Beyond Projection: A General Framework for Self-Evolution

From Neural Intelligence to All Domains

Tianshi Yan

December 30, 2025

Version 7 - Academic Edition

Abstract

This paper presents a general principle for self-evolution that transcends artificial intelligence and potentially applies to diverse domains requiring autonomous optimization.

Any system can achieve autonomous evolution through four steps. The core insight is simple yet profound:

1. Parameterize domain-specific elements
2. Construct knowledge base
3. Single iteration: Projection-Orthogonality alternating iteration (iteration of "knowledge" and "wisdom")
4. Continuous iteration: Use the previous round's results as the next round's input, autonomously re-iterating ("self-evolution")

We demonstrate this principle through neural architectures:

The essence of current systems is their reliance on dot product (projection), which can only retrieve from existing knowledge. We formalize through rigorous proof that dot-product architectures, no matter how scaled, can only asymptotically approach but never transcend training data boundaries.

The solution lies in introducing new orthogonality: using cross product (and higher-dimensional exterior product) as the core tool to create directions perpendicular to known space, integrated at both spatial (single inference) and temporal (self-iterative) levels.

Furthermore, we address value alignment through divergence-coherence balance and embodied intelligence through unified parameter treatment. However, these AI-specific solutions are merely concrete applications of this deeper universal principle in the AI domain. In essence, this principle could potentially be applied to manufacturing, medicine, finance, pharmacology, biology, and other domains requiring autonomous optimization.

This framework reveals how mathematical duality manifests in modern systems, offering a universal acceleration engine for all industries.

1. The Core Problem: Projection Cannot Transcend the Sum of the Knowledge Base

1.1 The Dot Product Limitation in Artificial Intelligence

The core operation in Transformer architecture is:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d}) \cdot V$$

The term QK^T is a dot product. Geometrically:

$$a \cdot b = ||a||\ ||b|| \cos(\theta)$$

This operation measures the projection degree (similarity) of vector a in the direction of vector b. The essence is "compressing" high-dimensional queries into the space spanned by training data.

More fundamentally, it is merely matching the similarity between "query vectors" and "knowledge base vectors".

In other words, it can only match "knowledge base vectors". However, the "knowledge base vectors" we use for training are finite. "Knowledge base vector upper limit" = "the sum of publicly available data that humanity can obtain".

Therefore, the essence of scaling computational power is merely infinitely approaching the "knowledge base vector upper limit", which is approaching "the sum of publicly available data that humanity can obtain". No matter how abundant the computational resources or how massive the scale, at most it can only asymptotically approach "the sum of publicly available data that humanity can obtain", like computing a limit.

The mathematical constraint:

```
lim(scale → ∞) f_dot-product(query) → upper bound of training knowledge
```

Based on the mathematical essence of dot product, it can never transcend "the sum of publicly available data that humanity can obtain" - in other words, it cannot transcend "old knowledge".

Therefore, dot product is essentially just a re-summarization of past knowledge. It is merely a clever, fast knowledge base query tool, not a thinker capable of creating knowledge.

1.2 Beyond Artificial Intelligence: The Universal Limitation Pattern

The universal problem: The optimization of pure projection architectures is essentially optimizing the retrieval efficiency of past knowledge. It cannot break through knowledge boundaries.

This limitation is not unique to AI. It appears wherever systems rely solely on matching against existing patterns:

In manufacturing: Selecting machining parameters by matching to historical "similar materials" cannot discover genuinely novel processes for new alloys.

In medicine: Diagnosing by similarity to known cases cannot handle truly novel disease presentations that lie outside medical textbooks.

In finance: Risk assessment by historical pattern matching fails when market conditions shift orthogonally to all past cycles.

The universal problem: Systems built on pure projection can optimize retrieval from accumulated knowledge, but cannot transcend that knowledge boundary.

