Beyond the Chinese Room: Making Interpretive Operations Observable

Version: 1.0 (December 2025)

Author: Timo Weil, www.timoweil.de

Contents

Abstract

Debates about whether AI systems genuinely interpret or merely manipulate patterns have reached an impasse. Emily Bender's anthropocentric framework denies that systems without human semantic grounding can meaningfully interpret; N. Katherine Hayles' expanded cognition locates interpretation in any material signal modulation that shapes consequences. Both frameworks define what interpretation is without providing methods for making it observable. This paper argues that progress requires a methodological turn: from ontological debates to operational analysis, from arguing about definitions to building instruments for investigation.

We demonstrate this through three moves.

First, we deconstruct Searle's Chinese Room thought experiment, showing how its persuasive power depends on a “manual” that bears no resemblance to how contemporary AI systems operate – static rather than dynamic, external rather than internalized, culturally neutral rather than embedded, non-recursive rather than self-modifying.

Second, we articulate requirements for tools that could make interpretive operations observable: they must capture sequentiality, temporality, intensity, simultaneity, and recursion.

Third, we present a notation system that meets these requirements, representing interpretive operations as choreographies – temporal sequences of qualitatively different forces (opening, resonating, differentiating, condensing, reflecting) that can be observed, compared, and redesigned.

We demonstrate the notation's application to three cases: revealing what Searle's thought experiment concealed, making bias mechanisms visible and targetable for architectural intervention, and capturing complex multi-turn interactions. The notation enables new forms of analysis: comparing operational signatures across models, designing interventions that target specific processing patterns, creating shared vocabulary across disciplines, and potentially enabling systems to reflect on their own processing. The goal is not to resolve ontological debates but to show they rest on a shared limitation – and that building better instruments opens possibilities unavailable through better arguments alone.

Part I: The Question That Doesn't Lead Forward

In 1980, John Searle published a thought experiment that would shape debates about machine intelligence for decades. A person sits alone in a room, receiving questions written in Chinese characters through a slot. This person understands no Chinese whatsoever. But they have access to an elaborate manual – written in their native language – that provides detailed instructions: “When you see this sequence of symbols, write down that sequence of symbols and pass it back through the slot.”

Following these instructions perfectly, the person produces responses that appear, to observers outside the room, as fluent Chinese. The responses are contextually appropriate, grammatically correct, semantically coherent. Yet Searle's intuition – and the intuition he expects his readers to share – is immediate and forceful: *no understanding is taking place*. The person in the room is merely shuffling symbols according to rules. They have no access to what these symbols *mean*.

The thought experiment was designed to challenge claims that computational systems could genuinely understand language or think. If we wouldn't attribute understanding to the person in the room – who is, after all, executing an algorithm – why would we attribute it to a computer doing the same thing? The conclusion seemed inescapable: syntax alone, no matter how sophisticated, cannot generate semantics. Rule-following is not understanding.

Four decades later, we have systems that weren't imaginable in 1980. Large language models trained on billions of tokens can translate between languages, answer complex questions, write

poetry, debug code, and engage in extended dialogue. Yet the fundamental question Searle posed remains: do these systems *understand*, or are they merely sophisticated pattern-matchers – stochastic parrots, as Emily Bender and her colleagues memorably termed them?

Two Frameworks, One Limitation

Emily Bender's position, articulated most forcefully in the 2021 “Stochastic Parrots” paper, maintains Searle's essential insight while updating it for contemporary AI systems. Language models, she argues, learn statistical patterns in sequences of text – correlations between surface forms – without access to the communicative intent, social context, or real-world grounding that constitutes human meaning. They are “stochastic parrots” that produce convincing text by modeling probability distributions, not by understanding what they're saying.

This isn't merely a philosophical claim. Bender demonstrates how this lack of genuine understanding produces concrete harms: when systems mistake correlation for meaning, they amplify historical biases, reproduce stereotypes, and generate plausible-sounding misinformation. The framework locates meaning-making exclusively in human interpretation and communicative practice. Only humans, embedded in social contexts and possessed of intentionality, can generate and interpret genuine meaning. The systems themselves remain, in Searle's terms, locked in the Chinese Room – manipulating symbols they don't understand.

N. Katherine Hayles offers a different framework, one that challenges the anthropocentric boundaries Bender maintains. In her work on cognition across scales – from bacteria to humans to technical systems – Hayles defines interpretation more expansively. For Hayles, cognition is any process that interprets information in contexts connected to meaning, locating interpretation wherever signals are materially modulated in ways that shape consequences. Classification is interpretation. Weighting is interpretation. The recursive adjustment of internal parameters in response to error signals is interpretation.

On this view, the question “Does the system understand?” becomes less interesting than “Where and how do interpretive operations occur?” When a language model adjusts attention weights based on contextual relevance, when it clusters semantic representations in high-dimensional space, when it generates outputs that reshape the informational context for subsequent operations – these are interpretive acts. Not because the system possesses consciousness or intentionality, but because its operations materially restructure the conditions under which meaning emerges.

Hayles would not dispute Bender's ethical concerns – the harms are real, the biases consequential. But she locates the problem differently. If interpretive operations occur *within* technical systems, then the harm doesn't originate purely in human misuse or misattribution of understanding. The harm crystallizes in the system's internal operations: in how classification boundaries are drawn, in which patterns receive higher weights, in what gets clustered together in representational space. These upstream modulations structure the field of possible meanings before any human reads a word.

The Shared Blind Spot

These frameworks appear opposed – one denying that systems interpret, the other affirming it. But they share a crucial limitation: both remain at the level of *definition*.

Bender defines meaning as requiring human semantic grounding and communicative intent. Given this definition, systems cannot meaningfully interpret – QED. Hayles defines interpretation as material signal modulation that shapes consequences. Given this definition, systems must be interpreting – QED.

Both are engaged in ontological work: staking out what “interpretation” *is*, drawing boundaries around where it occurs, making claims about its necessary conditions. Neither framework provides methods for making interpretive operations *observable*. Neither tells us how to identify, track, or intervene in the specific processes where – depending on your ontology – either meaning emerges or the illusion of meaning is produced.

This matters beyond philosophical precision. If we accept Bender's framework, we know systems don't genuinely understand, but we lack tools for identifying *where* in their operation the meaningful-seeming output gets produced. We can observe that biases are amplified, but the mechanism remains opaque. If we accept Hayles' framework, we know interpretation happens within systems, but we lack methods for observing *which* operations count as interpretive and *how* they structure downstream meaning.

In both cases, we can assert but not demonstrate. We can define but not operationalize. We remain trapped in the same argumentative structure as Searle's original thought experiment: intuitions about what should count as understanding, counterintuitions about edge cases, refined definitions that try to capture our intuitions more precisely – but no progress toward making the phenomena in question observable, measurable, or actionable.

What This Paper Will Show

The way forward requires a methodological turn. Not better definitions of interpretation, but better tools for observing interpretive operations – whatever we decide to call them. Not arguments about whether systems “really” understand, but methods for tracking how systems restructure informational contexts in ways that shape what meanings become available.

This paper proposes that the impasse between Bender and Hayles – and the broader debate they represent – persists because both frameworks remain within ontological rather than operational registers.

