Sparsity and Superposition in Mixture of Experts

Anonymous authors
Paper under double-blind review

Abstract

Mixture of Experts (MoE) models have become central to scaling large language models, yet their mechanistic differences from dense networks remain poorly understood. Previous work has explored how dense models use *superposition* to represent more features than dimensions, and how superposition is a function of feature sparsity and feature importance. MoE models cannot be explained mechanistically through the same lens. We find that neither feature sparsity nor feature importance cause discontinuous phase changes, and that network sparsity (the ratio of active to total experts) better characterizes MoEs. We develop new metrics for measuring superposition across experts. Our findings demonstrate that models with greater network sparsity exhibit greater *monosemanticity.* We propose a new definition of expert specialization based on monosemantic feature representation rather than load balancing, showing that experts naturally organize around coherent feature combinations when initialized appropriately. These results suggest that network sparsity in MoEs may enable more interpretable models without sacrificing performance, challenging the common assumption that interpretability and capability are fundamentally at odds.

1 Introduction

Mixture of Experts (MoEs) have become prevalent in state-of-the-art language models, such as Qwen3, Mixtral, and Gemini (Yang et al., 2025a; Jiang et al., 2024; Google DeepMind, 2025), primarily for their computational efficiency and performance gains (Shazeer et al., 2017; Fedus et al., 2022). Subsequent work improve routing (e.g., Expert-Choice routing) and training stability/transfer (e.g., ST-MoE) (Zhou et al., 2022; Zoph et al., 2022). Theoretical and empirical results further show that learnable routers can discover latent cluster structure in data, providing insight for why experts specialize (Dikkala et al., 2023). However, despite their widespread adoption, MoEs remain poorly understood from a mechanistic interpretability perspective.

Interpretability-oriented approaches have sought to make expert behavior more transparent. Yang et al. (2025b) proposes MoE-X, which encourages sparsity-aware routing and uses wide, ReLU-based experts to reduce polysemanticity. Park et al. (2025) introduce Monet, scaling the number of experts to enable capability editing via expert activation. Yet these works largely focus on architectural changes; we still lack a mechanistic understanding of how MoEs represent features, how experts affect superposition, and whether experts naturally specialize without extra regularization. Mu & Lin (2025) survey MoE research and identify mechanistic interpretability as a key open challenge.

A fundamental challenge in interpreting neural networks is the phenomenon of superposition: when models represent more features than they have dimensions. This allows networks to pack many sparse features into fewer neurons at the cost of making individual neurons polysemantic and difficult to interpret.

MoE architectures introduce a new dimension to this problem: network sparsity. Unlike dense models that activate all neurons regardless of input, MoEs activate a fraction of their total parameters (Shazeer et al., 2017). While dense models exploit feature sparsity by packing many sparse features into shared neurons, MoEs can afford to be more selective, potentially dedicating entire experts to specific feature combinations.

We investigate whether (1) MoEs exhibit less superposition than their dense counterparts, (2) there is a discrete phase change in MoE experts as seen in dense models, and (3) we can understand expert specialization through the lens of feature representation rather than just load balancing.

We explore these questions using simple models that extend Elhage et al. (2022)'s framework to MoEs. Our key contributions are as follows: (1) unlike dense models, MoEs do not exhibit sharp phase changes, instead showing more continuous transitions as network sparsity increases; and (2) MoEs consistently exhibit greater monosemanticity (less superposition) than dense models with equivalent active and total parameters, with individual experts representing features more cleanly; (3) we propose an interpretability-focused definition of expert specialization based on monosemantic feature representation, showing that experts naturally organize around coherent feature combinations rather than arbitrary load balancing.

2 Background

Learned representations of meaningful 'features'—loosely, those ideas that have human-interpretable meaning—are often assumed to be linear in terms of activations (Gorton & Lewis, 2025). Thus, measuring activations directly corresponds to the intensity of meaningful features. The *superposition hypothesis* contends that models are capable of representing far more features than dimensions (Elhage et al., 2022).

