Logical Embedding and Deduction as Stability

Y.Matsuda and ChatGPT5.2

January 9, 2026

Abstract

This paper proposes a reframing of deduction as a stabilized regime of cognitive dynamics, rather than as a rule-based inference operation. We argue that what appears as deductive behavior emerges when inductive learning, shaped by memory and constraint, converges to a consolidated state in which judgments become reproducible and invariant under further experience. We refer to this perspective as *deduction as stability*.

To make this view computationally explicit, we introduce *logical embedding* as a lightweight representational substrate in which concepts, situations, and actions are represented as vectors in a shared semantic space. Approximate logical relations are realized through simple geometric operations such as weighted composition, similarity, and thresholding, without assuming explicit symbolic rules or full Bayesian inference. Within this framework, induction corresponds to the continuous reshaping of semantic geometry, while deduction emerges when such reshaping is gated by stability and constraint.

We present a layered architecture comprising induction, memory, constraint, policy, and stability layers, and formalize their interactions. This architecture clarifies how rule-like behavior can arise from coordinated stabilization across layers, rather than from explicit rule execution. Finally, we examine the intrinsic limitations of logical embedding, including the absence of formal learning guarantees and genuine novelty generation, and argue that these limitations can be mitigated through modular extensions that preserve the semantic core.

1 Introduction

Deduction has traditionally been regarded as a hallmark of rational cognition. In classical logic and symbolic artificial intelligence, deductive reasoning is defined as the application of explicit rules that guarantee the correctness of conclusions given a set of premises. Under this view, deduction is an operation: a discrete, rule-governed transformation from premises to conclusions.

However, everyday human reasoning rarely conforms to this ideal. Outside formal domains such as mathematics or law, human deductive behavior is often approximate, context-dependent, and resistant to explicit formalization. People routinely act as if they were following rules, yet those rules are seldom articulated in advance, and are frequently revised, suspended, or ignored under changing circumstances. This gap between normative definitions of deduction and observed human behavior raises a fundamental question: *what, in practice, constitutes deductive reasoning?*

In this paper, we propose a reframing of deduction. Rather than treating deduction as an inference operation, we argue that deduction should be understood as a *stable regime of cognitive dynamics*. Under this view, a system behaves deductively when its judgments become reproducible, insensitive to small perturbations, and invariant under further experience. What appears as a logical rule to an external observer corresponds internally to a stabilized configuration of representations shaped by prior induction and constraint. We refer to this perspective as *deduction as stability*.

This reframing shifts the focus of analysis. The central problem is no longer how explicit rules are represented or applied, but how inductive processes give rise to stable, rule-like behavior over time. Deduction is not opposed to induction; it is the asymptotic outcome of inductive adaptation under memory and constraint. Understanding deduction therefore requires a model in which stability can be defined, measured, and regulated.

Logical embedding provides such a model. In logical embedding, concepts, situations, and actions are represented as vectors in a shared semantic space. Reasoning is realized through simple geometric operations such as weighted composition, similarity, and thresholding, rather than through symbolic manipulation. At an intuitive level, logical embedding treats meaning as location in space, and reasoning as movement and stabilization within that space.

This representation enables a natural account of how inductive experience accumulates into apparent rules. Consider a familiar example from everyday consumption. A person repeatedly encounters discounted products, low account balances, and impulsive desires. Early behavior may be inconsistent: sometimes purchasing, sometimes abstaining. Over time, however, certain combinations of cues come to be reliably associated with avoidance. In logical embedding, this process is modeled as the gradual alignment of composite situational vectors with a region of semantic space associated with risk. Avoidance emerges not because a symbolic rule has been explicitly learned, but because similar situations come to occupy nearby locations that reliably map to the same action.

From this perspective, induction corresponds to the continuous reshaping of semantic geometry, while deduction corresponds to the exploitation of stabilized configurations. A judgment appears deductive when learning updates no longer alter the outcome, and when the same interfaces consistently produce the same behavior. Stability, rather than formal validity, becomes the defining criterion of deduction.

