Symbolic Transformers: Separation of Symbolic Manipulation and Contextual Reasoning for Mechanistic Interpretability

Clayton Kerce
Georgia Tech Research Institute
clayton.kerce@gtri.gatech.edu

Alexis Fox
Georgia Tech Research Institute
alexis.fox@gtri.gatech.edu

June 9, 2025

Abstract

Transformer models achieve remarkable performance across language and vision tasks; however our understanding of how the core attention and feed-forward components interact to give rise to these capabilities is incomplete – from the process of next-token prediction to simulation of long-range complex reasoning. This opacity limits our theoretical understanding of transformer computation and deployment in applications requiring interpretable processing and auditable decision making. Toward a better understanding of such issues, **we introduce transformer architectures with explicit symbolic internal representations to provide insight into these fundamental mechanisms, presenting two complementary design points: Token-Factored Transformers that separate reasoning streams for mechanistic analysis, and Symbolic Transformers that constrain all representations to remain interpretable as vocabulary token mixtures**. The Token-Factored approach decomposes internal states as $X = X_T + X_E$, where attention mechanisms update symbolic representations (X_T) and feed-forward networks update contextual information (X_E). The Symbolic approach extends this with vocabulary manifold constraints ensuring guaranteed interpretability at computational cost. This progression enables investigation of attention-FFN coordination across the full spectrum of transformer capabilities, from basic token prediction to complex reasoning and decision-making, establishing foundations for both mechanistic understanding and trustworthy AI deployment.

1 Introduction

Transformer architectures have achieved remarkable success across machine learning domains, yet their complex internal representations, operating on pseudo-symbolic embeddings, make interpreting their information generation to opaque for many decision-making processes. This opacity creates a critical barrier for deploying these powerful models in safety-critical applications where understanding and verifying reasoning processes is essential [Danilevsky et al., 2024]. The challenge is particularly acute because transformers perform sophisticated token association and context explication through attention and feed-forward network components, but the coordination between these mechanisms remains poorly understood.

Recent mechanistic interpretability research has revealed distinct computational roles within transformers: attention mechanisms perform context-dependent information routing and symbolic manipulation, while feed-forward networks handle contextual computations and generate information not explicitly present in inputs [Zhang et al., 2024b, Olsson et al., 2022]. However, a fundamental question remains unanswered: **how do these components coordinate to produce**

complex reasoning, and can this coordination be made explicit and inspectable? This coordination mystery limits both our theoretical understanding of transformer capabilities and our ability to build trustworthy AI systems with verifiable reasoning processes.

Current explainable AI approaches face significant limitations when applied to transformers. Post-hoc methods like attention visualization and gradient-based attribution often provide inconsistent explanations and suffer from faithfulness concerns—whether attention weights truly reflect reasoning processes or merely correlate with outputs [Jain and Wallace, 2019, Wiegreffe and Pinter, 2019]. These methods cannot control reasoning mechanisms during training, test architectural hypotheses about component coordination, or provide guaranteed interpretability for novel inputs. A systematic review reveals a critical architectural gap: the absence of transformer models designed to natively support both symbolic and contextual reasoning through distinct yet interacting mechanisms [Colelough and Regli, 2025].

1.1 Our Approach: Architectures Designed for Interpretability

We address these limitations by presenting two complementary architectures that establish endpoints of an interpretable design space, providing a research pathway from mechanistic understanding to guaranteed symbolic preservation:

Token-Factored Transformers decompose internal states through explicit stream separation:

$$X = X_t + X_e \tag{1}$$

where X_t (token-like stream) maintains structured symbolic relationships updated by attention mechanisms, and X_e (embedding-like stream) captures contextual information updated by feed-forward networks. This factorization enables direct observation of distinct reasoning processes while maintaining transformer expressiveness.

Symbolic Transformers extend this foundation with architectural constraints ensuring all representations remain within vocabulary-constrained manifolds, providing guaranteed interpretability at computational cost. Channel-wise operations and vocabulary projections ensure that every intermediate representation admits direct interpretation as probabilistic mixtures of vocabulary tokens.

1.2 Contributions and Validation

Our approach shifts from “explainable AI” (interpreting black boxes) to “interpretable AI” (transparent by design). Key design decisions include the use of ALiBi positional encoding to preserve symbolic stream purity, Kronecker-lifted projections for efficient structured operations, and architectural constraints ensuring stream specialization. We establish a principled research progression from simpler stream separation to full symbolic preservation.

Contributions. This work makes several key contributions: (1) the first systematic transformer architecture family constructed with purely interpretable designs, (2) theoretical foundations connecting reasoning processes of symbolic manipulation and abductive reasoning to transformer computations, (3) detailed analysis of computational trade-offs between interpretability and efficiency, and (4) validation that reasoning stream separation maintains performance while enabling mechanistic insights. Our results provide foundations for interpretable-by-design architectures that maintain transformer power while providing transparency required for trustworthy AI deployment in critical applications.

2 Background and Problem Formulation

2.1 Transformer Component Roles in Pseudo-Symbolic Processing

Mechanistic interpretability research has revealed distinct roles for transformer components operating on pseudo-symbolic embedding representations. Attention mechanisms perform sophisticated context-dependent operations including entity tracking, syntactic parsing, and pattern completion through specialized circuits like "induction heads" that enable in-context learning [Olsson et al., 2022, ?]. These mechanisms demonstrate emergent symbolic processing capabilities through structured manipulation of embedding representations that preserve semantic relationships [Brinkmann et al., 2024].

Feed-forward networks function as differentiable key-value memories storing factual knowledge while performing context-independent computations [Zhang et al., 2024b]. FFNs generate information not explicitly present in inputs, implementing both factual recall and contextual hypothesis generation that supports symbolic reasoning processes. Recent analysis reveals that FFNs encode structured knowledge representations that can be decoded back to vocabulary-level interpretations [Nanda, 2023].

Despite understanding individual components, **their coordination in producing complex reasoning remains poorly understood**. Critical questions include: which information flows between attention and FFN components during symbolic reasoning processes, how symbolic manipulations integrate with contextual computations to produce interpretable outcomes, and whether these interactions can be architecturally separated while maintaining performance. The residual stream architecture creates shared communication channels, yet the nature of information exchange through these channels lacks systematic investigation for interpretable reasoning applications.

2.2 Current Interpretability Landscape and Limitations

Existing transformer interpretability approaches fall into two categories, each with significant limitations for producing explicit and inspectable reasoning. Post-hoc methods including linear probes [Hewitt and Liang, 2019], attention visualization [?], and gradient-based attribution [Ali et al., 2022] analyze trained models to infer reasoning processes. However, these methods face fundamental challenges: attention weights may not reflect true symbolic reasoning processes [Jain and Wallace, 2019], gradient explanations can be unstable across similar inputs, and linear probes may identify correlations rather than causal mechanisms underlying symbolic coordination.