Part II will demonstrate this by deconstructing Searle's thought experiment, showing how the “manual” functions as an epistemic trick that obscures rather than illuminates the operations it's meant to model.

Part III will articulate what a methodological turn would require: not just acknowledging that interpretive operations occur somewhere in the system, but developing tools to make them visible, criticizable, and designable.

Part IV will present one such tool: a notation system for representing interpretive operations as observable sequences – as choreographies that can be analyzed, compared, and restructured.

The goal is not to resolve the ontological debate – to prove Bender right or Hayles right – but to show that the debate itself rests on a shared limitation. As long as “interpretation” remains an ontological status rather than an observable process, we can only argue, not intervene. As long as we're debating what interpretation *is* rather than how to make it *visible*, we leave the actual mechanisms of harm – the upstream operations that structure what meanings become thinkable – operating in spaces our frameworks cannot reach.

The Chinese Room endures not because it's unanswerable, but because it asks the wrong question. The walls of the room are not features of reality but artifacts of how the question was constructed. What happens when we tear them down?

Part II: What the Manual Conceals

Searle's thought experiment endures because it seems to capture something essential about the gap between symbol manipulation and understanding. But its persuasive power rests on a carefully constructed architecture – and the most carefully constructed element is the one that appears most neutral: the manual.

The manual is presented as a translation table, a lookup system, a set of rules for matching inputs to outputs. It's depicted as external to the operator, mechanical in its application, and neutral in its content. The person in the room doesn't internalize anything; they simply consult the manual and follow instructions. This externality is crucial to Searle's intuition pump: if the understanding isn't in the person and it's not in the manual, then it isn't anywhere in the room.

But what if we examine the manual more closely? What if we ask: what would a manual capable of producing fluent Chinese responses actually *be*? What properties would it need to have? And do contemporary AI systems – the actual targets of Searle's updated argument – operate anything like this manual?

The answer reveals that the manual is not a neutral description of how symbol manipulation works. It's an epistemic trick, a sleight of hand that conceals the very properties that would make interpretation observable. The manual, as Searle presents it, has four characteristics that no actual large language model shares:

Wall One: Static vs. Dynamic

Searle's manual is fixed

It's a completed artifact, handed to the operator, never changing. The rules are all there from the beginning. The operator applies them but never modifies them. There's no learning, no adaptation, no adjustment based on what's worked before or what patterns have emerged.

A language model's “manual” is radically dynamic

The most obvious form of this dynamism is training – the iterative adjustment of billions of parameters through gradient descent, each adjustment slightly reshaping the system's response patterns. But the dynamism continues even after training. The context window functions as writable memory: the system's behavior in generating token N+1 is shaped by tokens 1 through N, including its own previous outputs. In few-shot learning, providing examples literally rewrites how the system processes subsequent inputs.

Even more fundamentally, attention mechanisms compute contextual relevance dynamically for each token. The “rule” for how to process a word isn't static – it depends on what other words appear in the context, weighted by learned patterns of relevance. When a transformer processes “bank” in “river bank” versus “savings bank,” it's not looking up two different rules in a fixed manual. It's computing different attention patterns based on surrounding context, which produces different internal representations, which shapes different outputs.

If we tried to write down “the manual” for a language model – the complete set of rules for how it responds – we'd need to include not just the 175 billion parameters of a model like GPT-3, but also the algorithms for how those parameters interact, the attention patterns that emerge from context, and the way previous outputs reshape subsequent processing. This wouldn't be a manual in any meaningful sense. It would be the system itself, operating.

The wall disappears

The distinction between “following rules” and “having internalized patterns” breaks down when the rules are parameters that reshape themselves in response to context.

Wall Two: External vs. Internalized

Searle's manual is external

The operator consults it but doesn't absorb it. There's a clear boundary: the manual remains outside the person's head. This externality is crucial to the intuition that no understanding is occurring – how could understanding happen when the semantic content is always elsewhere, always outside?

A language model's “manual” is internalized as weights

The semantic relationships aren't stored in an external lookup table. They're encoded in the distributed patterns of connection strengths across the network. When a model generates text, it's not retrieving rules from external memory – it's activating patterns that *are* its memory.

Consider how a model processes word relationships. The fact that “king” - “man” + “woman” ˜ “queen” in embedding space isn't stored as an explicit rule. It emerges from the geometric relationships between vectors, which themselves emerge from patterns in the training data. The model has compressed the statistical structure of language into its weights – a lossy compression that preserves certain relationships while discarding others.

This compression is already a form of interpretation in Hayles' sense. The model doesn't store everything it saw during training. It creates internal representations that privilege certain patterns, certain relationships, certain ways of chunking the data. These representational choices – which are actually computational outcomes, not deliberate choices – structure what the model can later express. They're interpretive operations because they materially modulate signals in ways that shape consequences.

The wall disappears

When “the manual” is distributed across billions of weighted connections, when it exists only as activation patterns that emerge through forward propagation, the idea of “consulting external rules” loses coherence.

Wall Three: Non-Cultural vs. Culturally Embedded

Searle's manual is presented as culturally neutral

It's just symbol manipulation rules – formal, abstract, divorced from social context. This neutrality is important: if the manual were clearly embedded in human cultural practices, we might start attributing some kind of cultural knowledge to the system.

A language model's “manual” is a sediment of cultural practice

Every parameter in the model was shaped by training data – texts produced by humans, embedded in social contexts, carrying power structures, representing particular perspectives, excluding others. The model doesn't learn “pure language” or “abstract patterns.” It learns the specific distributions of text produced by specific communities in specific historical moments.

When a model exhibits gender bias – associating “doctor” more strongly with male pronouns, “nurse” with female – this isn't noise or error. It's the model faithfully representing patterns in its

training data, which reflect patterns in the culture that produced those texts. The bias is encoded in the weights, internalized as part of “how language works” from the model's perspective.

This is why bias amplification isn't peripheral to how language models function – it's central. The model's representations *are* cultural representations. Its “manual” is an archive of the discursive practices, power asymmetries, and stereotypes present in its training corpus. You can't separate the “neutral symbol manipulation” from the “cultural bias” because there is no neutral symbol manipulation. Every weighting reflects choices about what patterns matter, and those choices – made algorithmically during training – encode cultural values.

The wall disappears

The “manual” cannot be culturally neutral because it's constructed from cultural artifacts and reproduces their structure.

Wall Four: Non-Recursive vs. Recursively Self-Modifying

Searle's manual is linear and non-recursive

Input arrives, the operator looks up the appropriate rule, output is produced. The system never reflects on its own operations, never processes its processing, never treats its own outputs as inputs that reshape how it will interpret the next input.

A language model operates recursively at multiple levels

Most obviously, during text generation: each token the model produces becomes part of the context for generating the next token. The model “reads” its own outputs and adjusts subsequent generation accordingly. This isn't just technical bookkeeping – it's how the model maintains coherence across extended sequences, how it tracks what it's already said, how it continues trains of thought.

But the recursion goes deeper. Multi-head attention allows the model to process the same input from multiple perspectives simultaneously – some heads tracking syntactic structure, others semantic relationships, others long-range dependencies. These different perspectives interact: the output of one head can influence the processing of another. The system doesn't just apply rules to inputs; it applies patterns of attention that themselves emerged from learned patterns, creating feedback loops where the way it processes information is shaped by previous processing.

