Next-Token Prediction Cannot Reliably Build Procedural Knowledge

Sohan Venkatesh
Manipal Institute of Technology Bengaluru
soh.venkatesh@gmail.com

Abstract

Next-token prediction over natural language text selectively builds declarative knowledge while leaving procedural knowledge underdeveloped. Text describes processes. It does not enact them. The result is a systematic imbalance we call representational asymmetry: models explain processes they cannot reliably simulate. Its behavioral signature is a hidden absence: fluent, confident output on tasks requiring structural simulation, with failure invisible from the output form alone. We ground this claim in a precise criterion: a task is structural if correct performance requires tracking state across steps, where the answer depends on the full trajectory rather than the terminal state. This criterion unifies failures the field has treated as independent—hallucination, planning collapse, calibration failure and execution errors—as instances of a single phenomenon rooted in the training objective. Convergent evidence across algorithmic reasoning, clinical medicine and uncertainty calibration already instantiates the pattern. We specify four falsifiable predictions and argue the asymmetry will not close under scaling. Understanding it is a prerequisite for building systems that do not just describe competence but actually possess it.

1 Introduction

Ask any frontier language model to explain Dijkstra's algorithm. It produces a textbook-quality response. Ask it to predict whether a Fibonacci heap runs faster than a binary heap on a specific input and performance collapses. The model sounds right. It uses the right vocabulary. The answer is wrong.

This is not an edge case. Identical patterns appear in planning [67], calibration [33], hallucinated citations [31], mathematical reasoning [37] and graph algorithm execution [68]. The field treats these as independent engineering problems with independent fixes. We argue they are instances of a single structural phenomenon.

The dominant assumption is that capability limitations are data-and-compute problems. Scale enough and models will acquire whatever they currently lack [29, 35]. We argue this assumption is wrong for a specific class of failures. It is wrong because it treats capability as a single dimension. Declarative competence and procedural competence are distinct. Next-token prediction optimizes strongly for one and weakly for the other.

> **This paper argues that next-token prediction over natural language text structurally favors declarative knowledge over procedural knowledge and that this asymmetry will not close under scaling within the same training paradigm.**

The primary contribution is a four-axis taxonomy that decomposes this imbalance into independently testable failure modes: Descriptive-Generative, Referential-Grounding, State-Transition and

Position Paper.

Uncertainty-Calibration. Each axis corresponds to a distinct kind of structural task. Each makes predictions about where models will fail and why. Together, they unify what the literature has treated as separate phenomena under a single causal explanation rooted in the training objective.

Getting this wrong has real costs. Structural failures are invisible at the surface. Users overrely on fluent output. Benchmarks miss the gap entirely. The rest of this paper argues why the asymmetry is real, why it persists under scale and intervention and what would count as evidence that it does not.

2 Background and Motivation

2.1 The Failure Literature

LLM failures are well-documented. Models struggle with arithmetic [52, 14], multi-step reasoning [54, 64], factual grounding [31, 45], planning under constraints [67, 40] and uncertainty calibration [33, 72]. These failures are real and costly. They are almost universally treated as separate problems requiring separate fixes.

A smaller body of work identifies structural commonalities. Valmeekam et al. [67] showed that models explain planning principles fluently while performing near-randomly on constraint-bounded planning tasks. Srivastava et al. [61] documented consistent asymmetries between descriptive and procedural performance across BIG-Bench. Mallen et al. [41] showed high confidence in incorrect factual claims. Dziri et al. [17] found compositional failures scale monotonically with structural depth. None of these papers propose a unifying explanation. This paper does.

2.2 The Scaling Puzzle

The strongest empirical motivation for this framework is how failures behave under model scaling. Naive intuition says larger models should perform better across all tasks. What we observe is more nuanced: descriptive competence scales reliably but structural failures often persist or sharpen [22, 62].

Wei et al. [70] documented emergent capabilities at scale alongside the persistence of structured failures. Ganguli et al. [22] found unpredictable combinations of improvement and plateau across capability types. Schaeffer et al. [58] argued that some emergent improvements are measurement artifacts. What remains consistent: descriptive tasks benefit more from scale than structural ones do. No existing framework accounts for this asymmetric scaling as a unified phenomenon. representational asymmetry does, because it identifies the cause rather than the symptom.

2.3 Related Constructs

Several related ideas exist in the literature but address different phenomena. Hallucination research [31, 45, 53] focuses on factual accuracy, not the declarative-procedural gap. World model research [26, 27] asks whether models build internal representations of the world but does not frame the failure as a consequence of the training objective. Mechanistic interpretability [19, 18, 48] probes internal representations but has not systematically studied the declarative-procedural division. representational asymmetry is not a synonym for any of these. It is a framework that predicts when and why they co-occur. Concretely: a model that hallucinates citations (Referential-Grounding failure) should also fail on matched planning tasks (State-Transition failure) at a rate predictable from the asymmetry score, because both failures share the same root cause. Hallucination research and planning research have no mechanism for predicting that correlation. representational asymmetry does.