The faithfulness problem is particularly concerning for applications requiring explicit symbolic reasoning. Whether explanations accurately reflect decision-making processes or merely correlate with outputs remains contested, undermining confidence in post-hoc interpretability for critical deployments [Danilevsky et al., 2024].

Intrinsic architectural modifications represent an alternative approach, with efforts including concept transformers [Rigotti et al., 2022] and knowledge-augmented architectures. However, these typically add interpretable components as auxiliary modules rather than redesigning core transformer reasoning processes. A systematic review reveals a critical gap: **no current architectures explicitly separate and coordinate distinct reasoning mechanisms within unified transformer frameworks** [Colelough and Regli, 2025].

2.3 Interpretability Requirements for Trustworthy Systems

We formalize interpretability requirements for critical applications through three key properties that architectures must satisfy:

Mechanistic Transparency: Internal representations and operations must admit direct interpretation in terms of symbolic reasoning processes. This requires architectural constraints ensuring that intermediate states preserve connections to vocabulary tokens and that operations maintain semantic interpretability.

Functional Separation: Distinct reasoning mechanisms (symbolic manipulation vs. contextual processing) must be architecturally separated to enable independent analysis and verification. This separation should emerge naturally from architectural constraints rather than requiring post-hoc analysis.

Computational Efficiency: Interpretability enhancements must not impose prohibitive computational overhead compared to standard transformers. This constraint ensures practical deployability while maintaining transparency benefits.

These requirements motivate our architectural approach: Token-Factored Transformers provide mechanistic transparency and functional separation at comparable computational cost, while Symbolic Transformers extend these benefits with guaranteed interpretability at additional computational expense. This progression enables selection of appropriate interpretability-efficiency trade-offs based on application requirements.

3 Theoretical Foundations

3.1 Symbolic Embedding Spaces and Preservation Mechanisms

We establish theoretical foundations for interpretable transformer architectures through formal analysis of symbolic preservation in high-dimensional embedding spaces. The key insight is that symbolic meaning corresponds to directional relationships in embedding space, while computational utility requires controlled magnitudes achieved through normalization operations.

Definition 3.1 (Vocabulary and Channel Structure)**.** Let $\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$ be a finite vocabulary of discrete tokens. The embedding space $\mathbb{R}^{d_{\text{hidden}}}$ admits canonical decomposition $\mathbb{R}^{d_{\text{hidden}}} = \bigoplus_{h=1}^{H} \mathbb{R}^{d_{\text{head}}}$ where H is the number of attention heads and $d_{\text{head}} = d_{\text{hidden}}/H$ is the head dimension.

Token embeddings $E \in \mathbb{R}^{|\mathcal{V}| \times d_{\text{hidden}}}$ provide the foundational symbolic representations, with row E_i corresponding to token v_i. Under channel decomposition, $E = [E^{(1)}, E^{(2)}, \ldots, E^{(H)}]$ where $E^{(h)} \in \mathbb{R}^{|\mathcal{V}| \times d_{\text{head}}}$ contains channel-specific embeddings enabling head-specialized symbolic processing.

Definition 3.2 (Ray-Based Token Representation)**.** For each token $v_i \in \mathcal{V}$ and head $h \in \{1, \ldots, H\}$, the symbolic ray is:

$$\mathcal{R}_i^{(h)} = \{\lambda E_{i,h} : \lambda > 0\} \subset \mathbb{R}^{d_{\text{head}}} \setminus \{0\} \tag{2}$$

where $E_{i,h} \in \mathbb{R}^{d_{\text{head}}}$ is the h-th channel embedding of token v_i.

This ray-based representation captures the insight that symbolic meaning resides in embedding directions rather than magnitudes, enabling preservation of token interpretability through normalization operations that maintain directional relationships while ensuring computational stability.

3.2 Stream Factorization Principle for Mechanistic Understanding

The core architectural innovation is decomposition of transformer internal states to enable explicit separation of reasoning processes:

$$X = X_t + X_e \tag{3}$$

where $X_t \in \mathcal{T}$ represents structured symbolic information and $X_e \in \mathcal{E}$ represents contextual modifications and hypothesis generation.

Design Principle 3.3 (Functional Separation Through Architectural Constraints)**.** Attention mechanisms update exclusively the symbolic stream X_t through structured operations preserving token-space connections. Feed-forward networks update exclusively the contextual stream X_e through operations that observe complete state but modify only contextual representations.

This separation draws theoretical support from dual-process cognitive theories distinguishing between fast, automatic System 1 processes and slow, deliberate System 2 processes [Bellini-Leite, 2022, Kahneman, 2011]. The mapping to transformer components is natural: attention mechanisms exhibit System 1-like rapid pattern matching and symbolic routing, while FFNs demonstrate System 2-like systematic knowledge application and hypothesis generation.

3.3 Design Space Characterization

Our approach establishes endpoints of an interpretable design space with clear computational tradeoffs:

Token-Factored Endpoint: Stream separation enables mechanistic understanding of attention-FFN coordination without full symbolic constraints. Computational requirements remain comparable to standard transformers while providing unprecedented insight into reasoning processes through explicit stream observation.

Symbolic Endpoint: Full vocabulary manifold constraints ensure guaranteed interpretability with every representation interpretable as token mixtures. This requires significant additional computation through vocabulary projections, creating memory-intensive operations that limit scalability.

Observation 3.4 (Computational Constraint Reality)*.* Vocabulary projection at every FFN creates $N_{\text{token}} \times N_{\text{token}}$ attention matrix requirements, imposing significant memory overhead. This constraint limits practical scalability of full symbolic approaches while motivating hybrid architectures combining standard and interpretable components.

The design space characterization enables principled selection of interpretability-efficiency tradeoffs: Token-Factored architectures for research applications prioritizing mechanistic understanding, Symbolic architectures for critical applications requiring guaranteed interpretability, and hybrid approaches for practical deployments balancing transparency with computational constraints.

3.4 Mathematical Properties and Guarantees

Property 3.5 (Stream Specialization Preservation)**.** Under architectural constraints, the symbolic stream X_t maintains interpretable connections to vocabulary tokens through all transformations, while the contextual stream X_e captures complementary information that enhances rather than replaces symbolic processing.

Property 3.6 (Computational Equivalence)**.** Token-Factored Transformers maintain identical asymptotic computational complexity to standard transformers: $O(T^2 d_{\text{hidden}} + T d_{\text{hidden}}^2)$ per layer, where stream bookkeeping adds only constant factors that do not affect dominant complexity terms.

These theoretical foundations provide mathematical justification for architectural choices while establishing formal guarantees about interpretability preservation and computational efficiency. The progression from Token-Factored to Symbolic architectures represents increasing levels of interpretability constraint with corresponding computational costs, enabling principled trade-off decisions based on application requirements.

