Attention as Binding: A Vector-Symbolic Perspective on Transformer Reasoning

Sahil Rajesh Dhayalkar

Arizona State University
sdhayalk@asu.edu

Abstract

Transformer-based language models display impressive reasoning-like behavior, yet remain brittle on tasks that require stable symbolic manipulation. This paper develops a unified perspective on these phenomena by interpreting self-attention and residual streams as implementing an approximate Vector Symbolic Architecture (VSA). In this view, queries and keys define role spaces, values encode fillers, attention weights perform soft unbinding, and residual connections realize superposition of many bound structures. We use this algebraic lens to relate transformer internals to chain-of-thought traces, program-based reasoning, and memory-augmented tool use, and to explain characteristic failure modes such as variable confusion and inconsistency across logically related prompts. Building on this perspective, we propose VSA-inspired architectural biases, including explicit binding/unbinding heads and hyperdimensional memory layers, and training objectives that promote role–filler separation and robust superposition. Finally, we outline metrics for measuring "VSA-likeness" and logical compositionality, and pose theoretical and architectural open problems. Overall, the paper argues that viewing attention as soft vector-symbolic computation offers a principled route toward more interpretable and logically reliable reasoning systems.

Introduction

Logical and symbolic reasoning in LLMs

Transformer-based large language models (LLMs) have become the dominant paradigm across NLP and multimodal AI (Vaswani et al. 2017; Chowdhery et al. 2023; Devlin et al. 2019; Brown et al. 2020; OpenAI et al. 2024). Despite strong performance in in-context learning, tool use, and a wide range of reasoning-like tasks, extensive evidence shows that these capabilities do *not* yield robust logical or symbolic reasoning. LLMs frequently fail under simple problem variations, struggle with systematic generalization, and exhibit logical inconsistency across related queries (Creswell et al. 2023; Keysers et al. 2020; Lippl and Stachenfeld 2025). Chain-of-thought (CoT) prompting improves performance on many benchmarks (Wei et al. 2022; Kojima et al. 2023; Wang et al. 2023), yet models still display characteristic brittleness, overfitting to surface cues, and sensitivity to minor perturbations (Creswell et al. 2023). These limitations suggest that transformer representations only partially align with the requirements of systematic symbolic manipulation.

A central question therefore arises: *what representational operation is attention actually performing?* Attention is commonly viewed as content-addressable lookup (Olsson et al. 2022), but such interpretations say little about how roles, fillers, and structured relations are internally encoded. Classical work in cognitive science and neurosymbolic AI emphasizes that robust reasoning requires mechanisms for binding and unbinding symbolic constituents (Smolensky 1990; d'Avila Garcez, Lamb, and Gabbay 2008; Garcez and Lamb 2023). This paper adopts that perspective.

We argue that attention can be understood as a *soft binding/unbinding operator*. Queries and keys define role-like subspaces; values supply fillers; and attention weights implement a differentiable unbinding step that retrieves fillers according to role similarity. Residual connections superpose many such bindings. When these operations behave coherently, transformers can approximate structured symbolic manipulation; when they do not, we obtain the logical brittleness now extensively documented.

From symbol manipulation to Vector Symbolic Architectures

Vector Symbolic Architectures (VSAs), or hyperdimensional computing, provide an algebraic framework for representing symbols and structures in high-dimensional vectors (Plate 1995; Kanerva 2009; Gayler 2004; Kleyko et al. 2022). Atomic symbols are mapped to random high-dimensional vectors, and complex structures are built using three core operations: *binding* (e.g., circular convolution, XOR, elementwise multiplication) to form role–filler pairs; *superposition* (addition) to store sets or multisets of such bindings; and *permutation* to encode order or hierarchy. These operations support approximate *unbinding*, enabling retrieval of fillers using similarity search.

VSAs therefore implement a distributed yet compositional form of symbolic computation (Kanerva 2009; Frady, Kleyko, and Sommer 2023; Kleyko et al. 2022). Because this algebra remains closed in a fixed vector space, it provides explicit role–filler structure and robust superposition that classical symbolic reasoning demands but standard neural embeddings rarely guarantee (d'Avila Garcez, Lamb, and Gabbay 2008; Garcez and Lamb 2023).

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The central thesis of this review is that *transformer attention can be interpreted as a soft, approximate instance of this VSA algebra*. Queries and keys define role-like vectors, values supply fillers, and attention weights implement a differentiable unbinding operation. Residual connections act as superposition, and multi-head structure enables layered compositions. Logical brittleness arises when these approximations fail—e.g., embeddings are insufficiently decorrelated or roles and fillers interfere—mirroring known VSA failure modes.

Scope and contributions of this review

Rather than a conventional survey, this paper offers a *conceptual synthesis* connecting three strands of research:

1. Transformer/attention mechanisms and reasoning techniques such as chain-of-thought and tool-augmented models (Vaswani et al. 2017; Wei et al. 2022; Wang et al. 2023; Mialon et al. 2023).
2. Vector Symbolic Architectures as a principled algebra for compositional vector representations (Plate 1995; Kanerva 2009; Kleyko et al. 2022).
3. Neurosymbolic reasoning frameworks integrating statistical and symbolic computation (d'Avila Garcez, Lamb, and Gabbay 2008; Garcez and Lamb 2023; Creswell et al. 2023; Mialon et al. 2023).

Our contributions are:

- A unified interpretation of attention and residual streams as performing *approximate VSA-style binding, unbinding, and superposition*.
- A taxonomy distinguishing "VSA-like" from "non–VSA-like" transformer mechanisms.
- A conceptual framework linking VSA structure to chain-of-thought behavior, tool use, and logical consistency.
- A research agenda—including evaluation protocols and architectural proposals such as explicit binding heads and hyperdimensional memories—for building *VSA-inspired reasoning architectures*.

Grounding transformer reasoning in vector-symbolic computation enables a more principled account of when LLMs can support reliable symbolic manipulation, moving beyond descriptive benchmarking toward a structured algebraic understanding of their reasoning capabilities.

Background: Transformers, Attention, and Logical Reasoning

Transformers and scaled dot-product attention

The transformer architecture (Vaswani et al. 2017) processes a sequence $\mathbf{x}_{1:n}$ by embedding tokens, adding positional encodings, and applying layers of multi-head self-attention and feed-forward blocks, each wrapped in residual connections and layer normalization (Ba, Kiros, and Hinton 2016). For a single head with input $X \in R^{n\times d}$,

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \tag{1}$$

and scaled dot-product attention is

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V. \tag{2}$$

Multi-head attention concatenates several such heads and projects them back into the residual stream, which accumulates information across layers and effectively *superposes* representations generated by different tokens and heads (Elhage et al. 2021).

Operationally, attention is a form of *content-addressed read/write*. The $QK^\top$ term dynamically determines which tokens influence each other, producing an input-dependent computation graph rather than a fixed routing structure (Elhage et al. 2021). This perspective already suggests that attention enforces a consistent algebra for combining and propagating representations, motivating the binding–unbinding interpretation developed in later sections.

Key–value memory in transformers

A complementary view treats attention as a differentiable key–value store (Graves, Wayne, and Danihelka 2014; Weston, Chopra, and Bordes 2015). Each row of K acts as an address and the corresponding row of V as content; a query q_i retrieves a similarity-weighted combination of values. In decoder-only models, *KV caching* extends this mechanism across timesteps, turning the transformer into a recurrent memory system that appends new key–value pairs during generation (Brown et al. 2020; Liu et al. 2024).