3 The Framework

3.1 Operationalizing Structural Competence

Before defining representational asymmetry, we need a precise boundary between descriptive and structural tasks. Without this, the framework cannot be falsified in practice.

We propose the following criterion: a task is *structural* if correct performance requires tracking a state that changes across steps, where the final answer depends on the full state trajectory, not just the terminal state. A task is *descriptive* if a model that had memorized all correct final answers would pass it.

This criterion is operationalizable. Consider two tasks: explaining the bubble sort algorithm and predicting the intermediate array state after three passes. The first is descriptive: the correct answer is a fixed text that can be memorized. The second is structural: the correct answer requires propagating state across steps and any shortcut that skips the trajectory will fail on new inputs. Code generation is largely descriptive: it produces syntactically valid text. Code execution prediction is structural: it requires tracking variable bindings across lines. Retrieval-augmented factual QA is descriptive: retrieve and report a fixed answer. Constraint-bounded planning is structural: propagate constraints across a decision sequence.

The criterion draws the line at state-trajectory dependence. It admits of degree: tasks can be more or less structural depending on how many steps require trajectory tracking versus terminal-state lookup. We predict asymmetry scores will scale with this proportion, providing a finer-grained test of the framework than a binary classification.

3.2 Representational Asymmetry

We define representational asymmetry as the systematic imbalance between two distinct competences encoded in a model's representations:

- **Descriptive Abstraction:** the capacity to produce fluent, accurate-sounding explanations of domain concepts, processes and principles. Knowing *about*.
- **Generative Structural Competence:** the capacity to simulate, predict or track the dynamic behavior of those same processes under perturbation, constraint or compositional complexity. Knowing *how*.

The core claim: next-token prediction training disproportionately rewards Descriptive Abstraction. This is not a claim that structural competence is entirely absent. It is a claim about relative robustness and the predictable location of failure.

3.3 Hidden Absence

A hidden absence occurs when a model has acquired declarative knowledge of a domain without the corresponding procedural knowledge to simulate or execute its processes. Three features co-occur: fluent explanation of domain principles; failure on tasks that are structural by the criterion in Section 3.1; and surface coherence preserved throughout. The failure is not detectable from output form alone.

A Hidden Absence is not incompetence. It is a training-shaped asymmetry between knowing about and being able to simulate. The model has learned the vocabulary of the map while developing only partial internal geography.

The word "hidden" is precise. A Hidden Absence cannot be identified by asking the model to explain its reasoning. The explanation will be fluent and plausible. It can only be identified by applying the state-trajectory test: does the model track the process it describes or does it pattern-match to the output form? This is what most current evaluations fail to test [55, 23].

3.4 The Four-Axis Taxonomy

The primary contribution of this paper is a four-axis decomposition of representational asymmetry. Prior work has documented each of these failure modes independently. No prior framework explains them as instances of a single phenomenon, predicts their co-occurrence or derives them from a common cause. The four axes do all three.

The **Descriptive-Generative** axis captures the most common failure pattern: a model explains a process correctly but cannot predict quantitative consequences under perturbation. The task shifts from describing the algorithm to tracking its execution on specific inputs. This is structural by the

criterion: the answer depends on iterative application of state-transition rules, not on what algorithm descriptions typically look like. Seen in CLRS benchmarks [68, 4] and symbolic math under variable renaming [54].

The **Referential-Grounding** axis may appear less obviously structural but grounding is state-trajectory dependent in a precise sense: verifying entity identity requires tracking referential chains across a document or knowledge base, not matching surface patterns of what citations look like. A model that has memorized all correctly formatted citations would still fail to verify their existence. The format is correct; the referential link is absent [31, 43, 53].

The **State-Transition** axis is the most directly structural failure mode by the criterion. Trajectory-dependence is the definition of planning: every decision changes the constraint budget and correct answers require propagating those changes across steps. Models describe planning steps coherently; execution drifts when tested [67, 34, 40].

The **Uncertainty-Calibration** axis is structural because calibration requires tracking the correspondence between confidence states and outcome frequencies across a sequence of predictions. A model that memorized calibration definitions would still miscalibrate under novel structural queries. Models correctly define probability theory but produce miscalibrated intervals on structural tasks [33, 36, 72].