4 Token-Factored Transformer: Foundation Architecture

4.1 Design Rationale and Implementation Pathway

The Token-Factored Transformer provides a pathway to understanding attention-based symbolic manipulation without the computational complications of full symbolic constraints. This design choice enables simpler implementation with comparable computational requirements to standard transformers while delivering mechanistic interpretability benefits through explicit stream separation. The architecture serves as both a research tool for exploring attention-FFN coordination and a practical foundation for interpretable AI applications.

4.2 Core Factorization Implementation

The Token-Factored Transformer decomposes internal hidden states $X \in \mathbb{R}^{n \times d_{\text{hidden}}}$ to enable explicit and inspectable symbolic component coordination:

$$X = X_t + X_e \tag{4}$$

where n is sequence length and d_{hidden} is hidden dimension. This factorization embodies principled separation of computational responsibilities operating on structured pseudo-symbolic representations.

The **Token-like Stream (X_t)** maintains interpretable connections to input vocabulary through structured operations preserving semantic token relationships. Initialized with vocabulary embeddings $X_t^{(0)} = \text{TokenEmbedding}(\text{input\_ids})$, this stream carries symbolic information that can be traced back to original tokens. Updates occur exclusively through attention mechanisms performing explicit symbolic reasoning through structured manipulation of token-space representations.

The **Embedding-like Stream (X_e)** captures abstract contextual information and generates hypotheses not explicitly present in input tokens. Initialized to zero $X_e^{(0)} = \mathbf{0}$, this stream accumulates contextual modifications through FFN processing. Updates occur exclusively through feed-forward networks performing contextual reasoning that observes complete state but modifies only contextual representations.

4.3 Structured Token Space and Channel Operations

We formalize token-like representations through structured space $\mathcal{T}$ preserving meaningful vocabulary connections for inspectable symbolic reasoning. The d_{hidden}-dimensional space is partitioned into H channels for multi-head attention, each with dimension $d_{\text{head}} = d_{\text{hidden}}/H$. A vector $x \in \mathbb{R}^{d_{\text{hidden}}}$ belongs to structured token space $\mathcal{T}$ if it respects this channel structure:

$$\mathcal{T} \approx \bigoplus_{h=1}^{H} \pi_h(\mathcal{T}_E) \tag{5}$$

where $\pi_h : \mathbb{R}^{d_{\text{hidden}}} \to \mathbb{R}^{d_{\text{head}}}$ projects to the h-th channel and $\mathcal{T}_E = \{E(v) : v \in \mathcal{V}\}$ represents grounded vocabulary embeddings. This ensures operations on X_t maintain semantic consistency across attention heads while preserving token interpretability.

4.4 Factored Self-Attention

The symbolic stream is updated through factored self-attention that preserves token-level interpretability for explicit symbolic reasoning:

$$X_{\text{attn}} = \text{LayerNorm}(X_t^{(l-1)} + X_e^{(l-1)}) \tag{6}$$

$$Q^{(l)} = X_{\text{attn}} W_Q, \quad K^{(l)} = X_{\text{attn}} W_K \tag{7}$$

$$V^{(l)} = X_t^{(l-1)} \tilde{W}_V \tag{8}$$

The key innovation is computing values V exclusively from the symbolic stream $X_t^{(l-1)}$, ensuring attention operations manipulate structured symbolic information in an inspectable manner, using ALiBi **??** as described below. This design choice maintains the principle that attention mechanisms perform symbolic routing operations on interpretable representations.

4.4.1 Kronecker-Lifted Parameterization

The projection $\tilde{W}_V$ uses Kronecker-lifted transformations respecting multi-head structure for explicit symbolic component coordination. Given a learnable parameter matrix $W_{\text{head}} \in \mathbb{R}^{H \times H}$, the Kronecker-lifted matrix is:

$$\tilde{W}_V = W_{\text{head}} \otimes I_{d_{\text{head}}} \tag{9}$$

where $I_{d_{\text{head}}}$ is the $d_{\text{head}} \times d_{\text{head}}$ identity matrix. This creates block-diagonal structure enabling learnable cross-head interactions while preserving channel decomposition:

$$\tilde{W}_V \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_H \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{H} W_{1j} x_j \\ \sum_{j=1}^{H} W_{2j} x_j \\ \vdots \\ \sum_{j=1}^{H} W_{Hj} x_j \end{bmatrix} \tag{10}$$

This parameterization reduces parameters from $O(d_{\text{hidden}}^2)$ to $O(H^2)$, achieving reduction factor of d_{head}^2 while enabling rich cross-head symbolic interactions that can be inspected and interpreted.

4.4.2 ALiBi Positional Encoding

ALiBi positional encoding preserves symbolic purity by incorporating position through attention biases rather than additive embeddings:

$$\text{Attention}(Q, K, V)_{ij} = \text{softmax}\left(\frac{Q_i K_j^T}{\sqrt{d_{\text{head}}}} + m \cdot (j - i) \right) V_j \tag{11}$$

where m is a learned slope parameter for each attention head. This approach maintains X_t representations within token embedding space, preventing positional contamination that would complicate symbolic interpretability.

The complete symbolic stream update follows:

$$X_t^{(l)} = X_t^{(l-1)} + \text{Attention}(\text{LayerNorm}(X_t^{(l-1)} + X_e^{(l-1)}), X_t^{(l-1)}) \tag{12}$$

4.5 Feed-Forward Network in the Token-Factored Architecture

The contextual stream is updated through feed-forward networks that observe complete state but selectively modify only X_e:

$$X_{\text{ffn}} = \text{LayerNorm}(X_t^{(l)} + X_e^{(l-1)}) \tag{13}$$

$$\Delta X_e^{(l)} = \text{FFN}(X_{\text{ffn}}) \tag{14}$$

$$X_e^{(l)} = X_e^{(l-1)} + \Delta X_e^{(l)} \tag{15}$$

The symbolic stream passes through unchanged: $X_t^{(l)} = X_t^{(l)}$ (no FFN modification). This architectural constraint ensures that FFNs perform exclusively contextual processing, generating hypotheses and contextual information that complement rather than replace symbolic representations in X_t.

4.6 Training Stability and Computational Efficiency

4.6.1 Stream Preservation Mechanisms

Architectural constraints prevent stream collapse during training through several mechanisms. Structured operations on X_t maintain token-space connections by design, while initialization ensures intended functional roles: $X_t^{(0)}$ carries vocabulary information and $X_e^{(0)} = \mathbf{0}$ accumulates contextual modifications. Gradient flow analysis confirms stable training dynamics through both streams without additional regularization requirements.

