Symbolic vs. Continuous Features in Transformers: A Digital Communication System's Explanation

Anonymous Author(s)
Affiliation
Address
email

Abstract

The term "feature" in mechanistic interpretability is ambiguous —- sometimes referring to symbolic properties (e.g., grammatical number), sometimes to neural activations (e.g., basis vectors). We clarify this distinction using communication theory: symbolic features are the *information* being transmitted, while neural features are the *signals* carrying that information. Through a toy transformer implementing subject-verb agreement, we demonstrate how linguistic properties can be encoded as orthogonal basis vectors, transmitted via attention, and decoded for grammatical decisions. This educational distillation provides a communication-theoretic lens for understanding transformer internals, offering conceptual clarity for mechanistic interpretability.

1 Introduction

The term "feature" in mechanistic interpretability is ambiguous: does it mean abstract linguistic properties (singular/plural) or neural activation patterns? Despite discoveries of circuits [10], superposition [6], and induction heads [12], this confusion persists. Communication theory distinguishes information (message) from signals (carrier)—we apply this to transformers.

Transformers' complex behavior emerges from simple operations, like communication systems that transmit digital information through linear filters and routing. The key is **layered abstraction** [16, 5]: separating physical signals from logical information. HPSG [13] similarly factorizes grammar through typed feature structures. We build a white-box toy model showing how transformers transmit symbolic linguistic features via orthogonal basis vectors and attention-based routing, providing ground-truth understanding of symbolic processing through continuous computation.

2 Related Work

Mechanistic interpretability reveals structured mechanisms: circuits [10], superposition [6], induction heads [12], and the IOI circuit [15]. The Transformer Circuits thread [7] provides mathematical foundations for understanding attention as information routing, while logit lens [11] reveals progressive refinement across layers. Sparse autoencoders [3] recover features from superposition, connecting to compressed sensing [4, 2]. Dictionary learning approaches [1] find interpretable directions in activation space, analogous to basis recovery in signal processing.

Tracr [9] compiles programs to weights but uses CFG which loses lexical information; HPSG [13] maintains it through typed features. The English Resource Grammar [8] demonstrates HPSG's practical scalability with thousands of lexical types. Information theory applications [14] analyze local mechanisms; we apply communication principles globally, treating the entire transformer as a coordinated communication network.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

3 Model Transformer Computation Graph as Physical Layer

The transformer architecture can be precisely modeled as a communication system's physical layer. Each operation has an exact mathematical equivalence in signal processing, enabling rigorous analysis of information flow.

Signal Modulation. The embedding layer maps discrete tokens to continuous signals. For vocabulary $\mathcal{V}$ and embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$:

$$\mathbf{s}_t = \mathbf{e}_{x_t} + \mathbf{p}_t \in \mathbb{R}^d \tag{1}$$

where $\mathbf{e}_{x_t}$ is the token embedding (codeword) and $\mathbf{p}_t$ is the position encoding (phase). This is identical to digital modulation in communication systems, mapping symbols to signal constellation points.

Attention as Matched Filtering. The QKV mechanism implements signal detection through correlation. For head h:

$$\alpha_{ij}^{(h)} = \frac{\langle \mathbf{W}_Q^{(h)} \mathbf{x}_i, \mathbf{W}_K^{(h)} \mathbf{x}_j \rangle}{\sqrt{d_k}} \tag{2}$$

This is mathematically equivalent to matched filtering, where $\mathbf{W}_K^{(h)}$ defines reference patterns and the dot product performs correlation detection. The QK circuit [7] computes pattern matching, while the OV circuit moves information:

$$\mathbf{z}_i = \sum_j \text{softmax}(\alpha_{ij}) \cdot \mathbf{W}_V^{(h)} \mathbf{x}_j \tag{3}$$

where $\mathbf{W}_V^{(h)}$ acts as feature extraction filters (demodulation carriers).

Multi-Head as Frequency Division. The multi-head mechanism divides the d-dimensional signal space into H parallel channels, each operating on a $d_h = d/H$ dimensional subspace. This is analogous to frequency division multiplexing, where different frequency bands carry independent information streams.

Residual Stream as Communication Bus. The residual connections create a communication path through layers:

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} + \text{Attention}^{(l)}(\mathbf{x}^{(l)}) + \text{MLP}^{(l)}(\mathbf{x}^{(l)}) \tag{4}$$

This additive structure ensures information from early layers remains accessible, enabling multi-hop communication across the network.

Feature Encoding. For symbolic features, we can use orthogonal encoding when the feature space is small:

$$\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_n] \in \mathbb{R}^{d \times n}, \quad \mathbf{B}^T \mathbf{B} = \mathbf{I}_n \tag{5}$$

Each feature f_i maps to basis vector $\mathbf{b}_i$, guaranteeing perfect recovery via matched filtering. When $n > d$, sparse overcomplete bases enable superposition [6], where k-sparse signals with $k \ll n$ can be recovered despite non-orthogonality, as demonstrated by sparse autoencoders [3].

