mathcal{G}_{\text{PKG}}\Big)\,dt, \quad t \in [0,1],$$

reflecting continuous geometric transformations between graph domains.

1.6 Infinite RoPE and Dynamic Vocabulary

- **RoPE Transformation:** $\psi : \mathbb{N} \times \mathbb{R}^{d_{\text{att}}} \to \mathbb{R}^{d_{\text{att}}}$.
- **Frequency Parameters:** θ_j^* for $j = 1, \ldots, \frac{d_{\text{att}}}{2}$.
- **Dynamic Vocabulary Update:** $\Delta_{\mathcal{V}} : \mathcal{V} \times \mathcal{D}_{\text{learn}} \to \mathcal{V}'$.

Hypergeometric Tokenization

A novel tokenization scheme minimizes an energy function:

$$\mathcal{T}_{\text{base}}(X) = \arg\min_{\{t_i\}} \sum_{k=1}^{K} \left[\mathrm{Re}(z_k) - \sum_{j=1}^{J} \alpha_j \, \phi_j(t_k) \right]^2 + \lambda \, \Omega(\{\alpha_j\}),$$

where ϕ_j are basis functions and Ω is a regularizer.

1.7 Core Model Architecture

- **Deep Model:** $f_\theta : (\mathbb{R}^{d_F})^n \to \mathcal{Y}$, with parameters $\theta \in \Theta \subset \mathbb{R}^p$.
- **Transformer Sub-module:** $f_{\theta_{\mathrm{T}}}^{\mathrm{T}}$.
- **RWKV Sub-module:** $f_{\theta_{\mathrm{R}}}^{\mathrm{RWKV}}$, featuring recurrence variables S_t, Z_t and a decay vector $\boldsymbol{\lambda}$.
- **Expert Coordinator:** $\Gamma : \{f_{\theta_i}^{(i)}\}_{i \in I_{\exp}} \to f_\theta^{\text{Unified}}$.

1.8 Titans Memory Architecture

- **Memory Bank:** $\mathcal{M} = \{(k_i^{\mathcal{M}}, v_i^{\mathcal{M}})\}_{i=1}^{M_{\mathcal{M}}}$.
- **Similarity Function:** $s : \mathbb{R}^{d_L} \times \mathbb{R}^{d_k} \to \mathbb{R}$.
- **Attention Weights:**

$$\alpha_i(x) = \frac{\exp\Big(s\big(r, k_i^{\mathcal{M}}\big)/\tau\Big)}{\sum_j \exp\Big(s\big(r, k_j^{\mathcal{M}}\big)/\tau\Big)}.$$

- **Aggregated Memory Read:** $m(x) = \sum_{i=1}^{M_{\mathcal{M}}} \alpha_i(x) \, v_i^{\mathcal{M}}$.
- **Meta-Learner:** $h : \mathbb{R}^{d_v} \to \mathbb{R}^p$, producing the parameter update $\Delta\theta(x)$.

1.9 Recursive Adaptive Feedback

- **Recursive Entities:** $\mathcal{E}^{\mathrm{R}} = \{E_i^{\mathrm{R}}\}_{i \in I_{\mathrm{R}}}$.
- **Trigger and Action Functions:** $\tau^{\mathrm{R}} : \mathcal{E}^{\mathrm{R}} \to \mathcal{T}^{\mathrm{R}}$ and $\alpha^{\mathrm{R}} : \mathcal{E}^{\mathrm{R}} \to \mathcal{A}^{\mathrm{R}}$.
- **Influence Function:** $\Delta^{\mathrm{R}} : \mathcal{E}^{\mathrm{R}} \times \mathcal{E}^{\mathrm{R}} \to \mathcal{F}^{\mathrm{R}}$.
- **Feedback Composition:**

$$\phi^{\mathrm{R}}\big(E_i^{\mathrm{R}}, E_j^{\mathrm{R}}\big) = \Delta^{\mathrm{R}}\big(E_i^{\mathrm{R}}, E_j^{\mathrm{R}}\big) \circ \beta^{\mathrm{R}}\big(E_j^{\mathrm{R}}, E_i^{\mathrm{R}}\big).$$

- **Overall Runtime State:** Σ^{R}, updated via

$$\mu^{\mathrm{R}} : \Sigma^{\mathrm{R}} \to \Sigma^{\mathrm{R}},$$

satisfying

$$\mu^{\mathrm{R}}\Big(\mu^{\mathrm{R}}(\Sigma^{\mathrm{R}})\Big) = \mu^{\mathrm{R}}(\Sigma^{\mathrm{R}}).$$