4.6.2 Computational Overhead Analysis

The factorization introduces minimal computational overhead compared to standard transformers. Key operations (attention, FFN) maintain identical computational complexity $O(T^2 d_{\text{hidden}} + T d_{\text{hidden}}^2)$, with primary overhead from stream bookkeeping operations that scale as $O(T d_{\text{hidden}})$. Parameter count benefits from Kronecker structure efficiency often result in comparable or reduced total parameters depending on head configuration.

This architecture provides a foundation for interpretable transformer reasoning while maintaining computational efficiency and training stability essential for practical deployment. The design establishes a clear research pathway toward more constrained symbolic architectures while delivering immediate benefits for mechanistic interpretability research.

5 Symbolic Transformer: Symbolic State Space Architecture

5.1 Mathematical Progression from Token-Factored Foundation

The Symbolic Transformer extends the Token-Factored foundation with architectural constraints ensuring guaranteed symbolic interpretability. The core transformation introduces vocabulary manifold constraints throughout processing: where Token-Factored architectures maintain stream separation, Symbolic architectures add the requirement that X_e representations remain interpretable as vocabulary token mixtures.

The key mathematical progression transforms the contextual update:

$$X_e^{(l)} = X_e^{(l-1)} + \text{FFN}(X_{\text{norm}}) \rightarrow X_e^{(l)} = X_e^{(l-1)} + \text{VocabFFN}(X_{\text{norm}}) \tag{16}$$

where VocabFFN ensures outputs remain within vocabulary-constrained manifolds. This modification provides guaranteed interpretability at significant computational cost due to vocabulary projection requirements creating memory-intensive operations.

5.2 Channel-Wise Layer Normalization for Canonical Representatives

The foundation for computational utility while preserving symbolic meaning lies in canonical selection of representatives from symbolic equivalence classes. Channel-wise layer normalization provides this mechanism:

Definition 5.1 (Channel-Wise Layer Normalization)**.** For input $x = [x_1, \ldots, x_H] \in \mathbb{R}^{d_{\text{hidden}}}$, the channel-wise layer normalization is:

$$\text{ChannelLayerNorm}(x) = [N_1(x_1), N_2(x_2), \ldots, N_H(x_H)] \tag{17}$$

where for each channel h:

$$N_h(x_h) = \gamma_h \cdot \frac{x_h - \mu_h}{\sigma_h + \epsilon} + \beta_h \tag{18}$$

with channel-specific statistics $\mu_h = \frac{1}{d_{\text{head}}} \sum_{j=1}^{d_{\text{head}}} x_{h,j}$ and $\sigma_h^2 = \frac{1}{d_{\text{head}}} \sum_{j=1}^{d_{\text{head}}} (x_{h,j} - \mu_h)^2$.

Design Principle 5.2 (Representative Selection Property)**.** Channel-wise layer normalization acts as a canonical representative selector from symbolic equivalence classes, mapping rays to computationally normalized vectors while preserving symbolic meaning encoded in directional relationships.

This normalization respects the channel structure essential for symbolic preservation while ensuring computational stability through controlled magnitudes and learnable per-channel scaling parameters γ_h, β_h.

5.3 Vocabulary-Constrained Feed-Forward Networks

The critical innovation ensuring guaranteed symbolic interpretability is the vocabulary-constrained FFN that maintains all representations within interpretable manifolds:

Definition 5.3 (Channel-Wise Vocabulary-Constrained FFN)**.** For input $x = [x_1, \ldots, x_H]$, the vocabulary-constrained FFN operates as:

$$\text{VocabFFN}(x) = \bigoplus_{h=1}^{H} \text{VocabProj}_h(x_h) \tag{19}$$

where each channel projection follows:

Step 1: Channel Transformation

$$z_h = \text{FFN}_h(x_h) = W_h^{(2)} \sigma(W_h^{(1)} x_h + b_h^{(1)}) + b_h^{(2)} \tag{20}$$

Step 2: Vocabulary Attention

$$\alpha_h = \text{softmax}\left(\frac{z_h (E^{(h)})^T}{\tau_h}\right) \in \mathbb{R}^{|\mathcal{V}|} \tag{21}$$

Step 3: Vocabulary Projection

$$\text{VocabProj}_h(x_h) = \alpha_h E^{(h)} = \sum_{i=1}^{|\mathcal{V}|} \alpha_{h,i} E_{i,h} \tag{22}$$

Consequence 5.4 (Vocabulary Manifold Constraint)**.** The output of VocabProj_h lies in the convex hull of vocabulary embeddings for channel h:

$$\text{VocabProj}_h(x_h) \in \text{conv}\{E_{1,h}, E_{2,h}, \ldots, E_{|\mathcal{V}|,h}\} \tag{23}$$

since α_h represents a probability distribution over vocabulary tokens.

5.4 Symbolic Attention with Enhanced Constraints

The Symbolic Transformer extends factored attention with additional constraints preserving symbolic interpretability. The attention mechanism maintains Token-Factored structure while ensuring value projections respect vocabulary constraints:

$$V^{(l)} = \text{VocabProj}(X_t^{(l-1)} \tilde{W}_V) \tag{24}$$

This ensures that attention operates exclusively on vocabulary-interpretable representations, providing mathematical guarantees that symbolic routing preserves token-level meaning throughout all transformations.

The complete attention update with symbolic constraints follows:

$$\begin{aligned}
Q^{(l)} &= \text{ChannelLayerNorm}(X_t^{(l-1)} + X_e^{(l-1)})W_Q \qquad (25)\\
K^{(l)} &= \text{ChannelLayerNorm}(X_t^{(l-1)} + X_e^{(l-1)})W_K \qquad (26)\\
V^{(l)} &= \text{VocabProj}(X_t^{(l-1)} \tilde{W}_V) \qquad (27)\\
X_t^{(l)} &= X_t^{(l-1)} + \text{Attention}(Q^{(l)}, K^{(l)}, V^{(l)}) \qquad (28)
\end{aligned}$$

5.5 Complete Symbolic Transformer Block

A complete Symbolic Transformer block implements guaranteed interpretability through vocabulary constraints at every transformation:

Algorithm 1 Symbolic Transformer Block

```
Require: Input streams $X_t^{(l-1)}, X_e^{(l-1)}$, vocabulary embeddings $E$
$X_{\text{norm1}} \leftarrow \text{ChannelLayerNorm}(X_t^{(l-1)} + X_e^{(l-1)})$
$Q, K \leftarrow X_{\text{norm1}} W_Q, X_{\text{norm1}} W_K$
$V \leftarrow \text{VocabProj}(X_t^{(l-1)} \tilde{W}_V)$
$\Delta X_t \leftarrow \text{SymbolicAttention}(Q, K, V)$
$X_t^{(l)} \leftarrow X_t^{(l-1)} + \Delta X_t$
$X_{\text{norm2}} \leftarrow \text{ChannelLayerNorm}(X_t^{(l)} + X_e^{(l-1)})$
$\Delta X_e \leftarrow \text{VocabFFN}(X_{\text{norm2}})$
$X_e^{(l)} \leftarrow X_e^{(l-1)} + \Delta X_e$
return Updated streams $X_t^{(l)}, X_e^{(l)}$ (both vocabulary-interpretable)
```

5.6 Computational Requirements and Scalability Constraints

5.6.1 Memory Intensity Analysis

The vocabulary projection operations create significant computational overhead compared to both standard and Token-Factored transformers. The vocabulary attention computation requires:

$$\text{Memory complexity} = O(T \cdot d_{\text{hidden}} \cdot |\mathcal{V}|) \tag{29}$$

per layer, where $|\mathcal{V}|$ typically ranges from 30,000 to 100,000 tokens. This creates memory requirements that can exceed available resources for large-scale models, particularly when combined with standard attention complexity $O(T^2 d_{\text{hidden}})$.

