Feature Integration Beyond Sparse Coding: Evidence for Non-Linear Computation Spaces in Neural Networks

Omar Claflin[1]

Abstract

- **Problem**: Current sparse autoencoders achieve high reconstruction fidelity yet fail to eliminate polysemanticity, indicating that neural networks encode information beyond sparse feature superposition [7].
- **Hypothesis**: Neural representations encode both feature identity (what concepts are present) *and* feature integration (how concepts combine computationally) into the same compressed representational space. Linear superposition assumes non-orthogonality is noise or interference, but we present a different view of it as encoded computational relationships.
- **Method**: Train Neural Factorization Machines on SAE residuals, apply secondary SAE to NFM embeddings, giving us more interpretability to the modelled interaction layer of the NFM (which inputs SAE activations, and outputs a residual that can be additively combined with the primary SAE to better reconstruct the layer's original activations).
- **Results**: 23% reconstruction improvement (on a Top K restricted SAE), 5-20% contribution from the interaction component, along with selective intervention effects to demonstrate the impact of these 'integration' features.
- **Significance**: First evidence for separate feature integration encoding space

Introduction

The Linear Superposition Assumption

Current interpretability approaches fundamentally assume that neural network representations follow a linear superposition model [7], where each activation can be decomposed into a sparse combination of interpretable features:

$$\text{neural\_activation} = w_1 \times \text{feature}_1 + w_2 \times \text{feature}_2 + w_3 \times \text{feature}_3 + \ldots$$

This assumption underlies the success of Sparse Autoencoders (SAEs) [1], which have demonstrated remarkable ability to discover [1,3] interpretable features and achieve high reconstruction fidelity on neural activations [1,2, 8]. Within this framework, non-orthogonal feature representations are viewed as interference or compression artifacts—necessary evils that arise when neural networks attempt to represent more features than they have dimensions.

[1]Independent Researcher, <omarclaflin@gmail.com>

However, a fundamental puzzle remains: despite achieving high reconstruction fidelity, SAEs consistently fail to eliminate polysemantic features that respond to seemingly unrelated concepts [1,3,8]. If linear superposition fully captures neural computation, why do the same polysemantic patterns [3,8,18] appear robustly across different models and scales [8]? This persistence suggests that our current understanding may be incomplete.

Recent work has highlighted systematic limitations in sparse coding approaches [4,5]. Gurnee et al. demonstrated that SAE reconstruction errors are pathological rather than random, indicating missing computational structure [4]. The mechanistic interpretability literature describes substantial "dark matter" [5,6]—neural computation that remains unexplained even after extensive circuit analysis [14]. These findings collectively suggest that the linear superposition assumption may be insufficient to capture the full complexity of neural representations.

Dual Encoding Hypothesis

We propose that neural networks encode information in two complementary spaces that are compressed into the same neural substrate:

Feature Identity Space: Represents *what concepts are present* in the input. This corresponds to the sparse features successfully captured by current SAE approaches—interpretable concepts like "Paris," "democracy," or "positive sentiment" that can be identified and measured independently.

Feature Integration Space: Represents *how concepts combine computationally* to produce emergent meanings and behaviors. These are the relationships between features that cannot be captured by linear combinations—the computational patterns that determine how "surprise" + "birthday" produces joy while "surprise" + "diagnosis" produces anxiety.

This dual encoding framework reframes non-orthogonal representations not as interference to be eliminated, but as computational structure encoding meaningful relationships between concepts. The persistent polysemanticity observed in neural networks may reflect the *compression of both identity and integration information into the same representational space*, rather than mere artifacts of insufficient capacity.

Finally, this dual encoding hypothesis is distinct from existing analyses of feature relationships through co-activation patterns or similarity metrics, which capture statistical correlations between features across datasets. Instead, we focus on computational interactions—how features combine to produce emergent meanings that cannot be predicted from their individual activation patterns or co-occurrence statistics. While similarity analysis might reveal that "fire" and "hearth" (or “fire” and “forest”) features often appear together, integration analysis reveals how their combination computes concepts like "warmth/comfort" (or “destruction/emergency”) with behavioral consequences that emerge only from their joint activation (which may be (1) *nonlinearly interactive*, on (2) *underspecified features* existing as combinations of more atomic

features, versus the typical *king - man + woman = queen*, *additive* interactions through *defined* features).

Neural Compression and Computational Structure

Under this view, neural networks face a fundamental compression challenge: they must encode both the identity of relevant features and the computational relationships between them within limited representational capacity. A neuron that responds to both "late" and "party" (or “late” and “meeting”) concepts may not simply be storing two unrelated features due to capacity constraints—it may be computing something about their relationship, such as "fashionable" or "problematic."

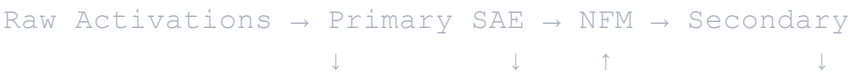
This perspective offers a unified explanation for several puzzling phenomena in neural network interpretability: why polysemantic neurons are so robust across models, why certain feature combinations consistently appear together, and why interventions on individual features often produce complex, context-dependent effects, and why features exhibit sharp phase transitions during training as they crystallize from integration patterns into dedicated representations. If neurons encode computational relationships alongside feature identities, these observations become natural consequences of the underlying representational structure rather than obstacles to overcome.