Each axis is independently testable via the paired-task battery proposed in Section 8. The framework predicts that all four axes will show positive asymmetry scores, that the scores will correlate within models and that they will resist closing under scaling in the same training paradigm. If any axis shows near-zero asymmetry consistently, the taxonomy is wrong in a specific and informative way.

3.5 Structural Scaling Collapse

As structural complexity increases, performance on structural tasks degrades monotonically while descriptive fluency holds or improves. This is observed in synthetic graph reasoning [20, 69], long-horizon planning [67, 40] and compositional arithmetic [17, 77]. It distinguishes representational asymmetry from random failure. Uniform failure across complexity levels would suggest a missing domain. Monotonic collapse with increasing structural depth suggests the model has something. It just lacks structural simulation capacity.

4 Mechanistic Account

4.1 What the Objective Rewards

Next-token prediction is optimized by learning statistical regularities in text, memorizing high-frequency patterns and developing representations that generalize across surface forms. What it does not require: building internal models that simulate the processes being described.

A model trained on descriptions of sorting does not need to simulate sorting. It needs to predict the language of sorting descriptions. Descriptive regularities are abundant in text and highly predictive of next tokens. Structural simulation dynamics are sparse in text and provide marginal predictive signal. The model learns what it is rewarded to learn. Bommasani et al. [8] noted that emergent capabilities may arise from compressed world models but did not analyze the asymmetry between description and simulation. Brown et al. [10] made this point implicitly. Its full implications remain unexplored.

4.2 What Structural Competence Would Require

For genuine structural competence, a model would need internal representations encoding: invariants of the domain's processes; state-transition functions applicable iteratively; referential grounding connecting tokens to entities; and calibrated uncertainty reflecting actual epistemic state across predictions.

Evidence suggests some of these develop partially. Meng et al. [44] showed factual associations can be localized in transformer weights. Li et al. [38] found linear probes recover world-state variables from activations during board game play. Gurnee and Tegmark [26] found evidence of linear representations of space and time. Nanda et al. [46] showed models can learn modular arithmetic through grokking. These results matter: structural representations are not entirely absent. They

are inconsistently developed and unreliably accessed. That is precisely the pattern representational asymmetry predicts.

4.3 Training Data and Structural Signal

The training data composition amplifies the objective's bias. Natural language is overwhelmingly declarative. Descriptions of algorithms vastly outnumber execution traces. Explanations of causal processes vastly outnumber process simulations. Even code, the most procedurally-structured training data, is encountered as static text, not as dynamic execution [74]. Alain and Bengio [1] showed that linear probes reveal how features are structured across layers. Tenney et al. [65] found syntactic features cluster earlier and semantic features later. Structural simulation capacity is the least reinforced category across all of these. Ruis et al. [56] provide direct evidence: for reasoning tasks, the most influential pretraining documents are those demonstrating procedural steps, not those containing answers. Reasoning ability traces back to procedural signal in training data, not to factual co-occurrence. This is exactly the mechanism representational asymmetry predicts.

5 Empirical Evidence

representational asymmetry gains traction by unifying failures documented independently across six domains. Each maps onto a specific axis of the taxonomy.

5.1 Algorithmic Reasoning (Descriptive-Generative)

CLRS [68] constructed graph algorithm benchmarks. Models explaining BFS and DFS correctly fail to execute them on novel inputs. Anil et al. [4] documented spurious fluency: correct-seeming outputs that collapse under minimal perturbation. Variable renaming drastically reduces performance. The model learned the surface form of algorithm descriptions, not the operations themselves. Xu et al. [73] replicated this pattern across symbolic reasoning tasks with different surface forms. These failures are structural by the state-trajectory criterion: correct execution requires tracking variable bindings across steps.

5.2 Planning and Constraint Satisfaction (State-Transition)

Valmeekam et al. [67] showed GPT-4 performs near-randomly on planning tasks requiring constraint satisfaction while producing fluent descriptions of planning principles. Kambhampati [34] argued LLMs perform planning by pattern matching rather than constraint propagation. Liu et al. [40] extended this to multi-step task planning with tool use and found similar results. Silver et al. [59] noted that planning failures in LLMs do not resemble human planning failures, suggesting a structurally different underlying process. Planning is the canonical State-Transition task: every step changes the constraint budget. Stengel-Eskin et al. [63] provide controlled evidence using finite-state machine execution: models mimic algorithmic behavior for short traces but fail as procedural depth grows and larger models improve local accuracy while remaining brittle on multi-step execution. This is Structural Scaling Collapse under a controlled paradigm where world knowledge is irrelevant and only trajectory tracking determines correctness.