1.10 Universal Training System

- **Loss Function:** $\mathcal{L} : \Theta \times \mathcal{D} \to \mathbb{R}_{\geq 0}$, with mini-batch loss

$$\mathcal{L}(\theta; B) = \frac{1}{|B|} \sum_{(x,y)\in B} \ell\Big(f_\theta(x), y\Big) + \lambda_W \, \|\theta\|^2 + \lambda_{\text{sparse}} \, \mathcal{R}_{\text{sparse}}(\theta).$$

- **Optimizer:** $\mathcal{O} : \Theta \times \nabla_\theta \mathcal{L} \times \Xi \to \Theta$, where Ξ represents the optimizer state (e.g., for AdamW).
- **Normalization Operator:** $N : \mathbb{R}^d \to \mathbb{R}^d$, defined as

$$N(x) = \gamma \odot \frac{x - \mu_x}{\sqrt{\sigma_x^2 + \epsilon}} + \beta.$$

- **Dropout Operator:** $D : \mathbb{R}^d \to \mathbb{R}^d$, using a Bernoulli mask $m \sim \text{Bernoulli}(1 - p)$.
- **Skip Connection:** $S(x, F(x)) = x + F(x)$.

1.11 Final Decoding and Multimodal Output

- **Decoding Function:** $\delta : \mathcal{Y} \to \mathcal{O}_{\text{mod}}$, where $\mathcal{O}_{\text{mod}}$ specifies the output modality (default: Text).
- **Modality Decision Function:** $\mu_{\text{mod}} : \mathcal{Y} \times \mathcal{C}_{\text{task}} \to \mathcal{O}_{\text{mod}}$, with $\mathcal{C}_{\text{task}}$ representing task requirements.

2 Enhanced Algorithmic Formalism

Algorithm 1 Eidos Quantum-Adaptive Inference with Module Annotations

1: **Input (Module A):** $|X_{\text{raw}}\rangle = \bigotimes_{k=1}^{K} |\sigma_k\rangle$, entangled dataset $\mathcal{D}_{\text{ent}}$
2: **Initialize (Module I):** $\rho_0 = |0\rangle\langle 0|^{\otimes n_{\text{qvm}}}$
3: **for** $t \in \mathbb{T}_{\text{adapt}}$ **(Dynamic Adaptivity) do**
4: $\quad |\psi_t\rangle \leftarrow \mathcal{F}_{\text{rope}}\Big(\bigotimes_i |z_i\rangle\Big)$ *(Module G: Infinite RoPE)*
5: $\quad$ Apply $\mathcal{U}_{\text{mem}} = \exp\big(-i\mathcal{H}_{\mathcal{M}} t\big)$, where $\mathcal{H}_{\mathcal{M}} = \sum_i \alpha_i \, \mathcal{M}_i$ *(Module I: Memory Integration)*
6: $\quad$ Measure: $\mathcal{G}'_{\text{Unified}} \leftarrow \Pi_{\text{ctx}} \circ \mathcal{M}_{\text{mem}}(\rho_t)$ *(Module F: Knowledge Graph Integration)*
7: $\quad$ Update: $\theta_{t+1} \leftarrow \theta_t \oplus \mathfrak{L}\big\{\mathcal{D}_{\text{ent}}, \mathcal{G}'_{\text{Unified}}\big\}$ *(Module K: Training System)*
8: $\quad \rho_{t+1} \leftarrow \mathcal{E}_{\text{noise}}(\rho_t) \otimes |\theta_{t+1}\rangle\langle\theta_{t+1}|$ *(Noise and Parameter Persistence)*
9: **end for**
10: **Output (Module L):** $\mathrm{Tr}\Big(\rho_{\text{final}}\Big) \otimes \mathcal{P}_{\text{out}}$

3 Complete Operator Taxonomy

Table 1: Eidos Operator Hierarchy

Symbol	Space	Properties
$\mathfrak{E}$	$\mathcal{L}(\mathbb{H}_{\text{emb}})$	Completely positive, trace-nonincreasing
$\mathcal{F}_{\text{rope}}$	$\mathbb{C}^{d_{\text{att}}}$	Rotational equivariance, $\mathfrak{so}(2n)$ representation
$\mathfrak{L}$	$\Theta \times \mathcal{D}^{\infty}$	Fréchet differentiable, ∇-parallel
$\mathcal{U}_{\text{mem}}$	$\mathbb{H}_{\mathcal{M}}$	Haar-random, ergodic