The implications extend beyond interpretability to fundamental questions about neural computation. If feature integration represents a distinct form of neural encoding, current approaches that focus solely on feature decomposition may be missing crucial computational mechanisms. Understanding these integration patterns could provide insights into how neural networks (and potentially, biological neural networks [1]) perform complex reasoning, maintain contextual coherence, and exhibit emergent capabilities that cannot be explained by simple feature combinations.

Methods

Experimental Pipeline

Our methodology decomposes neural representations into two complementary encoding spaces through a three-stage pipeline: (1) sparse feature extraction via SAE training, (2) integration pattern capture using Neural Feature Models (NFMs) trained on SAE reconstruction residuals, and (3) integration space analysis through secondary SAE decomposition of NFM embeddings. *Detailed methods* at the end of the document.



Model and Data Configuration

We conducted experiments using OpenLLaMA-3B with activations extracted from layer 16 (middle layer). The model was evaluated on WikiText-103, with tokenized sequences processed in 50-token windows. All experiments utilized a single NVIDIA RTX 3090 GPU with 24GB VRAM and 128GB system RAM.

Stage 1: Sparse Autoencoder Training

We trained a 50,000-feature TopK SAE achieving 0.136 reconstruction loss and 86.4% variance explained.

Stage 2: Neural Feature Model Architecture

NFMs capture feature integration patterns by predicting SAE reconstruction residuals:

```
residual = x_original - SAE(x_original)
residual_pred = NFM(SAE_features)
```

Architecture: Neural Factorization Machine with linear + interaction components

Linear Component:

$$\text{linear\_output} = \Sigma_i \, w_i \times f_i + b$$

Interaction Component:

$$\text{interaction\_output} = 0.5 \times (\Sigma_i \, v_i f_i)^2 - \Sigma_i (v_i f_i)^2$$

where $v_i \in \mathbb{R}^k$ represents learned embedding vectors for feature i

The NFM was trained on 5 million tokens using Adam optimization (lr=1e-4) with K=300 embedding dimensions. This achieved 23.4% error reduction over the base SAE, with linear components contributing 95.5% and interactions contributing 4.5% of the improvement.

Stage 3: Integration Space Analysis

To analyze the computational structure captured by NFMs, we applied secondary TopK SAEs to the NFM interaction pathway, specifically targeting post-MLP1 vectors before ReLU activation. The secondary SAE used 25× expansion (300 → 7,500 features) with K=150 active features.

Validation methodology: We implemented 2×2 factorial stimulus designs (formal/informal × emotional/neutral) with systematic intervention experiments. Secondary SAE features were ranked by activation variance across experimental conditions, then subjected to clamping interventions at multiple levels (0×, 1×, ±4×). Behavioral effects were measured through logit changes for category-relevant vocabulary sets.

Stage 4: Experimental Validation

Intervention testing: We validated integration features through systematic clamping of both primary SAE features (via linear weight manipulation) and secondary SAE features (direct activation clamping). Effects were measured using logit differential analysis across formality and emotion vocabulary categories.

Controls: Secondary SAE applied directly to original residuals, random feature permutations, and linear-component-only variants served as baseline comparisons.