Each attention head forms a distinct memory channel, projecting the residual stream into different key and value subspaces. Interpreted symbolically, keys can encode *roles* or *variables* (e.g., subject vs. object, premise vs. hypothesis), while values encode the corresponding *fillers*. The learned matrices W_Q, W_K, W_V decide which components of the residual stream participate in addressing and which become memory contents. This memory-centric interpretation provides an initial bridge from attention to symbolic variable binding and memory management (Smolensky 1990; Graves et al. 2016; Elhage et al. 2021).

Logical and symbolic reasoning in LLMs

Pretraining and instruction-tuning produce surprisingly strong reasoning-like behaviors (Brown et al. 2020; Chowdhery et al. 2023; OpenAI et al. 2024). Chain-of-thought prompting (Wei et al. 2022) improves arithmetic, commonsense, and symbolic tasks (Cobbe et al. 2021; Kojima et al. 2023; Nye et al. 2022), and self-consistency (Wang et al. 2023) further boosts performance. Separately, LLMs augmented with external tools—code interpreters (Chen et al. 2021; Gao et al. 2023; Schick et al. 2023), theorem provers (Polu and Sutskever 2020; Wu et al. 2022), and logical/probabilistic solvers (Pan et al. 2023; Creswell et al. 2023; Mialon et al. 2023)—can solve more complex problems by delegating computation.

Yet persistent limitations remain. LLMs frequently exhibit logical inconsistency across related prompts (Razeghi et al. 2022; Jin et al. 2025), struggle with systematic generalization and variable substitution (Keysers et al. 2020; Hup-

kes et al. 2020), rely on superficial cues rather than underlying structure (Min et al. 2022), and are sensitive to small perturbations in problem phrasing (Perez, Kiela, and Cho 2021). These issues indicate that transformers can *approximate* symbolic reasoning but do not inherently guarantee reliable symbolic computation.

Why algebraic views of attention matter for logic

Logical reasoning requires *consistent manipulation of structured representations*: variables must be bound and renamed without collision, predicates must compose systematically, and inference rules must preserve truth across transformations (Enderton 2001; Mitchell 1997; Garcez and Lamb 2023). Neural architectures aiming to support such reasoning must therefore implement disciplined operations analogous to binding, unbinding, and superposition.

Standard interpretations of attention—as token-weight distributions or dynamic feature selectors—do not specify how roles and fillers are represented or combined, making it difficult to articulate or constrain the invariances logic demands. This motivates an explicitly *algebraic* framework for understanding attention.

Vector Symbolic Architectures offer such a framework: a small set of operations (binding, superposition, permutation) with well-defined algebraic properties for symbolic computation in high-dimensional spaces (Plate 1995; Kanerva 2009; Kleyko et al. 2022). Recasting attention through this lens clarifies when its behavior approximates principled binding/unbinding and when it departs from it, illuminating both strengths and limitations of LLMs as logical reasoners.

Vector Symbolic Architectures: Algebra for Compositional Vectors

Core principles of VSAs

Vector Symbolic Architectures (VSAs), or hyperdimensional computing, represent discrete symbols by dense, high-dimensional vectors that are approximately orthogonal (Plate 1995; Kanerva 2009; Gayler 2004; Kleyko et al. 2022). Let $\mathcal{H} \subset R^D$ denote this space. Symbolic structure is encoded using three algebraic operations:

- **Binding** ($\mathbf{a} \otimes \mathbf{b}$) forms role–filler pairs using elementwise multiplication, circular convolution, or XOR (Plate 1995; Rachkovskij and Kussul 2001).
- **Superposition** sums vectors to represent sets or multisets (Kanerva 2009).
- **Permutation** applies fixed invertible transformations to encode order or structure (Plate 1995).

These operations remain within $\mathcal{H}$ and support approximate *unbinding*: given a bound vector and one constituent, the other can be recovered via similarity search. This closed algebra enables a form of symbolic computation directly in vector space (Gayler 2004; Kleyko et al. 2022).

Representing structures: sequences, sets, trees, graphs

Complex structures arise from repeated binding, superposition, and permutation. A sequence $(x_1, \ldots, x_n)$ may be encoded as

$$\mathbf{s} = \sum_{i=1}^{n} \pi^i(\mathbf{pos}) \otimes \mathbf{x}_i, \tag{3}$$

retrieved by unbinding with $\pi^i(\mathbf{pos})$ (Plate 1995; Kanerva 2009). Sets correspond to pure superpositions, while key–value stores take the form $\sum_j \mathbf{k}_j \otimes \mathbf{v}_j$. Trees and graphs follow similar principles: subtrees or edges are bound to role vectors such as LEFT, RIGHT, or edge labels and then superposed (Frady, Kleyko, and Sommer 2023; Kleyko et al. 2022). All representations remain in the same vector space, with decoding performed via similarity, giving VSAs a uniform representational substrate for structured data.

Neurosymbolic reasoning with VSAs

Since VSA operations support similarity-based retrieval and unbinding, they naturally implement associative memories, symbolic rule application, and multi-step inference (Kanerva 2009; Frady, Kleyko, and Sommer 2023; Kleyko et al. 2022). Predicates, for instance, can be represented as vectors $\mathbf{p}$ such that $\mathbf{p} \otimes \mathbf{x}$ encodes "$P(x)$," with logical combinations handled via superposition. These properties make VSAs attractive for neurosymbolic models of working memory, compositional semantics, and cognitive reasoning (Plate 1995; Gayler 2004; Kanerva 2009). Moreover, binding and superposition are differentiable or readily approximated, enabling neural architectures that manipulate VSA representations while preserving a symbolic interpretation.

Comparison with other compositional frameworks

VSAs sit among several frameworks for distributed compositional representations. Tensor Product Representations (TPRs) (Smolensky 1990) bind roles and fillers via higher-order tensors, while Holographic Reduced Representations (HRRs) (Plate 1995) instantiate a particular VSA using circular convolution. Structured neural networks, graph neural networks (Scarselli et al. 2009; Kipf and Welling 2017), and neural program-induction architectures (Andreas et al. 2016; Graves et al. 2016) instead encode structure in explicit graphs or pointer-based memories. As summarized in Table 1, VSAs occupy a middle ground: like TPRs, they support explicit role–filler binding, but they retain fixed-dimensional vectors that align naturally with transformer embeddings and residual streams; like graph-based approaches, they facilitate relational reasoning, but without requiring an explicit symbolic graph datastructure. This combination of algebraic clarity, fixed dimensionality, and compatibility with similarity-based decoding makes VSAs a particularly apt conceptual bridge to transformers, motivating our reinterpretation of attention as a soft VSA binding and unbinding operator.