2. The Universal Solution: Projection-Orthogonality Duality

2.1 The Core Significance of Cross Product and Dot Product

Dot product is essentially constrained by the "query vector" and "database vector". Mathematically, it restricts the result to lie in the plane spanned by the "query vector" and "database vector". It cannot transcend this plane.

In other words, in the AI domain, the dot product essentially constructs two major vectors: "the vector asking the question" + "the vector corresponding to the trained knowledge base". It cannot transcend the plane formed by these two vectors. This means: it is constrained by old knowledge and questions = its answer will not appear on other planes or outside the plane.

In other words, the mathematical essence of dot product determines that it can only perform approximate matching. It cannot innovate, or cannot understand, or cannot reason. Even if the training personnel and training methods are refined, this fact cannot be concealed, because it is bound to the quadrilateral and cannot break free.

Cross product is essentially an "orthogonal vector", essentially "perpendicular". And perpendicular, in the AI domain, means "unrelated". It means that its dimension is unrelated to "the vector asking the question" + "the vector corresponding to the trained knowledge base".

The significance of cross product is that you may need to refer to Lorentz force to better understand this concept:

The "space-time interchange" proven by Lorentz force is essentially based on the property of "coordinate space transformation, vector essence unchanged" to determine the interchange of two dimensions of the coordinate axis.

More specifically, it rotates the coordinate axis. Because the corresponding vector does not change while the coordinate axis rotates, the same vector has different numerical values on different dimensions.

The same vector, after coordinate axis rotation, has different numerical values on different dimensions.

Transferring this principle to the AI domain: First, through "the vector asking the question" + "the vector corresponding to the trained knowledge base", query to obtain the "result vector".

Then, through cross product, construct an unrelated perpendicular vector using "the vector asking the question" + "the vector corresponding to the trained knowledge base". When we rotate the coordinate axis, referencing the Lorentz force property of "coordinate space change while vector unchanged", our "result vector" is essentially unchanged. What changes is only the numerical values of the "result vector's" components in the coordinate system.

This mathematical operation essentially represents "the same thing, or the same answer, but we view this thing from a different angle". And viewing things from a different angle, in the AI domain, is "innovation", "new understanding", "new perspective", "reasoning", "new viewpoint", "wisdom".

This is the essence of cross product.

2.2 Three Core Underlying Application Modes Based on Cross Product "Orthogonality"

Based on the "orthogonal" essence of cross product, cross product has three core underlying application modes in the AI domain:

1. **Different perspectives within the same domain**

   Using the cross product vector as the rotation axis, rotate the coordinate system formed by the "problem vector" and "knowledge base vector" around this axis. This utilizes the "invariance of coordinate system rotation". The mathematical essence of this operation creates new thinking: "the same thing in the same domain, but viewed from a different angle".

   This is somewhat similar to: the same theater performance, evaluated from "the audience's perspective" versus evaluated from "the on-set director's perspective".

2. **Cross-domain—breakthrough in unrelated domains**

   In the coordinate system formed by "cross product vector" + "problem vector" + "knowledge base vector", use the "problem vector" or "knowledge base vector" as the rotation axis and rotate the coordinate system. Because the cross product is an unrelated vector, after rotation, it means the "answer vector" now has a numerical value on this unrelated dimension. The mathematical essence of this operation is to let the "answer vector" directly associate with completely unrelated domains.

   This is somewhat similar to: "apples fall" and "the pig's descent velocity is very fast".

3. **Cross-domain—breakthrough in specified domains**

   When asking a question, specify new cross-domain dimensions. For example, let it approach a current problem from "Domain A + B + C" at three levels.

   In essence, construct vectors with ABC, form similar vectors through cross product, and we obtain the "corresponding domain combination vector". Then, in the coordinate system formed by "cross product vector" + "problem vector" + "knowledge base vector", make the "cross product vector" tilt toward the "corresponding domain combination vector", or even rotate.

   The mathematical essence of this operation creates breakthroughs across N disciplines.