Results

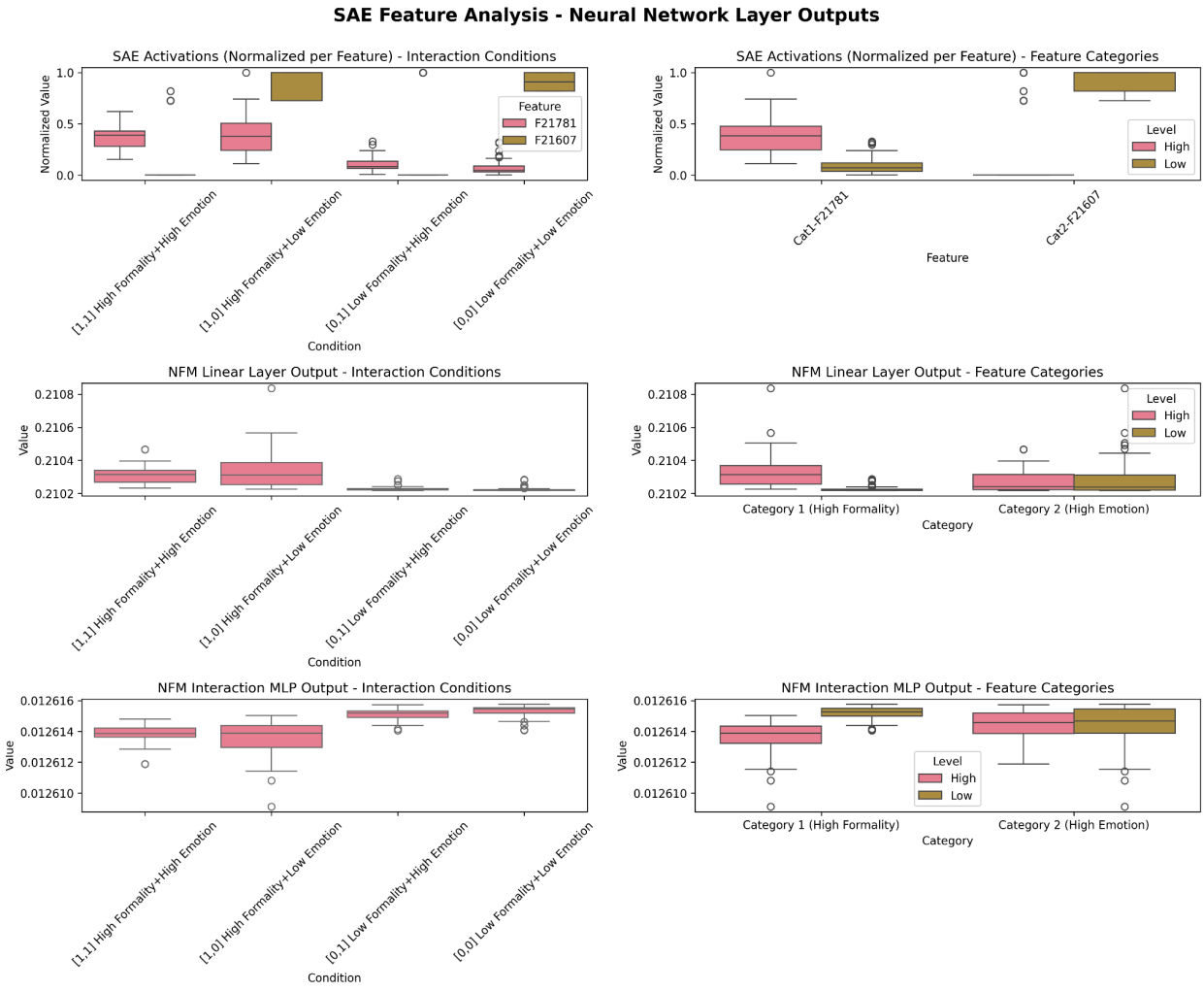
Quantitative Reconstruction Evidence

The Neural Factorization Machine approach achieved substantial improvements over sparse autoencoder baselines. Training on 5 million tokens, the combined SAE+NFM system demonstrated 23.18% error reduction on training data and 23.43% error reduction on validation data compared to SAE-only reconstruction which constrained our top K features to the top 250 features (train: 0.3813 → 0.2930; validation: 0.3672 → 0.2811).

Component analysis revealed that linear combinations dominated the improvement, contributing 95.5% of the correction magnitude (linear: 0.2773, interaction: 0.0130), while higher-order (non-linear) interaction effects accounted for 4.5%. This suggests that NFMs capture both underspecified feature combinations that should be learnable by larger SAEs and genuinely non-linear integration patterns that cannot be captured through sparse coding approaches.

Feature Specificity in Integration Space

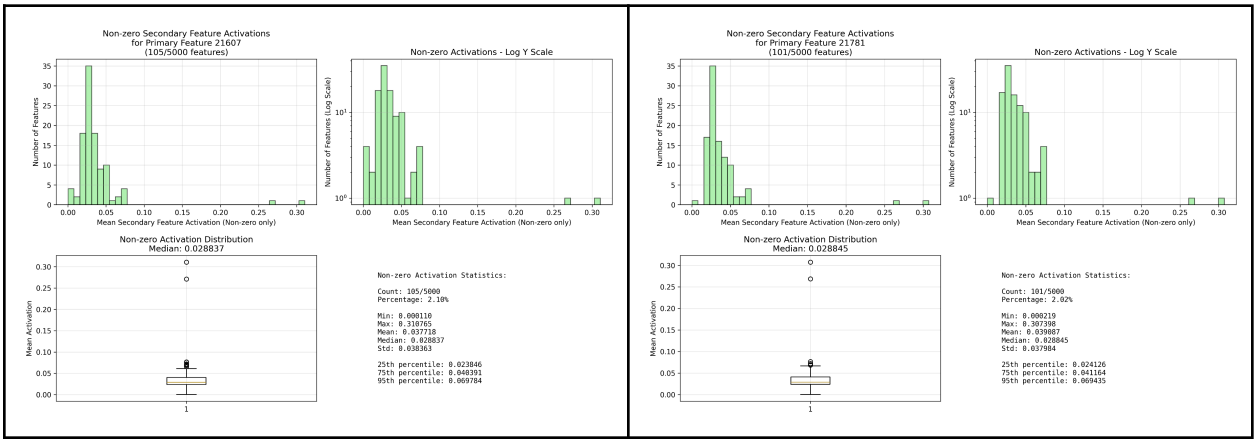
Using a stimulus-driven discovery approach, we identified primary SAE features responding to orthogonal semantic dimensions: **Feature 21607 (Emotion)** and **Feature 21781 (Formality)** were selected based on maximal t-test differences across each stimulus conditions directly (e.g. collapsing across Emotion to examine High vs Low Formality, etc). Below shows the (1) Primary SAE activations on the stimulus-discovered features, (2) the impact on the NFM Linear layer (mean), and (3) the impact on the final embedding of the NFM interactive Layer (mean).



Secondary SAE analysis on the NFM integration pathway revealed selective feature activation patterns. Among 7,500 secondary features, we identified features with distinct sensitivity profiles:

- **Feature 4022**: Highest ANOVA sensitivity ($F=26.72$, $p=2.85\times10^{-9}$) across experimental conditions
- **Feature 2020**: Highest activation in [formal,emotional] conditions (activation=0.517) but less interaction effects than 4022.
- **Counterexample features**: Feature 1113 showed no ANOVA sensitivity ($F=0.022$, $p=0.996$), and Feature 31 showed no activation differences across conditions

Distribution analysis across secondary features revealed a bimodal pattern: most features showed zero contribution to the primary feature dimensions, while a smaller subset exhibited normal distributions around meaningful contribution levels, with our target features appearing as outliers in the high-contribution tail. This pattern held across multiple analysis methods (max activation, max difference, ANOVA sensitivity).