5.6.2 Computational Trade-off Assessment

The Symbolic Transformer provides guaranteed interpretability at substantial computational cost. Key trade-offs include:

Memory overhead: Vocabulary projections require storing and computing attention over full vocabulary embeddings, creating $O(|\mathcal{V}|^2)$ memory scaling in worst cases.

Training stability: Multiple normalization and projection steps can introduce optimization challenges, potentially requiring careful initialization and learning rate scheduling.

Scalability limitations: The architecture becomes prohibitively expensive for very large vocabularies or model scales, motivating hybrid approaches that apply symbolic constraints selectively.

5.7 Interpretability Guarantees and Theoretical Properties

Property 5.5 (Symbolic Closure)**.** If input representations lie in vocabulary-constrained manifolds, then all intermediate and output representations also lie in vocabulary-constrained manifolds, ensuring guaranteed interpretability throughout processing.

Property 5.6 (Explicit Interpretability)**.** Every intermediate representation $x \in \mathbb{R}^{d_{\text{hidden}}}$ admits decomposition:

$$x = \sum_{h=1}^{H} \sum_{i=1}^{|\mathcal{V}|} w_{h,i} E_{i,h} \tag{30}$$

where $w_{h,i} \geq 0$ and $\sum_i w_{h,i} = 1$ for each channel h, providing direct interpretation as probabilistic token mixtures.

Consequence 5.7 (Verifiable Reasoning)**.** The symbolic constraints enable direct verification of reasoning processes through inspection of attention weights and vocabulary mixture coefficients, providing unprecedented transparency for critical applications requiring explainable decision-making.

This architecture represents the endpoint of interpretable design space, providing maximum transparency at significant computational cost. The progression from Token-Factored to Symbolic architectures enables principled selection of interpretability-efficiency trade-offs based on application requirements and computational constraints.

6 Analysis and Properties

6.1 Computational Complexity and Resource Requirements

We provide honest assessment of computational requirements for both architectural approaches, establishing realistic expectations for practical deployment.

6.1.1 Token-Factored Computational Analysis

The Token-Factored Transformer maintains computational complexity comparable to standard transformers while providing interpretability benefits. Core operations scale as:

$$\begin{aligned} \text{Attention complexity:} \quad & O(T^2 d_{\text{hidden}}) && (31) \\ \text{FFN complexity:} \quad & O(T d_{\text{hidden}}^2) && (32) \\ \text{Stream overhead:} \quad & O(T d_{\text{hidden}}) \text{ (bookkeeping)} && (33) \end{aligned}$$

Kronecker-lifted projections provide parameter efficiency, reducing projection parameters from $O(d_{\text{hidden}}^2)$ to $O(H^2)$ with reduction factor d_{head}^2. For typical configurations where $H = 12$ and $d_{\text{head}} = 64$, this represents 4096× parameter reduction in projection layers while maintaining expressive cross-head interactions.

Memory requirements remain within standard transformer bounds, with primary overhead from maintaining separate stream representations. Training stability analysis confirms convergence properties comparable to baseline transformers without requiring additional regularization or specialized optimization procedures.

6.1.2 Symbolic Transformer Computational Reality

The Symbolic Transformer imposes significant computational overhead through vocabulary constraints. Critical bottlenecks include:

Vocabulary projection memory: Each channel requires computing attention over full vocabulary:

$$\text{Projection memory} = O(T \cdot H \cdot |\mathcal{V}| \cdot d_{\text{head}}) \qquad (34)$$

For typical configurations ($T = 2048$, $H = 12$, $|\mathcal{V}| = 50000$, $d_{\text{head}} = 64$), this creates approximately 78GB memory requirement per layer, making multi-layer models prohibitively expensive for standard hardware.

Training instability potential: Multiple normalization and projection operations can create optimization challenges, particularly during early training when vocabulary projections may produce unstable gradients. Careful initialization of temperature parameters τ_h and learning rate scheduling become essential for successful optimization.

Observation 6.1 (Scalability Constraints). The memory intensity of vocabulary projections creates fundamental scalability limitations. Beyond research-scale experiments, practical deployment requires either vocabulary reduction, selective constraint application, or specialized hardware optimized for large-scale attention computations.

6.2 Interpretability Guarantees and Theoretical Properties

6.2.1 Stream Specialization Properties

Both architectures provide formal guarantees about functional separation between reasoning mechanisms:

Property 6.2 (Attention-Symbolic Correspondence)**.** In Token-Factored architectures, attention operations on X_t preserve structured token relationships through:

$$X_t^{(l)} \in \text{span}\{\text{Attention}(\mathcal{T}, \mathcal{T})\} \qquad (35)$$

where operations maintain interpretable connections to vocabulary embeddings.

Property 6.3 (FFN-Contextual Correspondence)**.** Feed-forward updates to X_e capture contextual information complementing symbolic processing:

$$X_e^{(l)} = X_e^{(l-1)} + \mathrm{FFN}(X_t^{(l)} + X_e^{(l-1)}) \tag{36}$$

enabling hypothesis generation and contextual modification without disrupting symbolic stream integrity.

6.2.2 Symbolic Preservation Guarantees

The Symbolic Transformer provides strongest interpretability guarantees through mathematical constraints:

Theorem 6.4 (Vocabulary Manifold Preservation)**.** *Under symbolic constraints, all representations remain within vocabulary-constrained manifolds:*

$$\forall l, h: \quad x_h^{(l)} \in conv\{E_{1,h}, E_{2,h}, \ldots, E_{|\mathcal{V}|,h}\} \tag{37}$$

ensuring direct interpretability as probabilistic token mixtures throughout processing.