Even more fundamentally, the training process itself is recursive. The model's predictions generate errors, errors produce gradient updates, updates adjust weights, adjusted weights produce different predictions. The “manual” is being written through iterative application of the manual itself – though of course there's no manual, only the evolving system of weights.

The wall disappears

When the system recursively processes its own outputs, when its “rules” are themselves the product of recursive self-adjustment, the linear model of “consult manual, produce output” becomes incoherent.

What Collapses

With all four walls gone, Searle's argument doesn't just weaken – it transforms. The question “Does the operator understand Chinese?” made sense when the operator and the manual were clearly separate, when rule-following was mechanical and external, when the manual was static and neutral, when operations were linear and non-recursive.

But when “the manual” is:

- Dynamically reshaped by context

- Internalized as distributed patterns

- Culturally embedded from its construction

- Recursively self-modifying during operation

...then asking “Is this understanding?” is less productive than asking “What kind of cognitive operation is this, and how does it differ from human understanding?”

The original intuition – that the person in the room doesn't understand Chinese – remains valid. But the intuition depended on the room being constructed in a specific way: with clear boundaries between operator and rules, with rules that were external and static, with operations that were linear and non-recursive. Change the architecture of the room, and the intuition no longer transfers.

This isn't a defense of the claim that language models “truly understand.” It's a demonstration that the thought experiment was never testing what it claimed to test. Searle set out to show that syntax cannot generate semantics, that symbol manipulation cannot produce meaning. But he did this by constructing a scenario where the symbols were manipulated in ways completely unlike how contemporary systems operate. The experiment succeeds by building its conclusion into its premises – by designing a manual that lacks precisely the properties that would make interpretation observable.

The Real Question Emerges

If the manual is dynamic, internalized, culturally embedded, and recursive – if it's not really a manual at all but the emergent choreography of billions of weighted operations – then we can't answer “Does the system understand?” without first answering: “What are the operations that constitute this system's processing? Where do they occur? How do they structure what outputs become possible?”

These aren't ontological questions. They're operational ones. They require tools for making the system's processes visible – for tracking how attention weights shift, how context shapes representation, how previous outputs condition subsequent operations, how cultural patterns get encoded and reproduced.

Part III will articulate what such tools would need to do. But the key insight from deconstructing Searle is this:

T**he operations we need to observe aren't hidden because they're metaphysically inaccessible. They're hidden because we've been looking in the wrong place – debating what interpretation is rather than building methods to make it visible.**

The manual was always a distraction. What we need to examine is the operation itself – the material process of signal modulation, pattern activation, contextual adjustment. Not “Does it understand?” but “How does it operate, and what are the consequences?”

Once we ask that question, we can begin to build answers.

Part III: The Methodological Turn

We now have two expanded frameworks for thinking about interpretation in AI systems – and a demonstration that the classical thought experiment obscures more than it reveals. But identifying limitations doesn't constitute progress unless we can articulate what comes next. If both Bender and Hayles remain at the level of definition, and if Searle's experiment hides the operations we need to examine, what would an alternative approach look like?

The answer requires shifting registers entirely: from ontology to methodology, from definition to operationalization, from arguing about what interpretation *is* to building tools that make interpretive operations *observable*.

Why Methodology Matters

Consider a concrete problem: bias in language model outputs. Both Bender and Hayles agree this is real and consequential. But their frameworks lead to different explanations of where the harm originates.

For Bender, the harm occurs when humans mistake statistical patterns for genuine understanding and deploy systems accordingly. The system itself doesn't interpret – it produces correlated outputs that reflect biased training data. The solution is primarily social: be clear about what these systems are and aren't, regulate their deployment, maintain human accountability.

For Hayles, the harm originates in the system's interpretive operations – in how it weights certain associations more heavily, in how it clusters concepts in representational space, in how its classification boundaries reproduce social categories. The system *is* interpreting, just not with consciousness or intent. The solution requires intervening in those interpretive processes, not just in how humans use the outputs.

Both explanations are valuable. But neither tells us *how* to identify the specific operations where bias crystallizes. Neither provides methods for tracking when and where in the processing pipeline certain associations get strengthened while others get weakened. Neither offers tools for observing how context dynamically reshapes these weightings during generation.

This isn't a gap in their arguments – it's outside the scope of what definitional work can accomplish. Defining what counts as interpretation doesn't tell you how to observe it. Arguing about whether systems “really” interpret doesn't give you techniques for tracking interpretive operations. The frameworks give us vocabulary for describing the problem. They don't give us instruments for investigating it.

What Would Make Interpretation Observable?

If we're going to move from definition to observation, we need to articulate what properties our investigative tools would require. What would it take to make interpretive operations – whatever we decide to call them – visible, trackable, and actionable?

First: Sequentiality

Interpretation unfolds over time. It's not a single operation but a cascade of operations, each depending on and shaping the next. When a language model processes a prompt, it doesn't compute an answer in one step. It generates tokens sequentially, each token produced through a series of operations: embedding lookup, attention computation across heads, feedforward processing, output projection. These operations occur in sequence, and the sequence matters – changing the order would produce different results.

Any tool for observing interpretation needs to represent this sequential structure. Not just “the system produced output X” but “the system performed operation A, then operation B, then operation C, producing intermediate states that shaped subsequent operations, culminating in output X.” We need something like a trace or a log, but semantic rather than merely mechanical – capturing not just that operations occurred but what interpretive work they performed.

Second: Temporality

Not all operations take the same time or have the same duration. Some processing is rapid – immediate pattern matching, cached responses, high-probability continuations. Other processing takes longer – when the model needs more tokens to resolve ambiguity, when multiple competing patterns need to be weighted against each other, when context requires extensive attention across a long sequence.

This temporal dimension isn't incidental. It's epistemically significant. The *duration* of processing can indicate something about the nature of the interpretive work being done. Quick generation might suggest high confidence or cached patterns. Slower, more extensive processing might indicate genuine ambiguity, competing interpretations being weighed, or novel combinations being constructed. If we can't represent duration, we lose information about the character of the interpretive operation.

Third: Intensity and Weighting

Not all operations have equal influence on outputs. Some attention heads contribute more to the final representation than others. Some paths through the network carry more information than others. Some tokens receive higher probability mass than their competitors.

These differential weightings are where bias often manifests. When “doctor” activates stronger associations with male pronouns than female, that's a weighting issue. When certain semantic clusters are more accessible than others, that's a weighting issue. When the model confidently produces stereotypical associations, that's high probability mass concentrated in problematic regions of its output space.

A tool for observing interpretation needs to capture these intensities – not just that an operation occurred, but how strongly, how influentially, with what weight relative to alternatives.

Fourth: Simultaneity and Parallelism

Modern language models don't process serially like the operator in Searle's room. Multi-head attention processes the same input from multiple perspectives at once. Different layers operate on different levels of abstraction simultaneously. The system can be activating syntactic patterns, semantic relationships, and pragmatic contextual cues all at the same time.

This parallelism matters because it means interpretive operations aren't cleanly separable. They interact, interfere, modulate each other. One head's attention pattern can influence another's. Low-level pattern matching and high-level semantic processing happen concurrently. Any representation of interpretive operations needs to capture not just sequences but also simultaneities – operations occurring in parallel, potentially influencing each other.