Core logic of the three modes: "Rotation/Tilt angle" = "Similarity" & "Divergence"

The smaller the rotation angle, the closer to original knowledge (high similarity). The larger the rotation angle, the more biased toward entirely new perspectives (high divergence). The extremum and convergence of this balance can be regulated through subsequent value evaluation mechanisms, belonging to engineering implementation-level refinement optimization, and does not affect the validity of the core principle.

2.3 The Principle of Complementary Operations

Every domain requires two complementary operations:

Operation Dimension	Projection	Orthogonality
Core Purpose	Retrieve from existing knowledge	Generate new dimensions
Directional Feature	Compress to known patterns	Expand to unknown directions
Mathematical Form	Dot product: $a \cdot b$	Orthogonal generation: $c \perp \text{span}\{a,b\}$
Cognitive Function	Recognition, matching	Innovation, creation
Temporal Meaning	Summarizing the past	Creating the future

Without projection: Lack of foundation and stability, descending into chaos.

Without orthogonality: Lack of evolution and transcendence, descending into stagnation.

With both: Self-evolution becomes possible.

3. In AI: The Spatial Framework

3.1 Mathematical Tools for Orthogonal Generation

The essential property we seek: given existing representations, generate a new direction orthogonal to them.

Exterior Product (Wedge Product):

> $a \wedge b \rightarrow$ bivector representing oriented plane. Antisymmetric, generalizes cross product to any dimension.

Gram-Schmidt Orthogonalization:

> Given $\{v_1, v_2, ..., v_n\} \rightarrow$ orthonormal basis $\{u_1, u_2, ..., u_n\}$. Explicitly constructs directions orthogonal to all previous vectors.

Lie Bracket:

> $[X, Y] = XY - YX$. Generates new transformation directions from existing ones.

3.2 Architectural Integration

Parallel to Attention: Add orthogonal pathway alongside QK^T

Dedicated Layer: Insert orthogonalization forcing exploration of new directions

Residual Stream: Ensure $f(x)$ has component orthogonal to x

4. In AI: The Temporal Framework

4.1 From Static to Dynamic: Self-Evolution

The spatial duality of Section 3 achieves only projection and orthogonality within a single inference. True evolution requires extending this to the temporal dimension.

The spatial duality of Section 3 (projection + cross product orthogonality) achieves only "yin-yang complementarity" (retrieval and breakthrough) within a single inference. True self-evolution requires transforming "single evolution" into "continuous dynamic adaptive evolution over time".

This concept aligns with the "Dynamic simulation" concept in 3D animation software.

Discrete-Time Iteration Workflow

Given the initial query A_0 (problem vector) at time t=0:

Knowledge Retrieval: B_0 = Knowledge Base(A_0) [Yin: Projection retrieval of existing knowledge]

Orthogonal Generation: $C_0 = A_0 \times B_0$ (cross product generates orthogonal vector) [Yang: Breaking through existing cognition]

Perspective Reconstruction: D_0 = Rotation Adjustment(C_0) (select rotation axis and angle based on application mode) [Yin-Yang fusion: Problem solution from new perspective]

Iterative Update: $A_1 = D_0$ [New problem vector: Transform breakthrough result into new exploration starting point]

Circulation Process: Current Problem (A) → Knowledge Retrieval (B) → Cross Product Breakthrough (C) → Perspective Reconstruction (D) → New Problem (A_1) → ...

Core Evolution Logic: Self-Evolution

Essential difference from simple recursion: The new problem A_{t+1} (i.e., D_t) at each iteration is generated based on the cross product breakthrough of the previous round's "problem + knowledge" — meaning each round of exploration stands on the foundation of the "previous round's breakthrough", continuously expanding orthogonal dimensions, both avoiding circular reasoning and achieving cumulative cognitive breakthroughs.