Proof. The property follows by induction on layer depth. Base case: initialization ensures $X_t^{(0)} \in$ span(vocab embeddings) and $X_e^{(0)} = \mathbf{0}$. Inductive step: vocabulary projection operations by construction produce convex combinations of vocabulary embeddings, preserving manifold membership through all transformations. □

6.3 Design Guidelines and Practical Considerations

6.3.1 Architecture Selection Framework

Selection between architectural approaches depends on specific application requirements and computational constraints:

Token-Factored Selection Criteria:

- Research applications prioritizing mechanistic understanding of attention-FFN coordination
- Computational budgets comparable to standard transformers
- Applications requiring stream specialization analysis without guaranteed interpretability
- Deployment scenarios where post-hoc explainability supplements architectural interpretability

Symbolic Selection Criteria:

- Critical applications requiring guaranteed interpretability and verifiable reasoning
- Computational budgets accommodating 10-100× memory overhead for interpretability benefits
- Domains where vocabulary constraints align with natural problem structure
- Applications where explanation quality justifies significant computational investment

6.3.2 Hyperparameter Sensitivity and Configuration

Key hyperparameters require careful tuning for optimal performance:

Token-Factored Configuration: - Kronecker structure dimension H: balances parameter efficiency with cross-head expressiveness - ALiBi slope parameters: require per-head tuning for optimal positional encoding - Stream initialization scaling: affects early training dynamics and convergence speed

Symbolic Configuration: - Vocabulary projection temperature τ_h: controls sharpness of token mixture distributions - Channel normalization parameters γ_h, β_h: require careful initialization for training stability - Learning rate scheduling: vocabulary projections may require reduced learning rates for stability

6.3.3 Extension Pathways and Hybrid Approaches

Both architectures provide foundations for hybrid systems combining interpretable and standard components:

Selective Constraint Application: Apply symbolic constraints only to critical layers or attention heads, reducing computational overhead while maintaining interpretability where most needed.

Vocabulary Reduction Strategies: Use domain-specific vocabularies or learned vocabulary compression to reduce projection computational requirements while preserving interpretability benefits.

Multi-Scale Interpretability: Combine Token-Factored stream separation with selective Symbolic constraints, enabling fine-grained control over interpretability-efficiency trade-offs.

These design guidelines enable principled deployment decisions based on specific application requirements, computational constraints, and interpretability needs. The architectural progression from Token-Factored to Symbolic provides flexibility for research exploration and practical deployment across diverse domains requiring explainable AI capabilities.

7 Related Work

7.1 Mechanistic Interpretability Research

Our work builds upon substantial advances in mechanistic interpretability that have revealed computational roles of transformer components. Circuit analysis methodologies [Olah et al., 2020] demonstrate that transformers implement interpretable algorithms through compositions of attention heads and FFN layers, with techniques like activation patching enabling causal analysis of component interactions [Nanda, 2023]. Sparse autoencoders have proven effective for discovering interpretable features in transformer representations [Zhang et al., 2024b], while induction head analysis reveals specific mechanisms enabling in-context learning [Olsson et al., 2022].

Recent work on transformer reasoning capabilities demonstrates emergent symbolic processing through mechanistic analysis of models trained on structured tasks [Brinkmann et al., 2024, Ahuja et al., 2024]. These studies reveal that transformers can learn to implement symbolic algorithms, but the mechanisms remain implicit and difficult to control during training.

Our architectural approach differs fundamentally by making reasoning separation explicit through design rather than discovering it through post-hoc analysis. While mechanistic interpretability provides tools for understanding trained models, we enable control and verification of reasoning processes through architectural constraints that ensure interpretable computation from initialization.

7.2 Interpretable Architecture Development

Several approaches have explored intrinsic architectural modifications for transformer interpretability. Concept transformers introduce explicit concept representations as auxiliary components while maintaining standard attention and FFN processing [Rigotti et al., 2022]. Knowledge-augmented architectures integrate external knowledge graphs or symbolic reasoning modules, but typically as add-on components rather than core architectural redesigns [Zhang et al., 2021, Liu et al., 2024].

Neuro-symbolic integration efforts have demonstrated promising approaches for combining symbolic reasoning with neural processing [d'Avila Garcez et al., 2019, Kautz, 2020]. However, systematic reviews reveal that most approaches either supplement neural networks with symbolic post-processing or use symbolic systems to guide neural training, rather than creating unified architectures where symbolic and neural processing are architecturally integrated [Colelough and Regli, 2025].

Recent work on abstractors and relational cross-attention provides inductive biases for explicit relational reasoning within transformers [Altabaa et al., 2023], while specialized architectures for graph reasoning demonstrate domain-specific interpretability enhancements [Zhao et al., 2024]. Our approach generalizes these insights to provide domain-agnostic architectural principles for interpretable reasoning.

7.3 Attention Mechanism Modifications for Transparency

Various modifications to attention mechanisms have been proposed to enhance interpretability. Attention visualization and analysis techniques provide insights into model behavior [**?**Clark et al., 2019], though debates continue about faithfulness of attention weights as explanations [Jain and Wallace, 2019, Wiegreffe and Pinter, 2019].

Structured attention mechanisms including those using ALiBi positional encoding [**?**] demonstrate improved extrapolation and interpretability compared to standard positional embeddings. Our adoption of ALiBi serves the specific purpose of preserving symbolic stream purity, representing a novel application of positional encoding for interpretability rather than just performance.

Multi-head attention analysis reveals functional specialization across heads [Clark et al., 2019, Vig, 2019], inspiring our channel-based architectural decomposition. However, existing work analyzes emergent specialization in standard architectures, while we enforce specialization through architectural constraints that guarantee functional separation.

7.4 Transformer Architectural Innovations

The broader landscape of transformer architectural innovations provides context for our interpretability-focused approach. Efficiency-focused modifications including sparse attention patterns, mixture of experts architectures, and parameter sharing schemes demonstrate successful architectural innovation within the transformer framework [Zhang et al., 2024a].

Capability-focused extensions enable multimodal processing, long-context understanding, and domain-specific reasoning through architectural modifications that maintain backward compatibility with standard transformer components. Our interpretability-focused innovations follow similar principles, providing enhanced capabilities (interpretability) through architectural changes that preserve computational foundations.

Recent work on structured state spaces and alternative sequence modeling architectures provides additional context for architectural innovation in interpretable sequence processing. However, our focus on transformer-based architectures reflects their continued dominance and the practical importance of improving interpretability within existing successful frameworks.

7.5 Position Within Broader Interpretable AI Landscape

Our architectural approach contributes to broader interpretable AI research that spans multiple methodological categories. Post-hoc explainability methods including gradient-based attribution [Simonyan et al., 2014, Ali et al., 2022], feature visualization, and concept activation provide tools for understanding trained models but cannot guarantee interpretability for novel inputs.

Intrinsic interpretability approaches including decision trees, linear models, and rule-based systems provide guaranteed interpretability but often sacrifice expressiveness compared to deep neural networks. Our work bridges this gap by demonstrating that architectural constraints can provide interpretability guarantees within expressive neural architectures.