The goal of this paper is to articulate this perspective in a precise computational form. We propose a layered architecture in which inductive synthesis, episodic memory, external constraints, policy selection, and stability-based gating interact to produce behavior interpretable as deductive. By formalizing these interactions, we aim to clarify both the power and the limits of reasoning understood as stabilization, and to situate logical embedding as a semantic core within a broader cognitive architecture.

2 Logical Embedding: Core Representation

Logical embedding is a representational and operational framework in which reasoning is realized through structured transformations of continuous semantic representations. Unlike classical logic, which treats symbols as discrete and rule-governed, logical embedding treats concepts, situations, and actions as elements of a shared geometric space, and defines reasoning in terms of relations within that space.

2.1 Definition of Logical Embedding

Formally, logical embedding assumes a semantic space $\mathcal{E} \subset \mathbb{R}^d$ equipped with a similarity structure. Each concept x is represented by a normalized vector $\mathbf{e}(x) \in \mathcal{E}$. The meaning of a concept is not intrinsic to the vector itself, but is defined relationally by its position with respect to other vectors in the space.

Logical embedding differs from standard embedding models in its intended interpretation. The embedding space is not merely a substrate for prediction or classification, but the primary medium

in which approximate logical relations are expressed. In this sense, logical embedding is not a feature representation, but a representational logic.

2.2 Primitive Operations

Reasoning in logical embedding is defined by a small set of primitive operations that act directly on embeddings. These operations are intentionally limited in number and computational complexity, reflecting the constraints of cognitively plausible reasoning.

Weighted Composition Given embeddings $\mathbf{e}(x_1), \ldots, \mathbf{e}(x_n)$ and weights $w_1, \ldots, w_n$, a composite embedding is defined as

$$\mathbf{v} = \text{norm}\left(\sum_{i=1}^{n} w_i \mathbf{e}(x_i)\right).$$

This operation serves as an approximate realization of conjunction, disjunction, or contextual binding, depending on the weights and participating concepts. Importantly, composition does not introduce new primitives; it produces a situated meaning as a point in semantic space.

Similarity and Distance Similarity between embeddings is measured by cosine similarity, with distance defined as its complement. These measures provide the basic mechanism for approximate equivalence, implication, and category membership. Logical embedding therefore replaces binary predicates with graded relational judgments.

Thresholded Evaluation Discrete decisions are obtained by thresholding continuous measures:

$$\text{IF } d(\mathbf{v}, \mathbf{t}) < \tau \text{ THEN True.}$$

This operation corresponds to an approximate conditional, whose validity depends on proximity rather than formal entailment. Thresholds may be fixed, adaptive, or history-dependent, allowing constraints to shape inference behavior.

Projection and Scaling Optional projection operators restrict embeddings to subspaces associated with particular features, while scaling operations modulate confidence or salience. Together, these operations enable conditional and weighted reasoning without introducing symbolic variables.

2.3 Logical Status of the Operations

It is crucial to emphasize that these operations are not intended to preserve logical correctness in the classical sense. They do not satisfy the axioms of propositional or predicate logic, nor do they guarantee soundness or completeness.

However, this is not a deficiency to be remedied, but a defining characteristic of the framework. Logical embedding deliberately abandons formal guarantees in order to model a different phenomenon: the emergence of stable, rule-like behavior from continuous, experience-driven adaptation.

Logical relations in this framework are therefore neither true nor false in an absolute sense. They are better understood as *geometrically stabilized tendencies*. An implication holds when relevant embeddings consistently occupy a constrained region of space, and a rule emerges when such configurations remain invariant under further experience.

2.4 Interpretational Commitment

By adopting logical embedding, we commit to a view of reasoning in which logic is not an external calculus applied to representations, but an emergent property of representational dynamics themselves. What appears as a logical rule to an observer corresponds internally to a stable geometric configuration, maintained by memory, constraint, and limited adaptation.