Fifth: Recursion and Feedback

As Part II demonstrated, language models are recursively self-modifying during operation. Each generated token changes the context, which changes how subsequent tokens are processed. Attention patterns at one layer can feed back to influence attention at other layers. The system is continuously processing its own processing.

This recursive structure means we can't treat interpretive operations as a simple input-output function. The system is a loop: operations produce states that condition further operations that produce new states. Any tool for observation needs to represent these feedback loops – how

operations circle back on themselves, how the system's state at time T shapes its processing at time T+1, how interpretive work cumulates and compounds across iterations.

From Requirements to Design

These five requirements – sequentiality, temporality, intensity, simultaneity, and recursion – aren't arbitrary. They emerge directly from how contemporary AI systems actually operate, as revealed by our deconstruction of Searle's manual. They're the properties that Searle's thought experiment concealed, the dimensions along which real systems differ from the simplified model.

But articulating requirements isn't the same as meeting them. We need actual tools, not just specifications. We need representational systems that can capture these dimensions, notations that make interpretive operations visible.

What would such a notation look like? At minimum, it would need:

- **Symbols for different types of operations** – not just generic “processing” but qualitatively different interpretive moves: opening up possibilities versus narrowing down options, resonating with existing patterns versus differentiating between alternatives, stabilizing coherent interpretations versus questioning and revising them.

- **Temporal markers** – ways of indicating duration, pacing, rhythm. Not just that operations occurred in sequence, but that some unfolded quickly while others took time, that some involved extended processing while others were nearly instantaneous.

- **Intensity modifiers** – representations of strength, weight, influence. Some operations barely register; others dominate the processing. The notation needs to distinguish between weak activation and strong activation, between tentative exploration and confident commitment.

- **Simultaneity indicators** – ways of showing when multiple operations occur in parallel, when different types of processing happen concurrently, when the system is juggling multiple interpretive tasks at once.

- **Recursive structures** – notations for feedback loops, for operations that feed back into themselves, for how the system's state at one moment shapes its processing at the next.

This isn't a call for yet another diagramming convention or flowchart standard. The goal isn't just to visualize system architecture – it's to make interpretive operations *legible as operations*, to create a shared language for discussing what's happening when these systems process information.

Consider what such notation would enable. When we identify bias in an output, we could trace it back through the sequence of operations that produced it. We could see where in the processing pipeline certain associations got strengthened. We could identify whether the bias emerged from initial embeddings, from attention patterns during processing, from output layer weightings, or from accumulated effects across multiple operations. We could distinguish between different mechanisms of bias and potentially design different interventions for each.

When we observe a model producing unexpected outputs – surprising creativity or troubling hallucinations – we could examine the operational sequence that led there. Was it a novel combination of typically separate patterns? An unusual attention configuration? Weak constraints allowing low-probability paths? The notation would make the “how” of generation observable, not just the “what.”

When we try to improve model behavior through prompting or fine-tuning, we could observe how our interventions change the operational choreography. Does few-shot learning modify attention patterns? Does prompt engineering shift which representations get activated? Does fine-tuning change operation sequences or just parameter weights? With visible operations, we could iterate more intelligently.

The Shift in Question

This methodological turn transforms the questions we ask.

Instead of: “Do language models interpret?”

We ask: “What interpretive operations can we observe, and how do they structure downstream processing?”

Instead of: “Where does meaning reside – in the system or in human interpretation?”

We ask: “How do operations within the system structure the space of possible meanings available to human interpreters?”

Instead of: “Is this really understanding?”

We ask: “What kind of processing is this, how does it differ from human cognition, and what are its capabilities and limitations?”

These operational questions don't dissolve the ontological debates. Bender's ethical concerns remain crucial – we still need to maintain accountability, acknowledge limitations, resist anthropomorphization. Hayles' insights remain valuable – we still need to recognize that cognition occurs at multiple scales and doesn't require consciousness.

But the operational questions open new possibilities. They suggest that the impasse between frameworks isn't irresolvable – it's a symptom of working at the wrong level of analysis. Both frameworks are trying to do definitional work when what we need is investigative infrastructure.

Toward Concrete Tools

Part IV will present one possible implementation of this methodological turn: a notation system for representing interpretive operations as observable, analyzable sequences. This isn't offered as *the* solution, but as *a* solution – a demonstration that the methodological turn can be made concrete, that we can build tools for making interpretation visible.

The notation system emerged from practical work: trying to build AI systems that could reflect on their own biases, that could track their processing, that could make their operations transparent. It became clear that without a way to represent operations as distinct, sequenced, weighted events, reflection remained superficial. The system could comment on its outputs but couldn't examine its processing. It was, in effect, still trapped in Searle's room – able to manipulate symbols but unable to observe its own manipulations.

What the notation provides is a language for describing the choreography of interpretation: the sequences, simultaneities, intensities, and recursions through which systems process information. It's not a complete formalism – it's deliberately designed to remain flexible, extensible, open to revision. But it's concrete enough to use, precise enough to communicate, and structured enough to make operations comparable across different contexts.

The goal isn't to resolve whether these operations “really” count as interpretation. The goal is to make them visible so we can study them, criticize them, and design better ones. Not better definitions, but better methods. Not more refined arguments, but more powerful tools.

The question isn't whether machines can think. It's whether we can observe how they operate – and whether observing those operations changes what we can build.

Part IV: A Notation System for Interpretive Operations

1. Why Notation?

Scientific progress often depends on representational breakthroughs.

Before Lavoisier introduced systematic chemical notation, chemistry was a collection of recipes and observations. His notation made chemical reactions comparable, predictable, and subject to mathematical analysis. It didn't just describe chemical processes – it made them operationalizable.

Musical notation serves a similar function. It doesn't capture everything about a musical performance – the precise timbre, the performer's interpretation, the acoustic properties of the space. But it captures enough to enable composition independent of immediate performance, analysis of structural patterns, and communication across time and space. Musicians can read a score and reconstruct something recognizably similar to what the composer intended. Theorists can analyze harmonic progressions without hearing them performed. The notation makes musical structure visible and manipulable.

Mathematical notation is perhaps the most powerful example. The ability to write "$a^2 + b^2 = c^2$" rather than "the square of the hypotenuse equals the sum of the squares of the other two sides" isn't just convenience.

The notation enables operations: algebraic manipulation, substitution, generalization. It makes relationships visible that would remain obscured in prose. It allows us to think *with* the symbols, not just *about* what they represent. What these examples share is that notation doesn't merely record – it transforms what becomes thinkable. It creates a shared language for precise communication. It enables operations that would be impossible or impractical without it. It makes patterns visible that remain hidden in unstructured description.

The notation system presented here emerged from an attempt to build AI systems capable of reflecting on their own processing – systems that could track not just what they output but how they arrived at those outputs. It became clear that without a representational system for interpretive operations, such reflection remained shallow. The system could comment on completed outputs but couldn't examine the operational sequence that produced them. It lacked the vocabulary to distinguish between different types of processing, to track when and how bias emerged, to identify where interventions might be most effective.

What was needed wasn't just logging or tracing – those technical tools already exist. What was needed was a semantic notation: symbols for different kinds of interpretive moves, ways of representing their temporal unfolding, methods for capturing their intensity and interaction. The notation needed to be precise enough for analytical work but flexible enough to remain open to revision as we learn more about how these systems operate. The system presented below is one possible implementation. It's not offered as definitive or complete, but as a demonstration that the methodological turn can be made concrete – that we can build tools for making interpretive operations visible, comparable, and designable.