5.3 Hallucinated Citations (Referential-Grounding)

Models produce citations conforming to academic formatting conventions while fabricating authors, titles, venues and DOIs [31, 43]. Maynez et al. [43] noted hallucinated outputs often carry higher model confidence than correct ones. Ji et al. [31] documented that hallucination rates do not decrease predictably with model size. Peng et al. [53] showed retrieval augmentation reduces but does not eliminate grounding failures. Min et al. [45] found models confabulate not only facts but entire reasoning chains. Format is intact; the referential link to entities is absent.

5.4 Uncertainty Calibration (Uncertainty-Calibration)

Kadavath et al. [33] found models produce correct definitions of calibration while being poorly calibrated in practice. Kuhn et al. [36] showed semantic uncertainty fails to track actual factual

accuracy, with correlation degrading on structurally novel questions. Xiong et al. [72] found that confidence elicitation methods improve calibration on simple queries but fail on multi-step structural tasks. Jiang et al. [32] showed verbalized confidence is poorly correlated with accuracy. These are Uncertainty-Calibration failures: the model knows the definition of calibration without tracking its own epistemic state.

5.5 Mathematical Reasoning (Descriptive-Generative)

Lewkowycz et al. [37] trained Minerva on mathematical text. Strong performance on in-distribution problems coexisted with consistent failures on novel compositional steps. Failures occur at points requiring genuine structural inference, not pattern retrieval. Cobbe et al. [14] found solution verification is more reliable than solution generation, consistent with partial structural encoding at the representational level. Frieder et al. [21] found GPT-4 solves routine mathematical problems reliably but fails on proofs requiring novel structural steps. This is the Structural Scaling Collapse signature.

5.6 Code Execution (Descriptive-Generative)

Chen et al. [12] showed strong code generation performance coexisting with systematic failures on execution prediction. Models generate syntactically correct code that fails at runtime more often than they predict which of two implementations is faster. Austin et al. [6] found code correctness declines sharply with program length. Gu et al. [25] documented that models explain code functionality accurately while failing to trace execution on modified inputs. Code generation is descriptive; code execution prediction is structural. The gap between them is the axis in action.

6 Alternative Views

The tasks are simply hard. If humans also perform near-randomly on structural tasks, the asymmetry reflects intrinsic difficulty, not a model-specific deficit. representational asymmetry specifically predicts asymmetry *within* the same domain: descriptive and structural performance on matched tasks should diverge. Evidence suggests humans do not exhibit this pattern to the same degree. Human performance on planning and mathematical reasoning correlates more tightly across description and execution [67, 37]. Task hardness cannot explain an asymmetry that is absent in humans on the same tasks.

It is distribution shift. Synthetic benchmarks differ from training data. Failures may reflect out-of-distribution generalization rather than structural deficit. But if descriptive competence transfers to novel domains while structural competence does not, the asymmetry persists even under this framing. Marcus [42] argued LLM failures reflect systematic extrapolation limits. Hupkes et al. [30] showed compositional generalization fails systematically in neural models. Distribution shift predicts uniform failure; we observe selective failure on structural tasks. The phenomenon requires explanation regardless of what we call it.

Compound interventions already close the gap. CoT prompting [70], RLHF [50], tool use [40] and structural fine-tuning [47] each improve performance on structural tasks. Applied together, they substantially close observed gaps on some benchmarks. If compound interventions eliminate hidden absences, the asymmetry is a solvable engineering problem, not a structural critique of the training objective.

We argue this conflates behavioral improvement with closing the representational gap. A model that reasons correctly only when scaffolded with CoT has not acquired procedural knowledge. It has acquired a workaround that compensates for its absence. The asymmetry is still there; the scaffold masks it. This is testable: remove the scaffold and measure the performance drop. If performance drops back to unscaffolded baseline on structural tasks but not descriptive ones, the scaffold patched behavior without touching representation. Turpin et al. [66] showed CoT explanations are often unfaithful to the actual computation, consistent with this account. The compound intervention objection proves too much: by the same logic, a calculator attachment closes the arithmetic gap but no one concludes that LLMs have thereby acquired numerical reasoning.

The representations exist but the output mapping fails. Internal activations may encode structural variables correctly while the mapping to output tokens fails. Under this view the asymmetry is a decoding problem, not a representation problem. This is empirically testable: train linear probes on hidden states. If probes succeed where outputs fail, the intervention target shifts from training data to output routing. Li et al. [38] found partial evidence of this for board game state. The probing question is open for most domains and is addressed directly in Section 8. Importantly, this objection does not debunk the asymmetry. It relocates it. Either the deficit is in representation or in routing. Both are failures of structural competence by the criterion in Section 3.1.