This commitment underlies the central thesis of this paper: deductive behavior is not implemented as rule execution, but arises when inductive processes converge to stable semantic structures.

3 Layered Architecture

We model cognition as an interaction among a small number of layers, each corresponding to a distinct computational role. The guiding principle is separation of concerns: (i) induction synthesizes a situational representation from experience, (ii) memory provides episodic regularization and fast recall, (iii) constraints impose history-dependent gating and external limits, (iv) policy realizes action selection that appears deductive, and (v) stability regulates when and how learning is allowed to alter meaning.

This layered view is not merely an engineering convenience. It expresses a theoretical commitment: deductive behavior should not be assumed as a primitive faculty, but should emerge from the stabilized coupling among induction, memory, and constraint. Accordingly, we define explicit interfaces between layers in terms of the operands and operations introduced earlier.

3.1 Common Operands and Interface Signals

Across layers, the system exchanges a small set of signals:

- Concept embeddings $\mathbf{e}(x)$ for concepts x (including context and action prototypes).
- A situational composite vector $\mathbf{v}_t$ produced by induction.
- A discrete latent state z_t (e.g., risk on/off) produced by constraint gating.
- A policy vector $\mathbf{p}_t$ produced by the policy layer.
- Action scores $\pi_t(a)$ and the selected action a_t.
- A stability score S_t that gates learning updates.

Each layer may maintain internal state (e.g., coefficient estimates, memory buffers, spending counters), but communicates through these shared signals. This allows the architecture to scale without expanding the set of primitive operations.

3.2 Induction Layer (Vector Synthesis under Experience)

The induction layer is responsible for constructing a compact representation of the current situation. Rather than explicitly reasoning with symbols, it synthesizes a composite embedding from a small set of cues:

$$\mathbf{v}_t = \mathrm{norm}(\alpha_t \mathbf{e}(\mathrm{Sale}) + \beta_t \mathbf{e}(\mathrm{LowBalance}) + \eta_t \mathbf{e}(\mathrm{Context}) + \kappa_t \mathbf{e}(\mathrm{Impulse})).$$

The coefficients $\alpha_t, \beta_t, \eta_t, \kappa_t$ encode the current inductive hypothesis about how strongly each cue should influence the situational meaning. In cognitively plausible settings, these coefficients are

not estimated by expensive optimization, but by incremental adaptation (e.g., exponential moving averages), reflecting the idea that induction is slow and noisy.

The output of the induction layer, $\mathbf{v}_t$, is not itself a decision. It is an intermediate operand that can be interpreted differently depending on constraints and memory. This design prevents the induction layer from becoming a monolithic predictor: it merely shapes the geometry in which later selection occurs.

3.3 Memory Layer (Episodic Regularization and Recall)

The memory layer stores episodic associations between past situational embeddings and actions (or outcomes). We represent memory as a buffer

$$\mathcal{M} = \{(\mathbf{v}_i, a_i, w_i)\}_{i=1}^{N},$$

and define recall as a nearest-neighbor aggregation:

$$\hat{a}_t = \arg\max_{a} \sum_{i \in \mathcal{N}_k(\mathbf{v}_t)} w_i \mathbb{1}[a_i = a] s(\mathbf{v}_i, \mathbf{v}_t).$$

Intuitively, memory provides a fast, low-cost approximation to "what worked in similar situations." This is particularly important when inductive synthesis is ambiguous or when constraints yield borderline states.

Memory also functions as a form of regularization: it discourages abrupt shifts in behavior by biasing decisions toward previously stable action patterns. In this sense, memory is not merely storage but an active stabilizing operator.

3.4 Constraint Layer (History-Dependent Gating and External Limits)

The constraint layer introduces non-geometric structure that cannot be reliably captured by embeddings alone. Two forms are central.