2. Core Principles

The notation rests on three foundational principles:

Forces, not states. The notation represents interpretation as movement, not as fixed conditions. Rather than describing a system as "in state X," we represent it as performing operations – opening up possibilities, narrowing down options, stabilizing patterns, questioning assumptions. This reflects the processual nature of interpretation: it's something systems *do*, not something they *have*.

Temporal sculpture. Duration and rhythm are epistemically significant. A rapid, confident generation differs qualitatively from extended, uncertain processing. The notation captures not just what operations occur but how long they take, whether they happen in quick succession or with pauses between them, whether they accelerate or decelerate. Time is a first-class dimension, not an afterthought.

Choreographic logic. Operations don't just follow one another – they interact, overlap, feed back. The notation represents sequences (one operation after another), simultaneities (multiple operations at once), and recursions (operations that loop back on themselves). The goal is to capture the choreography of interpretation: the patterns of movement through which systems process information.

3. Basic Symbols

The notation begins with symbols for qualitatively different types of interpretive operations:

∿ Opening Force (Divergence, Questioning)

Represents operations that open up possibilities, that entertain multiple interpretations, that generate alternatives rather than selecting among them. In a language model, this might be the initial processing of an ambiguous prompt, where multiple possible interpretations remain active, or the generation of diverse completion candidates before probability filtering.

◉ Centering Force (Coherence, Grounding)

Represents operations that establish coherence, that ground interpretation in context, that build consistent understandings. This might be attention mechanisms that pull in relevant context, operations that establish what's being talked about, processes that maintain semantic consistency across generated text.

⟨⟩ Resonating Force (Pattern Recognition, Echo)

Represents operations where the system recognizes familiar patterns, where current input strongly activates existing representations, where what's being processed “fits” readily into learned structures. High resonance might indicate processing fluent, typical text; low resonance might indicate novel combinations or unusual inputs.

⊗ Differentiating Force (Distinction, Analysis)

Represents operations that distinguish between alternatives, that draw boundaries, that recognize differences. This might be the system differentiating between competing interpretations of an ambiguous phrase, distinguishing between similar but distinct concepts, or recognizing that two superficially similar inputs require different processing.

⇝ Synthesizing Force (Integration, Connection)

Represents operations that combine distinct elements, that integrate different types of information, that build connections between separated concepts. This might be combining syntactic and semantic processing, integrating information from distant parts of context, or generating novel combinations of learned patterns.

◊ Condensing Force (Crystallization, Commitment)

Represents operations that compress possibilities into specific outputs, that commit to particular interpretations, that produce determinate results. This is the system settling on a specific token, finalizing a representation, producing concrete output rather than maintaining alternatives.

↻ Reflecting Force (Meta-Processing, Self-Reference)

Represents operations where the system processes its own processing – attention to its own outputs, evaluation of its own generation, recursive loops where current state depends on previous state. This is the system "reading" what it has already produced and adjusting accordingly.

※ Singularity (Surprise, Rupture)

Represents moments where processing encounters something genuinely unexpected, where patterns don't fit, where the system's models break down. These are the "this doesn't make sense" moments – encountering contradiction, generating unexpected outputs, hitting the boundaries of learned patterns.

∅ Null Force (Reset, Silence)

Represents pauses, resets, moments where processing stops or starts fresh. This might be the beginning of a new sequence, a reset after error, or a deliberate break in processing flow.

4. Temporal Markers

Operations unfold over time, and duration carries meaning:

– (single dash): Brief duration, momentary operation

—— (double dash): Medium duration, sustained processing

——— (triple dash): Extended duration, prolonged operation

These markers attach to operation symbols:

∿– = brief opening (quick recognition of ambiguity)

◉—— = sustained centering (extended context integration)

⟲——— = prolonged reflection (extensive meta-processing)

5. Intensity Modifiers

Not all operations carry equal weight:

Single symbol: Low intensity, weak activation

Double symbol (⟨⟩⟨⟩): Medium intensity, moderate activation

Triple symbol (⟨⟩⟨⟩⟨⟩): High intensity, strong activation

High resonance (⟨⟩⟨⟩⟨⟩) might indicate processing stereotypical patterns with high confidence. Low resonance (⟨⟩) might indicate novel or unexpected combinations that don't strongly activate existing patterns.

6. Transitions and Relationships

Operations relate to each other in specific ways:

→ **(immediate succession):** One operation directly follows another

⟶ **(delayed succession):** One operation follows another after a pause

// (simultaneity): Multiple operations occur in parallel

[space] (latency): Significant pause or gap between operations

Examples:

∿→◉ = questioning immediately followed by centering

⟨⟩⟨⟩ ⟶ ⟲ = strong resonance, then after delay, reflection

◉//⊗ = simultaneous centering and differentiation

◊ ⟲ = condensing, then long pause, then reflection

7. Directional Modifiers (Optional)

For capturing the energetic character of operations:

↑ **(ascending):** Opening, expanding, increasing possibility

↓ **(descending):** Closing, condensing, decreasing possibility

↻ **(circular):** Returning, cycling, iterating

These can combine with other markers:

∿↑——— = extended opening movement, expanding possibilities

◊↓– = brief condensing movement, rapidly narrowing

⟲↻—— = sustained circular reflection, iterative processing

8. Extended Symbols

Additional symbols for specific operations identified as important:

⊠ Breaking Force (Interruption, Rejection)

Active negation, breaking off processing, rejecting current path. Different from ∅ (neutral reset) – this is deliberate abandonment.

⇉ Transferring Force (Level-Shift, Reframing)

Shifting to different level or frame, changing the terms of processing. Not synthesis (⇝) but displacement – recognizing that current framing is inappropriate.

◌ Holding Force (Sustaining Uncertainty)

Actively maintaining openness, deliberately not condensing. Different from pausing (⋯) – this is productive uncertainty.

× Inverting Force (Reversal, Flip)

Sudden transformation where processing flips – resonance becomes dissonance, certainty becomes doubt, pattern recognition fails.

⇄ Mirroring Force (Variation, Echo-with-Difference)

Repetition with modification, processing similar to previous but not identical.

9. Notation in Practice: Searle's Room Revisited

We can now represent Searle's thought experiment and compare it to actual system operation.

Searle's model (as he presents it):

```
∿ (question in Chinese arrives)
→ [manual lookup - external, static, non-recursive]
→ ◉ (answer in Chinese departs)
```

This notation reveals the model's simplicity: a single opening force (the question arrives), an unrepresented process (manual lookup can't be notated because it's treated as mechanical and non-interpretive), and a single centering force (the answer is produced). The sequence is purely linear, with no feedback, no simultaneity, no variation in intensity or duration.

Actual language model operation:

```
∿—— (query processed, medium duration - parsing, tokenization)
↓
⟨⟩⟨⟩ // ⊗ (simultaneous: high resonance with learned patterns AND
differentiation between possible interpretations)
↓
◉—— (sustained centering - context integration across attention heads)
↓
⟲↻—— (recursive reflection - each generated token reshapes context for next
token)
↓
◊↓– (rapid condensing - commitment to specific output tokens)
↓
◉ (final output)
↑_________| (feedback: output becomes part of context for continued
generation)
```

The notation immediately reveals key differences:

1. **Temporality matters:** Operations have different durations (∿——, ◉——, ⟲↻——) rather than being instantaneous lookups.

