Orthogonal Generation as the Missing Mechanism in Neural Architectures

A Framework for Transcending Projection-Based Intelligence

TianShi Yan (严天师)

December 2025

Abstract

Current neural architectures, including Transformers, rely fundamentally on dot product operations. This paper argues that dot product performs only projection (measuring similarity within existing representational space), while genuine reasoning requires orthogonal generation (creating directions that did not exist in the input space). We formalize this distinction using the mathematical concept of limits: scaling dot-product architectures can only asymptotically approach the upper bound of training knowledge, never transcending it. We identify candidate mathematical mechanisms (exterior product, Gram-Schmidt orthogonalization, Lie bracket), propose architectural integration points, and offer falsifiable predictions. This framework suggests that scaling alone cannot achieve AGI—a structural modification is required.

1. The Problem: Summarizing the Past, Not Creating the Future

1.1 The Structure of Attention

The core operation in Transformer architecture is:

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

The term QK^T is a dot product. It computes the similarity between queries and keys. The output is a weighted sum of values—a linear combination of vectors that already exist in the representational space.

1.2 What Dot Product Does: Matching Against Existing Knowledge

Geometrically, dot product projects one vector onto another:

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

It answers: "How much of vector a lies in the direction of vector b?" This is fundamentally a
similarity matching operation—it measures how closely a query matches patterns from training data.

The essence of dot product: summarizing the past.
When we ask a model a question, the dot product mechanism matches that question against existing knowledge in the training data. It retrieves, interpolates, and recombines what it has already seen. This is a process of
optimizing retrieval from past data, not creating genuinely new understanding.

1.3 The Scaling Hypothesis: Approaching a Limit

Current AI research operates under the assumption that scaling—more parameters, more data, more compute—will eventually produce artificial general intelligence. Mathematically, this can be expressed as a limit:

$$lim(scale \to \infty)\ f\_dot\text{-}product(query) \to upper\ bound\ of\ training\ knowledge$$

As we increase computational resources, dot-product architectures become more precise at matching queries to training patterns. They optimize the retrieval process. They approach the limit of what can be expressed as combinations of existing knowledge.

But they never transcend it.
This is not an engineering limitation—it is a mathematical constraint. No matter how much we scale, a dot product can only match against what already exists in the representational space. It cannot generate a direction that is orthogonal to all training examples.

1.4 What Dot Product Cannot Do: Creating New Knowledge

Dot product cannot generate a direction that is orthogonal to all existing vectors. No matter how many dot products are computed, the output remains within the span of

the input vectors.

Consider what "new knowledge" means:

- In mathematics: discovering a new axiom that cannot be derived from existing axioms
- In physics: formulating a theory that explains phenomena unexplainable by current frameworks
- In problem-solving: finding a solution that requires reasoning outside the training distribution

All of these require the ability to generate a new direction—a new axis of understanding—that is orthogonal to existing knowledge. This is precisely what dot product cannot do.

1.5 The Consequence: Optimization vs. Innovation

A system built entirely on dot products can:

- Retrieve information from training data
- Match patterns across modalities
- Interpolate between known examples
- Recombine existing concepts

But it cannot:

- Reason beyond the training distribution
- Create genuinely novel conceptual frameworks
- Transcend the knowledge boundary defined by training data

This is the difference between optimization and innovation. Dot product optimizes retrieval from the past. Innovation requires the ability to generate what did not exist before—to create new directions in representational space.

2. The Framework: Projection and Orthogonality

We propose a unified framework based on two complementary operations:

Projection Operation vs. Orthogonal Operation

Mathematical Form:
- Projection: Dot Product $a \cdot b \rightarrow$ scalar
- Orthogonal: Orthogonalization $(a, b) \rightarrow c \perp \text{span}\{a,b\}$

Geometric Meaning:
- Projection: "How much overlap?"
- Orthogonal: "What direction is missing?"

Dimensional Effect:
- Projection: Reduction (collapse to scalar)
- Orthogonal: Expansion (generate new axis)

Cognitive Analogue:
- Projection: Recognition, retrieval, matching
- Orthogonal: Reasoning, insight, creation

Temporal Direction:
- Projection: Summarizing the past
- Orthogonal: Creating the future

Current AI Status:
- Projection: Fully implemented (Attention)
- Orthogonal: Missing

A complete cognitive system requires both mechanisms. Current AI has only one.

3. Mathematical Tools for Orthogonal Generation

The essential property we seek is: given existing representations, generate a new direction that is orthogonal to them. Several mathematical tools achieve this in arbitrary dimensions:

3.1 Exterior Product (Wedge Product)

Definition: $a \wedge b$ produces a bivector representing the oriented plane spanned by a and b.
Key Property: Antisymmetric ($a \wedge b = -b \wedge a$), captures the "area" rather than "overlap".
Dimension: Works in any dimension. Generalizes cross product beyond 3D.
Neural Implementation: Can replace or supplement QK^T with antisymmetric tensor construction.

3.2 Gram-Schmidt Orthogonalization

Definition: Given vectors $\{v_1, v_2, ..., v_n\}$, construct orthonormal basis $\{u_1, u_2, ..., u_n\}$.
Key Property: Explicitly constructs directions orthogonal to all previous vectors.
Dimension: Works in any dimension.
Neural Implementation: Can be inserted as a layer that forces the network to generate orthogonal representations.