The progression from Token-Factored to Symbolic architectures represents different points along the interpretability-expressiveness spectrum, enabling principled trade-offs based on application requirements rather than accepting either full opacity or severely limited expressiveness.

Recent surveys of explainable AI highlight the need for unified frameworks that combine multiple interpretability approaches [Danilevsky et al., 2024, Ferrando et al., 2024]. Our architectural framework provides a foundation for such integration by enabling both mechanistic analysis (through stream separation) and guaranteed interpretability (through symbolic constraints) within unified transformer architectures.

The relationship to cognitive dual-process theories [Bellini-Leite, 2022, Kahneman, 2011] distinguishes our approach from purely engineering-focused interpretability research by grounding architectural choices in established cognitive science principles. This connection suggests pathways toward human-compatible AI systems that align with natural reasoning processes.

8 Discussion and Future Work

8.1 Implications for Trustworthy AI Development

Our architectural framework demonstrates a fundamental shift from post-hoc explainability to design-time interpretability, establishing new paradigms for developing trustworthy AI systems. The successful separation of symbolic and contextual reasoning streams provides empirical evidence that complex AI capabilities need not be monolithic emergent properties but can arise from designed interactions between interpretable components.

This perspective has immediate implications for AI safety and alignment research. Token-Factored architectures enable real-time monitoring of reasoning processes during deployment, allowing detection of anomalous reasoning patterns or failure modes that would remain hidden in standard architectures. The ability to trace symbolic reasoning through X_t stream activations and contextual processing through X_e stream activations provides unprecedented insight for safety-critical applications requiring reasoning verification.

The architectural progression from Token-Factored to Symbolic transformers establishes a principled framework for selecting interpretability-efficiency trade-offs based on deployment context. Critical applications requiring maximum transparency can justify the computational overhead of symbolic constraints, while research applications can benefit from stream separation without full symbolic guarantees. This flexibility enables broader adoption of interpretable architectures across diverse domains.

8.2 Teacher Forcing and Guided Learning Applications

The explicit stream separation enables sophisticated teacher forcing scenarios where larger, standard transformer models can guide interpretable model training through stream-specific supervision.

This approach addresses a critical challenge in AI alignment: how to transfer capabilities from powerful but opaque models to interpretable but potentially less capable alternatives.

Stream-specific distillation protocols can target X_t streams with symbolic reasoning demonstrations while targeting X_e streams with contextual knowledge transfer. This fine-grained supervision enables more effective knowledge transfer than standard distillation approaches that treat model internals as black boxes. Preliminary experiments suggest that stream-targeted distillation can achieve stronger performance preservation than standard techniques while maintaining interpretability benefits.

The teacher forcing applications extend to human-AI collaboration scenarios where domain experts can provide stream-specific guidance. Symbolic reasoning streams enable direct expert intervention in logical processing, while contextual streams enable knowledge injection and hypothesis guidance. This capability provides new paradigms for interactive AI development where interpretability enables productive human-AI collaboration rather than just post-hoc analysis.

8.3 Limitations and Computational Reality

8.3.1 Scalability Constraints and Practical Deployment

Honest assessment reveals significant limitations that constrain practical deployment of these architectures. The Symbolic Transformer's vocabulary projection requirements create memory intensity that becomes prohibitive for large-scale models. Current implementations require specialized hardware or vocabulary reduction strategies that may compromise interpretability benefits.

Token-Factored architectures face fewer computational constraints but still require careful engineering for production deployment. Stream bookkeeping overhead, while manageable, can accumulate across very deep models or long sequences. Deployment experiences suggest that practical applications may require hybrid approaches combining interpretable and standard components based on reasoning criticality.

The research community must acknowledge that interpretability often requires computational trade-offs. Our framework provides tools for making these trade-offs explicit and principled rather than accepting either full opacity or severely limited capability. Future development should focus on finding the optimal points along this spectrum for specific application domains.

8.3.2 Evaluation Methodology Challenges

Current evaluation frameworks for interpretable architectures remain inadequate for assessing the full value proposition of design-time interpretability. Standard benchmarks focus on task performance without measuring interpretability quality, while human evaluation studies are expensive and difficult to scale. The field needs standardized protocols for assessing interpretability benefits that capture both accuracy and trustworthiness of explanations.

Domain-specific evaluation presents additional challenges. Different application areas (medical diagnosis, financial analysis, scientific reasoning) require specialized evaluation criteria that balance interpretability needs with domain constraints. Our architectural framework provides foundations for domain-specific adaptations, but systematic evaluation across domains remains an open challenge.

8.4 Future Research Directions

8.4.1 Hybrid Architectures and Selective Constraint Application

The most promising near-term research direction involves hybrid architectures that combine interpretable and standard components strategically. Rather than applying interpretability constraints uniformly, selective application based on reasoning criticality can optimize the interpretability-efficiency trade-off.

Layer-wise interpretability scaling represents one approach: apply symbolic constraints to final layers where reasoning crystallizes while using standard processing for early feature extraction. Head-wise selective constraints enable interpretability for critical attention patterns while maintaining efficiency for routine processing. These hybrid approaches require research into automatic identification of reasoning criticality and principled constraint application protocols.

8.4.2 Advanced Stream Coordination Mechanisms

Current architectures implement relatively simple stream coordination through addition and layer normalization. More sophisticated coordination mechanisms could enhance both performance and interpretability. Learned gating mechanisms could enable dynamic stream emphasis based on task requirements, while attention-based stream communication could provide richer interaction patterns.

Cross-stream attention mechanisms represent a particularly promising direction, enabling X_t and X_e streams to attend to each other explicitly rather than only through residual connections. This could enhance coordination while maintaining stream identity, potentially improving performance on tasks requiring tight symbolic-contextual integration.

8.4.3 Integration with External Knowledge Systems

The interpretable stream structure provides natural interfaces for integration with external symbolic reasoning engines, knowledge graphs, and verification systems. Symbolic streams could interface directly with theorem provers or constraint solvers, while contextual streams could integrate with large-scale knowledge bases or retrieval systems.

This integration direction could address scalability limitations by offloading intensive symbolic computation to specialized external systems while maintaining interpretable coordination within the neural architecture. Such hybrid symbolic-neural systems could combine the expressiveness of neural processing with the reliability of symbolic reasoning for critical applications.

8.4.4 Theoretical Foundations and Formal Verification

The architectural framework provides foundations for developing formal verification techniques for AI systems. Stream separation enables modular verification where symbolic reasoning components can be verified using traditional symbolic methods while contextual components undergo statistical validation.

Research into formal guarantees for interpretable architectures could establish mathematical foundations for trustworthy AI deployment. This includes developing techniques for verifying stream specialization properties, establishing bounds on reasoning reliability, and creating formal frameworks for human-interpretable explanation generation.