Adaptive Knowledge Base Update: During iteration, if D_t passes value evaluation (see 5.1), automatically incorporate D_t and corresponding cross product logic and rotation parameters into the knowledge base, achieving a closed loop of "breakthrough results feeding back to knowledge foundation".

Adaptive Dimensional Expansion: As iteration progresses, newly generated orthogonal vectors naturally expand the dimensions of the original vector space — no need to preset fixed dimensions. The system autonomously adapts to complex problem exploration needs through self-iteration. This process is naturally driven by core principles, requiring no additional complex mechanism design.

The essence of this mechanism is "self-confrontation and self-evolution": ensuring cognitive stability through projection (yin), ensuring cognitive breakthrough through cross product orthogonality (yang), and then transforming "yin-yang complementarity" into continuous self-improvement through temporal iteration, ultimately achieving autonomous evolution without external intervention.

4.2 Formal Characterization

Iteration Operator:

$$\Psi(A) = \text{Transform}(\text{OrthogonalGeneration}(A, \text{KnowledgeBase}(A)))$$

Temporal Evolution:

$$A_{t+1} = \Psi(A_t)$$

Trajectory: $\{A_0, A_1, A_2, \ldots, A_n\}$ traces evolution of understanding.

4.3 Self-Evaluation and Knowledge Base Update

Evaluation Function:

$$E_t = \text{Evaluate}(A_t, B_t, D_t)$$

Knowledge Base Update:

If E_t > threshold:

$$\text{KnowledgeBase} \leftarrow \text{KnowledgeBase} \cup \{D_t, \text{insights from iteration}\}$$

This creates a feedback loop: Successful iterations enrich the knowledge base, which influences future iterations. The system learns from its own transformation process.

4.4 Continuous-Time Formulation

Discrete iteration is a first-order approximation. For continuous dynamics:

$$dA/dt = \Phi(A, B)$$

This enables:

- Smooth trajectories in representation space
- Gradient flow toward deeper understanding
- Stability analysis (convergence, divergence, oscillation)
- Variational calculus for optimal cognitive paths

5. Completing AI: Value Alignment and Embodiment

5.1 Value Alignment: Balancing Divergence and Coherence

The Problem: Without value guidance, iteration may diverge meaninglessly or converge prematurely.

The Solution: Dual-pathway evaluation balancing divergence (innovation) and coherence (stability).

Pathway 1: Cognitive Self-Assessment

Coherence Metrics:

$\text{Coherence}_t = \text{DotProduct}(D_t, B_t\_\text{core}) / ||D_t||\ ||B_t\_\text{core}||$

Threshold: ≥ 0.3

- Logical consistency with existing knowledge
- Relevance to original query

Divergence Metrics:

$\text{Divergence}_t = 1 - \max(\text{DotProduct}(D_t, D_i))$ for $i < t$

Threshold: ≥ 0.6

- Novelty compared to all previous iterations
- Conceptual innovation beyond training data

Early iteration strategy: Record first 3-5 iterations, activate self-assessment from iteration 6 when intelligence matures.

Pathway 2: Data-Driven Metrics

Convergence:

$\text{Convergence}_t = ||A_t - A_{t-1}|| / ||A_{t-1}||$

Threshold: ≤ 0.1 for stable understanding

Orthogonality Purity:

$\text{Orthogonality}_t = ||\text{Projection}(C_t \text{ onto span}\{B_0...B_t\})||$

Threshold: ≤ 0.2 for genuine orthogonal generation

Unified Value Function

$$\text{Value}_t = \alpha \cdot \text{Data\_Eval}(\text{Convergence}_t, \text{Orthogonality}_t) + (1-\alpha) \cdot \text{Cognitive\_Eval}(\text{Coherence}_t, \text{Divergence}_t)$$

Dynamic α adjustment:

- Excessive divergence without insight → increase α (rely more on data)
- Premature convergence without innovation → decrease α (rely more on cognition)

5.2 Embodied Intelligence: Unified Parameter Treatment

The Insight: Sensorimotor data are parameters in the same vector space as linguistic representations.