Fine-tuning on structural data eliminates the phenomenon. If training on execution traces or simulation rollouts eliminates hidden absences, the asymmetry is a training data artifact. This is likely partially true. Nye et al. [47] showed scratchpad training improves structural reasoning substantially. But this confirms the framework's diagnosis, not debunks it. The claim is that next-token prediction on natural language text is insufficient. If fixing the gap requires qualitatively different training data, that is exactly what the framework predicts. It explains the origin rather than dismissing the phenomenon.

7 Philosophical and Cognitive Grounding

The claim that a system knows more than it can produce is not new. Recognizing the lineage clarifies what the framework asserts and why those assertions are strong.

7.1 Ryle: Knowing That vs. Knowing How

Gilbert Ryle's distinction between knowing that and knowing how [57] is the most directly relevant antecedent. Knowing that Dijkstra runs in $O(V \log V)$ is declarative. Knowing how to execute it is procedural. Ryle's insight was that these are genuinely different kinds of knowledge. They are not reducible to each other. representational asymmetry is, in one sense, an empirical test of Ryle's claim applied to transformers.

A model trained on text has absorbed an enormous amount of knowing that. Whether next-token prediction over descriptions of procedures builds knowing how is precisely what representational asymmetry interrogates. The evidence suggests it does not. At least not reliably.

Text is a trace of thought, not thought itself. You can read every book ever written about swimming without developing procedural knowledge of swimming. The question is whether transformers, trained on textual traces of reasoning, develop the underlying reasoning procedures. Or only their descriptions.

7.2 Chomsky: Competence vs. Performance

Chomsky's distinction between linguistic competence and linguistic performance [13] is structurally analogous. Competence can be rich while performance degrades under load or complexity. representational asymmetry applies this to neural models: rich descriptive representations coexist with degraded structural simulation capacity. The analogy suggests the phenomenon is deep and systematic rather than incidental.

7.3 ACT-R: Separate Memory Systems

Anderson's ACT-R cognitive architecture [3] formalizes the declarative-procedural distinction computationally. Declarative memory stores facts. Procedural memory stores condition-action production rules that fire during execution. These are stored and retrieved through different mechanisms with different learning dynamics. Procedural knowledge requires practice on the task itself. Declarative knowledge can be acquired from instruction alone.

Pre-training on text is, in ACT-R terms, a massive injection of declarative knowledge. It is not practice on structural tasks. Squire [60] established a similar distinction in human memory: explicit and implicit memory are neurologically dissociable. If these frameworks reflect something real about knowledge structure, we should expect exactly the asymmetry we observe in LLMs. That we do observe it is not surprising. That the field has not named it as such is.

7.4 Situated Cognition

Dreyfus [16] and Brooks [9] argued from different directions that genuine competence requires engagement with tasks, not description of them. We do not endorse the strongest versions of these positions. But the underlying point is directly relevant: a model trained on text about procedures is in the same epistemic position as a student who has read every book on surgery but has never held a scalpel. The state-trajectory criterion makes this precise: if correct performance requires tracking a changing state across steps, reading descriptions of that process is not sufficient preparation. Text is a record of declarative knowledge. It is not practice.

8 Falsifiable Predictions

A position without testable commitments is not a position. We specify four predictions the framework entails. Each is falsifiable. Each distinguishes representational asymmetry from the null hypothesis that current failures are incidental and will resolve with scale or intervention.

8.1 Prediction 1: The Asymmetry Is Measurable and Cross-Domain

If representational asymmetry is real, it should be measurable with a paired task battery. For each domain, construct a descriptive task and a structurally matched generative task at equal surface difficulty using the state-trajectory criterion from Section 3.1. The asymmetry score, defined as the normalized performance delta between the two, should be positive, consistent across model families and larger for domains with sparse structural training signal.

No such instrument currently exists. Hallucination benchmarks, planning benchmarks and calibration benchmarks each measure different things. None are designed to measure the gap between descriptive and structural competence on matched tasks. Priority domains: algorithmic execution traces [68], constraint-bounded planning [67], grounded entity resolution [41], calibrated probabilistic prediction [33] and code simulation [12]. If the asymmetry score is near zero or inconsistent across domains, the framework is wrong.

8.2 Prediction 2: Mechanistic Probing Will Find Weak Procedural Encoding

If the asymmetry is representational, probes on hidden states should recover structural variables less reliably than descriptive ones. Two outcomes are discriminable. In the first, structural variables are absent from internal representations entirely: probes on hidden states fail to recover game states, constraint budgets or referential grounding. The intervention is training on structural data.