4 Complete Commutative Diagram

$$\begin{tikzcd}
\Sigma^* \arrow[r, "\mathcal{P}_{\text{in}}"] \arrow[d, "\textit{Entangle}"'] & \mathcal{X}_{\text{proc}} \arrow[r, "\mathcal{T}_{\text{base}}"'] & \mathcal{V}^{\otimes n} \arrow[d, "\mathfrak{E}"] \\
\mathbb{H}_{\text{raw}} \arrow[r, "\mathcal{U}_{\text{prep}}"'] & \mathbb{H}_{\text{emb}} \arrow[r, "\mathcal{F}_{\text{rope}}"'] & \mathbb{H}_{\text{ctx}} \arrow[d, leftrightarrow, "\mathcal{A}_{\text{fb}}\ \mathcal{M}_{\text{mem}}"] \\
& \mathbb{H}_{\mathcal{M}} \arrow[r, "\mathfrak{L}"'] & \Theta_{\text{adapt}}
\end{tikzcd}$$

5 Formal Verification

Consistency Proof

Consistency of the Recursive Feedback. Assume $\mu^{\mathrm{R}} : \Sigma^{\mathrm{R}} \to \Sigma^{\mathrm{R}}$ is a contraction mapping under the norm $\|\cdot\|$. Then, by the Banach Fixed-Point Theorem,

$$\|\mu^{\mathrm{R}}(\Sigma) - \mu^{\mathrm{R}}(\Sigma')\| \leq L\|\Sigma - \Sigma'\|, \quad \text{with } L < 1.$$

Since the idempotence axiom (A2) holds, we have $L = 0$, guaranteeing the existence of a unique fixed point Σ_{fix}. $\square$

Proof. Under the contraction mapping assumption on μ^{R}, the Banach Fixed-Point Theorem implies a unique fixed point exists. Given the idempotence condition,

$$\mu^{\mathrm{R}}(\mu^{\mathrm{R}}(\Sigma)) = \mu^{\mathrm{R}}(\Sigma),$$

conclude that $\Sigma_{\text{fix}} = \mu^{\mathrm{R}}(\Sigma_{\text{fix}})$ is unique. □

Universal Approximation Property

Universal Approximation. For any Borel measure ν defined on $(\mathbb{R}^{d_F}, \mathcal{B})$, there exists a parameter vector $\theta^* \in \Theta$ satisfying

$$d_{\mathrm{TV}}\Big(f_{\theta^*}(\mathcal{X}), \nu\Big) < \epsilon + \lambda_{\text{sparse}} \|\theta^*\|_{\ell^0},$$

where d_{TV} denotes the total variation distance. □

Proof. This follows from the Γ-density property of the mixture-of-experts in the function space $\mathcal{C}(\mathbb{R}^{d_F}, \mathcal{Y})$. By suitably choosing the expert parameters and sparse regularization, the target distribution ν can be approximated arbitrarily closely. □

6 End-to-End Data Flow and Processing Sequence

This section delineates the complete data and model flow within the Eidos framework.

Step 1: Input Processing (Module A)

- **Raw Input:** $X_{\text{raw}} \in \Sigma^*$ (e.g., a text document).
- **Preprocessing:** Compute $X_{\text{proc}} \leftarrow \mathcal{P}_{\text{in}}(X_{\text{raw}})$, where $X_{\text{proc}} \in \mathcal{X}_{\text{proc}}$.

Step 2: Universal Communication and Data Handling (Module B)

- Encapsulate each message as a universal data packet

$$\mathcal{P} = \Big(\text{ID}_{\mathcal{P}},\ \text{Payload}_{\mathcal{P}},\ \text{Meta}_{\mathcal{P}}\Big).$$

- Utilize communication channels $\mathcal{C}_{ij}$ with routing function $\mathcal{R}_{\text{comm}}$ to orchestrate module interactions.
- The coordination manager Ω registers modules and ensures reliable packet delivery.

Step 3: Universal Streaming, Loading, and Chunking (Module C)

- Partition model parameters as $\theta = \bigcup_{j \in J} T_j$.
- Index each chunk T_j by $\mathcal{I}(T_j) = \ell_j \in \mathcal{S}_{\text{disk}}$.
- Dynamically stream chunks into memory via σ and cache them using μ.