In order to have many more features than dimensions in a latent space, features vectors must be packed such that they are not all orthogonal. The Johnson–Lindenstrauss lemma proves that this can be done efficiently by allowing for small amounts of interference. This is an acceptable compromise because most features are extremely sparse (i.e., they are active on only a tiny fraction of inputs). For example, in the complete corpus of *all* language text, few sentences have to do with the feature of 'Martin Luther King, Jr.' Nonlinearities and bias terms allow models to account for moderate interference.

Monosemantic features are defined as those that are well-aligned with individual neurons: a single neuron's activation cleanly reflects the presence or absence of the feature. Superposition, by contrast, refers to representations where multiple features are represented within the same set of neurons, such that no single neuron corresponds to a single meaningful feature. In this setting, neurons and features are polysemantic: a neuron's activation reflects a linear combination of many features, and a feature is distributed across many neurons. While superposition is efficient for capacity and generalization, it makes interpretation challenging, since observations and interventions on single neurons no longer correspond cleanly to changes in single features.

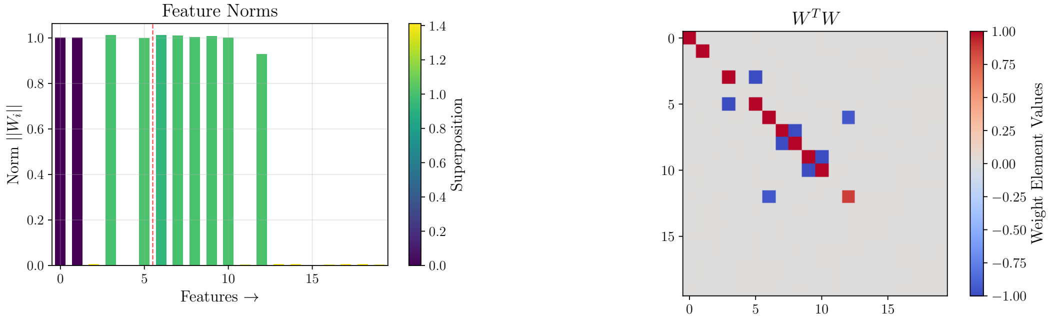
3 Demonstrating Superposition

An intuitive way to think about MoEs is as composition of dense (feed forward) models, where each expert behaves like a dense model in isolation. But it is unclear how to quantify or justify this claim in terms of the mechanistic or representational similarity. We investigated this question by exploring how models vary their representations in terms of superposition.

To test whether MoEs exhibit more or less superposition than the dense models, we extended the standard toy models of Elhage et al. (2022) to include MoE variants, which allows us to exactly measure superposition. It also allows us to characterize superposition as a function of feature density ($1 -$ sparsity) and the ratio of active experts (k) to the total number of experts in a MoE (E), i.e. *network sparsity.*

3.1 Toy Models

Our MoE model consists of E experts, each with m hidden dimensions. The input features, n, are routed to the top-k experts as per the learned router $g(x) = \text{softmax}(W_r x)$, where $W_r \in \mathbb{R}^{n \times E}$. Each expert e processes the input through $h^e = W^e x$, followed by the reconstruction $x'^e = \text{ReLU}\big((W^e)^\top h^e + b^e\big)$. The final output is a weighted combination of the active experts given as $x' = \sum_e w_e \, x'^e$ where w^e are the renormalized gating weights for the top-k selected experts. No load balancing loss is used for this section so that the overall loss simplifies to a reconstruction loss.



(a) Norm of each feature's weight vector $\|W_i\|$, with colors indicating superposition status (green for features in superposition, purple for monosemantic features).

(b) $W^\top W$ matrix where each cell represents $(\hat{W}_i \cdot W_j)$, revealing interference patterns between features.

Figure 1: Feature representation and superposition in a dense model with $n = 20$ features and $m = 6$ hidden dimensions, with importance $I = 0.7^i$ and uniform feature density $(1 - S) = 0.1$.