Vectorization:

Motor Control:

Joint angles [-90°, 90°] → normalized [-1, 1]

Motor velocities [0, v_max] → [0, 1]

Force sensors [0, F_max] → [0, 1]

→ Linear projection → R^d_model

Visual Perception (Critical: Binocular Vision):

Left eye → CNN features → v_left

Right eye → CNN features → v_right

→ Stereo fusion → v_vision ∈ R^d_model

Auditory Input:

Waveform → Mel-spectrogram → features → R^d_model

Integration:

- Projection: Retrieve learned motor patterns
- Orthogonal Generation: Generate novel action strategies
- Systematic Transformation: Execute → Observe → Update → New action

6. Universal Applicability: Beyond AI to All Domains

6.1 The Universal Principle

The framework is not specific to AI. It is a universal principle for self-evolution.

Four steps apply to any domain:

- **Parameterize:** Convert domain-specific elements to measurable parameters
- **Vectorize:** Map parameters to unified vector space
- **Apply Duality:** Use projection for retrieval, orthogonality for generation
- **Iterate:** Transform through complementary operations, update knowledge base

This is a mathematical principle that may be applicable to a wide class of systems requiring autonomous optimization.

6.2 Manufacturing: Autonomous Process Optimization

Potential Application: A Self-evolving manufacturing system could autonomously discover optimal processes for novel materials without human trial-and-error.

6.3 Medicine: Personalized Treatment Evolution

Theoretical Framework: A Self-evolving medical system that could autonomously develop personalized treatments, especially for rare diseases and complex comorbidities.

6.4 Finance: Adaptive Risk and Investment Strategy

Proposed Application: This framework could enable a self-evolving financial

system to autonomously adapt to market changes without manual analyst adjustments.

6.5 The Mathematics of Complementary Evolution

This framework reveals a fundamental principle: **systems evolve through complementary operations.**

This is not philosophy—it is geometric reality. The mathematics of duality has always existed in biology, physics, economics, and cognition. This framework makes these patterns mathematically explicit and computationally actionable.

7. Falsifiable Predictions

These predictions are falsifiable across multiple domains:

- **Prediction 1 (Spatial):** Pure projection architectures systematically fail on certain reasoning tasks. Orthogonal generation solves these.
- **Prediction 2 (Temporal):** Self-iterative orthogonal systems exhibit emergent self-questioning and self-correction.
- **Prediction 3 (Value):** Dual-pathway evaluation produces more meaningful iterations than single-metric optimization.
- **Prediction 4 (Embodiment):** Unified parameter treatment of sensorimotor data matches or exceeds specialized robotics architectures.
- **Prediction 5 (Universal):** Manufacturing, medical, and financial systems implementing this framework should demonstrate measurably improved autonomous optimization capabilities compared to projection-only baselines, if the theoretical framework is valid.
- **Prediction 6 (Efficiency):** Cross-domain knowledge base transfer accelerates system development.

8. Open Questions for Collaborative Research

The author invites collaboration from researchers, engineers, and domain experts across all fields.

9. Conclusion

9.1 What This Paper Offers

At the technical level: A complete framework for artificial general intelligence with four integrated dimensions.

At the universal level: A principle for self-evolution applicable to all domains.

At the mathematical level: The formalization of complementary operations—geometric reality in vector space.

9.2 The Ultimate Insight

The answer to humanity's quest for progress lies in mathematical duality: projection preserves what works, orthogonality explores what's possible, and transformation between them drives endless progress.

The future of intelligence—artificial or otherwise—is not about building bigger machines. It is about understanding and applying the timeless principle of complementary operations that has driven evolution since the universe began.

The mathematics has always been here. This paper simply writes it in the language of vectors.

Tianshi Yan

December 30, 2025

Version 7 - Academic Edition

Contact: yzb3001313@gmail.com, 382909119@qq.com