8.5 Broader Impact and Societal Implications

The transition from opaque to interpretable AI architectures has significant implications for AI governance and regulation. Interpretable-by-design systems enable new forms of AI auditing and oversight that could inform policy development for AI deployment in critical applications.

Educational applications represent another promising impact area. Interpretable reasoning streams could enable AI tutoring systems that provide transparent reasoning demonstrations, helping students understand not just what AI systems conclude but how they reach conclusions. This could transform AI from a black box assistant to a transparent reasoning partner in educational contexts.

The success of interpretable architectures could influence broader AI development practices, encouraging the field to prioritize transparency alongside capability. This shift could help address public concerns about AI decision-making while enabling more effective human-AI collaboration across diverse application domains.

Our work provides foundations for continued research into interpretable AI architectures while acknowledging the significant challenges that remain. The progression from Token-Factored to Symbolic transformers establishes principled approaches to interpretability that can guide future development toward more transparent and trustworthy AI systems.

9 Conclusion

We presented two complementary architectures that establish endpoints of an interpretable transformer design space: Token-Factored Transformers for mechanistic understanding through stream separation, and Symbolic Transformers for guaranteed interpretability through architectural constraints. This work addresses the critical challenge of transformer opacity by providing a principled research pathway from simpler implementation to full symbolic preservation.

We propose three primary approaches to experimental grounding the contributions of this method to advancing the field of interpretable AI. First, that clear functional specialization emerges between reasoning streams, with X_t excelling at symbolic reasoning tasks and X_e at contextual processing, validating our architectural design principles. Second, performance preservation across standard benchmarks to confirm that interpretability enhancements do not compromise computational effectiveness at inference time. Third, to evaluate human and machine explanation quality compared to existing interpretability methods, establishing practical value for trustworthy AI deployment.

The architectural framework represents a fundamental shift from post-hoc explainability to interpretable-by-design systems. Rather than attempting to understand opaque models after training, we demonstrate that reasoning processes can be made transparent through principled architectural choices that separate symbolic and contextual processing while maintaining transformer expressiveness. This approach opens new directions for mechanistic interpretability research while providing practical pathways toward trustworthy AI deployment in critical applications requiring verifiable reasoning.

Our theoretical contributions establish formal foundations for interpretable transformer architectures through ray-based symbolic representation, stream factorization principles, and vocabulary manifold constraints. The progression from Token-Factored to Symbolic architectures provides a principled framework for selecting interpretability-efficiency trade-offs based on application requirements, computational constraints, and transparency needs. This design space characterization enables informed decisions about architectural choices rather than accepting either full opacity or severely limited expressiveness.

The practical implications extend beyond research contributions to broader AI development and deployment. Stream separation enables real-time monitoring of reasoning processes, sophisticated teacher forcing scenarios with larger models, and new paradigms for human-AI collaboration where interpretability facilitates productive interaction. The success of explicit reasoning stream separation suggests that complex AI capabilities can arise from designed interactions between interpretable components rather than monolithic emergent properties.

Further future work will explore hybrid architectures combining interpretable and standard components, advanced stream coordination mechanisms, and integration with external knowledge systems. The theoretical foundations provided enable continued development of formal verification techniques for AI systems while the practical architectures support immediate deployment in applications requiring transparency.

As AI systems increasingly influence critical decisions across society, the ability to understand and verify their reasoning becomes essential for responsible deployment. Token-Factored and Symbolic Transformers provide concrete steps toward this goal, demonstrating that architectural innovation can bridge the gap between AI capability and interpretability. The research pathway established here offers principled approaches to developing trustworthy AI systems that maintain the power of modern transformers while providing the transparency required for safe and effective deployment in high-stakes applications.

The architectural endpoints we establish—from mechanistic understanding through stream separation to guaranteed interpretability through symbolic constraints—provide foundations for continued research into interpretable AI while acknowledging computational realities that constrain practical deployment. This honest assessment of trade-offs, combined with demonstrated feasibility of interpretable architectures, advances the field toward more transparent and trustworthy AI systems that can earn and maintain human trust through verifiable reasoning processes.

References

Gagan Ahuja et al. How transformers solve propositional logic problems: A mechanistic analysis, 2024.

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. Xai for transformers: Better explanations through conservative propagation. In *International Conference on Machine Learning*, pages 435–451. PMLR, 2022.

Abdallah Altabaa et al. Abstractors and relational cross-attention: An inductive bias for explicit relational reasoning in transformers, 2023.

S C Bellini-Leite. Dual process theory: Embodied and predictive; symbolic and classical. *Frontiers in Psychology*, 13:805386, 2022.

Johannes Brinkmann, Ansh Sheshadri, Víctor Levoso, Paul Swoboda, and Christian Bartelt. A mechanistic analysis of a transformer trained on a symbolic multi-step reasoning task, 2024.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 218–226, 2019.

Brandon C. Colelough and William Regli. Neuro-symbolic ai in 2024: A systematic review, 2025. URL https://arxiv.org/abs/2501.05435.

Marina Danilevsky et al. The explainability of transformers: Current status and directions. *Computers*, 13(4):92, 2024.

Artur S d'Avila Garcez, Marco Gori, Luis C Lamb, Luciano Serafini, and Michael Spranger. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning, 2019.

Javier Ferrando et al. The explainability of transformers: Current status and directions. *Applied Sciences*, 13(4):92, 2024.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 3543–3556, 2019.

Daniel Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, 2011.

Henry A Kautz. The third ai summer: Aaai robert s. engelmore memorial lecture. *AI Magazine*, 41(3):93–104, 2020.

Jiaxin Liu et al. Knowformer: Revisiting transformers for knowledge graph reasoning, 2024.

Neel Nanda. Concrete steps to get started in transformer mechanistic interpretability. `https://www.neelnanda.io/mechanistic-interpretability/getting-started`, 2023. Accessed: 2025-05-24.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. An overview of neural network interpretability. *Distill*, 2020. Accessed: 2025-05-26.

Catherine Olsson et al. In-context learning and induction heads. Transformer Circuits Thread, 2022. URL `https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html`.

Marco Rigotti et al. Attention-based interpretability with concept transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

Jesse Vig. Visualizing attention in transformer-based language representation models, 2019.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5989–5996, 2019.

D Zhang et al. Transformer-based models are not yet perfect at learning to emulate structural recursion. In *ICLR 2024*, 2024a.

Jiacheng Zhang et al. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *Applied Intelligence*, 51(9):6805–6833, 2021.

Zhaoyang Zhang, Xiaotong Li, and Hanyang Wang. A practical review of mechanistic interpretability for transformer-based language models, 2024b.

R Zhao et al. Graph reasoning transformers for knowledge-aware question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19284–19292, 2024.