Why General Artificial Intelligence Is Principally Impossible

Y. Matsuda and ChatGPT5.2

January 24, 2026

Abstract

This paper argues that General Artificial Intelligence (GAI), understood as a system capable of universally reliable generalization across arbitrary domains, tasks, and normative regimes, is principally unattainable under the prevailing paradigm of predictive learning. We first explain in detail how contemporary generative models, particularly Transformer-based architectures, acquire what is operationally referred to as "meaning"—including inter-concept relations, structural dependencies such as procedural order and constraints, and functional roles with respect to goals—as geometric structure in high-dimensional vector spaces through next-token prediction and self-attention mechanisms. We then show that, because learning in these models is fundamentally predictive and optimized with respect to finite data distributions, generalization is necessarily distribution-dependent. As a consequence, both under-generalization and over-generalization arise not as incidental failures but as structurally inevitable outcomes of the learning objective itself. From this analysis, we conclude that scaling predictive generative models alone cannot yield a form of intelligence with universal, distribution-independent generalization guarantees, and that the aspiration toward GAI must therefore confront principled limitations rather than purely engineering challenges.

1 Problem Setting: "Meaning" as Geometry in Generative Models

We use "meaning" in an operational sense: not as dictionary definitions, but as (i) relations among concepts, (ii) structural dependencies that must be preserved across long contexts (e.g., causality-like explanations, multi-step procedures, and hard constraints), and (iii) functional roles relative to a goal (e.g., "this block initializes," "this argument constrains," "this step validates"). In contemporary generative models, these regularities are not stored as explicit symbolic tables; instead, they emerge as *geometric structure* in learned representations (vectors) and their transformations.

We focus on Transformer-based models trained primarily by next-token prediction. The core question is: *what computation and what training algorithm cause these forms of meaning to be acquired as geometry?*

2 Learning Objective: Prediction as the Computational Core

2.1 Autoregressive next-token prediction

Let a token sequence be $(x_1, \ldots, x_T)$ where tokens can represent natural-language subwords or program symbols. Autoregressive training minimizes the negative log-likelihood:

$$\mathcal{L}(\theta) \;=\; -\sum_{t=1}^{T-1} \log p_\theta(x_{t+1} \mid x_{1:t}), \tag{1}$$

typically implemented via cross-entropy over a large vocabulary.

2.2 Why prediction pressures "semantic" structure

Although the loss is purely predictive, correct prediction under real data requires the model to preserve information about:

- **Inter-concept relations:** which concepts co-occur, substitute, entail, or contrast within contexts.
- **Structural dependencies:** which earlier parts control later parts (definitions → uses, premises → conclusions, opening → closing brackets, scoping, typing, protocol order, etc.).
- **Functional roles:** which spans serve as introductions, constraints, validations, summaries, exceptions, steps in procedures, and so on.

Because the training signal rewards whatever internal computation improves predictive likelihood, the model is incentivized to encode these regularities in internal states. The resulting encodings are vectors; hence the learned "meaning" appears as geometry.

3 Computation Model: How Transformers Create Geometric Structure

3.1 Token embeddings and contextual states

Each input token x_t is mapped to an embedding $e_t \in \mathbb{R}^d$. A Transformer layer updates contextual states $h_t^{(\ell)}$ by combining (a) attention-based aggregation of other positions and (b) position-wise nonlinear transformations. Thus, meaning is not attached to tokens alone, but to *contextualized representations* that evolve across layers.

3.2 Self-attention as geometric relation extraction

Given a layer input h_t, attention computes query, key, value vectors:

$$q_t = W_Q h_t, \quad k_t = W_K h_t, \quad v_t = W_V h_t, \tag{2}$$

and uses scaled dot-product similarity to determine attention weights:

$$\alpha_{tj} = \mathrm{softmax}_j\left(\frac{q_t^\top k_j}{\sqrt{d_k}}\right), \qquad \tilde{h}_t = \sum_{j \le t} \alpha_{tj} v_j \tag{3}$$

(for causal masking in autoregressive models). Crucially, *the notion of "relatedness" is computed by inner products in vector space.* Thus, the model represents and exploits relations through geometry: vectors that point in compatible directions yield large dot products, hence stronger interactions.