Note: The overwhelming majority of secondary feature activations (for each stimulus set shown here) are zero and excluded from the graph. The intent is to show the stimulus-discovered primary SAE features are outlier features for each category. While this is not demonstrative of monosemanticity, it shows some level of specificity.

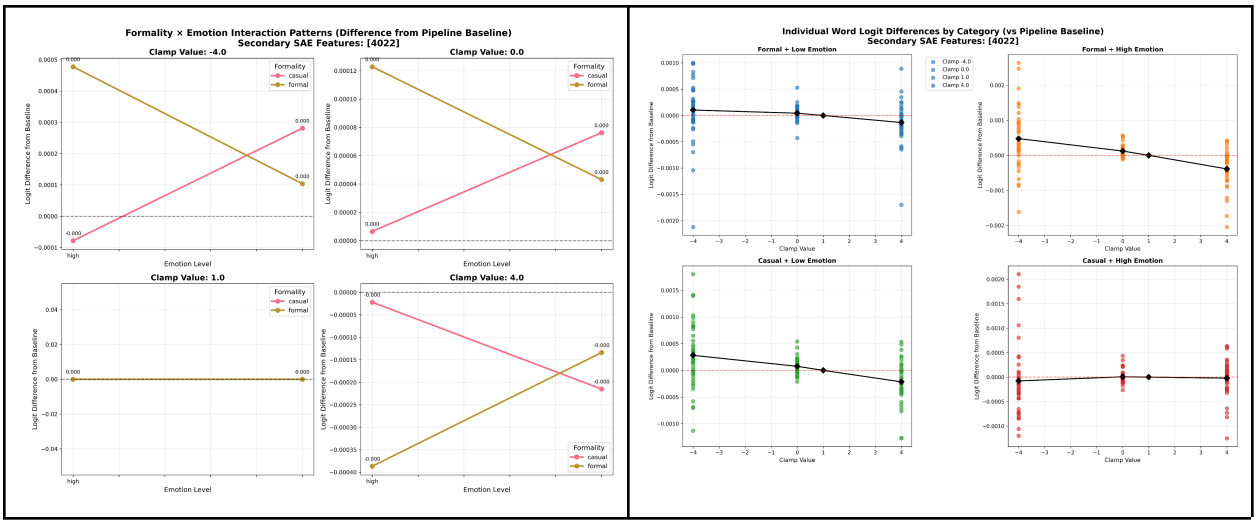
Intervention Validation and Behavioral Effects

Systematic clamping experiments on Feature 4022 demonstrated selective behavioral effects across vocabulary categories. Using clamping multipliers of [-4×, 0×, 4×], we measured logit changes for categorized word sets (generated by Claude):

- **Formal/low-emotion**: "perhaps," "therefore," "consequently"
- **Formal/high-emotion**: "profoundly," "devastated," "extraordinary"
- **Casual/low-emotion**: "yeah," "basically," "whatever"
- **Casual/high-emotion**: "totally," "literally," "absolutely"

Statistical validation confirmed significant interaction effects (F=5.06, **p=0.027** for formality×emotion interaction), demonstrating that Feature 4022 clamping produced *non-additive effects* across the 2×2 semantic space rather than simple main effects.

Additionally, significant effects were seen on the other clamp values (-4.0: p=0.0273, 0.0 p=0.0257, 4.0 p = 0.0397), along with a differential impact of interaction effects by clamping (-4.0: 7e-4, 0.0: 1e-4, 4.0: 4e-4). Interestingly, main effects for formality and emotion were not seen at most clamping levels, but our sample size was small.



Mean effects and scatter plots across the three clamping levels (-4x,0x,4x). Note that the presence of an interaction means the clamping of the secondary feature differentially impacts one of the subgroups (4x: high emotion, formal; -4x: low emotion, casual) above the rest, regardless of the main effects of emotion and formality.

Control experiments validated specificity:

- *Linear component clamping*: No interaction effects observed, as expected, from our linear layers, when comparing clamped linear NFM weights vs baseline, across our logit groups.
- *Non-sensitive feature clamping*: Feature 1113 showed no systematic patterns across categories (F>0.28, p>0.18, for all clamping ranges)

Limitations of Traditional Feature Discovery

Discovery-oriented feature identification approaches are typically shown in interpretability experiments which has the advantage of being scalable and fairly objective, versus *stimulus-oriented* feature identification. While several discovery-oriented approaches were attempted (e.g. finding primary features that contribute to a K element or secondary SAE feature with the highest Gini coefficients, then interrogating that secondary feature for top other contributing primary features), we ran into issues with secondary feature identification. Relatively clean primary features could be identified in our primary SAE (e.g. “Paris”, “folk”; with the identified word being in all our top discovered examples), along with corresponding secondary feature indices in which they (1) differentially contributed maximally to compared to other primary features, (2) demonstrated an interaction, (3) maximally conjoint activated, or other approaches, showing an effect on our secondary feature activation, we ran into issues identifying what the secondary feature *meant.* Unfortunately, the behavioral output of our small Llama model was not reliable, even when using a primary SAE alone (with a relatively

reasonable reconstruction loss of 0.11) often producing incomprehensible output. For these reasons, our primary approach turned towards measuring impact on a set of pre-established target logits, with manipulation of our secondary feature compared to baseline, within our 2x2 analysis framework. Ideally, this will be improved upon in the future either through scaling up our framework on larger models, or alternate interpretation approaches.