Step 4: Multidimensional Vocabulary and Tokenization (Module D)

- Define the complete vocabulary as $\mathcal{V} = \mathcal{V}^{(0)} \cup \mathcal{V}^{(1)}$.
- Assign each token $t \in \mathcal{V}$ a unique identifier, $\mathrm{ID}(t) = \eta(t)$.
- Segment the preprocessed input X_{proc} into tokens $(t_1, \ldots, t_n)$ using the base tokenizer $\mathcal{T}_{\text{base}}$.

Step 5: Contextual Embedding and Tokenization (Module E)

- For each token t_i, compute its base embedding $\mathbf{e}_i = E_{\mathrm{B}}(t_i) \in \mathbb{R}^{d_E}$.
- Process the sequence $(\mathbf{c}_1, \ldots, \mathbf{c}_n) = E_{\mathrm{C}}(\mathbf{e}_1, \ldots, \mathbf{e}_n)$.
- Fuse base and contextual embeddings: $\mathbf{z}_i = g(\mathbf{e}_i, \mathbf{c}_i) \in \mathbb{R}^{d_F}$.
- Integrate dual-layer representations via

$$\mathbf{z}_i = g\Big(E_{\mathrm{B}}(t_i),\, E_{\text{sup}}(t_i, \xi)\Big).$$

Step 6: Deep Knowledge Graph Construction (Module F)

- Build the Base Knowledge Graph:

$$\mathcal{G}_{\text{BKG}} = \Big(\mathcal{N}_{\text{BKG}},\, \mathcal{E}_{\text{BKG}}\Big),$$

  where nodes represent tokens t with embeddings $E_{\mathrm{B}}(t)$ and edges are defined by $\rho_{\text{base}}(t_i, t_j) \subset \mathcal{R}_{\text{base}}$.
- Construct the Personal Knowledge Graph:

$$\mathcal{G}_{\text{PKG}} = \Big(\mathcal{N}_{\text{PKG}},\, \mathcal{E}_{\text{PKG}}\Big),$$

  utilizing personalized embeddings $E_{\text{sup}}(t, \xi)$.
- Fuse the graphs via $\oplus_{\mathcal{K}}$ to obtain

$$\mathcal{G}_{\text{Unified}} = \mathcal{G}_{\text{BKG}} \cup \mathcal{G}_{\text{PKG}}.$$

Step 7: Infinite RoPE and Dynamic Vocabulary Updating (Module G)

- For each token position i and each 2D subspace j, define the rotation matrix

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

- Form the block-diagonal rotation:

$$R(i) = \mathrm{diag}\Big(R^{(1)}(i), \ldots, R^{(d_{\text{att}}/2)}(i)\Big), \quad \psi(i, v) = R(i)v.$$

- Apply $\psi(i, \cdot)$ to the query/key vectors in the attention mechanism.
- Dynamically integrate newly learned tokens via

$$\Delta_{\mathcal{V}} : \mathcal{V} \times \mathcal{D}_{\text{learn}} \to \mathcal{V}'.$$

Step 8: Core Model Processing (Module H)

- Process the fused embeddings through the deep model:

$$f_\theta : (\mathbb{R}^{d_F})^n \to \mathcal{Y}, \quad \theta \in \Theta.$$

- Sub-modules include:
  - **Transformer:** $f^{\mathrm{T}}_{\theta_{\mathrm{T}}}$, leveraging multi-head self-attention (enhanced with RoPE).
  - **RWKV:** $f^{\mathrm{RWKV}}_{\theta_{\mathrm{R}}}$ with recurrence defined as

$$S_t = \boldsymbol{\lambda} \odot S_{t-1} + \exp(k_t) \odot v_t, \quad Z_t = \boldsymbol{\lambda} \odot Z_{t-1} + \exp(k_t).$$

- Coordinate the expert models via

$$f^{\mathrm{Unified}}_\theta = \Gamma\Big(\{f^{(i)}_{\theta_i}\}_{i \in I_{\exp}}\Big).$$

Step 9: Titans Memory Architecture (Module I)

- Define the memory bank:

$$\mathcal{M} = \{(k^{\mathcal{M}}_i, v^{\mathcal{M}}_i)\}_{i=1}^{M_{\mathcal{M}}}.$$