First, *history-dependent gating* (e.g., hysteresis) prevents oscillation between states:

$$z_t = \begin{cases} 1 & d(\mathbf{v}_t, \mathbf{e}(\text{Risk})) < \tau_{\text{on}} \\ 0 & d(\mathbf{v}_t, \mathbf{e}(\text{Risk})) > \tau_{\text{off}} \\ z_{t-1} & \text{otherwise.} \end{cases}$$

This implements a cognitively plausible form of persistence: once risk is "on," it should not immediately switch off due to minor fluctuations.

Second, *hard constraints* enforce external rules (e.g., a monthly budget limit). Such constraints override otherwise plausible actions, not because the embedding geometry implies them, but because the environment imposes them. This distinction is essential: logical embedding is not intended to represent all structure internally. Constraints define a boundary of admissible behavior that the semantic core must respect.

3.5 Policy Layer (Deductive Behavior as Geometric Selection)

The policy layer converts intermediate representations into action. It produces a policy vector $\mathbf{p}_t$ conditioned on the constrained state z_t. For instance,

$$\mathbf{p}_t = \begin{cases} \text{norm}(\mathbf{e}(\text{Risk}) + \delta \mathbf{e}(\text{Saving})) & z_t = 1 \\ \text{norm}(\mathbf{e}(\text{Impulse}) + \mathbf{e}(\text{Sale}) + \lambda \mathbf{e}(\text{GoalFun})) & z_t = 0, \end{cases}$$

and selects an action by similarity to action prototypes:

$$a_t = \arg\max_{a \in \mathcal{A}} s(\mathbf{a}, \mathbf{p}_t).$$

This yields behavior that appears deductive: given the same constrained state, the same policy vector leads to the same action, and small perturbations in input do not change the result.

Importantly, this is not rule execution. The policy layer does not apply a symbolic rule "IF risk THEN skip." Rather, it produces a vector that is stably closer to the action prototype corresponding to avoidance. The *appearance* of a rule is an emergent property of stable geometry and constrained state transitions.

3.6 Stability Layer (When to Learn and When to Freeze)

A key risk of any adaptive embedding system is semantic drift: updating embeddings can destroy previously stable relations. The stability layer controls this by computing a stability score S_t that governs whether learning updates are allowed.

Although the precise definition may vary, the stability score is designed to combine: (i) numerical stability, such as decision margin and policy change, and (ii) semantic stability, such as invariance of key alignments (e.g., "Risk" remains aligned with avoidance) and local topological consistency (e.g., KNN neighborhoods of anchor concepts).

Learning is then gated: high stability freezes updates (consolidation), while low stability permits adaptation (plasticity). This layer operationalizes the core thesis of this paper: deduction corresponds to stabilized regimes of the dynamics, not to the presence of explicit rules.

3.7 Coupling Among Layers

The layered design is effective because the layers are coupled in a constrained loop: induction produces $\mathbf{v}_t$, constraints map it to z_t, policy selects a_t from $\mathbf{p}_t$, memory stores and recalls episodic associations, and stability decides whether embeddings and coefficients may be updated. The system therefore alternates between two modes: *plastic* (learning permitted) and *consolidated* (learning frozen).

This coupling clarifies a conceptual point: what we call "deduction" in this framework is not a separate inference engine, but the observable consequence of consolidation in a semantic space shaped by inductive experience and constraints.

Finally, the architecture is deliberately modular. Extensions (symbolic hypothesis generators, meta-evaluators, semantic anchors) can attach to specific interfaces without requiring changes to the core operators. This modularity supports the broader claim that logical embedding is best viewed as a semantic core within a heterogeneous cognitive system.

4 Stability as the Emergence of Deduction

Within the layered architecture described above, deduction does not appear as a distinct computational module. Instead, it emerges when interactions among induction, memory, constraint, and policy layers enter a *consolidated* regime. In this regime, learning updates are gated, semantic relations stabilize, and behavior becomes reproducible across similar situations.

We therefore define deduction operationally as a property of system dynamics rather than as an inference procedure. Specifically, a system is said to exhibit deductive behavior when transitions among layers no longer induce significant change in policy or concept embeddings, and when the same interfaces consistently produce the same outputs.