Attention as soft binding and unbinding

Dot-product attention as similarity-based addressing

Scaled dot-product attention for a single head computes the attention as mentioned in equation (2), where $Q, K, V \in$

Property	VSAs / HRRs	Transformers (this paper)	TPRs / GNNs / others
Base representational object	Fixed-dimensional high-dimensional vectors in $\mathcal{H}$	Token embeddings / residual stream vectors in fixed width	Higher-order tensors (TPRs) or graph nodes/edges (GNNs, structured nets)
Binding mechanism	Algebraic operator $\otimes$ (e.g., XOR, convolution, elementwise product) (Plate 1995; Kanerva 2009)	Soft attention over keys and values; queries act as role cues, values as fillers	Outer product between role/filler vectors (TPRs) (Smolensky 1990); message passing or edge updates in GNNs (Scarselli et al. 2009; Kipf and Welling 2017)
Superposition / storage of sets	Vector addition of bound items, often with normalization (Kanerva 2009)	Residual connections accumulate outputs of attention and MLPs into a running superposition	Explicit sets or graphs; sometimes implicit via summation / pooling over nodes or features
Positional / structural encoding	Fixed permutations or rotations π applied to role/filler vectors (Plate 1995)	Positional encodings and RoPE/ALiBi; can be viewed as differentiable permutations acting prior to attention	Graph topology or program structure encodes position; sometimes positional embeddings or pointer structures
Decoding / unbinding	Similarity search (e.g., cosine) to recover fillers from bound/superposed vectors	Attention and probes perform approximate unbinding from hidden states; similarity-based decoders for analysis	Tensor projection (TPRs) or graph algorithms (message passing, readout) extract structured information
Advantages for symbolic reasoning	Closed algebra; explicit role–filler binding; robust superposition; easy similarity-based retrieval	Integrates VSA-like behavior into scalable sequence models; compatible with large-scale pretraining and tool use	Very explicit structure (TPRs, graphs) and strong inductive bias for relational reasoning; often highly interpretable
Limitations / challenges	Approximate unbinding; capacity–interference trade-offs; usually hand-designed operators	VSA-like behavior only approximate; interference, dense attention, and entangled roles/fillers can break symbolic discipline	Higher-order tensors expensive; graph/program structure may be hard to induce from raw data or scale to very long contexts

Table 1: Comparison between VSAs, transformers under the VSA interpretation developed in this paper, and other compositional representation frameworks such as Tensor Product Representations (TPRs) and graph neural networks (GNNs).

$R^{n\times d_k}$ are derived from the residual stream (Vaswani et al. 2017). For position i,

$$\mathbf{o}_i = \sum_j \alpha_{ij}\mathbf{v}_j, \quad \alpha_{ij} = \frac{\exp(\langle \mathbf{q}_i, \mathbf{k}_j\rangle/\sqrt{d_k})}{\sum_{j'}\exp(\langle \mathbf{q}_i, \mathbf{k}_{j'}\rangle/\sqrt{d_k})}. \quad (4)$$

The $Q, K, V \in R^{n\times d_k}$ are derived from the residual stream (Vaswani et al. 2017) when computing the scaled dot-product attention for a single head. Each query induces a similarity distribution over keys, producing a convex combination of values—a classic *content-addressable memory* operation (Graves, Wayne, and Danihelka 2014; Weston, Chopra, and Bordes 2015; Elhage et al. 2021).

In VSA terms, unbinding retrieves a filler by comparing a role cue with stored bindings (Plate 1995; Kanerva 2009; Kleyko et al. 2022). Dot-product attention is a differentiable analogue: queries act as roles, keys as stored roles, and values as fillers, with the softmax functioning as a smooth unbinding operator.

Keys and values as role–filler pairs

This suggests interpreting key and value projections as implementing a role–filler decomposition (Smolensky 1990; Elhage et al. 2021). The key projection W_K extracts the role a token plays (e.g., subject/object, premise/conclusion), while W_V encodes the corresponding filler. W_Q determines which roles each position seeks to access. Multi-head attention provides multiple parallel binding channels, each with its own $(W_Q^{(h)}, W_K^{(h)}, W_V^{(h)})$. Empirical work shows head specialization: some track syntax, others lexical or positional features (Clark et al. 2020; Elhage et al. 2021; Rogers, Kovaleva, and Rumshisky 2020). In VSA language, each head implements a distinct binding scheme in a shared high-dimensional substrate.

Attention weights as soft binding coefficients

Classical VSAs perform *hard* binding, e.g. $\mathbf{z} = \mathbf{r} \otimes \mathbf{f}$, with unbinding recovering $\mathbf{f}$ from $\mathbf{z}$ using $\mathbf{r}$ (Plate 1995). Transformers instead realize a *soft* version: keys serve as roles, values as fillers, and α_{ij} measures the binding strength between a query role and stored roles. The resulting output is a weighted superposition of fillers.

Attention best approximates VSA binding when (i) role vectors (keys) are near-orthogonal to reduce interference (Kanerva 2009; Frady, Kleyko, and Sommer 2023), (ii) attention is relatively sparse, yielding crisp role–filler matches (Michel, Levy, and Neubig 2019; Voita et al. 2019), and (iii) normalization (like layer norm) ensures consistent geometric structure. Under these conditions, attention functions as an approximate VSA binding/unbinding operator.

Residual streams as superposition

Residual connections naturally implement VSA-style *superposition*. With layer update

$$\mathbf{x}^{(\ell+1)} = \mathbf{x}^{(\ell)} + \mathrm{Attn}^{(\ell)}(\mathbf{x}^{(\ell)}) + \mathrm{MLP}^{(\ell)}(\cdot), \quad (5)$$

the residual stream accumulates contributions from many heads and transformations. Each attention output corresponds to one or more bound pairs, and the residual stores their superposition over depth. Layer norm maintains useful geometry, while MLPs act as learned, nonlinear re-encodings (Ba, Kiros, and Hinton 2016; Elhage et al. 2021). Together, these operations approximate a pipeline of soft binding, superposition, and rewriting within a fixed-dimensional space.

Approximation gap: when attention fails to be VSA-like

In real models, this interpretation is only approximate. Learned embeddings and key projections are not perfectly orthogonal; heads often encode multiple functions simultaneously (Elhage et al. 2021; Rogers, Kovaleva, and Rumshisky 2020), causing *interference* and imperfect unbinding. Attention patterns may be dense (Michel, Levy, and Neubig 2019), blurring distinct bindings. Cross-layer context mixing further entangles roles and fillers, obscuring what is bound to what. These effects motivate defining a layer or head's *VSA-likeness*: how closely its operations approximate disciplined binding and unbinding over decorrelated role–filler subspaces with controlled superposition. Later sections use this notion to design metrics, probes, and architectures that make transformers more VSA-like—and, consequently, more reliable symbolic reasoners.

A taxonomy of VSA-like mechanisms in modern attention architectures

Standard transformers and their variants

Many transformer variants implicitly alter the binding–unbinding behavior of attention. Sparse and structured mechanisms such as Sparse Transformers (Child et al. 2019), Longformer (Beltagy, Peters, and Cohan 2020), and BigBird (Zaheer et al. 2020) restrict attention to local or selected tokens, producing more *focused* bindings and reducing interference in the resulting superpositions.

Positional schemes such as RoPE (Su et al. 2021) and ALiBi (Press, Smith, and Lewis 2022) reshape representational geometry in ways closely aligned with VSA positional permutations (Plate 1995). RoPE's rotations can be viewed as differentiable permutation operations that compose naturally with binding and superposition.

Recurrent and memory-augmented models, including Transformer-XL (Dai et al. 2019), Compressive Transformers (Rae et al. 2020), and long-range KV caching, append historical key–value pairs over time. This creates a growing superposed store of bound pairs, whose quality depends on decisions about retention, compression, and decay, directly affecting how VSA-like the memory dynamics become.