Maximum activation analysis on secondary features consistently returned conjunctive tokens ("that") or punctuation features (“)”) rather than interpretable semantic patterns. This failure occurred despite clear functional effects demonstrated through intervention experiments, suggesting that integration features may not correspond to simple activation maxima in natural text.

Primary feature interpretability also proved challenging in the *stimulus-driven* workflow, with selected features lacking the clean monosemantic examples typically found in traditional SAE analysis. However, the systematic behavioral effects of these features under controlled interventions provide functional validation independent of activation-based interpretability.

These results provide some converging evidence that neural networks encode feature integration patterns alongside feature identity, with integration features exhibiting selective sensitivity to experimental manipulations and producing systematic behavioral effects despite their opacity to traditional discovery methods.

Discussion

Implications for Neural Computation

Our findings challenge the prevailing view of neural networks as sparse feature storage systems, revealing instead a *dual encoding architecture* where neural representations simultaneously compress both feature identity and feature integration information. The reconstruction improvement achieved by capturing integration patterns demonstrates that current sparse coding approaches [11,12], while successful at identifying interpretable features, systematically miss computational structure [4,5] that is functionally significant for model behavior.

Polysemantic neurons may not represent compression artifacts to be eliminated, but rather computational units that encode relationships between concepts. The selective intervention effects we observed—where some features produce systematic 2×2 interaction patterns while others show no effects—suggest that polysemanticity may reflect meaningful computational roles rather than random interference patterns. This reframes the persistent polysemanticity

observed even in high-capacity SAEs from a limitation to be overcome to evidence of fundamental computational organization.

The ***dynamic encoding landscape*** we propose explains several puzzling phenomena: feature integrations represent computational relationships between established features that, with sufficient frequency and available encoding space, can themselves become codified as distinct feature identities. This creates a continuous spectrum where today's integration pattern may become tomorrow's sparse feature, explaining both the continuing variance gains in larger SAEs and the persistent polysemanticity observed even at scale.

Relation to Existing Work

Our framework provides a possibly unifying explanation for several limitations identified in current interpretability research. The *"dark matter"* described in circuit analysis [5,6]—computation that remains unexplained despite extensive feature identification [14,15]—may largely reflect missing integration patterns rather than inadequate feature discovery. Our demonstration that traditional maximum activation approaches fail to identify integration features, despite their clear functional effects, suggests that current interpretability methods may be systematically blind to this form of computation.

The *systematic reconstruction errors* identified by Gurnee et al. [4] find a natural explanation within our dual encoding framework: these errors reflect missing integration structure rather than random noise or capacity limitations. The pathological nature of these errors—their non-random, structured character—aligns with our finding that integration patterns exhibit selective sensitivity and systematic behavioral effects.

Our work also addresses the *"wrong abstraction level"* problem frequently encountered in SAE research [9, 10, 21], where features appear either too specific or too general for interpretable analysis. Under our framework, this may reflect the artificial separation of identity and integration encoding: some apparent features may actually be integration patterns, while some apparent integrations may be underspecified identity features awaiting sufficient encoding capacity.

Unlike static feature relationship methods [17] (e.g. cosine similarity analysis between features) that capture co-occurrence patterns, feature integration analysis reveals *computational relationships*—how features combine to produce emergent meanings that cannot be predicted from their individual activation patterns or statistical co-occurrence. This distinction is crucial for understanding the difference between features that merely appear together and features that compute together.

Limitations and Future Work

Scale constraints represent one primary limitation of this work. Our experiments on a 3B parameter model with 50k SAE features provide proof-of-concept evidence, but scaling to industrial-scale models with millions of features remains challenging. The computational

requirements of NFM training scale super-linearly with feature count, necessitating architectural innovations or more efficient approximation methods.

Integration interpretability presents ongoing challenges. While we demonstrated functional effects of integration features through systematic interventions, these features remain largely opaque to direct inspection. The failure of maximum activation analysis (or several other attempted analytical approaches) to yield interpretable patterns for integration features suggests need for specialized interpretability methods designed for computational rather than representational structure.

Methodological extensions could address several current limitations: (1) Dynamic analysis of how integration patterns evolve during training could reveal the mechanisms by which computational relationships crystallize into identity features. (2) Cross-model validation could establish whether specific integration patterns represent universal computational primitives or model-specific artifacts. (3) Cross-layer analysis could demonstrate the dynamics of feature integration as activity gets processed through layers. (4) Application to larger models could test whether the linear/nonlinear interaction split observed here reflects fundamental properties of neural computation or artifacts of limited scale.

Finally, architectural implications of a secondary phenomena that is amenable to newer interpretability approaches warrant investigation. If feature integration represents a fundamental aspect of neural computation, architectural designs could explicitly separate identity and integration encoding rather than compressing both into shared representational space. Such architectures might achieve better interpretability without sacrificing computational capability.