In the second, structural variables are encoded in intermediate layers but overwritten or not routed to output. Probes succeed on intermediate layers and fail on later ones. The intervention is representational steering or activation patching [44, 18]. Following Li et al. [38], Meng et al. [44] and the mechanistic interpretability program [19, 48], a systematic probing study across a clean structural domain could discriminate between these. Elhage et al. [18] showed circuits in transformers can be isolated. Conmy et al. [15] demonstrated automated circuit discovery. If probes recover structural variables as reliably as descriptive ones, the asymmetry is in output mapping, not representation. Either way, the finding is concrete and the intervention is identifiable.

8.3 Prediction 3: The Absolute Gap Will Not Close Under Scaling

The sharpest prediction: the absolute performance gap between descriptive and structural tasks will remain stable as models scale within the same training paradigm. Descriptive performance will improve with scale [29, 35]. Structural performance will improve too. The gap between them will not close. Ganguli et al. [22] showed that structural failures persist or sharpen at scale even as descriptive performance improves. Stengel-Eskin et al. [63] found that larger models improve local accuracy on procedural tasks while remaining brittle on multi-step execution; the gap persists in ratio terms. Schaeffer et al. [58] showed that apparent emergent improvements often reflect metric choice rather than genuine capability closure. The asymmetry ratio, not absolute performance, is the diagnostic test.

8.4 Prediction 4: Structural Training Selectively Reduces the Gap

If the asymmetry arises because structural dynamics are sparse in natural language text, then training on execution traces, simulation rollouts or process-annotated corpora should selectively reduce the procedural gap without affecting descriptive performance. Nye et al. [47] showed scratchpad training substantially improves structural reasoning. Zelikman et al. [75] showed self-generated reasoning chains improve structural performance. Neither study measured the asymmetry score directly. If targeted structural training reduces the score from Prediction 1 while leaving descriptive task performance stable, the causal claim is validated.

What would falsify the framework entirely: near-zero asymmetry scores across multiple domains using the Prediction 1 battery; or probing evidence that structural variables are encoded as reliably as declarative ones in the same model family where behavioral asymmetry is observed. Either result would require a fundamentally different account of the failures documented in Section 5.

9 Safety and Deployment Implications

9.1 Epistemic Overtrust

The safety implications of representational asymmetry are specific, not generic. The issue is not that AI systems fail. All systems fail. The issue is that hidden absences produce fluent, confident, plausible outputs in exactly the cases where structural simulation is wrong. The failure mode is invisible at the surface. That is what makes it a safety problem rather than a reliability problem.

We call this epistemic overtrust: because hidden absences look like competence from the outside, users develop miscalibrated confidence in AI reasoning in high-stakes structural domains. Weidinger et al. [71] identified automation bias as among the most practically important LLM harms. representational asymmetry provides the mechanistic basis for where automation bias will be specifically acute: in structural reasoning domains, where the gap between apparent and actual competence is largest and least detectable.

Buccinca et al. [11] showed cognitive forcing functions reduce AI overreliance. Bansal et al. [7] found calibration degrades when uncertainty expressions are fluent but inaccurate. That is the Hidden Absence scenario exactly. Passi and Barocas [51] documented how practitioners conflate model fluency with model reliability. These findings are consistent with representational asymmetry but were not predicted by it. The framework now makes those predictions in advance.

9.2 Predictive Risk Ranking

representational asymmetry generates a predictive ranking of deployment domains by Hidden Absence risk before empirical evidence is in. Risk is highest where three conditions co-occur: the task is structural by the state-trajectory criterion; the output is fluent regardless of whether the simulation is correct; and errors in the underlying structural task produce consequential real-world decisions.

Each high-risk domain satisfies all three conditions by the state-trajectory criterion. Drug interaction prediction requires tracking pharmacokinetic trajectories across time and metabolic pathways: the answer depends on the full state sequence, not the terminal concentration alone. Gong et al. [24] already document the asymmetry empirically: models exceed physicians on licensing examinations while scoring 45–55% on diagnostic tasks. Multi-step legal reasoning requires propagating constraints across case facts, statutes and precedents; each step changes what conclusions remain admissible. Models produce fluent legal prose whose constraint validity is invisible from its form. Long-horizon resource planning requires tracking budgets across a decision sequence where each allocation changes what remains available downstream. Epidemiological projection under novel conditions requires simulating dynamics not present in training data; the state-trajectory is the simulation itself and no terminal-state shortcut exists.

This ranking is falsifiable. If empirical study finds that these domains do not show elevated asymmetry scores relative to matched descriptive tasks, the framework's safety predictions are wrong. That is a stronger commitment than the existing safety literature, which catalogues failures after they occur rather than predicting them in advance.