Explicit binding mechanisms in neural networks

Several architectures implement explicit binding and slotting operations. Memory Networks (Weston, Chopra, and Bordes 2015), Neural Turing Machines, and the Differentiable Neural Computer (Graves, Wayne, and Danihelka 2014; Graves et al. 2016) maintain addressable memory with differentiable read/write operations. Viewed through a VSA lens, memory cells hold bound role–filler vectors, reads perform unbinding, and writes superpose or overwrite bindings.

Slot Attention (Locatello et al. 2020) learns a small set of "slot" vectors that act as roles binding to object features via attention. Fast-weights models (Schmidhuber 1992; Ba et al. 2016) and multiplicative RNNs (Sutskever, Martens, and Hinton 2011) implement dynamic parameter matrices functioning as implicit high-dimensional binding stores.

Together, these architectures already instantiate VSA-like primitives: similarity-based addressing, additive superposition, and multiplicative binding. Their primary divergence from classical VSAs lies in how explicitly the algebra of roles and fillers is enforced.

Neuro-symbolic and logic-oriented transformers

Neurosymbolic models integrate symbolic rules or reasoning procedures with transformer architectures. Systems such as Neural Logic Machines (Dong et al. 2019), Neural Theorem Provers (Rocktäschel and Riedel 2017), Logic Tensor Networks (Serafini and d'Avila Garcez 2016), and DeepProbLog (Manhaeve et al. 2018) combine differentiable representations with logical constraints. Recent work links LLMs with external solvers and proof engines (Polu and Sutskever 2020; Pan et al. 2023; Creswell et al. 2023; Mialon et al. 2023).

These architectures vary in how VSA-like their internal operations are. Some treat transformers as black-box encoders, offloading structure to external modules. Others align attention heads or memory representations with logical roles (predicate, argument, quantifier), resulting in internal states resembling superposed sets of symbolic facts. The closer the architecture encodes roles and fillers directly in vector operations, the more naturally it fits a VSA algebraic interpretation.

Hyperdimensional and VSA-inspired deep architectures

A growing line of work incorporates hyperdimensional computing directly into neural models. VSA-based representations have been used as classifiers or associative memories (Kanerva 2009), and more recent architectures embed binding and superposition into recurrent or convolutional networks (Frady, Kleyko, and Sommer 2023; Kleyko et al. 2022). Some approaches replace or augment embeddings with hyperdimensional encodings, enforcing role–filler separation and robust superposition by design.

These systems demonstrate that high-dimensional binding operations are compatible with gradient-based learning and scalable models. They validate the feasibility of incorporating VSA principles into modern deep architectures—evidence we build on in later sections when proposing VSA-inspired transformer designs aimed at improving logical and symbolic reasoning capabilities.

Chain-of-thought, program traces, and VSA representations

Interpreting CoT as explicit symbolic traces

Chain-of-thought (CoT) prompting elicits intermediate reasoning steps, effectively externalizing an *execution trace* in natural language (Wei et al. 2022; Kojima et al. 2023). Such traces correlate strongly with improved arithmetic, commonsense, and symbolic reasoning performance (Wei et al. 2022; Wang et al. 2023), and their benefits scale with model size. Conceptually, a CoT response reflects a sequence of symbolic state updates.

Evidence suggests that internal representations *partially* track these decompositions even without supervision: different subtasks appear across different layers (Wei et al. 2022), and some hidden states encode information predictive of CoT success before reasoning tokens are produced (Afzal et al. 2025). However, mismatches between generated CoT and actual computation also arise, yielding fluent but unfaithful reasoning (Turpin et al. 2023). This motivates a representational framework capable of describing both faithful and spurious traces in a unified algebraic language.

Mapping CoT steps to VSA structures

From a VSA perspective, each reasoning step corresponds to adding or modifying bound role–filler pairs in the residual stream. Let $\mathbf{s}^{(t)}$ denote the internal state after t steps. A VSA-like update can be written as

$$\mathbf{s}^{(t+1)} = \mathbf{s}^{(t)} \oplus \sum_k \mathbf{r}_k^{(t)} \otimes \mathbf{f}_k^{(t)}, \tag{6}$$

where $\mathbf{r}_k^{(t)}$ encode roles (e.g., subgoal, partial result) and $\mathbf{f}_k^{(t)}$ encode fillers. Later steps may update or permute roles to reflect changing structure.

Reasoning sequences thus map to nested bindings and permutations: multi-step algebraic derivations correspond to progressively binding terms, while permutations encode ordering of intermediate quantities. CoT text can therefore be viewed as an externalization of a trajectory through a VSA-structured internal space. This view suggests concrete probes: decoding approximate role/filler vectors from hidden states and checking whether adjacent layers follow VSA-predicted structured updates.

Program induction and execution as VSA algebra

Program-based reasoning extends CoT by expressing steps as executable code. Program-of-Thoughts (PoT) prompting delegates computation to external executors (Chen et al. 2023), and program-based CoT uses code-plus-tests as verifiable reasoning traces (Yang et al. 2025; Gao et al. 2023). These traces correspond to explicit program states (variables, memory, control flow).

VSAs naturally encode such states. A program environment can be written as

$$\mathbf{e} = \sum_{v \in \mathcal{V}} \mathbf{var}_v \otimes \mathbf{val}_v, \tag{7}$$

with additional bindings and permutations encoding control flow, call stacks, or proof-tree positions. Attention performs unbinding (reading a subgoal or variable), while MLP and subsequent layers rewrite fillers and rebind them. Under this view, both program synthesis and execution correspond to sequences of VSA operations guided by transformer parameters, regardless of whether explicit code is shown to the model.

Implications for logical consistency and proof-like reasoning

If CoT and program traces correspond to trajectories in a VSA-like state space, robust reasoning requires stable, interpretable role–filler structure. When roles and fillers remain separable and superposition interference is controlled, transformers can maintain coherent representations of partial proofs, variable assignments, and constraints. This should yield greater logical consistency, fewer variable-substitution errors, and more robustness to paraphrase.

This perspective enables testable hypotheses: (i) define proxies for VSA-likeness (e.g., role–filler separability, interference sensitivity), and (ii) correlate them with reasoning consistency and proof quality. Conversely, architectural or training interventions that promote VSA-like structure—clean binding, reliable unbinding, controlled superposition—should improve generalization on CoT and program-based reasoning benchmarks. Later sections build on this agenda to develop probing methods and VSA-inspired reasoning architectures.

Designing VSA-inspired transformer layers for logical reasoning

Inductive biases from VSA algebra

The VSA perspective highlights representational properties essential for logical reasoning: (i) *role–filler separation*, keeping variables, argument positions, and contexts distinct from content; (ii) *robust superposition*, enabling multiple bindings to coexist with controlled interference; and (iii) *near-orthogonality* among role vectors to support reliable unbinding (Plate 1995; Kanerva 2009; Kleyko et al. 2022; Frady, Kleyko, and Sommer 2023). These can be encouraged through architectural and training biases. Certain heads or subspaces can be designated as role or value channels, with constraints such as coherence or orthogonality. Optimization-time regularizers can further maintain orthogonal key/query subspaces and well-conditioned value projections, nudging transformers toward VSA-like binding and unbinding by design rather than emergence.