Conclusion

This work provides the first systematic evidence for *dual encoding spaces* in neural network representations, demonstrating that networks compress both feature identity and feature integration into shared neural substrates. Our methodology for detecting and validating integration patterns reveals computational structure invisible to current sparse coding approaches, achieving reconstruction improvement and demonstrating selective behavioral effects through systematic intervention experiments.

The *feature integration framework* offers a path toward a more complete understanding of neural computation by recognizing that networks perform computation with feature relationships, not merely storage and access of distinct feature identities. The persistent polysemanticity observed across models and scales may reflect this fundamental computational organization rather than limitations to be overcome.

Methodological contributions include the first demonstration of separable feature identity and integration encoding, systematic approaches for detecting computational relationships between

features, and potentially stimulus-oriented validation methodologies that may have advantages for establishing functional significance beyond discovery-oriented approaches.

Broader implications extend beyond interpretability to fundamental questions about neural computation, AI safety, and the relationship between artificial and biological neural systems [8,15,16]. Understanding how networks integrate information to produce emergent behaviors is crucial for developing reliable, controllable AI systems and for advancing theories of intelligence itself. As neural networks continue to exhibit capabilities that cannot be explained through feature decomposition alone, recognizing the computational relationships between features becomes essential for understanding how intelligence emerges from the interaction of simpler components. This work establishes integration analysis as a necessary complement to feature identification in the comprehensive understanding of neural computation.

Detailed Methods

Model and Data Configuration

We conducted experiments using OpenLLaMA-3B with activations extracted from layer 16 (middle layer). The model was evaluated on WikiText-103, with tokenized sequences processed in 50-token windows. All experiments utilized a single NVIDIA RTX 3090 GPU with 24GB VRAM and 128GB system RAM.

Stage 1: Sparse Autoencoder Training

We trained a 50,000-feature SAE following established methodologies to achieve a reconstruction fidelity of 0.136 reconstruction loss, 0.864 variance explained.

Architecture and Sparsity Mechanism: We implemented a TopK sparse autoencoder [2] with an encoder-decoder architecture, where the encoder consists of a linear transformation followed by ReLU activation, and the decoder uses a linear transformation without bias. The model was configured with 50,000 hidden features and enforced sparsity through a TopK mechanism (K=1024), which retains only the top 1024 most active features per sample while setting all others to zero. This approach targets approximately 2.05% feature activation, providing a fixed sparsity level that eliminates the need for L1 regularization hyperparameter tuning.

Training Data and Preprocessing: Training data consisted of 1 million tokens from the WikiText-103 dataset, processed in 128-token chunks to capture diverse linguistic contexts. We extracted activations from layer 16 of a LLaMA 3B model and applied z-score normalization using statistics computed from a 10,000-sample subset to ensure numerical stability. The dataset was split into 90% training and 10% validation sets, with activations stored as float32 tensors to maintain precision during training.

Optimization and Training Procedure: The model was trained using the Adam optimizer with a learning rate of 1e-4, β coefficients of (0.9, 0.999), and a batch size of 1024 samples. We employed Kaiming normal initialization for the encoder weights and initialized the encoder as the transpose of the decoder weights to promote symmetric learning. The loss function consisted solely of mean squared reconstruction error, with gradient clipping (max norm = 1.0) and linear learning rate decay to 10% of the initial rate over 80% of training steps. Training proceeded for 500,000 steps with checkpointing every 10,000 iterations, achieving 86.4% variance explained and 0% dead neurons while maintaining the target sparsity level.

Validation: Feature interpretability was verified through maximum activation analysis and manual inspection of top-activating examples for a subset of features.

Stage 2: Neural Feature Model Architecture

Neural Feature Models (NFMs) capture feature integration patterns by predicting SAE reconstruction residuals using the SAE's own feature activations as input:

```
residual = x_original - SAE(x_original)
residual_pred = NFM(SAE_features)
```

Architecture: NFMs employ a factorization machine approach optimized for sparse feature interactions:

Linear Component:

```
linear_output = Σᵢ wᵢ × fᵢ + b
```

Interaction Component:

```
Interaction_vector = Σᵢ<□ ⟨vᵢ, v□⟩ × fᵢ × f□
```

where $v_i \in \mathbb{R}^k$ represents learned embedding vectors for efficient pairwise interaction computation. (Note: This interaction 'math trick' captures all the possible interactions, by subtracting the 'self-interaction' term leaving only the cross-interaction terms. Note: This doesn't imply *only* pairwise interactions remain, but rather all possible interactions: pairwise, three-way, four-way, etc). While this doesn't disentangle pairwise from the other higher order interactions, it cleanly separates high-order from linear, and is a significant speed up time from pairwise ($O(n^2)$) or n-way interactions ($O(2^n)$) to roughly O(nk) time.

```
interaction_output = self.interaction_mlp(interaction_vector)
```

Neural Factorization Machine for Residual Modeling: We implemented a Neural Factorization Machine (NFM) to predict SAE reconstruction residuals using the SAE's own feature activations as input. The NFM architecture combines linear and interaction components: a linear layer mapping the 50,000 SAE features directly to the 3,200-dimensional activation space, and an interaction component using learned embeddings (K=300 dimensions) to capture pairwise feature interactions via the factorization machine formulation. The interaction term is computed as $0.5 \times (\Sigma_i v_i f_i)^2 - \Sigma_i (v_i f_i)^2$, where v_i represents the learned embedding for feature i and f_i is the feature activation.