4 Design a Toy Communication Protocol for Subject-Verb Agreement

4.1 Symbolic Linguistic Features for Grammar Analysis

Consider the minimal pair "cat meows" (grammatical) vs. "cat meow" (ungrammatical). To understand how transformers might process this, we need to distinguish between **symbolic linguistic features** (abstract grammatical properties) and their **neural representations** (directions in activation space).

Context-Free Grammar (CFG): Syntactic Categories Only. CFG uses production rules:

$$S \to NP_{sg}\, VP_{sg} \quad | \quad NP_{pl}\, VP_{pl} \tag{6}$$

$$NP_{sg} \to \text{cat} \quad VP_{sg} \to \text{meows} \tag{7}$$

"Cat meows" parses as $S \to NP_{sg}\, VP_{sg}$. But CFG only captures syntactic categories—the actual word "cat" is lost after parsing, making it impossible to reconstruct the original sentence.

Universal Dependencies (UD): Post-hoc Feature Annotation. UD annotates with dependency relations:

```
cat --nsubj--> meows    [NUMBER=sg on both]
```

Agreement is checked after parsing through feature annotations. While UD preserves lexical items, it doesn't explain the mechanism of feature checking.

HPSG: Lexical Features as First-Class Citizens. HPSG represents each word with typed feature structures that include both syntactic and semantic information:

$$\text{cat} : \begin{bmatrix} \text{PHON} & \langle \text{cat} \rangle \\ \text{HEAD} & \text{noun}[3sg] \\ \text{VAL|SUBJ} & \langle\rangle \end{bmatrix} \quad \text{meows} : \begin{bmatrix} \text{PHON} & \langle \text{meows} \rangle \\ \text{HEAD} & \text{verb}[3sg] \\ \text{VAL|SUBJ} & \langle NP[3sg] \rangle \end{bmatrix} \tag{8}$$

The key insight: HPSG treats linguistic features as **atomic symbolic information** attached to each lexical item. Each token carries multiple features (POS, NUM, semantic role) that must be transmitted and checked. This naturally maps to a communication protocol where each token broadcasts multiple feature "carriers"—one per linguistic property.

4.2 From Symbolic Features to Neural Transmission

We now design a transformer that implements HPSG-style feature checking through a communication protocol. The key innovation: each token transmits multiple orthogonal signals, one for each linguistic feature.

The Codebook-Basis Connection. In mechanistic interpretability, a "basis" is a set of directions in activation space. In communication theory, a "codebook" maps discrete symbols to continuous signals. These are the same concept:

- **Basis vectors** $\{\mathbf{b}_1, \mathbf{b}_2, ...\}$ define orthogonal directions in $\mathbb{R}^d$
- **Codewords** are specific points/vectors assigned to symbols
- The embedding matrix is literally a codebook: column i is the codeword for token i

Multi-Carrier Transmission Protocol. Instead of encoding just the token identity (e.g., "cat"), we transmit structured information:

$$\text{Token Embedding} = \underbrace{\mathbf{b}_{\text{POS}} \otimes \mathbf{c}_{\text{noun}}}_{\text{POS carrier}} + \underbrace{\mathbf{b}_{\text{NUM}} \otimes \mathbf{c}_{\text{sg}}}_{\text{NUM carrier}} + \underbrace{\mathbf{b}_{\text{LEX}} \otimes \mathbf{c}_{\text{cat}}}_{\text{LEX carrier}} \tag{9}$$

Each token transmits three orthogonal carriers:

- $\mathbf{b}_{\text{POS}}$: Direction for part-of-speech information
- $\mathbf{b}_{\text{NUM}}$: Direction for number agreement
- $\mathbf{b}_{\text{LEX}}$: Direction for lexical identity

The codewords $\mathbf{c}_{\text{noun}}, \mathbf{c}_{\text{sg}}, \mathbf{c}_{\text{cat}}$ encode the specific feature values. This is why transformers have high-dimensional embeddings—they need space for multiple orthogonal feature directions.

Concrete Example: "Cat Meows" Processing.

Stage 1: Modulation (Embedding Layer). Each word is encoded as multiple features:

$$\mathbf{e}_{\text{cat}} = \mathbf{b}_{\text{POS}} \otimes \mathbf{c}_{\text{noun}} + \mathbf{b}_{\text{NUM}} \otimes \mathbf{c}_{\text{sg}} + \mathbf{b}_{\text{LEX}} \otimes \mathbf{c}_{\text{cat}} \tag{10}$$

$$\mathbf{e}_{\text{meows}} = \mathbf{b}_{\text{POS}} \otimes \mathbf{c}_{\text{verb}} + \mathbf{b}_{\text{NUM}} \otimes \mathbf{c}_{\text{sg}} + \mathbf{b}_{\text{LEX}} \otimes \mathbf{c}_{\text{meows}} \tag{11}$$

Message Encoding as Feature Decomposition: Instead of treating "cat" as an atomic token, we decompose it into linguistic features. Each feature gets its own basis direction (carrier), enabling independent transmission and processing.