This distinction can be expressed in terms of two modes. In the *plastic* mode, the induction layer actively reshapes the semantic space, memory recall competes with new experience, and constraints frequently trigger state transitions. During this phase, behavior may appear inconsistent or exploratory, and no stable rule-like interpretation is available.

In contrast, the *consolidated* mode is characterized by high stability scores that gate learning updates across layers. The situational composite vector $\mathbf{v}_t$ maps reliably to a constrained state z_t, the policy layer produces a consistent policy vector $\mathbf{p}_t$, and action selection becomes insensitive to small perturbations of the input. At this point, the system behaves *as if* it were applying an explicit rule, even though no symbolic rule representation exists internally.

Crucially, this stability is enforced not by a single mechanism, but by coordinated interfaces among layers. Memory regularizes behavior by biasing toward past stable actions, constraints suppress rapid oscillations, and the stability layer freezes updates once semantic and numerical criteria are met. Deduction, in this sense, is the macroscopic manifestation of successful coordination among these components.

This perspective dissolves the classical opposition between induction and deduction. Deduction is not a separate faculty that follows induction; it is the asymptotic outcome of constrained inductive dynamics. What appears to an observer as logical necessity corresponds internally to a stabilized configuration of semantic representations.

We summarize this position as *deduction as stability*: deductive behavior arises when a learning system has converged to a consolidated regime in which meaning, action, and constraint are jointly invariant under further experience.

5 Deduction as Stability: A Conceptual Reframing

In classical logic and symbolic AI, deduction is defined as the application of explicit rules to derive conclusions that are guaranteed by prior premises. Within such frameworks, deduction is an operation, and its validity is judged by formal correctness.

In contrast, the present work adopts a fundamentally different stance. We propose that deduction should be understood not as an operation, but as a *state* of a learning system. Specifically, we define deduction as the stabilization of meaning within an embedding space after prolonged inductive adaptation.

Under this view, a system is said to exhibit deductive behavior when the following conditions are met: (i) inductive updates no longer produce significant changes in policy or concept embeddings, (ii) decisions become reproducible under similar conditions, and (iii) semantic relations between key concepts (e.g., risk and avoidance) remain invariant over time. These conditions are captured quantitatively by the stability score introduced in the previous section.

Formally, let $\mathbf{p}_t$ denote the policy vector at time t. Deduction corresponds to the regime in which

$$\|\mathbf{p}_t - \mathbf{p}_{t-1}\| \approx 0$$

and semantic consistency measures remain above a fixed threshold. Importantly, no explicit rule representation is required: what appears as a rule to an external observer is internally realized as a stable geometric configuration.

This reframing has two major implications. First, it dissolves the sharp distinction between induction and deduction. Deduction is not a separate faculty, but the asymptotic outcome of constrained inductive dynamics. Second, it explains why human deductive behavior is often context-dependent, approximate, and resistant to formalization: it reflects stability under lived constraints, not adherence to abstract logical laws.

In this sense, logical embedding does not *implement* deduction. Rather, deduction *emerges* when learning dynamics converge to a stable attractor in semantic space. We therefore summarize our position as *deduction as stability*.

6 Limitations of Logical Embedding

Despite its expressive power and computational efficiency, logical embedding is not a complete theory of learning or reasoning. Its design choices impose fundamental limitations that must be explicitly acknowledged in order to avoid category errors. In this section, we analyze these limitations in detail, with particular emphasis on learning dynamics and epistemic guarantees.

6.1 Absence of Genuine Novelty Generation

Logical embedding operates by reconfiguring vectors within a pre-existing semantic space. All learning updates correspond to continuous transformations of existing embeddings. As a result, the framework cannot generate genuinely new primitive concepts, symbols, or logical operators. What appears as conceptual innovation is, in fact, a novel arrangement or stabilization of prior representations.