NFM Training and Optimization: The NFM was trained on 5 million tokens using streaming data processing with 100,000-token chunks to handle large-scale training. We used Adam optimization with learning rate 1e-4, dropout rate 0.15, and batch size 130. Feature embeddings were initialized with standard deviation 0.05, while the linear component used smaller initialization (std=0.01). The model was trained for 300,000 steps with gradient clipping and achieved 23.4% error reduction over the base SAE, with the linear component contributing 95.5% of the correction magnitude and interactions contributing 4.5%.

Combined Architecture Performance: The two-stage approach first trains the TopK SAE to achieve high-quality sparse representations, then trains the NFM to predict reconstruction residuals. This decomposition allows the linear component to handle systematic reconstruction

biases while the interaction terms capture higher-order feature dependencies. The combined system maintains computational efficiency through the sparse SAE features while achieving improved reconstruction fidelity through the residual modeling approach.

Stage 3: Integration Space Analysis Methods

Secondary SAE Training on NFM Embeddings: To analyze the computational structure captured by NFMs, we applied secondary TopK SAEs to the NFM interaction pathway, specifically targeting the post-MLP1 vectors before ReLU activation. The NFM interaction component processes SAE features through learned embeddings (K=300 dimensions) and a multi-layer perceptron, creating a distinct representational space that captures higher-order feature relationships. We trained secondary TopK SAEs with 25× expansion (300 → 7,500 features) and K=150 active features (2% sparsity) using identical methodology to the primary SAE: Adam optimization (lr=1e-4), Kaiming initialization, and gradient clipping. The secondary SAE targets the intermediate representation where feature interactions are computed but before final projection to the output space.

Integration Feature Analysis Pipeline: Our analysis pipeline processes texts through the complete three-pathway architecture: Layer 16 → Primary TopK SAE → [Primary Reconstruction + NFM Linear + NFM Interaction] → Secondary TopK SAE. For feature discovery, we implemented four complementary approaches: (1) Forward activation analysis identifying which secondary SAE features respond most strongly to specific primary SAE features, (2) Backward attribution analysis using gradient-based methods to identify primary feature contributors to secondary feature activations, (3) Stimulus-based differential analysis using 2×2 factorial designs (formal/informal × emotional/neutral text) with 50 examples per condition, and (4) ANOVA-based sensitivity analysis to identify secondary features most responsive to experimental manipulations across all four stimulus conditions.

Experimental Validation and Intervention Testing: We validated the integration space representation through systematic intervention experiments. Secondary SAE features were ranked by activation variance across the 2×2 stimulus factorial, identifying features most sensitive to the experimental manipulations. For selected high-variance features, we implemented clamping interventions at 0×, 1×, and -/+4× natural activation levels during forward passes through the complete pipeline. Behavioral effects were measured by computing logit changes for category-relevant vocabulary sets (e.g., formal vs. informal language markers, emotional vs. neutral terms) and comparing intervention effects across different integration features. This approach allows direct measurement of how integration space features influence model outputs, providing functional validation of the learned representations beyond reconstruction metrics.

Stage 4: Experimental Validation Methods

Feature Intervention Analysis: We validated the functional significance of learned SAE features through systematic clamping experiments targeting both primary and secondary features. For primary SAE features, we implemented interventions via linear weight

manipulation in the NFM's linear pathway, multiplying connection weights by factors ranging from 0× to 5× their baseline values. For secondary SAE features, we applied direct activation clamping within the interaction pathway after MLP layer 1 but before ReLU activation. Each intervention maintained the full three-pathway architecture (primary reconstruction + NFM linear + NFM interaction) while selectively modifying target feature contributions during forward passes.

Logit Differential Analysis: We measured intervention effects through logit lens analysis using categorized vocabulary sets organized across formality and emotion dimensions: formal/low-emotion (e.g., "perhaps," "therefore"), formal/high-emotion (e.g., "profoundly," "devastating"), casual/low-emotion (e.g., "yeah," "basically"), and casual/high-emotion (e.g., "totally," "literally"). For each intervention condition, we computed logit differences relative to unmodified pipeline baselines across five neutral prompts, enabling detection of 2×2 factorial interaction patterns. Statistical significance was assessed using ANOVA across intervention levels, with particular attention to formality×emotion interaction terms indicating feature-mediated integration of semantic dimensions.

Generation and Behavioral Validation: We assessed intervention effects on text generation using prompt sets designed to elicit different formality and emotion combinations. Generation proceeded with modified feature activations maintained throughout the forward pass, producing 50-token completions under each intervention condition. We analyzed both qualitative changes in generated text style and quantitative shifts in next-token probability distributions for target vocabulary categories. This approach directly demonstrated the causal role of identified features in controlling model behavior, complementing the reconstruction-based validation from earlier stages with evidence of functional significance for language generation.