Stage 2: Channel Probing and Routing (Attention). Attention heads act as specialized receivers that probe specific feature channels:

Head 1 (Syntactic Router): This head probes the POS channel to identify subject-verb pairs:

$$\mathbf{W}_Q^{(1)} = [\mathbf{b}_{\text{POS}}, ...] \quad \text{(probe POS channel in query)} \tag{12}$$

$$\mathbf{W}_K^{(1)} = [\mathbf{b}_{\text{POS}}, ...] \quad \text{(probe POS channel in key)} \tag{13}$$

When query extracts "verb" and key extracts "noun", high attention score triggers information routing. The value matrix then extracts and routes the NUM feature:

$$\mathbf{W}_V^{(1)} = [\mathbf{b}_{\text{NUM}}, ...] \quad \text{(extract NUM channel for routing)} \tag{14}$$

Channel Probing as Feature Extraction: Different attention heads probe different feature channels. This is exactly what sparse autoencoders do—they learn basis vectors (matched filters) to extract specific features from superposed representations.

Head 2 (Agreement Checker): This head probes the NUM channel to check compatibility:

$$\mathbf{W}_Q^{(2)} = [\mathbf{b}_{\text{NUM}}, ...] \quad \text{(probe NUM from verb position)} \tag{15}$$

$$\mathbf{W}_K^{(2)} = [\mathbf{b}_{\text{NUM}}, ...] \quad \text{(probe NUM from routed subject)} \tag{16}$$

The dot product $\langle \mathbf{c}_{\text{sg}}, \mathbf{c}_{\text{sg}} \rangle = 1$ indicates agreement (both singular). For "cat meow", we'd get $\langle \mathbf{c}_{\text{sg}}, \mathbf{c}_{\text{pl}} \rangle = 0$ (mismatch).

Routing Protocols for Feature Unification: Attention implements routing based on feature compatibility. When patterns match (e.g., both singular), information flows. When they mismatch, routing is blocked. This implements HPSG unification through neural computation.

Stage 3: Demodulation (MLP + Output). The MLP acts as a matched filter bank, extracting the agreement signal:

$$\mathbf{W}_{\text{MLP}}[i, :] = \mathbf{b}_{\text{AGREE}}^T \quad \Rightarrow \quad \text{activation}_i = \langle \mathbf{b}_{\text{AGREE}}, \text{residual stream} \rangle \tag{17}$$

High activation indicates agreement detected $\rightarrow$ output "grammatical". Low activation indicates mismatch $\rightarrow$ output "ungrammatical".

Superposition and Overcomplete Coding: When we have more features than dimensions ($n > d$), features share directions:

$$\text{Signal} = \sum_{i \in \text{Active}} \alpha_i \mathbf{b}_i, \quad |\text{Active}| \ll n \tag{18}$$

This is superposition—multiple features encoded in overlapping directions. Recovery works when features are sparse (few active per token), as shown by compressed sensing theory.

The transformer doesn't just process tokens—it transmits and processes structured symbolic information through continuous signals. Understanding this distinction between symbolic features (the information) and neural features (the signal carriers) is key to mechanistic interpretability.

5 Limitations

Our toy model uses designed weights, not learned ones, yielding artificially clean protocols. We demonstrate only simple subject-verb agreement, not long-range dependencies or nested structures. We assume perfect orthogonality while real transformers exhibit partial superposition with interference. The model lacks autoregression and the communication framework may not capture all emergent phenomena in large-scale transformers. Despite providing conceptual clarity, empirical investigation remains essential.

6 Conclusion

This educational distillation clarifies a fundamental ambiguity: symbolic features (linguistic properties) are the *information*, while neural features (basis vectors) are the *signals*. Through a communication lens, we show how transformers transmit structured information via orthogonal carriers and routing protocols, bridging digital communication and neural computation for mechanistic interpretability.

References

[1] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. `https://transformer-circuits.pub/2023/monosemantic-features`, 2023. Transformer Circuits Thread, Anthropic.

[2] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[3] Hoagy Cunningham, Aiden Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[4] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[5] Yonina C Eldar and Moshe Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Transactions on Information Theory*, 55(11):5302–5316, 2009.

[6] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Superposition, memorization, and double descent. *Distill*, 7(5):e20220048, 2022.

[7] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

[8] Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28, 2000.

[9] David Lindner, Ján Kramár, Matthew Rahtz, Thomas McGrath, and Vladimir Mikulik. Tracr: Compiled transformers as a laboratory for interpretability. *Advances in Neural Information Processing Systems*, 36, 2023.

[10] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

[11] nostalgebraist. The logit lens. *LessWrong*, 2020.

[12] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

[13] Carl Pollard and Ivan A Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994.

[14] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[15] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2023.

[16] Lloyd R Welch. Lower bounds on the maximum cross correlation of signals. *IEEE Transactions on Information Theory*, 20(3):397–399, 1974.