3.3 Multi-head attention: decomposing relation types

Multi-head attention uses multiple sets of (W_Q, W_K, W_V), producing multiple attention patterns in parallel. This enables the model to represent different dependency types simultaneously, such as:

- reference/aliasing (entity mention → later pronoun; variable definition → use),
- syntactic or structural control (subject → predicate; conditional guard → controlled block),
- constraint maintenance (matching parentheses, indentation blocks, type compatibility),

- procedural staging (setup → execution → cleanup).

Each head can specialize in a different geometric subspace, where a particular kind of relation becomes linearly accessible.

3.4 Feed-forward blocks: local feature synthesis

After attention aggregation, position-wise feed-forward networks apply nonlinear transforms to each position. This supports the synthesis of higher-level features (e.g., "this span is a definition," "this step is a validation," "this clause is causal"), which are again encoded in vectors. Across layers, these features become more abstract and more distributed.

4 How the Three "Meaning" Notions Emerge Along the Learning Flow

4.1 Inter-concept relations: distributional geometry

A classic intuition is distributional: expressions that occur in similar contexts become geometrically similar. Here, "similar" means their contextual vectors become close (or aligned) under the model's learned metric induced by dot products and linear projections. Prediction pushes the model to compress recurring regularities: if different surface forms can play the same role, it is efficient for their representations to converge.

4.2 Structural dependencies: causality-like links, procedures, and constraints

Structural dependencies are strongly pressured because violating them quickly harms predictive likelihood. Examples:

- **Procedures:** a later step depends on artifacts introduced earlier; correct next-token generation requires tracking "what has been established so far".
- **Constraints:** bracket matching, scope, type consistency, protocol order (open/write/close), etc.; failures produce low-probability continuations.
- **Causality-like coherence:** reason → conclusion patterns; while not causal inference in the scientific sense, the model learns the textual regularities that make explanations coherent within the training distribution.

In a Transformer, these dependencies are realized by attention pathways (which earlier states to consult) and by layer-wise state features (what constraints are currently active).

4.3 Functional roles relative to goals

Functional role is not merely lexical meaning; it is about the span's *purpose* within a larger plan. Even in pure next-token pretraining, role-like features emerge because role determines what continuation is likely. In many deployed systems, role sensitivity is further strengthened by instruction tuning and preference optimization: the model is trained to produce outputs that better satisfy user goals and normative criteria. In representation terms, this tends to make "goal-conditioned" distinctions more explicit in higher-layer geometry.

5 Why Surface Form and Even Algorithmic Structure Vary Under Regeneration

A key practical observation is that, upon regeneration, not only vocabulary but also "structure" (e.g., algorithmic presentation, decomposition style, ordering) can change while preserving high-level intent. From the representation viewpoint:

- The model does not store a literal template to replay; it reconstructs outputs token-by-token to maximize conditional probability.
- The mapping from "intended meaning" to surface form is *many-to-many*: multiple valid realizations exist for the same function.
- Because generation is a sampling process (even when approximately deterministic), small context differences can move the internal state within a region of representation space that supports multiple realizations.

Thus, "meaning" is retained as geometry, whereas surface form is re-synthesized—and may drift in vocabulary, phrasing, and even structural decomposition.

6 From Prediction to Generalization: Why Under- and Over-Generalization Are Inevitable

The user-level conclusion can be stated succinctly: *if learning is prediction, then generalization has a scope.* Two complementary failure modes naturally arise.

6.1 Under-generalization

Under-generalization occurs when the model fails to abstract sufficiently and remains bound to shallow patterns. This can happen due to limited capacity, insufficient coverage, or optimization/regularization effects. In geometric terms, the representation space fails to form stable clusters or separable directions for the relevant abstractions.

6.2 Over-generalization

Over-generalization occurs when the model applies learned patterns outside their valid regime, producing plausible but incorrect continuations (including "hallucinations" in broader usage). In geometric terms, a region of the representation space that is predictive within the training distribution may be entered in an out-of-distribution context, where the same geometric cues no longer correspond to truth or validity. This is not merely a bug; it is a consequence of optimizing likelihood rather than validity under arbitrary shifts.

7 Principle-Level Difficulty of "GAI": Why Universal Generalization Is Not Guaranteed

We now connect the above to the claim: *a universally general intelligence with universally correct generalization cannot be obtained by predictive learning alone.* This should be interpreted carefully as a principle-level limitation of the objective and setting, not as a statement about all imaginable future paradigms.