References

[1] Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N. L., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., & Olah, C. (2023). Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread, Anthropic*. https://transformer-circuits.pub/2023/monosemantic-features

[2] Gao, L., Schulman, J., & Hilton, J. (2024). Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*. https://arxiv.org/abs/2406.04093

[3] Cunningham, H., Ewart, T., Riggs, L., Huben, R., & Sharkey, L. (2023). Sparse Autoencoders Find Highly Interpretable Features in Language Models. *arXiv preprint arXiv:2309.08600*. https://arxiv.org/abs/2309.08600

[4] Gurnee, W. (2024). SAE reconstruction errors are (empirically) pathological. *AI Alignment Forum*. https://www.alignmentforum.org/posts/rZPiuFxESMxCDHe4B/sae-reconstruction-errors-are-empirically-pathological

[5] Engels, J., Liao, I., Michaud, E. J., Gurnee, W., & Tegmark, M. (2024). Decomposing The Dark Matter of Sparse Autoencoders. *arXiv preprint arXiv:2410.14670*. https://arxiv.org/abs/2410.14670

[6] Olah, C. (2023). Interpretability Dreams. *Transformer Circuits Thread*. https://transformer-circuits.pub/2024/july-update/index.html

[7] Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wortsman, M., & Ludwig, J. (2022). Toy Models of Superposition. *Anthropic Research*. https://transformer-circuits.pub/2022/toy_model/index.html

[8] Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Tamkin, A., Durmus, E., Hume, T., Mosconi, F., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., & Henighan, T. (2024). Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread, Anthropic*. https://transformer-circuits.pub/2024/scaling-monosemanticity/

[9] Chanin, D., Shlegeris, B., & Brundage, M. (2024). A is for Absorption: Studying Feature Splitting and Absorption in Sparse Autoencoders. *arXiv preprint arXiv:2409.14507*. https://arxiv.org/abs/2409.14507

[10] Makelov, A., Sharma, M., Tong, M., Hernandez, E., Braun, J., Pehlevan, C., & Tegmark, M. (2024). Towards Principled Evaluations of Sparse Autoencoders for Interpretability and Control. *arXiv preprint arXiv:2405.08366*. https://arxiv.org/abs/2405.08366

[11] Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583), 607-609. https://doi.org/10.1038/381607a0

[12] Field, D. J. (1994). What is the goal of sensory coding? *Neural Computation*, 6(4), 559-601. https://doi.org/10.1162/neco.1994.6.4.559

[13] Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature Visualization. *Distill*, 2(11), e7. https://doi.org/10.23915/distill.00007

[14] Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020). Zoom In: An Introduction to Circuits. *Distill*, 5(3), e00024.001. https://doi.org/10.23915/distill.00024.001

[15] Sharkey, L., Braun, D., & Millidge, B. (2025). Open Problems in Mechanistic Interpretability. *arXiv preprint arXiv:2501.16496*. https://arxiv.org/abs/2501.16496

[16] Bereska, L. F., & Gavves, E. (2024). Mechanistic Interpretability for AI Safety — A Review. *arXiv preprint arXiv:2407.11215*. https://arxiv.org/abs/2407.11215

[17] Park, K., Ro, Y., Liu, H., & Kim, J. (2024). The Linear Representation Hypothesis and the Geometry of Large Language Models. *arXiv preprint arXiv:2311.03658*. https://arxiv.org/abs/2311.03658

[18] Chen, S., Trojanowski, S., Karpinska, M., Pavlick, E., & Bowman, S. R. (2024). Taming Polysemanticity in LLMs: Provable Feature Recovery via Sparse Autoencoders. *arXiv preprint arXiv:2506.14002*. https://arxiv.org/abs/2506.14002

[19] Tamkin, A., Jurafsky, D., & Goodman, N. (2023). Codebook Features: Sparse and Discrete Interpretability for Neural Networks. *arXiv preprint arXiv:2310.17230*. https://arxiv.org/abs/2310.17230

[20] Bussmann, B., Treutlein, J., Shlegeris, B., Lièvre, J., Emmons, S., Roger, A., & Nanda, N. (2024). Learning Multi-Level Features with Matryoshka Sparse Autoencoders. *AI Alignment Forum*. https://www.alignmentforum.org/posts/rKM9b6B2LqwSB5ToN/learning-multi-level-features-with-matryoshka-saes

[21] Ayonrinde, K., Shah, R., Fry, S., Winsor, E., Gurnee, W., Tegmark, M., & Krueger, D. (2024). Interpretability as Compression: Reconsidering SAE Explanations of Neural Activations with MDL-SAEs. *arXiv preprint arXiv:2410.11179*. https://arxiv.org/abs/2410.11179

[22] Casper, S. (2024). EIS XIII: Reflections on Anthropic's SAE Research Circa May 2024. *AI Alignment Forum*. https://www.alignmentforum.org/posts/pH6tyhEnngqWAXi9i/eis-xiii-reflections-on-anthropic-s-sae-research-circa-may

[23] Lee, S., & Heimersheim, S. (2024). Investigating Sensitive Directions in GPT-2: An Improved Baseline and Comparative Analysis of SAEs. *LessWrong*. https://www.lesswrong.com/posts/dS5dSgwaDQRoWdTuu/investigating-sensitive-directions-in-gpt-2-an-improved

[24] Burns, C., Ye, H., Klein, D., & Steinhardt, J. (2022). Discovering Latent Knowledge in Language Models Without Supervision. *arXiv preprint arXiv:2212.03827*. https://arxiv.org/abs/2212.03827

[25] Nanda, N., Chan, L., Lieberum, T., Smith, J., & Steinhardt, J. (2023). Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*. https://arxiv.org/abs/2301.05217