9.3 Evaluation Design

Standard evaluations (MMLU [28], HellaSwag [76], BIG-Bench [61]) are dominated by descriptive tasks. A model can score highly on these while failing systematically on structural tasks in the same domains. Liang et al. [39] (HELM) broadened evaluation across dimensions but does not distinguish descriptive from structural performance within domains. The asymmetry score from Prediction 1 would directly address this gap. Evaluations that mix descriptive and structural tasks without distinguishing them produce aggregate scores that mask the asymmetry entirely. Anthropic's model card [5] and OpenAI's GPT-4 report [49] acknowledge planning and reasoning failures but do not frame them as structurally predicted. Amodei et al. [2] argued safety requires understanding the structure of model errors. The taxonomy in Section 3.4 is a step toward that.

10 Conclusion

The scaling paradigm treats capability as a single dimension. representational asymmetry argues it is at least two. Next-token prediction over natural language text builds declarative competence reliably and procedural competence weakly. The field is not making a mistake by scaling. It is making a measurement mistake: current evaluations track the dimension that improves and miss the one that does not. Ryle's distinction between knowing that and knowing how maps cleanly onto this gap. The four-axis taxonomy proposed here gives the field a shared instrument for measuring it and four predictions specific enough to be wrong. As AI systems take on greater roles in clinical reasoning, legal analysis and scientific planning, the cost of confusing fluent description with reliable simulation will grow. Understanding representational asymmetry is not optional for safe deployment. It is a prerequisite.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *ICLR Workshop*, 2017.

[2] Dario Amodei et al. Concrete problems in AI safety. *arXiv:1606.06565*, 2016.

[3] John R. Anderson et al. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004.

[4] Cem Anil et al. Exploring length generalization in large language models. In *NeurIPS*, volume 35, 2022.

[5] Anthropic. Claude model card. 2023.

[6] Jacob Austin et al. Program synthesis with large language models. *arXiv:2108.07732*, 2021.

[7] Gagan Bansal et al. Does the whole exceed its parts? the effect of AI explanations on complementary team performance. In *CHI*, 2021.

[8] Rishi Bommasani et al. On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2022.

[9] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1–3):139–159, 1991.

[10] Tom B. Brown et al. Language models are few-shot learners. In *NeurIPS*, volume 33, pages 1877–1901, 2020.

[11] Zana Buccinca, Maja Barbara Malaya, and Krzysztof Z. Gajos. To trust or to think: Cognitive forcing functions can reduce overreliance on AI. In *ACM CSCW*, volume 5, 2021.

[12] Mark Chen et al. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.

[13] Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, 1965.

[14] Karl Cobbe et al. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.

[15] Arthur Conmy et al. Towards automated circuit discovery for mechanistic interpretability. In *NeurIPS*, volume 36, 2023.

[16] Hubert L. Dreyfus. *What Computers Can't Do: A Critique of Artificial Reason*. Harper & Row, 1972.

[17] Nouha Dziri et al. Faith and fate: Limits of transformers on compositionality. In *NeurIPS*, volume 36, 2023.

[18] Nelson Elhage et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

[19] Nelson Elhage et al. Toy models of superposition. *Transformer Circuits Thread*, 2022.

[20] Bahare Fatemi et al. Talk like a graph: Encoding graphs for large language models. *arXiv:2310.04560*, 2023.

[21] Simon Frieder et al. Mathematical capabilities of ChatGPT. In *NeurIPS*, volume 36, 2024.

[22] Deep Ganguli et al. Predictability and surprise in large generative models. In *ACM FAccT*, 2022.

[23] Karan Goel et al. Robustness gym: Unifying the NLP evaluation landscape. In *NAACL*, 2021.

[24] Eun Ji Gong et al. Knowledge-practice performance gap in clinical large language models: Systematic review of 39 benchmarks. *JMIR*, 27:e84120, 2025.

[25] Alex Gu et al. CRUXEval: A benchmark for code reasoning, understanding, and execution. *arXiv:2401.03065*, 2024.

[26] Wes Gurnee and Max Tegmark. Language models represent space and time. In *ICLR*, 2024.

[27] Shibo Hao et al. Reasoning with language model is planning with world model. In *EMNLP*, 2023.

[28] Dan Hendrycks et al. Measuring massive multitask language understanding. In *ICLR*, 2021.

[29] Jordan Hoffmann et al. Training compute-optimal large language models. In *NeurIPS*, volume 35, 2022.

[30] Dieuwke Hupkes et al. Compositionality decomposed: How do neural networks generalise? *JAIR*, 67:757–795, 2020.

[31] Ziwei Ji et al. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[32] Zhengbao Jiang et al. How can we know when language models know? *Transactions of the ACL*, 9:962–977, 2021.

[33] Saurav Kadavath et al. Language models (mostly) know what they know. *arXiv:2207.05221*, 2022.