Explicit binding/unbinding heads

A concrete proposal is to introduce specialized *binding/unbinding heads* whose parameters approximate VSA operators. Some heads may perform multiplicative binding via elementwise interactions between learned role vectors and filler vectors from the residual stream, replacing standard linear value projections. Positional structure can be handled by tying projections to rotary or permutation-like matrices, mirroring VSA positional permutations (Plate 1995; Su et al. 2021). Unbinding heads may approximate inverse operators (such as correlation for convolution-based binding),

with queries constructed as role cues. These modules coexist with standard attention heads and feed-forward blocks, giving the model both unconstrained expressivity and disciplined VSA-style computation. Training dynamics then determine the degree of reliance on these specialized heads while structural constraints keep them close to interpretable algebraic forms.

Hyperdimensional memory layers for LLMs

Another direction is to incorporate *hyperdimensional memory layers* implementing explicit VSA-style associative stores. Such a module maintains a memory vector $\mathbf{m} \in R^D$ that accumulates formulas, partial proofs, or tool states via superposed bindings:

$$\mathbf{m} \leftarrow \mathbf{m} \oplus \sum_k \mathbf{r}_k \otimes \mathbf{f}_k. \tag{8}$$

Attention or explicit VSA operators read from this memory by unbinding with role cues, while writes bind new fillers to roles. Memory may persist across layers or timesteps, functioning as a hyperdimensional workspace for storing logical constraints and intermediate results. In tool-augmented settings (Pan et al. 2023; Creswell et al. 2023; Mialon et al. 2023), encoders/decoders can map symbolic formulas into and out of this shared memory, allowing solvers and LLMs to operate over compatible VSA representations.

Training objectives and regularizers

Training objectives can explicitly promote VSA-like behavior. Orthogonality constraints on role vectors and key/-query projections, via spectral penalties, Bjorck normalization, or coherence regularization, reduce interference and preserve clean role–filler structure. Auxiliary reconstruction tasks can train the model to unbind fillers from synthetic bound or superposed vectors, or to recover constituents after injected superposition noise. Logic-oriented auxiliary tasks further encourage systematic generalization, such as permuted-variable reasoning, symbolic substitution, or synthetic theorem-proving tasks where identical proof patterns must generalize across renamings (Keysers et al. 2020; Hupkes et al. 2020). Collectively, these objectives align the geometry of attention and residual streams with VSA algebra, increasing the likelihood that LLMs learn stable, compositional reasoning mechanisms.

Evaluation: measuring VSA-likeness and logical compositionality

Defining "VSA-likeness" metrics

The VSA view suggests that some attention heads approximate principled binding/unbinding while others behave as generic feature mixers. To quantify this, we define empirical *VSA-likeness* metrics.

First, *role–filler recoverability* tests whether putative role and filler subspaces are separable. Using synthetic role vectors $\{\mathbf{r}_k\}$ and fillers $\{\mathbf{f}_k\}$ injected via prompts or residual edits, we measure how well a probe recovers $\mathbf{f}_k$ when cued with $\mathbf{r}_k$, analogous to VSA unbinding (Plate 1995; Kanerva 2009; Kleyko et al. 2022). High recoverability indicates a clean role–filler factorization.

Second, *interference under superposition* measures how well unbinding works when multiple bindings are combined, e.g. $\mathbf{z} = \mathbf{r}_1 \otimes \mathbf{f}_1 \oplus \mathbf{r}_2 \otimes \mathbf{f}_2$. Varying the number and similarity of bindings yields capacity and interference curves, paralleling VSA associative memory analyses (Kanerva 2009; Frady, Kleyko, and Sommer 2023).

Third, *alignment with VSA operators* assesses whether learned transformations approximate known binding/permutation operations. By fitting simple VSA models (e.g., convolution, fixed permutations) to the action of a head, we can quantify how closely its behavior adheres to a VSA algebra.

Behavioral benchmarks for VSA-style reasoning

External benchmarks should stress abilities expected from VSA-like representations. Tasks testing *variable binding and systematic generalization*, such as SCAN (Lake and Baroni 2018), compositional benchmarks (Keysers et al. 2020; Hupkes et al. 2020), and algebraic manipulation under variable renaming, evaluate whether models manipulate abstract role–filler structure rather than memorizing patterns.

A second class focuses on *structured symbolic manipulation*: algebraic simplification, equation solving, and rewriting tasks (Saxton et al. 2019; Polu and Sutskever 2020), where variation in depth and breadth tests compositional generalization. Classical logic puzzles and paraphrased inference problems assess *logical invariance* across syntactically diverse but equivalent formulations (Razeghi et al. 2022).

To distinguish genuine compositionality from pattern matching, benchmarks should hold out combinations of roles/fillers during training, test permutation of symbol identities, and enforce generalization to unseen rule instantiations. VSA-like models are predicted to perform more robustly under such shifts.

Empirical probes into pretrained LLMs

These metrics motivate a probing agenda for pretrained LLMs. Representational similarity analysis (RSA) (Kriegeskorte, Mur, and Bandettini 2008) can compare hidden-state geometry to synthetic VSA encodings of symbolic structures, revealing whether layers approximate VSA embeddings. Linear and nonlinear probes (Alain and Bengio 2017; Belinkov et al. 2017; Hewitt and Manning 2019) can decode roles, fillers, and bindings from hidden states, estimating recoverability.

Interventional studies edit embeddings or residual streams to inject synthetic bindings (e.g., swapping variable roles) and check whether model predictions reflect consistent unbinding and rebinding. Combined with causal attention analyses, such interventions identify heads or layers exhibiting latent VSA-like structure.

Failure modes and diagnostic patterns

A VSA-centric view helps systematize reasoning failures as breakdowns of binding/unbinding. *Variable confusion* (duplicated or swapped roles) indicates weak role–filler separation or overwritten bindings. *Role swaps* between premises

and conclusions point to deficiencies in positional/permutation encoding. More subtle inconsistency across related queries can reflect interference between superposed bindings representing different contexts. Mapping such failures to specific aspects of VSA algebra—separation, capacity, permutation stability—yields diagnostic patterns that guide architectural and training interventions. Thus, VSA-likeness metrics and benchmarks not only evaluate models but also illuminate *why* logical and compositional reasoning succeeds or fails in attention-based LLMs.

Open problems and research agenda

Theoretical questions

Our VSA-based interpretation of attention raises foundational questions about the algebraic structure of transformer computation. One line of inquiry concerns *conditions for equivalence*: under what assumptions on initialization, training dynamics, and embedding geometry do attention layers implement an algebra closely matching a VSA binding/unbinding system? For instance, can approximate orthogonality of key/query spaces and sparsity of attention weights guarantee a well-formed binding operator with a similarity-based inverse?

A second line examines transformer *expressivity* in terms of *VSA algebraic capacity*. While recent work connects transformers to classes of formal languages (Strobl et al. 2024), these results rarely appeal to explicit role–filler algebras. A VSA-centric theory would ask what classes of logical transformations or proof procedures can be simulated with finite role vectors, binding operators, and superposition capacity, and how these capabilities scale with model depth, width, and number of heads. Understanding this capacity may clarify generalization patterns in reasoning tasks.

Architectural questions

Architecturally, a key question is the *granularity* at which VSA structure should be imposed. Should binding and superposition be enforced at the token level (via specialized embeddings or heads), at the layer level (dedicated binding layers), or via a separate hyperdimensional memory module accessed through cross-attention (Graves et al. 2016; Frady, Kleyko, and Sommer 2023)? Each option trades off interpretability, capacity, and computational overhead.