- For a latent representation r, compute similarities $s\big(r, k^{\mathcal{M}}_i\big)$ and attention weights

$$\alpha_i(x) = \frac{\exp\Big(s\big(r, k^{\mathcal{M}}_i\big)/\tau\Big)}{\sum_j \exp\Big(s\big(r, k^{\mathcal{M}}_j\big)/\tau\Big)}.$$

- Aggregate the memory read:

$$m(x) = \sum_{i=1}^{M_{\mathcal{M}}} \alpha_i(x) v^{\mathcal{M}}_i.$$

- Obtain the parameter update via the meta-learner:

$$\Delta\theta(x) = h\big(m(x)\big), \quad \theta_x = \theta + \Delta\theta(x).$$

Step 10: Recursive Adaptive Feedback (Module J)

- Define runtime recursive entities: $\mathcal{E}^{\mathrm{R}} = \{E^{\mathrm{R}}_i\}$.
- Construct feedback using trigger τ^{R}, action α^{R}, and influence Δ^{R}:

$$\phi^{\mathrm{R}}\big(E^{\mathrm{R}}_i, E^{\mathrm{R}}_j\big) = \Delta^{\mathrm{R}}\big(E^{\mathrm{R}}_i, E^{\mathrm{R}}_j\big) \circ \beta^{\mathrm{R}}\big(E^{\mathrm{R}}_j, E^{\mathrm{R}}_i\big).$$

- Update the overall state: $\Sigma^{\mathrm{R}} \leftarrow \mu^{\mathrm{R}}(\Sigma^{\mathrm{R}})$ (ensuring idempotence).

Step 11: Universal Training System (Module K)

- For a mini-batch $B \subset \mathcal{D}$, compute the loss:

$$\mathcal{L}(\theta; B) = \frac{1}{|B|} \sum_{(x,y) \in B} \ell\Big(f_\theta(x), y\Big) + \lambda_W \, \|\theta\|^2 + \lambda_{\text{sparse}} \, \mathcal{R}_{\text{sparse}}(\theta).$$

- Update parameter chunks via the optimizer:

$$T_j \leftarrow \mathcal{O}\Big(T_j, \nabla_{T_j} \mathcal{L}, \Xi_j\Big).$$

- Within the forward pass, apply normalization N, dropout D, and skip connections S.
- Continuously stream and commit parameter updates using σ.

Step 12: Final Decoding and Multimodal Output (Module L)

- The deep model produces a latent output $y_{\text{latent}} \in \mathcal{Y}$.
- Decode the latent representation using $\delta : \mathcal{Y} \to \text{Text}$ to obtain $\hat{y}_{\text{text}} = \delta\big(y_{\text{latent}}\big)$.
- Use $\mu_{\text{mod}} : \mathcal{Y} \times \mathcal{C}_{\text{task}} \to \mathcal{O}_{\text{mod}}$ to determine any additional modalities (e.g., images, audio).
- Package the final output as

$$\mathcal{P}_{\text{out}} = \Big(\text{ID}_{\text{out}}, \hat{y}, \text{Meta}_{\text{out}}\Big),$$

  and route it via Ω for logging and feedback.

7 Integrated End-to-End Algorithm

Algorithm 2 Eidos Integrated Inference and Training Pipeline

Input: Raw input $X_{\text{raw}} \in \Sigma^*$, training set $\mathcal{D}$, task context $\mathcal{C}_{\text{task}}$
Preprocess: $X_{\text{proc}} \leftarrow \mathcal{P}_{\text{in}}(X_{\text{raw}})$
Tokenize: $(t_1, \ldots, t_n) \leftarrow \mathcal{T}_{\text{base}}(X_{\text{proc}})$
for $i = 1$ to n **do**
    $\mathbf{e}_i \leftarrow E_{\text{B}}(t_i)$
end for
$(\mathbf{c}_1, \ldots, \mathbf{c}_n) \leftarrow E_{\text{C}}(\mathbf{e}_1, \ldots, \mathbf{e}_n)$
for $i = 1$ to n **do**
    $\mathbf{z}_i \leftarrow g(\mathbf{e}_i, \mathbf{c}_i)$
    $\mathbf{z}'_i \leftarrow \psi(i, \mathbf{z}_i)$
end for
Knowledge Graph: Update $\mathcal{G}_{\text{BKG}}$ and $\mathcal{G}_{\text{PKG}}$, compute

$$\mathcal{G}_{\text{Unified}} \leftarrow \mathcal{G}_{\text{BKG}} \cup \mathcal{G}_{\text{PKG}}.$$