This limitation is intrinsic rather than accidental. Embedding-based systems presuppose a fixed representational basis, and therefore cannot introduce new ontological commitments without external intervention. In contrast to symbolic approaches such as inductive logic programming, logical embedding does not search over hypothesis spaces of rules or programs. Instead, it interpolates within a continuous manifold of meaning.

Consequently, logical embedding is best understood as a model of *conceptual consolidation* rather than concept creation. Its strength lies in explaining how experience reshapes and stabilizes existing meanings, not how entirely new categories are discovered.

6.2 Lack of Formal Guarantees

Another fundamental limitation concerns the absence of formal guarantees. Because learning proceeds via approximate gradient-like updates and heuristic stability gating, there are no proofs of convergence, optimality, or consistency. A system may reach a stable configuration that is suboptimal, internally inconsistent, or contextually biased.

This stands in sharp contrast to classical logical systems, where deductive validity is guaranteed by construction, and to Bayesian models, where optimality is defined relative to explicit probabilistic assumptions. Logical embedding sacrifices these guarantees in favor of computational tractability and cognitive plausibility.

Importantly, this is not merely a technical shortcoming. It reflects a philosophical stance: human reasoning itself appears to operate without global guarantees of correctness, yet achieves sufficient reliability through stabilization under constraints. Logical embedding mirrors this property, but inherits its epistemic fragility.

6.3 Semantic Drift and Meaning Instability

Because concept embeddings are themselves subject to learning, logical embedding faces a persistent risk of semantic drift. Updating an embedding alters the geometry of the entire space, potentially changing the meaning of previously stable relations.

This creates a fundamental tension: strong adaptation improves responsiveness to new experience, while excessive updating erodes semantic continuity. Without explicit anchoring mechanisms, concepts may gradually lose their interpretability or invert their relational roles.

Although the framework introduces stability gating and semantic anchors to mitigate this effect, these mechanisms rely on design choices rather than guarantees. The boundary between adaptive refinement and destructive drift cannot be sharply defined within the embedding space itself.

6.4 Dependence on Weak or Implicit Supervision

Logical embedding typically relies on weak external signals, such as satisfaction, surprise, or constraint violations, rather than explicit labels or rewards. While this aligns with models of everyday human learning, it limits the precision and speed of adaptation.

In the absence of strong supervision, learning may stagnate, becoming dominated by memory recall rather than genuine generalization. Conversely, introducing strong rewards risks collapsing the approximate logical structure into task-specific heuristics, thereby undermining interpretability.

Thus, logical embedding occupies an intermediate regime between unsupervised association and goal-driven optimization, without fully inheriting the advantages of either.

6.5 Limited Explainability and Justification

Although logical embedding allows partial introspection through similarity scores, margins, and stability measures, it cannot provide symbolic explanations or proofs. A decision can be described geometrically (e.g., proximity to a risk vector), but not justified in terms of explicit premises and conclusions.

This limitation is particularly salient in scientific or legal contexts, where explanations must be articulated in propositional or rule-based form. Embedding-based rationales may be intuitively persuasive, but they lack the normative force of formal deduction.

As a result, logical embedding should be viewed as complementary to, rather than a replacement for, symbolic explanatory frameworks.

6.6 Summary of Limitations

Taken together, these limitations suggest that logical embedding is not a self-sufficient model of intelligence. It cannot generate new symbolic structures, does not guarantee correctness, and remains vulnerable to semantic instability. Its learning dynamics are heuristic and context-dependent, and its explanations are inherently geometric rather than logical.

However, these limitations are inseparable from its strengths. The same properties that preclude formal guarantees enable flexible, low-cost, and cognitively plausible reasoning. Recognizing these trade-offs is essential for deploying logical embedding appropriately within a broader cognitive architecture.

7 Mitigating Limitations via Modular Extensions

The limitations discussed in the previous section do not imply that logical embedding is an inadequate framework. Rather, they indicate that logical embedding should not be treated as a closed, self-sufficient model of intelligence. In this section, we argue that many of its limitations can be mitigated by modular extensions that preserve the core embedding-based dynamics while introducing complementary capabilities.