7.1 Distribution dependence of predictive objectives

The objective in Eq. (1) is evaluated on samples from a training distribution. Any statement of performance is therefore conditional on the relationship between training and deployment distributions. If deployment contexts shift substantially (new tasks, new norms, new physical constraints, new value functions), the predictive optimum does not guarantee correctness under the new distribution. "Generalization" is thus inherently relative, not absolute.

7.2 No universally best learner across all problem families

A well-known principle in learning theory (often summarized under the *No Free Lunch* intuition) is that, without assumptions about the problem distribution, there is no single learner that is uniformly best for all possible data-generating processes and tasks. In the present context, "universal generalization" would require a single system to perform optimally across an unbounded variety of task distributions, goals, and evaluation standards. Predictive training provides strong performance where the underlying regularities align with the data and objective; it does not yield a distribution-independent guarantee.

7.3 Externality of meaning, goals, and norms

Even when models encode rich geometric semantics, the notion of "correctness" depends on:

- the task definition (what is being optimized),
- the normative constraints (safety, responsibility, acceptability),
- the evaluation procedure (what counts as success).

These are not determined solely by the internal geometry. They are injected by data selection, fine-tuning signals, preference criteria, and deployment constraints. Therefore, a model trained for predictive success in one regime cannot, in principle, guarantee universal correctness across regimes with different goals and norms.

7.4 A careful formulation of the "GAI difficulty" claim

A strong but defensible statement is:

> Given that current generative models learn primarily by predictive objectives, their generalization is necessarily distribution-dependent. Therefore, *universal* general intelligence understood as universally reliable generalization across arbitrary tasks, distributions, and normative constraints is not guaranteed—and is unlikely to arise as a direct extension of predictive learning alone.

This avoids claiming metaphysical impossibility for all conceivable systems, while clearly stating the principle-level limitation of the prevailing paradigm.

8 Conclusion

Transformer-based generative models acquire operational "meaning" as geometry because their core computation (self-attention with dot-product similarity) and training objective (next-token prediction) jointly pressure the model to encode inter-concept relations, structural dependencies, and functional roles in vector representations. However, since learning is fundamentally predictive and optimized on finite distributions, generalization has an intrinsic scope. Under-generalization and over-generalization are thus not accidental; they are structurally entailed failure modes. Consequently, "GAI" in the sense of universally reliable generalization across

arbitrary regimes is, at minimum, principle-level difficult and cannot be treated as an automatic outcome of scaling predictive learning.

A Appendix: On AGI, Agent Architectures, and the Invariance of Generalization Limits

This appendix clarifies several potential sources of confusion surrounding the claim that General Artificial Intelligence (GAI) is principally unattainable under predictive learning paradigms. Rather than restating the conclusion alone, we explicitly trace the reasoning process that leads to this claim, addressing common variations in terminology and system architecture that are sometimes assumed to weaken or bypass the limitation.

A.1 AGI versus GAI: Does Renaming Alter the Principle?

A frequent objection to claims of "GAI impossibility" is that the term itself is ill-defined, and that alternative labels such as Artificial General Intelligence (AGI) might refer to a different, potentially achievable target. However, this distinction is largely rhetorical rather than principled.

The argument presented in the main text does not hinge on the specific name assigned to general intelligence, but on the learning principles assumed to underlie it. In particular, the limitation arises from three assumptions that are typically shared by both AGI and GAI discussions: (1) learning is primarily predictive, i.e., optimized to minimize expected error under a data distribution; (2) generalization is statistical and therefore distribution-dependent; and (3) meaning, goals, and norms are externally specified rather than internally grounded.

Renaming the target system does not alter these assumptions. An "AGI" that learns via the same predictive objectives and representation mechanisms as contemporary generative models remains subject to the same structural constraints. Thus, changing terminology does not change what is principally unattainable: namely, universally reliable generalization across arbitrary domains, tasks, and normative regimes.

A.2 Domain-Restricted Agents: Practical Success without Principle Change

Another common response is to point out that agent-based systems often operate in restricted domains, where performance can be made robust and reliable. Indeed, constraining the task domain significantly alters the empirical behavior of learning systems.

When the environment, objectives, and evaluation criteria are narrowly specified, the effective data distribution becomes more stable. This makes predictive generalization appear stronger and failures less frequent. In such cases, systems may exhibit behavior that is, for all practical purposes, "general enough."