$y_{\text{latent}} \leftarrow f_\theta(\mathbf{z}'_1, \ldots, \mathbf{z}'_n)$
Test-Time Adaptation:
$r \leftarrow g_\theta(X_{\text{proc}})$
for $i = 1$ to $M_{\mathcal{M}}$ **do**
    $s_i \leftarrow s\big(r, k_i^{\mathcal{M}}\big)$
end for
Compute $\alpha_i \leftarrow \frac{\exp(s_i/\tau)}{\sum_j \exp(s_j/\tau)}$, and $m(x) \leftarrow \sum_i \alpha_i \, v_i^{\mathcal{M}}$
$\Delta\theta(x) \leftarrow h\big(m(x)\big), \quad \theta_x \leftarrow \theta + \Delta\theta(x)$
Recursive Feedback: $\Sigma^{\text{R}} \leftarrow \mu^{\text{R}}(\Sigma^{\text{R}})$
Decoding: $\hat{y} \leftarrow \delta\big(y_{\text{latent}}\big)$
if multimodal output needed **then**
    $\hat{y}_{\text{mod}} \leftarrow \mu_{\text{mod}}(y_{\text{latent}}, \mathcal{C}_{\text{task}})$
end if
$\mathcal{P}_{\text{out}} \leftarrow \Big(\text{ID}_{\text{out}}, \hat{y}, \text{Meta}_{\text{out}}\Big)$
if training mode **then**
    /* **Execute training loop (refer to supplementary Algorithm ??)** */
end if

8 Conclusion

We have presented the **Eidos** framework with exhaustive technical definitions and rigorous mathematical formalism, ensuring that every component is fully integrated and systematically defined. The document now features:

(A) **Input Processing:** The raw input X_{raw} is transformed into a processed form X_{proc} via the operator $\mathcal{P}_{\text{in}}$, with precise domain and range definitions.

(B) **Universal Communication and Data Handling:** Data packets $\mathcal{P}$ and robust communication channels $\mathcal{C}_{ij}$ are employed, with the coordination manager Ω ensuring integration across modules.

(C) **Streaming, Loading, and Chunking:** Model parameters θ, partitioned into chunks $\{T_j\}$, are systematically streamed and cached, ensuring consistent updates.

(D) **Multidimensional Vocabulary and Tokenization:** A comprehensive vocabulary $\mathcal{V}$ is structured with unique identifiers η, and tokenization via $\mathcal{T}_{\text{base}}$ is robustly defined.

(E) **Contextual Embedding and Tokenization:** Dual-layer representations are obtained by fusing E_{B} and E_{C} with the fusion operator g, integrating enhanced adaptability through E_{sup}.

(F) **Knowledge Graphs:** Graphs $\mathcal{G}_{\text{BKG}}$ and $\mathcal{G}_{\text{PKG}}$ are meticulously constructed and fused into $\mathcal{G}_{\text{Unified}}$, with non-Euclidean geometric integration.

(G) **Infinite RoPE and Dynamic Vocabulary:** The RoPE operator ψ and dynamic update operator $\Delta_{\mathcal{V}}$ ensure that positional encoding and vocabulary refinement occur seamlessly.

(H) **Core Model Architectures:** The deep model f_θ integrates both Transformer ($f^{\text{T}}_{\theta_{\text{T}}}$) and RWKV ($f^{\text{RWKV}}_{\theta_{\text{R}}}$) sub-modules, coordinated by Γ, for a unified inference process.

(I) **Titans Memory Architecture:** Memory bank $\mathcal{M}$, with similarity measures and meta-learner h, provides adaptive parameter updates $\Delta\theta(x)$.

(J) **Recursive Adaptive Feedback:** The operator μ^{R} enforces idempotence and systematic integration of recursive feedback.

(K) **Universal Training System:** Comprehensive loss functions $\mathcal{L}$, coupled with optimizers $\mathcal{O}$, and auxiliary operators N, D, and S, form a robust training loop.

(L) **Final Decoding and Multimodal Output:** The function δ and modality decision operator μ_{mod} ensure that the final output is delivered in the desired format, with extensibility to other modalities.

This rigorously defined and fully integrated framework offers a reliable blueprint for constructing a persistent, adaptive, and multimodal intelligent system, ensuring every aspect is consistent and systematically utilized from the input stage to the final output.