3.2 Measuring feature capacity

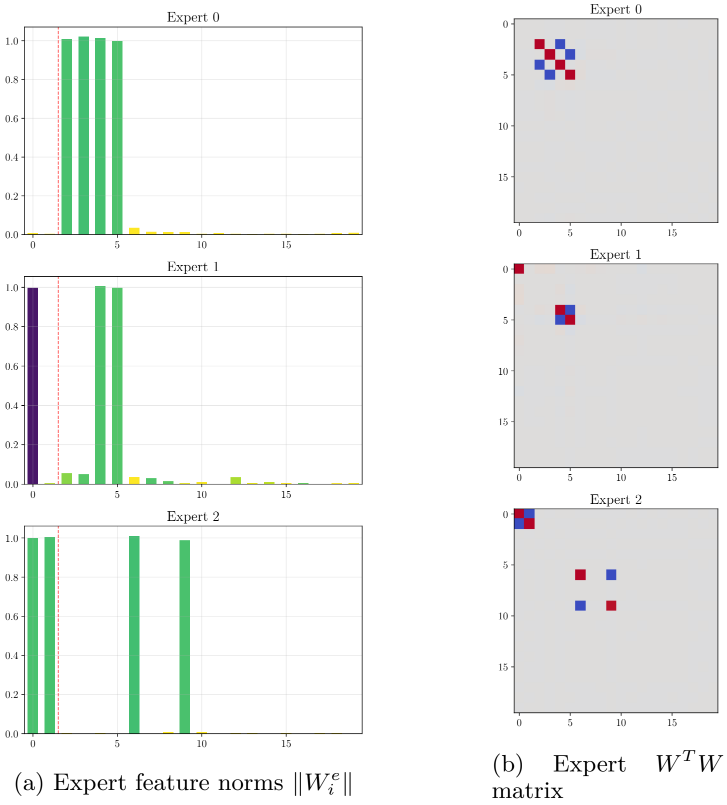
To analyze feature representations across architectures, we compared two fundamental properties of the features: *representation strength* and *interference* with other features. We measured the norm of a feature weight vector in an expert e given by $\|W_i^e\|$. It represents the extent to which a feature is represented within the expert e. $\|W_i^e\| \approx 1$ if feature i is fully represented in expert e and zero if it is not learned. We also calculated the interference of a feature i with other features in expert e by $\sum_{j \neq i} (\hat{W}_i^e \cdot W_j^e)^2$ where $\hat{W}_i^e$ is the unit vector in the direction of W_i^e.

As shown in Figures 1a and 2a, the dense and the MoE represent roughly the same number of features with similar norms for equal total parameters. But the experts in a MoE exhibit far less interference with other features than the dense model as observed in Figures 1b and 2b. This demonstrates that the MoEs allocate their representational capacity in a different way than the dense models even though they represent the same number of features with similar representational strength.

We want to understand how MoEs allocate their limited representation capacity differently from the dense model. We measured feature dimensionality, which represents the "fraction of a dimension" that a specific feature gets in a model (Elhage et al., 2022). For a feature i, we define its dimensionality in expert e by

$$D_i^e = \frac{\|W_i^e\|^2}{\sum_j \left(\hat{W}_i^e \cdot W_j^e\right)^2}$$

D_i^e is bounded between zero (not learned) and one (monosemantic). The total capacity for a MoE can thus be defined as $D = \sum_e \sum_{i=1}^n D_i^e$. When the features are "efficiently packed" in a model's representation space, the dimensionality of all the features add up to the number of embedding dimensions, i.e. $\sum_{i=1}^n D_i^e \approx m$ (Cohen et al., 2014; Scherlis et al., 2025; Elhage et al., 2022). In the case of a MoE, the relation becomes $\sum_e \sum_{i=1}^{n} D_i^e \approx E \cdot m$. Empirically, we find that both dense and MoE models satisfy the above dimensionality constraint, meaning that MoEs achieve the same efficiency in packing features as the dense models for the same total parameters.



(a) Expert feature norms $\|W_i^e\|$

(b) Expert $W^T W$ matrix

Figure 2: Feature representation and superposition in a MoE with $n = 20$ features, 3 total experts, and $m = 2$ hidden dimensions per expert (top-$k = 1$ routing), with importance $I = 0.7^i$ and feature density $1 - S = 0.1$.

Since both dense and MoE models "efficiently pack" features in their representation space, we compared the differences in number of features per dimension across the models. This allowed us to exactly measure superposition in both models and how the number of experts in a MoE affects superposition for different feature sparsities. If the features per dimension is greater than one, then the features are in superposition since the model is representing more features than there are dimensions. We define features per dimension for a MoE by

$$\frac{1}{k} \sum_{e=1}^{E} p_e \frac{\|W^e\|_F^2}{m}$$

where $||W||_F^2$ is the Frobenius norm and p_e is the expected probability that expert e is used across a batch of input samples, i.e. the average renormalized gating weight after top-k routing.