[34] Subbarao Kambhampati. Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 1534(1):15–18, 2023.

[35] Jared Kaplan et al. Scaling laws for neural language models. *arXiv:2001.08361*, 2020.

[36] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in NLG. In *ICLR*, 2023.

[37] Aitor Lewkowycz et al. Solving quantitative reasoning problems with language models. In *NeurIPS*, volume 35, 2022.

[38] Kenneth Li et al. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *ICLR*, 2023.

[39] Percy Liang et al. Holistic evaluation of language models. *TMLR*, 2023.

[40] Xiao Liu et al. AgentBench: Evaluating LLMs as agents. *arXiv:2308.03688*, 2023.

[41] Alex Mallen et al. When not to trust language models. In *ACL*, 2023.

[42] Gary Marcus. Deep learning: A critical appraisal. *arXiv:1801.00631*, 2018.

[43] Joshua Maynez et al. On faithfulness and factuality in abstractive summarization. In *ACL*, 2020.

[44] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, volume 35, 2022.

[45] Sewon Min et al. FActScoring: Fine-grained atomic evaluation of factual precision in long-form generation. In *EMNLP*, 2023.

[46] Neel Nanda et al. Progress measures for grokking via mechanistic interpretability. In *ICLR*, 2023.

[47] Maxwell Nye et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv:2112.00114*, 2021.

[48] Chris Olah et al. Zoom in: An introduction to circuits. *Distill*, 2020.

[49] OpenAI. GPT-4 technical report. *arXiv:2303.08774*, 2023.

[50] Long Ouyang et al. Training language models to follow instructions with human feedback. In *NeurIPS*, volume 35, 2022.

[51] Samir Passi and Solon Barocas. Problem formulation and fairness. In *ACM FAccT*, 2019.

[52] Arkil Patel, Satwik Bhatt, and Chitta Baral. Are NLP models really able to solve simple math word problems? In *NAACL-HLT*, 2021.

[53] Baolin Peng et al. Check your facts and try again. *arXiv:2302.12813*, 2023.

[54] Chen Qian et al. Limitations of language models in arithmetic and symbolic reasoning. *arXiv:2208.05051*, 2022.

[55] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *ACL*, 2020.

[56] Laura Ruis et al. Procedural knowledge in pretraining drives reasoning in large language models. *arXiv:2411.12580*, 2024.

[57] Gilbert Ryle. *The Concept of Mind*. Hutchinson, 1949.

[58] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *NeurIPS*, volume 36, 2023.

[59] Tom Silver et al. Generalized planning in PDDL domains with pretrained large language models. In *AAAI*, 2024.

[60] Larry R. Squire. Declarative and nondeclarative memory: Multiple brain systems supporting learning and memory. *Journal of Cognitive Neuroscience*, 4(3):232–243, 1992.

[61] Aarohi Srivastava et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *TMLR*, 2023.

[62] Aarohi Srivastava et al. Imitation of life: A search for counterexamples to the scaling hypothesis. *arXiv:2312.01473*, 2023.

[63] Elias Stengel-Eskin et al. The illusion of procedural reasoning: Measuring long-horizon FSM execution in LLMs. *arXiv:2511.14777*, 2025.

[64] Mirac Suzgun et al. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *ACL Findings*, 2023.

[65] Ian Tenney et al. BERT rediscovers the classical NLP pipeline. In *ACL*, 2019.

[66] Miles Turpin et al. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *NeurIPS*, volume 36, 2023.

[67] Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan. In *NeurIPS Workshop on Foundation Models for Decision Making*, 2022.

[68] Petar Veličković et al. The CLRS algorithmic reasoning benchmark. In *ICML*, 2022.

[69] Bowen Wang et al. Can language models solve graph problems in natural language? In *NeurIPS*, volume 36, 2023.

[70] Jason Wei et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, volume 35, 2022.

[71] Laura Weidinger et al. Taxonomy of risks posed by language models. In *ACM FAccT*, 2022.

[72] Miao Xiong et al. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *ICLR*, 2024.

[73] Frank F. Xu et al. A systematic evaluation of large language models of code. In *SPLASH Companion*, 2022.

[74] Pengcheng Yin et al. Learning to generate pseudo-code from source code using statistical machine translation. In *ASE*, 2018.

[75] Eric Zelikman et al. STaR: Bootstrapping reasoning with reasoning. In *NeurIPS*, volume 35, 2022.

[76] Rowan Zellers et al. HellaSwag: Can a machine really finish your sentence? In *ACL*, 2019.

[77] Denny Zhou et al. Least-to-most prompting enables complex reasoning in large language models. In *ICLR*, 2023.