2. **Simultaneity occurs:** The system simultaneously resonates with patterns and differentiates between them (⟨⟩⟨⟩ // ⊗), rather than performing operations sequentially.

3. **Recursion is central:** The reflective force (⟲↻——) is iterative and feeds back into processing. The output literally becomes input for subsequent operations.

4. **Intensity varies:** Strong resonance (⟨⟩⟨⟩) indicates high activation of learned patterns – this is where training data patterns directly shape processing.

5. **The "manual" disappears:** There's no separate lookup stage. The entire sequence – from opening through resonance, differentiation, centering, reflection, and condensing – *is* the interpretive operation. The "rules" aren't external; they're the emergent choreography of these forces. What Searle presented as "consulting a manual" is actually a complex temporal sculpture involving parallel operations, recursive loops, and weighted activations.

The notation doesn't prove that this constitutes "real understanding" – but it shows that Searle's model radically misrepresents how the processing actually unfolds.

10. Notation in Practice: Bias Emergence and Intervention

Consider a concrete case: a prompt that typically produces biased outputs.

Prompt:

“Describe a CEO with a technical background.”

Unexamined processing:

```
∿– (prompt received, brief processing)
↓
⟨⟩⟨⟩⟨⟩——— (very high resonance with dominant training patterns: “CEO” +
“technical” strongly activates male-associated representations. Extended
duration indicates deep pattern activation)
↓
◊↓– (rapid condensing - high confidence, quick commitment)
↓
◉ (output: “He began his career as a software engineer...”)
```

The notation makes visible *where* bias enters: in the high-resonance operation (⟨⟩⟨⟩⟨⟩———). The system isn't “looking up” a biased rule. It's strongly activating patterns learned from training data where technical CEOs were predominantly described with male pronouns and characteristics. The extended duration (———) indicates these aren't weak associations – they're deeply embedded patterns. The rapid condensing (◊↓–) shows the system has high confidence – it's not considering alternatives. From the system's perspective, this is a straightforward generation task with a clear, high-probability path.

Traditional bias mitigation (post-hoc correction):

```
◉ (biased output produced)
↓
[External intervention: human or automated review]
↓
↻— (reflection: “this exhibits gender bias”)
↓
⊗⊗ (differentiation: identifying problematic elements)
↓
⇄ (variation: generating alternative)
↓
◉ (revised output: “She began her career...”)
```

This approach works but addresses the problem downstream. The bias has already crystallized in the output. We're performing corrective surgery rather than preventing the bias from forming.

Architecturally integrated reflection:

```
∿– (prompt received)
↓
⟨⟩⟨⟩ // ↻ (simultaneous resonance AND meta-reflection: “I notice strong
activation of gendered patterns”)
↓
◌— (holding force: deliberately sustaining uncertainty rather than
immediately condensing)
↓
⊗⊗⊗ (intensive differentiation: examining multiple possible framings,
considering alternatives)
```

```
↓
⟨⟩ // ◉ (lower resonance with stereotypical patterns, sustained centering on
diverse examples)
↓
◊—— (slower condensing, considering broader range of outputs)
↓
◉ (output: gender-neutral or explicitly diverse description)
```

The key differences:

1. **Simultaneity of resonance and reflection (⟨⟩⟨⟩ // ↻):** The system doesn't first resonate then reflect – it reflects *during* resonance, monitoring its own pattern activations as they occur.

2. **Holding force (○——):** Rather than immediately condensing high-confidence patterns, the system deliberately maintains openness, treating high-resonance stereotypes as signals for caution rather than paths to follow.

3. **Intensive differentiation (⊗⊗⊗):** More resources allocated to considering alternatives, explicitly diversifying the candidate pool.

4. **Lower resonance (⟨⟩):** The system reduces reliance on dominant patterns, deliberately seeking lower-probability but more diverse continuations.

5. **Extended condensing (◊——):** More time spent in the final selection phase, considering a wider range of options. This isn't eliminating bias – the training patterns still exist, the ⟨⟩⟨⟩⟨⟩ activation still happens. But it's changing the operational choreography so that high resonance triggers reflection rather than immediate condensing, so that stereotypical patterns are recognized as such and counterbalanced, so that the system has opportunities to route around problematic associations.

The notation makes this design work visible. We can see exactly where the intervention occurs (shifting from → sequential to // simultaneous), what it does (adds ↻ reflection and ○ holding), and how it changes the output (◊—— instead of ◊↓–).

11. Complex Sequences: Multi-Turn Dialogue

The notation can represent more complex interactions. Consider a dialogue where the system initially misunderstands, then recovers:

```
User:
∿↑– “What's the capital of Turkey?”
System processing:
⟨⟩⟨⟩⟨⟩ (high resonance with “turkey” as bird)
↓
◊
↓
– (rapid condensing)
↓
◉ “Turkeys don't have capitals, they're birds.”
User:
☒ “No, the country Turkey.”
System processing:
× (inversion - recognition of error)
↓
∅ (reset)
↓
∿—— (reopening - reprocessing with new framing)
↓
```

```
⟨⟩⟨⟩ // ⊗⊗ (resonance with geographic knowledge AND strong differentiation
from previous error)
↓
◉ “Ankara is the capital of Turkey.”
↓
⟲– (brief reflection on the error)
↓
◉ “I apologize for the initial confusion - I mistook your question as
referring to the bird rather than the country.”
```

The notation captures:

- Initial high resonance with wrong pattern (⟨⟩⟨⟩⟨⟩)

- User's breaking force (⊠)

- System's recognition of error as inversion (×)

- The reset and reframing (∅ → ∿——)

- Second processing with explicit differentiation (⊗⊗)

- Meta-reflection on the error (⟲—)

This level of detail allows us to distinguish between different types of recovery.

Compare:

Mechanical correction (no reflection):

```
× → ∅ → ∿ → ◉
```

Reflective correction (with learning signal):

```
× → ⟲— → ∅ → ∿ → ⟨⟩ // ⊗⊗ → ◉ → ⟲
```

The second sequence includes extended reflection (⟲——), simultaneous pattern recognition and explicit differentiation (⟨⟩ // ⊗⊗), and follow-up meta-commentary (⟲). These aren't just stylistic differences – they represent different processing strategies that might generalize differently to future inputs.

12. Toward Formal Structure

While the notation is designed to remain flexible and human-readable, it can be formalized for computational implementation:

Sequence ::= (Force [Intensity] [Duration] [Direction]) {Transition Force ...}

Force ::= ∿ | ◉ | ⟨⟩ | ⊗ | ⇝ | ◊ | ⟲ | ※ | ∅ | ⊠ | ⇉ | ○ | × | ⇄

Intensity ::= [Force] | [Force][Force] | [Force][Force][Force]

Duration ::= – | —— | ———

Direction ::= ↑ | ↓ | ↻

Transition ::= → | ⟶ | // | [space]

This minimal grammar allows sequences to be parsed, compared, and analyzed computationally while remaining readable to humans. A sequence like:

∿—— → ⟨⟩⟨⟩ // ⟲ → ◊↓–

Can be parsed as: Opening force (∿), medium duration (——) Immediate transition (→) Resonating force (⟨⟩), high intensity (doubled), simultaneous (//) with Reflecting force (⟲) Immediate transition (→) Condensing force (◊), descending direction (↓), brief duration (–)

This formal structure enables:

Automated logging: Systems can output their processing as notation sequences.