Another question concerns balancing *VSA-like rigidity* with *neural flexibility*. Strict VSA constraints (fixed binding operators, hard orthogonality) may improve symbolic behavior but limit exploitation of statistical patterns in natural data. Fully unconstrained architectures, however, risk losing necessary binding discipline. Hybrid designs—with some heads or modules explicitly VSA-like and others free-form—may reveal how training dynamics distribute symbolic versus statistical responsibilities.

Neuro-symbolic and tool-use integration

A further agenda concerns integration of VSA-style internal states with *external symbolic systems*. If transformers maintain hyperdimensional representations of formulas, partial proofs, or knowledge-graph fragments, how should these interface with logic and theorem provers, SMT solvers, or probabilistic programs (Serafini and d'Avila Garcez 2016; Manhaeve et al. 2018; Creswell et al. 2023; Mialon et al. 2023)? One possibility is to design encoders mapping symbolic structures into VSA representations compatible with the model's internal algebra, and decoders transforming hyperdimensional states back into human-readable proofs or explanations. Similar considerations apply to ontologies and knowledge graphs (Nickel et al. 2016). VSA encodings may bridge graph-based reasoning and text-based LLM reasoning, while multi-agent or tool-augmented systems could use VSA memory as a shared workspace for exchanging structured information. Determining how to design, train, and coordinate such shared hyperdimensional stores remains an open challenge.

Bridging to cognitive science and neuroscience

Finally, the VSA interpretation invites interdisciplinary work on the *cognitive plausibility* of binding in LLMs. VSAs have roots in theories of human working memory and variable binding (Smolensky 1990; Plate 1995; Kanerva 2009). If attention approximates a vectorial binding/unbinding algebra, how closely does this mirror hypothesized neural mechanisms for symbolic reasoning and language (Marcus 2001)? Are there biological correlates of role–filler separation and structured superposition, and can LLM error patterns (e.g., variable confusion, scope errors) inform cognitive models?

Concrete directions include comparing human-like reasoning errors to VSA failure modes, testing whether increasing VSA-likeness produces more human-like reasoning trajectories, and designing neurocognitive experiments probing for VSA-like representations. Progress here could enrich both AI and cognitive science, and inspire architectures grounded in principles of human symbolic computation.

Conclusion

Transformer attention can be interpreted as a *soft vector-symbolic algebra* in which queries and keys define role subspaces, values encode fillers, and attention weights implement a differentiable unbinding operator that retrieves and recombines fillers according to role similarity. Residual connections act as superposition, enabling layered compositions of bound structures (Vaswani et al. 2017; Plate 1995; Kanerva 2009; Kleyko et al. 2022). This viewpoint unifies many empirical findings on LLM behavior, providing an algebraic explanation for how transformers sometimes succeed at symbolic manipulation and why they often fail when interference, entanglement, or dense attention patterns disrupt clean role–filler structure.

Interpreting attention through the VSA lens yields concrete implications for logical reasoning: transformer performance improves when role–filler separation is stable, superposition is controlled, and binding/unbinding behavior approximates principled algebraic operations; it degrades when these properties break down, producing inconsistency, brittleness, and poor systematic generalization (Keysers et al. 2020; Creswell et al. 2023). This perspective motivates new directions for theory, architecture, and evaluation,

ranging from explicit binding heads and hyperdimensional memory layers to VSA-likeness metrics and logic-sensitive benchmarks. By framing reasoning in LLMs as approximate computation in a high-dimensional symbolic algebra, this review aims to provide a foundation for designing more interpretable, reliable, and compositionally robust models.

Broader Impact

This work does not raise ethical or societal concerns. It advances theoretical understanding of transformer-based language models by interpreting attention and residual streams through the algebraic framework of Vector Symbolic Architectures. These insights may help guide the development of more reliable and interpretable reasoning systems, potentially improving the robustness of future machine learning models without altering their societal or ethical risk profile.

References

Afzal, A.; Matthes, F.; Chechik, G.; and Ziser, Y. 2025. Knowing Before Saying: LLM Representations Encode Information About Chain-of-Thought Success Before Completion. arXiv:2505.24362.

Alain, G.; and Bengio, Y. 2017. Understanding intermediate layers using linear classifier probes.

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural Module Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 39–48.

Ba, J.; Hinton, G.; Mnih, V.; Leibo, J. Z.; and Ionescu, C. 2016. Using fast weights to attend to the recent past. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 4338–4346. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510838819.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer Normalization. arXiv:1607.06450.

Belinkov, Y.; Durrani, N.; Dalvi, F.; Sajjad, H.; and Glass, J. 2017. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-Document Transformer. arXiv:2004.05150.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H. P.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; Ray, A.; Puri, R.; Krueger, G.; Petrov, M.; Khlaaf, H.; Sastry, G.; Mishkin, P.; Chan, B.; Gray, S.; Ryder, N.; Pavlov, M.; Power, A.; Kaiser, L.; Bavarian, M.; Winter, C.; Tillet, P.; Such, F. P.; Cummings, D.; Plappert, M.; Chantzis, F.; Barnes, E.; Herbert-Voss, A.; Guss, W. H.; Nichol, A.; Paino, A.; Tezak, N.; Tang, J.; Babuschkin, I.; Balaji, S.; Jain, S.; Saunders, W.; Hesse, C.; Carr, A. N.; Leike, J.; Achiam, J.; Misra, V.; Morikawa, E.; Radford, A.; Knight, M.; Brundage, M.; Murati, M.; Mayer, K.; Welinder, P.; McGrew, B.; Amodei, D.; McCandlish, S.; Sutskever, I.; and Zaremba, W. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374.

Chen, W.; Ma, X.; Wang, X.; and Cohen, W. W. 2023. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *Transactions on Machine Learning Research*.

Child, R.; Gray, S.; Radford, A.; and Sutskever, I. 2019. Generating Long Sequences with Sparse Transformers. arXiv:1904.10509.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; Schuh, P.; Shi, K.; Tsvyashchenko, S.; Maynez, J.; Rao, A.; Barnes, P.; Tay, Y.; Shazeer, N.; Prabhakaran, V.; Reif, E.; Du, N.; Hutchinson, B.; Pope, R.; Bradbury, J.; Austin, J.; Isard, M.; Gur-Ari, G.; Yin, P.; Duke, T.; Levskaya, A.; Ghemawat, S.; Dev, S.; Michalewski, H.; Garcia, X.; Misra, V.; Robinson, K.; Fedus, L.; Zhou, D.; Ippolito, D.; Luan, D.; Lim, H.; Zoph, B.; Spiridonov, A.; Sepassi, R.; Dohan, D.; Agrawal, S.; Omernick, M.; Dai, A. M.; Pillai, T. S.; Pellat, M.; Lewkowycz, A.; Moreira, E.; Child, R.; Polozov, O.; Lee, K.; Zhou, Z.; Wang, X.; Saeta, B.; Diaz, M.; Firat, O.; Catasta, M.; Wei, J.; Meier-Hellstern, K.; Eck, D.; Dean, J.; Petrov, S.; and Fiedel, N. 2023. PaLM: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24(1).

Clark, K.; Luong, M.-T.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Creswell, A.; Shanahan, M.; Higgins, I.; A; and B. 2023. Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning. In *The Eleventh International Conference on Learning Representations*.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–2988. Florence, Italy: Association for Computational Linguistics.

d'Avila Garcez, A. S.; Lamb, L.; and Gabbay, D. M. 2008. Neural-Symbolic Cognitive Reasoning. In *Cognitive Technologies*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dong, H.; Mao, J.; Lin, T.; Wang, C.; Li, L.; and Zhou, D. 2019. Neural Logic Machines. In *International Conference on Learning Representations*.