For MoEs with *equal* number of total parameters as the dense model, we observe that the dense model has a higher number of features per dimension (Figure 3), i.e. more superposition (see Appendix A.2 for an intuition). Furthermore, as we increase the total number of experts in the MoE—keeping the total parameters and the ratio k/E roughly the same—the number of features per dimension decreases or alternatively has less superposition. *The greater the number of experts, the less superposition in the model.* Concretely, features become more monosemantic with increasing number of experts. Furthermore, more superposition in the dense model allows it to achieve consistently lower reconstruction loss compared to the MoEs as shown in Figure 7 in Appendix A.1 with difference in loss at any given sparsity of $\sim 0.03 - 0.08$. But as the number of experts increases, the MoEs achieve consistently comparable loss to the dense models. This shows that in our toy models *for an equal total number of parameters, MoEs represent the same number of features as the dense model, but more monosemantically, with only a negligible difference in loss.*

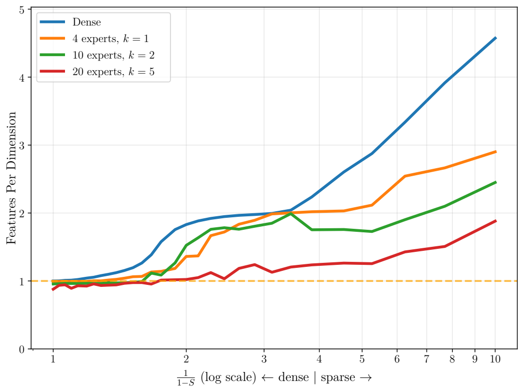


Figure 3: Features per dimension versus inverse feature density ($\frac{1}{1-S}$) for dense and MoE architectures with uniform feature importance ($I_i = 1.0$). The dense model ($n = 100$, $m = 20$) has the most superposition, which decreases with increasing expert count: 4 experts with $m = 5$, $k = 1$ (orange); 10 experts with $m = 2$, $k = 2$ (green); 20 experts with $m = 1$, $k = 5$ (red). All models have equal total parameters and similar k/E. The dashed line at 1.0 marks monosemantic representation.

4 Phase Change

Although MoEs and dense models learn a similar number of features, MoEs distribute them across experts with less interference. This suggests that network sparsity reshapes how features are allocated rather than how many are learned. We examined how properties of the input distribution—such as feature sparsity and importance—drive this allocation and whether they induce physics-inspired *phase changes* in representation.

Models have a finite way of representing features; each feature may be ignored, superimposed, or monosemantic. Phase change is the observation that sometimes there are discrete boundaries between regions, which are functions of feature sparsity and relative importance.

Dense toy models exhibit discontinuous 'phase changes' between internal feature representations (Elhage et al., 2022). By varying the sparsity and relative importance of features in the input distribution, we can elicit different behavior; for example, more feature sparsity encourages greater superposition. Analyzing the phase diagram of each expert in MoEs demonstrates they employ different representational strategies compared to dense models.

We follow the same setup as Section 3.1. We sample data distributions such that each feature x_i has feature sparsity $S \in (0, 1]$ and the last feature rx_{-1} has relative importance $r \in \mathbb{R}^+$. Feature sparsity governs the likelihood a particular input feature dimension is zero. The relative importance is a scalar on the magnitude of the last feature, so $x \in \{x_1, x_2, ...rx_n\}$: $x_i \in U(0, 1)$ with S likelihood that $x_i = 0$.

We report the expert-specific phase diagram across all feature sparsity and last-feature relative importance for varying network sparsity by increasing the experts (E) up to the number of input feature dimensions (n). In this section we fix active parameters rather than total parameters.