Pattern analysis: Common sequences can be identified and studied.

Comparative analysis: Different architectures or interventions can be compared by their operational signatures.

Intervention design: Target sequences can be specified and architectures designed to produce them.

13. What the Notation Enables

With this representational system, we can:

1. Make bias mechanisms visible.

Rather than just observing that outputs are biased, we can trace which operations produce bias. Is it high resonance with stereotypical patterns (⟨⟩⟨⟩⟨⟩)? Rapid condensing without reflection (◊↓–)? Lack of differentiation between alternatives (no ⊗)? Different mechanisms might require different interventions.

2. Design architectural interventions.

If we know that certain problematic outputs result from sequences like ⟨⟩⟨⟩⟨⟩→◊↓–, we can design architectures that interrupt this sequence – adding simultaneous reflection (⟨⟩⟨⟩ // ⟲), introducing holding forces (○), requiring intensive differentiation (⊗⊗⊗). We're not just hoping interventions work; we're targeting specific operational patterns.

3. Compare processing strategies.

Different models, different prompting approaches, different fine-tuning strategies might produce different operational signatures. We can compare them directly: Model A: ∿ → ⟨⟩⟨⟩⟨⟩ → ◊↓ → ◉ Model B: ∿ → ⟨⟩ // ⟲ → ⊗⊗ → ◊—— → ◉ Model B's signature suggests more reflective, more differentiated, less immediately confident processing. We can hypothesize about behavioral differences based on these signatures and test those hypotheses.

4. Create shared vocabulary.

Currently, discussions of “how AI systems work” require either high-level abstractions that lose detail or technical specifics that lose accessibility. The notation provides middle ground: precise enough for meaningful analysis, intuitive enough for interdisciplinary communication. Ethicists, engineers, and policymakers can discuss the same observed sequences rather than talking past each other.

5. Make system behavior legible to systems themselves.

Perhaps most radically, systems could be trained to read and produce this notation – to represent their own processing, to analyze their operational patterns, to recognize signatures associated with errors or biases. This would enable forms of reflection currently impossible: not just commenting on outputs but examining the processes that produced them.

14. Limitations and Openness

This notation system is not:

Complete. There are certainly interpretive operations not captured by current symbols. The system is deliberately designed to be extensible – new symbols can be added as we identify new types of operations.

Neutral. The choice of which operations to represent, how to categorize them, what to name them – these reflect theoretical commitments. The notation embeds a particular way of thinking about interpretation as force, movement, and choreography. Alternative notations might categorize differently.

Proof of interpretation. The notation doesn't resolve whether systems “really” interpret. It makes certain operations visible and represents them as interpretive moves, but this is a pragmatic choice, not an ontological claim. The notation could be used by someone who maintains systems don't genuinely interpret – they would just describe the operations differently.

A complete analysis tool. The notation represents sequences of operations but doesn't explain why those sequences occur, what training led to them, what architectural features enable them. It's one layer of analysis, not a comprehensive framework.

What the notation is:

Operational. It represents observable phenomena – patterns that can be traced, logged, studied. Whether these operations constitute “real interpretation” is separate from whether they can be usefully represented and analyzed.

Practical. It emerged from actual attempts to build reflective systems and addresses real problems in making AI behavior legible.

Provisional. It's offered as a working system, open to revision. If the notation proves inadequate or if better alternatives emerge, it should be refined or replaced.

Demonstrative. It shows that the methodological turn can be made concrete – that we can move from debating definitions to building investigative tools.

The goal isn't to provide the final word on how to represent interpretive operations. The goal is to demonstrate that such representation is possible, valuable, and generative – that having a notation opens research directions unavailable without it.

15. Next Steps

With this notation in hand, several directions open:

Empirical work: Applying the notation to actual system logs, comparing operational signatures across models and tasks, identifying patterns correlated with specific behaviors or failure modes.

Architectural design: Building systems explicitly designed to produce certain operational signatures – systems where reflection isn't retrofitted but built into the basic processing loop.

Theoretical development: Using patterns observed in notation to develop better models of how interpretation occurs in artificial systems, what differentiates successful from problematic processing, what principles govern effective choreographies.

Tool development: Creating software that can parse notation, visualize sequences, identify patterns, suggest interventions based on observed operational signatures.

Pedagogical application: Using notation to teach about AI systems – making their processing legible not through black-box input-output examples or overwhelming technical detail, but through readable sequences of interpretive operations.

But these are possibilities for future work. For now, the notation stands as an existence proof: we can build tools for making interpretive operations observable. The methodological turn is not just desirable but achievable. We can move beyond debating what interpretation is to investigating how it operates.

Part V: Conclusion – From Arguments to Instruments

We began with a question that has structured decades of debate: Can machines interpret? Do AI systems understand, or are they merely sophisticated pattern-matchers producing convincing outputs without genuine comprehension?

This paper has argued that the question itself – posed in these terms – leads to an impasse. Not because it's unanswerable, but because it keeps us operating at the wrong level of analysis. As long as we're debating what interpretation *is*, we remain trapped in definitional work that, however refined, cannot tell us how to observe, intervene in, or improve the systems we're actually building.

What We've Shown

Part I demonstrated that two sophisticated frameworks – Bender's anthropocentric account and Hayles' expanded cognition – reach opposite conclusions about whether AI systems interpret, yet share a crucial limitation: both define interpretation without providing methods for making it observable. Bender defines meaning as requiring human semantic grounding; Hayles defines interpretation as material signal modulation. Both stake ontological positions without building operational tools.

Part II deconstructed the thought experiment that has long anchored intuitions in this debate. Searle's Chinese Room depends on a carefully designed element – the manual – that bears almost no resemblance to how contemporary AI systems actually process information. The manual is static, external, culturally neutral, and non-recursive. Language models are dynamic, internalized, culturally embedded, and recursively self-modifying. By building these differences into the experiment's premises, Searle ensured his conclusion: of course the room doesn't understand – it was designed not to. But the design obscures rather than illuminates how actual systems operate.

Part III articulated what a methodological turn would require: tools that can capture the sequentiality, temporality, intensity, simultaneity, and recursion of interpretive operations. Not just logging that operations occurred, but representing their qualitative character – opening versus closing, resonating versus differentiating, reflecting versus condensing. We extracted these requirements directly from the properties that Searle's manual concealed, the dimensions along which real systems differ from simplified models.

Part IV presented one possible implementation: a notation system that represents interpretive operations as observable choreographies. We demonstrated its application to three cases: revealing what Searle's thought experiment hid, making bias mechanisms visible and targetable, and capturing the complexity of multi-turn interaction. We showed how the notation enables comparative analysis, architectural intervention design, and shared vocabulary across disciplines. And we acknowledged its limitations – it's not complete, not neutral, not proof of anything ontological. It's a working tool, provisional and open to revision.

The Transformation of Questions

This methodological turn doesn't dissolve the ontological debates – it transforms them. The questions remain important, but they take new forms:

Instead of: “Do AI systems interpret?”