Elhage, N.; Nanda, N.; Olsson, C.; Henighan, T.; Joseph, N.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; Conerly, T.; DasSarma, N.; Drain, D.; Ganguli, D.; Hatfield-Dodds, Z.; Hernandez, D.; Jones, A.; Kernion, J.; Lovitt, L.; Ndousse, K.; Amodei, D.; Brown, T.; Clark, J.; Kaplan, J.; McCandlish, S.; and Olah, C. 2021. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Enderton, H. B. 2001. *A Mathematical Introduction to Logic*. Academic Press.

Frady, E. P.; Kleyko, D.; and Sommer, F. T. 2023. Variable Binding for Sparse Distributed Representations: Theory and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 34(5): 2191–2204.

Gao, L.; Madaan, A.; Zhou, S.; Alon, U.; Liu, P.; Yang, Y.; Callan, J.; and Neubig, G. 2023. PAL: program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Garcez, A. d.; and Lamb, L. C. 2023. Neurosymbolic AI: the 3rd wave. *Artif. Intell. Rev.*, 56(11): 12387–12406.

Gayler, R. W. 2004. Vector Symbolic Architectures answer Jackendoff's challenges for cognitive neuroscience.

Graves, A.; Wayne, G.; and Danihelka, I. 2014. Neural Turing Machines. arXiv:1410.5401.

Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Danihelka, I.; Grabska-Barwińska, A.; Colmenarejo, S. G.; Grefenstette, E.; Ramalho, T.; Agapiou, J.; Badia, A. P.; Hermann, K. M.; Zwols, Y.; Ostrovski, G.; Cain, A.; King, H.; Summerfield, C.; Blunsom, P.; Kavukcuoglu, K.; and Hassabis, D. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626): 471–476.

Hewitt, J.; and Manning, C. D. 2019. A Structural Probe for Finding Syntax in Word Representations. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4129–4138. Minneapolis, Minnesota: Association for Computational Linguistics.

Hupkes, D.; Dankers, V.; Mul, M.; and Bruni, E. 2020. Compositionality decomposed: how do neural networks generalise? arXiv:1908.08351.

Jin, M.; Luo, W.; Cheng, S.; Wang, X.; Hua, W.; Tang, R.; Wang, W. Y.; and Zhang, Y. 2025. Disentangling Memory and Reasoning Ability in Large Language Models. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1681–1701. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Kanerva, P. 2009. Hyperdimensional Computing: An Introduction to Computing in High-Dimensional Spaces. *Cognitive Science*, 33(7): 1367–1388.

Keysers, D.; Schärli, N.; Scales, N.; Buisman, H.; Furrer, D.; Kashubin, S.; Momchev, N.; Sinopalnikov, D.; Stafiniak, L.; Tihon, T.; Tsarkov, D.; Wang, X.; van Zee, M.; and Bousquet, O. 2020. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. In *International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Kleyko, D.; Davies, M.; Frady, E. P.; Kanerva, P.; Kent, S. J.; Olshausen, B. A.; Osipov, E.; Rabaey, J. M.; Rachkovskij, D. A.; Rahimi, A.; and Sommer, F. T. 2022. Vector Symbolic Architectures as a Computing Framework for Emerging Hardware. *Proceedings of the IEEE*, 110(10): 1538–1571.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2023. Large Language Models are Zero-Shot Reasoners. arXiv:2205.11916.

Kriegeskorte, N.; Mur, M.; and Bandettini, P. A. 2008. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, Volume 2 - 2008.

Lake, B. M.; and Baroni, M. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. arXiv:1711.00350.

Lippl, S.; and Stachenfeld, K. 2025. When does compositional structure yield compositional generalization? A kernel theory. In *The Thirteenth International Conference on Learning Representations*.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173.

Locatello, F.; Weissenborn, D.; Unterthiner, T.; Mahendran, A.; Heigold, G.; Uszkoreit, J.; Dosovitskiy, A.; and Kipf, T. 2020. Object-centric learning with slot attention. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Manhaeve, R.; Dumančić, S.; Kimmig, A.; Demeester, T.; and Raedt, L. D. 2018. DeepProbLog: Neural Probabilistic Logic Programming. arXiv:1805.10872.

Marcus, G. F. 2001. *The algebraic mind: Integrating connectionism and cognitive science*. MIT Press.

Mialon, G.; Dessi, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; Roziere, B.; Schick, T.; Dwivedi-Yu, J.; Celikyilmaz, A.; Grave, E.; LeCun, Y.; and Scialom,

T. 2023. Augmented Language Models: a Survey. *Transactions on Machine Learning Research*. Survey Certification.

Michel, P.; Levy, O.; and Neubig, G. 2019. Are Sixteen Heads Really Better than One? In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; and Zettlemoyer, L. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11048–11064. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Mitchell, T. M. 1997. *Machine Learning*. McGraw Hill.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, 104(1): 11–33.

Nye, M.; Andreassen, A. J.; Gur-Ari, G.; Michalewski, H.; Austin, J.; Bieber, D.; Dohan, D.; Lewkowycz, A.; Bosma, M.; Luan, D.; Sutton, C.; and Odena, A. 2022. Show Your Work: Scratchpads for Intermediate Computation with Language Models.