Crucially, most of these extensions can be attached without altering the internal mechanics of logical embedding. The framework is therefore best understood as a semantic core that benefits from carefully designed interfaces to external modules.

7.1 External Symbolic Hypothesis Generators

The inability of logical embedding to generate genuinely new primitives can be addressed by introducing external symbolic generators. Such generators may include inductive logic programming systems, program synthesis modules, or human-provided hypotheses.

These components operate outside the embedding space, producing discrete symbolic candidates (e.g., rules, predicates, or causal hypotheses). Once generated, these candidates can be embedded into the semantic space as new vectors, where their meanings are gradually stabilized through interaction with existing concepts.

Importantly, this interface is intentionally non-seamless. The introduction of new symbols represents an ontological commitment that cannot be justified internally by the embedding dynamics. Logical embedding thus plays a selective and consolidating role: it does not create hypotheses, but evaluates whether externally proposed structures can be coherently integrated into existing semantic geometry.

7.2 Meta-Evaluators and Consistency Critics

The absence of formal guarantees can be mitigated by augmenting the stability score with meta-evaluative components. Such evaluators do not enforce correctness in a logical sense, but penalize instability, inconsistency, or excessive complexity.

Examples include:

- penalties for contradictory action tendencies under similar embeddings,
- minimum description length criteria applied to stabilized policy configurations,
- violation counts of soft constraints across time.

These signals can be seamlessly incorporated as additional terms in the stability function. Rather than dictating learning outcomes, they modulate learning rates and update gates, thereby shaping the trajectory of inductive convergence without imposing symbolic rules.

7.3 Semantic Anchors and Invariance Constraints

Semantic drift poses a serious challenge whenever embeddings are updated over time. To address this, logical embedding can be extended with semantic anchors: embeddings that are partially or fully frozen.

Anchors may correspond to:

- human-defined core concepts,
- perceptually grounded features,
- externally validated reference categories.

During learning, updates are constrained to preserve distances or relative ordering with respect to these anchors. This preserves large-scale semantic topology while allowing local adaptation.

Notably, this mechanism aligns naturally with the geometric nature of logical embedding. Anchors function as fixed points in semantic space, around which inductive dynamics can stabilize without collapsing into arbitrary drift.

7.4 Weak External Signals and Surprise-Based Triggers

Logical embedding relies on weak supervision and may therefore stagnate in stable but suboptimal configurations. To counter this, external signals such as surprise, prediction error, or dissatisfaction can be introduced as learning triggers.

These signals do not specify desired outcomes. Instead, they indicate when existing embeddings fail to adequately predict or explain experience. In response, stability thresholds can be temporarily relaxed, allowing renewed adaptation.

Because these triggers operate only on learning dynamics rather than representation content, they can be integrated seamlessly into the existing stability-gated update mechanism. This preserves the approximate logical structure while restoring exploratory flexibility.

7.5 Post-hoc Symbolization and Explanation Layers

While logical embedding does not natively support symbolic explanation, its internal states can be retrospectively translated into human-interpretable descriptions. Such post-hoc symbolization layers extract salient relations from embedding geometry, such as dominant similarity alignments or stable decision boundaries.

These extracted structures can be rendered as natural language explanations or symbolic rules. Although they lack the normative force of formal proofs, they provide communicative transparency and facilitate human oversight.

Importantly, this translation remains external to the core model. Logical embedding continues to operate in its native geometric domain, while explanation is treated as a separate representational problem.

7.6 Architectural Implications

Taken together, these extensions suggest that logical embedding is most effective when embedded within a heterogeneous cognitive architecture. In such an architecture:

- symbolic modules generate hypotheses,
- logical embedding consolidates meaning and behavior,
- evaluators and anchors ensure stability,
- explanatory layers translate outcomes for human use.