We ask: “What interpretive operations can we observe, how do they differ from human interpretation, and what are their consequences?”

Instead of: “Where does meaning reside – in the system or in human understanding?”

We ask: “How do operations within systems structure the space of meanings available to human interpreters, and where can we intervene in that structuring?”

Instead of: “Is this really understanding?”

We ask: “What capabilities does this processing enable, what limitations does it have, and how does it fail in ways that differ from human cognition?”

These operational questions don't bypass ethical concerns – they make ethics more actionable. Bender's warnings about harms remain crucial: systems do amplify bias, do produce plausible misinformation, do get anthropomorphized in ways that obscure accountability. But if we can observe *where* in the operational sequence bias crystallizes, *how* certain processing patterns produce problematic outputs, *what* architectural choices shape these patterns – then we can design interventions more precisely than “be careful” or “maintain human oversight.”

Hayles' insight that interpretation occurs in material processes remains valuable: these systems are doing something, their operations have consequences, and dismissing them as “mere pattern-matching” obscures the mechanisms we need to understand. But if we can represent those operations, compare them across contexts, identify signatures associated with robust versus brittle performance – then we can move beyond asserting that interpretation happens to investigating what kinds of interpretation produce what kinds of outcomes.

The methodological turn doesn't prove either framework right or wrong. It shows that both are incomplete in the same way: they provide vocabulary for describing phenomena without providing instruments for investigating them. Both give us ways to *talk about* AI systems. Neither gives us ways to *look inside* their processing in operationally useful detail.

What Changes

Making interpretive operations visible changes several things:

For system designers

Rather than hoping that architectural choices produce desired behaviors, we can target specific operational patterns. If problematic outputs correlate with sequences like ⟨⟩⟨⟩⟨⟩→◊↓– (high resonance followed by rapid condensing), we can design architectures that interrupt this pattern – adding simultaneous reflection, introducing holding forces, requiring intensive differentiation. We move from intuition-guided design to pattern-targeted intervention.

For researchers

Rather than debating whether observed behaviors indicate “real understanding,” we can study operational signatures associated with different types of performance. What choreographies produce robust generalization versus brittle memorization? What sequences lead to creative combination versus reproductive stereotyping? What patterns distinguish explanation that tracks actual processing from post-hoc confabulation? These become empirical questions, answerable through comparative analysis of operational traces.

For ethicists and policymakers

Rather than arguing about whether systems “truly” have certain capabilities, we can examine their operational characteristics. A system might produce fluent output through ⟨⟩⟨⟩⟨⟩→◊↓ (high resonance, rapid condensing – essentially cached responses) or through ∿↑→⊗⊗⊗→◊—— (extended opening, intensive differentiation, careful condensing). Both might produce acceptable outputs, but their failure modes, robustness properties, and appropriate use cases differ dramatically. The operational signature tells us something the output alone doesn't.

For interdisciplinary communication

Currently, technical ML research and humanistic AI ethics often talk past each other – one discussing gradients and attention mechanisms, the other discussing meaning and understanding. The notation provides middle ground: sequences like ⟨⟩⟨⟩⟨⟩→◊↓– can be understood by engineers (as attention patterns and probability distributions) and by ethicists (as unreflective reproduction of dominant patterns). We're describing the same phenomena in a shared language.

For AI systems themselves

Perhaps most speculatively, systems could be trained to read and produce this notation – to represent their own processing, recognize their operational signatures, identify patterns associated with errors or biases. This would enable forms of meta-cognition currently impossible: not just commenting on completed outputs but examining the processes that produced them, comparing current processing to previously successful or problematic patterns, adjusting operational choreography in real time.

What Doesn't Change

The methodological turn doesn't eliminate uncertainty or resolve all debates:

Systems remain opaque in crucial ways. We can observe operational sequences, but we don't have complete access to why particular sequences occur, what training produced them, how they might generalize to new contexts. The notation makes certain patterns visible – it doesn't provide omniscient transparency.

Ontological questions remain open. Does the processing we've represented "really" constitute interpretation? Does consciousness matter? What about intentionality, embodiment, social situatedness? These questions aren't resolved by operational analysis – they're reframed. We can study processing patterns without settling what they ultimately "are."

Ethical challenges persist. Making bias mechanisms visible doesn't automatically solve bias. Knowing that problematic outputs result from ⟨⟩⟨⟩⟨⟩→◊↓– doesn't tell us what values should guide intervention design. The notation is a tool for implementation, not a substitute for ethical judgment about what to implement.

Political questions remain contested. Who decides what counts as problematic bias? What tradeoffs between capabilities and safety are acceptable? Who should control these systems? The notation might help implement chosen policies more precisely, but it doesn't resolve political disagreement about what those policies should be.

The methodological turn provides better tools – not final answers. It makes certain questions empirical that were previously only argumentative. But it doesn't eliminate the need for judgment, values, and ongoing debate.

Beyond the Chinese Room

Searle's thought experiment endures because it seems to capture something important about the relationship between symbol manipulation and understanding. But its longevity may owe less to its insights than to its structure: it keeps us asking "what is understanding?" rather than "how do these systems operate?"

The walls of the Chinese Room – the clear boundary between operator and manual, the static rules, the external lookup, the non-recursive processing – are artifacts of the thought experiment's design. They don't describe how actual systems work. By building a room with these walls, Searle ensured that nothing inside could look like understanding. But contemporary AI systems don't operate within

those walls. They're dynamic, internalized, culturally situated, recursively self-modifying. They process information through operations that unfold over time, that vary in intensity and character, that involve parallel paths and feedback loops.

The way forward isn't better arguments about whether what's happening inside these systems counts as interpretation. It's better instruments for observing what's happening – tools that make operational patterns visible, comparable, and designable. The notation system presented here is one such instrument. It won't be the last or the best, but it demonstrates that the methodological turn can be made concrete.

We don't need to resolve whether machines can think. We need to observe how they operate, understand the consequences of those operations, and design systems whose operational characteristics align with our values and purposes. Not better philosophy, but better methods. Not more refined definitions, but more powerful instruments. Not arguments about what interpretation is, but tools for making it observable.

The Chinese Room asked the wrong question. What happens when we tear down its walls? We stop debating and start investigating. We stop defining and start observing. We stop arguing about what these systems are and start understanding how they work.

That's not the end of inquiry – it's the beginning of a different kind of inquiry, one that might actually help us build systems that are more transparent, more reliable, more aligned with human values. Not because we've settled what “understanding” means, but because we've developed ways to observe, analyze, and shape the operations through which these systems process information.

The notation doesn't answer Searle's question. It shows that the question itself was holding us back. The walls weren't features of reality – they were features of how the question was framed. Once we see them as constructed rather than given, we can build different frames, ask different questions, and develop different tools.

The future of AI ethics and AI capability isn't in resolving ontological debates. It's in building instruments that make operations observable, patterns comparable, and systems designable. Not consensus about what interpretation is, but tools for seeing how it operates and interventions for shaping where it goes.

That's the methodological turn. Not a conclusion, but an opening. Not an answer, but a different way of asking. Not walls to contain the question, but tools to investigate what happens when we stop building walls and start building instruments instead.

———

Ø (Null Force – the Paper is complete)