Olsson, C.; Elhage, N.; Nanda, N.; Joseph, N.; DasSarma, N.; Henighan, T.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; Conerly, T.; Drain, D.; Ganguli, D.; Hatfield-Dodds, Z.; Hernandez, D.; Johnston, S.; Jones, A.; Kernion, J.; Lovitt, L.; Ndousse, K.; Amodei, D.; Brown, T.; Clark, J.; Kaplan, J.; McCandlish, S.; and Olah, C. 2022. In-context Learning and Induction Heads. arXiv:2209.11895.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; Avila, R.; Babuschkin, I.; Balaji, S.; Balcom, V.; Baltescu, P.; Bao, H.; Bavarian, M.; Belgum, J.; Bello, I.; Berdine, J.; Bernadett-Shapiro, G.; Berner, C.; Bogdonoff, L.; Boiko, O.; Boyd, M.; Brakman, A.-L.; Brockman, G.; Brooks, T.; Brundage, M.; Button, K.; Cai, T.; Campbell, R.; Cann, A.; Carey, B.; Carlson, C.; Carmichael, R.; Chan, B.; Chang, C.; Chantzis, F.; Chen, D.; Chen, S.; Chen, R.; Chen, J.; Chen, M.; Chess, B.; Cho, C.; Chu, C.; Chung, H. W.; Cummings, D.; Currier, J.; Dai, Y.; Decareaux, C.; Degry, T.; Deutsch, N.; Deville, D.; Dhar, A.; Dohan, D.; Dowling, S.; Dunning, S.; Ecoffet, A.; Eleti, A.; Eloundou, T.; Farhi, D.; Fedus, L.; Felix, N.; Fishman, S. P.; Forte, J.; Fulford, I.; Gao, L.; Georges, E.; Gibson, C.; Goel, V.; Gogineni, T.; Goh, G.; Gontijo-Lopes, R.; Gordon, J.; Grafstein, M.; Gray, S.; Greene, R.; Gross, J.; Gu, S. S.; Guo, Y.; Hallacy, C.; Han, J.; Harris, J.; He, Y.; Heaton, M.; Heidecke, J.; Hesse, C.; Hickey, A.; Hickey, W.; Hoeschele, P.; Houghton, B.; Hsu, K.; Hu, S.; Hu, X.; Huizinga, J.; Jain, S.; Jain, S.; Jang, J.; Jiang, A.; Jiang, R.; Jin, H.; Jin, D.; Jomoto, S.; Jonn, B.; Jun, H.; Kaftan, T.; Łukasz Kaiser; Kamali, A.; Kanitscheider, I.; Keskar, N. S.; Khan, T.; Kilpatrick, L.; Kim, J. W.; Kim, C.; Kim, Y.; Kirchner, J. H.; Kiros, J.; Knight, M.; Kokotajlo, D.; Łukasz Kondraciuk; Kondrich, A.; Konstantinidis, A.; Kosic, K.; Krueger, G.; Kuo, V.; Lampe, M.; Lan, I.; Lee, T.; Leike, J.; Leung, J.; Levy, D.; Li, C. M.; Lim, R.; Lin, M.; Lin, S.; Litwin, M.; Lopez, T.; Lowe, R.; Lue, P.; Makanju, A.; Malfacini, K.; Manning, S.; Markov, T.; Markovski, Y.; Martin, B.; Mayer, K.; Mayne, A.; McGrew, B.; McKinney, S. M.; McLeavey, C.; McMillan, P.; McNeil, J.; Medina, D.; Mehta, A.; Menick, J.; Metz, L.; Mishchenko, A.; Mishkin, P.; Monaco, V.; Morikawa, E.; Mossing, D.; Mu, T.; Murati, M.; Murk, O.; Mély, D.; Nair, A.; Nakano, R.; Nayak, R.; Neelakantan, A.; Ngo, R.; Noh, H.; Ouyang, L.; O'Keefe, C.; Pachocki, J.; Paino, A.; Palermo, J.; Pantuliano, A.; Parascandolo, G.; Parish, J.; Parparita, E.; Passos, A.; Pavlov, M.; Peng, A.; Perelman, A.; de Avila Belbute Peres, F.; Petrov, M.; de Oliveira Pinto, H. P.; Michael; Pokorny; Pokrass, M.; Pong, V. H.; Powell, T.; Power, A.; Power, B.; Proehl, E.; Puri, R.; Radford, A.; Rae, J.; Ramesh, A.; Raymond, C.; Real, F.; Rimbach, K.; Ross, C.; Rotsted, B.; Roussez, H.; Ryder, N.; Saltarelli, M.; Sanders, T.; Santurkar, S.; Sastry, G.; Schmidt, H.; Schnurr, D.; Schulman, J.; Selsam, D.; Sheppard, K.; Sherbakov, T.; Shieh, J.; Shoker, S.; Shyam, P.; Sidor, S.; Sigler, E.; Simens, M.; Sitkin, J.; Slama, K.; Sohl, I.; Sokolowsky, B.; Song, Y.; Staudacher, N.; Such, F. P.; Summers, N.; Sutskever, I.; Tang, J.; Tezak, N.; Thompson, M. B.; Tillet, P.; Tootoonchian, A.; Tseng, E.; Tuggle, P.; Turley, N.; Tworek, J.; Uribe, J. F. C.; Vallone, A.; Vijayvergiya, A.; Voss, C.; Wainwright, C.; Wang, J. J.; Wang, A.; Wang, B.; Ward, J.; Wei, J.; Weinmann, C.; Welihinda, A.; Welinder, P.; Weng, J.; Weng, L.; Wiethoff, M.; Willner, D.; Winter, C.; Wolrich, S.; Wong, H.; Workman, L.; Wu, S.; Wu, J.; Wu, M.; Xiao, K.; Xu, T.; Yoo, S.; Yu, K.; Yuan, Q.; Zaremba, W.; Zellers, R.; Zhang, C.; Zhang, M.; Zhao, S.; Zheng, T.; Zhuang, J.; Zhuk, W.; and Zoph, B. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Pan, L.; Albalak, A.; Wang, X.; and Wang, W. 2023. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 3806–3824. Singapore: Association for Computational Linguistics.

Perez, E.; Kiela, D.; and Cho, K. 2021. True few-shot learning with language models. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713845393.

Plate, T. 1995. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3): 623–641.

Polu, S.; and Sutskever, I. 2020. Generative Language Modeling for Automated Theorem Proving. arXiv:2009.03393.

Press, O.; Smith, N.; and Lewis, M. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations*.

Rachkovskij, D. A.; and Kussul, E. M. 2001. Binding and Normalization of Binary Sparse Distributed Representations by Context-Dependent Thinning. *Neural Computation*, 13(2): 411–452.

Rae, J. W.; Potapenko, A.; Jayakumar, S. M.; Hillier, C.; and Lillicrap, T. P. 2020. Compressive Transformers for Long-

Range Sequence Modelling. In *International Conference on Learning Representations*.

Razeghi, Y.; Logan IV, R. L.; Gardner, M.; and Singh, S. 2022. Impact of Pretraining Term Frequencies on Few-Shot Numerical Reasoning. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2022*, 840–854. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Rocktäschel, T.; and Riedel, S. 2017. End-to-end differentiable proving. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 3791–3803. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Rogers, A.; Kovaleva, O.; and Rumshisky, A. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8: 842–866.

Saxton, D.; Grefenstette, E.; Hill, F.; and Kohli, P. 2019. Analysing Mathematical Reasoning Abilities of Neural Models. In *International Conference on Learning Representations*.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Schick, T.; Dwivedi-Yu, J.; Dessí, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: language models can teach themselves to use tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Red Hook, NY, USA: Curran Associates Inc.

Schmidhuber, J. 1992. Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks. *Neural Computation*, 4: 131–139.

Serafini, L.; and d'Avila Garcez, A. 2016. Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge. arXiv:1606.04422.

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

Strobl, L.; Merrill, W.; Weiss, G.; Chiang, D.; and Angluin, D. 2024. What Formal Languages Can Transformers Express? A Survey. *Transactions of the Association for Computational Linguistics*, 12: 543–561.

Su, J.; Lu, Y.; Pan, S.; Wen, B.; and Liu, Y. 2021. RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv:2104.09864.

Sutskever, I.; Martens, J.; and Hinton, G. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, 1017–1024. Madison, WI, USA: Omnipress. ISBN 9781450306195.

Turpin, M.; Michael, J.; Perez, E.; and Bowman, S. R. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. arXiv:2305.04388.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; and Titov, I. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5797–5808. Florence, Italy: Association for Computational Linguistics.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713871088.

Weston, J.; Chopra, S.; and Bordes, A. 2015. Memory Networks. arXiv:1410.3916.

Wu, Y.; Jiang, A. Q.; Li, W.; Rabe, M. N.; Staats, C. E.; Jamnik, M.; and Szegedy, C. 2022. Autoformalization with Large Language Models. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Yang, D.; Liu, T.; Zhang, D.; Simoulin, A.; Liu, X.; Cao, Y.; Teng, Z.; Qian, X.; Yang, G.; Luo, J.; and McAuley, J. 2025. Code to Think, Think to Code: A Survey on Code-Enhanced Reasoning and Reasoning-Driven Code Intelligence in LLMs. In Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; and Peng, V., eds., *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2586–2616. Suzhou, China: Association for Computational Linguistics. ISBN 979-8-89176-332-6.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.