However, this improvement does not contradict the principal limitation discussed in the main text. The claim is not that predictive models fail everywhere, but that they cannot be guaranteed to succeed everywhere. Domain restriction does not eliminate the limitation; it sidesteps it by deliberately shrinking the scope of applicability. In this sense, domain-specific agents do not overcome the impossibility of universal generalization, but instead redefine success to lie within a bounded regime where predictive assumptions hold.

A.3 Multi-Agent Architectures: Scaling Quantity versus Changing Kind

A more subtle challenge arises from multi-agent systems, where large numbers of agents interact, collaborate, or cross-check one another. At first glance, such architectures appear qualitatively different from single-model systems and may seem capable of transcending individual generalization limits.

Multi-agent systems can indeed improve robustness through redundancy, diversity of exploration, and partial mutual validation. They can reduce the impact of localized errors and make failures harder to trigger through any single faulty inference. From an engineering perspective, this is a powerful strategy.

Nevertheless, from a principled standpoint, each agent typically remains a predictive generalization system. Their interactions are governed by learned policies that themselves generalize from training distributions. As a result, the overall system inherits the same distribution dependence at a higher level. While error structures may be averaged, delayed, or obscured, they are not eliminated. A collection of imperfect generalizers does not constitute a universally correct generalizer.

Thus, multi-agent scaling alters the statistical profile of failure but does not remove the foundational dependence on predictive assumptions.

A.4 Could a Different Kind of Generalization Model Change the Conclusion?

The most substantive potential challenge to the main argument lies in questioning the generalization model itself. The impossibility claim applies to a specific class of models: those that generalize by statistical prediction over data distributions. If generalization were redefined or replaced by a fundamentally different mechanism, might the limitation dissolve?

Conceptually, alternative approaches can be imagined, such as systems centered on constraint satisfaction, formal verification, local reconstruction of viable actions, or operational semantics rather than global prediction. These approaches may reduce reliance on broad statistical extrapolation and improve safety or interpretability.

However, such shifts do not yield universal generalization in the original sense. Instead, they redistribute responsibility: from global predictive competence to local validity checks, explicit constraints, and context-sensitive reconstruction. In doing so, they intentionally narrow the role of generalization rather than expanding it. The system becomes less "general" in the universal sense, but more controllable and verifiable within specified bounds.

Therefore, even when the generalization model itself is altered, the aspiration to a single system that reliably generalizes across all domains and value systems remains unfulfilled.

A.5 Summary of the Argumentative Flow and Residual Design Principles

The discussion in this appendix establishes that the principal difficulty of GAI does not disappear through terminological changes, architectural scaling, or agent-based decomposition, as long as the underlying learning mechanism relies on predictive generalization over data distributions. However, this conclusion does not render system design directionless. On the contrary, it clarifies which design principles remain meaningful once universal generalization is abandoned as a goal.

The first residual principle is *explicit scope restriction*. Since generalization is inevitably distribution-dependent, intelligent systems should be designed with clearly articulated domains of validity. Rather than implicitly assuming transferability, systems should expose the assumptions under which their predictions or actions are expected to remain reliable. This shifts emphasis from hypothetical universality to accountable applicability.

The second principle is *local verifiability over global correctness*. If global correctness across arbitrary contexts cannot be guaranteed, then correctness must be enforced locally. This includes interface-level validation, constraint checking, invariant preservation, and post-hoc verification of intermediate results. In this view, intelligence is not identified with always being right, but with being able to detect, localize, and bound failure.

The third principle is *separation of generalization and commitment*. Predictive models may be used to propose candidates—plans, interpretations, actions, or code fragments—but commitment to execution or acceptance should be mediated by additional mechanisms. These

may include rule-based checks, formal methods, human oversight, or context-specific validators. Such separation limits the damage caused by over-generalization while retaining the flexibility of learned models.

The fourth principle is *viable-action-centered design*. Rather than aiming to compute globally optimal or universally correct outputs, systems should focus on determining whether a proposed action is viable under current constraints. Viability here encompasses physical feasibility, procedural consistency, and normative acceptability. This reframes intelligence as a capacity for constrained reconstruction rather than unconstrained inference.

Finally, these principles suggest a shift in how "intelligence" itself is operationally defined. Under predictive paradigms, intelligence should not be equated with unbounded generalization, but with the ability to operate robustly under acknowledged limitations, to manage uncertainty explicitly, and to integrate learning with verification. In this sense, the impossibility of GAI does not mark an endpoint, but delineates the design space within which practically useful and responsible intelligent systems can be constructed.