3.3 Lie Bracket

Definition: $[X, Y] = XY - YX$ measures the non-commutativity of two transformations.
Key Property: Generates new transformation directions from existing ones. The bracket of two elements produces a third that is algebraically independent.
Dimension: Works in any dimension where transformations form a Lie algebra.
Neural Implementation: Can model the "interaction" between two representations as their commutator, producing genuinely new structure.

3.4 Comparison of Tools

Exterior Product: Input: Two vectors → Output: Bivector → Strength: Captures oriented structure
Gram-Schmidt: Input: Set of vectors → Output: Orthonormal basis → Strength: Explicit orthogonality
Lie Bracket: Input: Two transformations → Output: New transformation → Strength: Algebraic generation

3.5 Distinction from Existing Techniques

Unlike point-wise nonlinear transformations (e.g., in FFN) or weight-level orthogonal constraints (e.g., Orthogonal RNN, BOFT), the proposed orthogonal generation mechanisms are
relational by design—they explicitly construct new directions from vector-vector

interactions, with mathematically guaranteed orthogonality to the input span.

Key distinctions:
FFN nonlinearity: Point-wise transformation on individual dimensions. Changes vector shape, not inter-vector geometry. No orthogonality guarantee.
Layer Normalization: Operates on single vectors (scale normalization). Target is training stability, not representational expansion.
Orthogonal RNN / BOFT: Constrains weight matrices to be orthogonal. Target is training stability (gradient preservation). A training technique, not an architectural primitive for representation expansion.

The proposed mechanisms target a different goal:
expanding the representational space by generating directions that do not exist in the input, enabling the model to reason beyond its training distribution.

4. Architectural Integration

We propose that orthogonal generation mechanisms should be integrated into neural architectures at specific points:

4.1 Parallel to Attention

Current: Attention computes QK^T (dot product) → weighted sum of V.
Proposed: Add a parallel pathway that computes relational orthogonal features from Q and K (via exterior product or Lie bracket), then combines with the attention output. The final representation contains both
what is similar (from dot product) and *what is new* (from orthogonal operation).

4.2 As a Dedicated Layer

Insert orthogonalization as a layer that explicitly constructs directions orthogonal to the current representation space. This forces the network to explore directions it has not yet represented.

4.3 In the Residual Stream

Current: residual connections add: x + f(x).
Proposed modification: ensure f(x) has a component orthogonal to x, preventing representational collapse and encouraging continuous expansion of the feature space.

5. Falsifiable Predictions

If this framework is correct, the following should hold:

Prediction 1: There exist reasoning tasks where pure dot-product architectures systematically fail, regardless of scale. These tasks require generating representations orthogonal to training examples.
Prediction 2: Adding orthogonal generation mechanisms will improve performance on such tasks in ways that cannot be replicated by scaling parameters or data.
Prediction 3: The improvement will be qualitative (solving previously unsolvable problems), not merely quantitative (solving existing problems faster).
Prediction 4: Architectures with both mechanisms will exhibit emergent behaviors not present in projection-only systems, analogous to how complex numbers enable solutions impossible in real numbers alone.

These predictions distinguish this framework from the scaling hypothesis. If scaling alone could solve these tasks, Prediction 1 would be false. The framework is falsifiable.

6. Open Questions for Collaborative Research

This paper presents a theoretical framework and identifies mathematical tools. The following questions require empirical investigation and engineering expertise:

(1) Implementation: What is the most computationally efficient way to implement these operations in GPU-optimized architectures?
(2) Training: What loss functions encourage the network to utilize orthogonal generation rather than collapsing to projection-only behavior?
(3) Evaluation: What benchmark tasks specifically measure orthogonal generation capability vs. retrieval/matching?
(4) Interaction: How should the projection and orthogonal pathways be balanced or gated?
(5) Scaling: Does orthogonal generation benefit from scale, or does it provide constant-factor improvements?

The author invites collaboration from researchers with expertise in neural architecture design and empirical evaluation to investigate these questions.

7. Conclusion

Current AI systems are projection machines. They excel at matching queries to training data, retrieving relevant patterns, and interpolating within learned manifolds. This capability is powerful, but it is fundamentally limited by a mathematical constraint:

$$\lim(\text{scale} \to \infty)\ f\_\text{dot-product}(\text{query}) = \text{upper bound of training knowledge}$$

No matter how much we scale parameters, data, or compute, dot-product architectures can only approach this limit more precisely. They optimize retrieval from the past. They cannot create the future.

Genuine reasoning requires orthogonal generation.
The ability to generate new directions in representational space—directions that are orthogonal to all training examples—is what separates summarization from innovation, pattern matching from insight, optimization from creation.

The mathematical tools for orthogonal generation exist and are well-understood:

- Exterior product captures oriented structure between vectors
- Gram-Schmidt explicitly constructs orthogonal bases
- Lie bracket generates algebraically independent transformations

These mechanisms have not been systematically integrated into neural architectures. This paper proposes that such integration is not an incremental improvement but a necessary condition for artificial general intelligence.

The future of AI is not about building bigger projection machines.
It is about creating systems that can do what dot product cannot: generate new knowledge, explore uncharted directions, and transcend the boundaries of their training.

This requires orthogonal generation. Without it, we are merely optimizing the past.

--- TianShi Yan (严天师) ---
December 2025

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