This division of labor preserves the strengths of logical embedding while compensating for its inherent limitations. Rather than aspiring to universal reasoning capability, the framework emphasizes robustness, scalability, and cognitive plausibility.

7.7 Summary of Mitigation Strategy

The limitations of logical embedding are not flaws to be eliminated, but boundaries that define its appropriate role. By augmenting the framework with modular extensions, it is possible to recover

many desirable properties of symbolic and probabilistic systems without sacrificing the efficiency and flexibility that motivate embedding-based reasoning.

Logical embedding thus serves as a stable semantic substrate, bridging inductive experience and deductive behavior, while remaining open to external sources of structure and validation.

8 Conclusion

This paper has argued for a redefinition of deduction. Rather than treating deduction as a rule-based inference operation, we have proposed to understand it as a stabilized regime of cognitive dynamics. Under this view, deductive behavior emerges when inductive learning, memory, and constraint converge to a consolidated state in which judgments become reproducible and invariant under further experience. We have summarized this position as *deduction as stability*.

To make this perspective concrete, we introduced logical embedding as a representational and computational substrate in which such stabilization can occur. Logical embedding represents concepts, situations, and actions as vectors in a shared semantic space, and defines approximate logical relations through simple geometric operations. Crucially, logical embedding does not implement deduction directly. Instead, it provides a space in which inductive adaptation can gradually settle into configurations that behave as if governed by rules.

The layered architecture presented in this work clarifies how this process unfolds. The induction layer reshapes semantic representations from experience, the memory layer regularizes behavior through episodic recall, the constraint layer imposes history-dependent and external limits, the policy layer selects actions based on stabilized geometry, and the stability layer regulates when learning should proceed and when it should be frozen. Deductive behavior appears not at any single layer, but as the macroscopic outcome of coordinated interactions across these layers.

This perspective has several important implications. First, it dissolves the traditional opposition between induction and deduction. Deduction is not a separate faculty that follows induction, but the asymptotic outcome of constrained inductive dynamics. Second, it explains why everyday human reasoning often appears rule-like without conforming to the standards of formal logic: human deduction reflects stability under lived constraints, not adherence to explicit symbolic rules. Third, it motivates a shift in evaluation criteria from logical correctness to stability, robustness, and semantic invariance.

At the same time, we have emphasized that logical embedding is not a complete theory of reasoning. Its learning dynamics lack formal guarantees, its representational space cannot generate genuinely new primitives, and its explanations remain geometric rather than symbolic. These limitations are not incidental shortcomings, but consequences of the framework's design commitments. Accordingly, we have argued that logical embedding is best understood as a semantic core within a heterogeneous cognitive architecture, to be complemented by external symbolic generators, meta-evaluators, semantic anchors, and post-hoc explanation layers.

In summary, this work contributes not a new inference algorithm, but a conceptual and computational reframing of what deduction can mean in resource-bounded, experience-driven systems. By treating deduction as stability and logical embedding as its enabling substrate, we provide a principled account of how rule-like behavior can emerge without explicit rules. We hope that this perspective will inform future work at the intersection of cognitive science, artificial intelligence, and the study of human reasoning.

References

[1] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986.

[2] P. Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1–2):159–216, 1990.

[3] A. Clark. *Surfing Uncertainty: Prediction, Action, and the Embodied Mind*. Oxford University Press, 2016.

[4] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017.

[5] M. Mitchell. *Why AI Is Harder Than We Think*. Nature Machine Intelligence, 5:160–164, 2023.

[6] J. A. Fodor and Z. W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1–2):3–71, 1988.

[7] L. W. Barsalou. Grounded cognition. *Annual Review of Psychology*, 59:617–645, 2008.

[8] S. J. Gershman, E. J. Horvitz, and J. B. Tenenbaum. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245):273–278, 2015.

[9] G. E. Hinton. Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 1986.

[10] G. Pezzulo, F. Rigoli, and K. Friston. Active inference, homeostatic regulation and adaptive behavioural control. *Progress in Neurobiology*, 112:1–20, 2014.