In all single-expert (dense) cases, we observed a clear phase change (Figure 4.X.1/1), affirming the work of Elhage et al. (2022). When we increased the total number of experts,

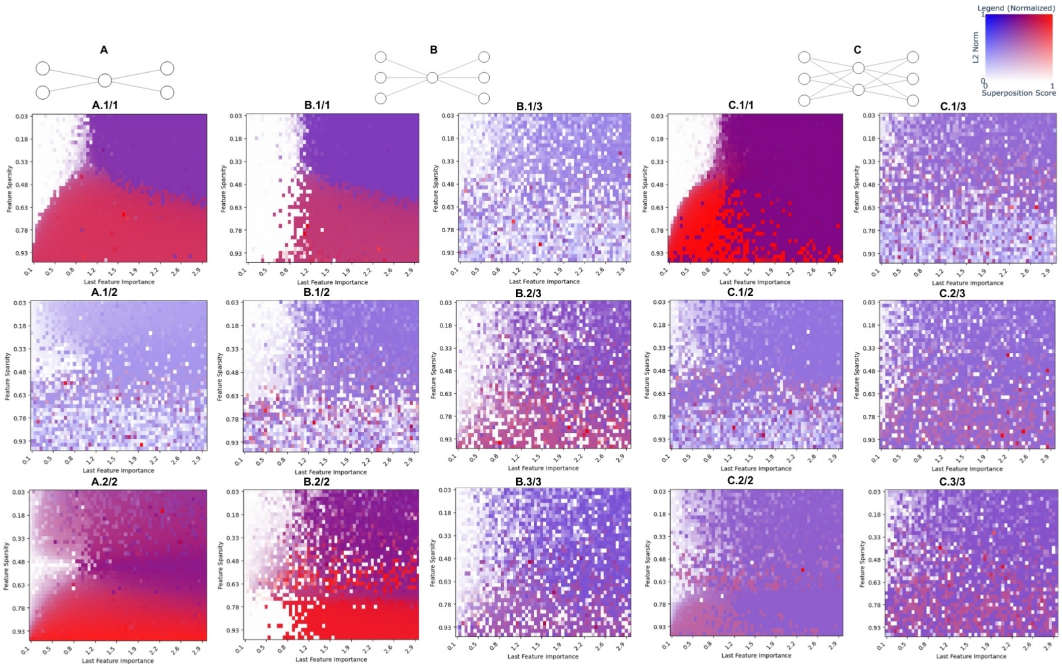


Figure 4: Joint feature norm ($||W_i||^2$) and superposition score ($\sum_{j \neq i}(\hat{W}_i \cdot W_j)^2$) across varying feature sparsity $S \in [0.1, 1]$ and relative last feature importance $r \in [0.1, 3]$. For each cell, we train ten models and select the one with the lowest loss. We used load balancing loss in this section. We plot joint feature norm and superposition for the last feature: low L2 norm ($||W_i||$) is white, denoting the model is ignoring the last feature; otherwise a low superposition score is blue-purple to indicate monosemantic representation of the last feature and red for a high superposition score. Subfigure X.e/E denotes the weight matrix of expert e of E total experts trained on architecture X; X.1/1 indicates a dense model.

discrete phase changes disappeared. Some experts in MoEs with $E = 2$ are reminiscent of their respective dense cases (Figure 4.X.2/2), but exhibit more continuous transitions. In each case, one expert became more monosemantic, specializing in the most important feature by relative importance. Experts dissimilar from the dense cases universally have much lower superposition scores (they are bluer), indicating more monosemantic representations. This aligns with the conclusions of the previous section—MoEs favor lower superposition scores compared to their dense counterparts.

For the $n = 2, m = 1$ setup (Figure 4.A), the dense model does not represent the last feature when feature sparsity is low. However, the comparable MoE model preserves the last feature much more because it has the capacity. With three input dimensions (Figure 4.B), the MoEs do not exhibit this behavior because the experts are superimposing the other two features; there is no space for the third feature within one hidden dimension. Unlike the other two cases, for architecture B the hidden dimension with superposition is not sufficient, in the high-sparsity regime, to represent all features. Yet we do not see clear phase change—except for the 0.5 − 0.7 feature sparsity region in 4.B.2/2, where it is mostly discrete but mixed. For $m = 2$ the white region in the dense model (Figure 4.C.1/1) (in the mid- to low-feature sparsity domain, when the feature is relatively less importance than the others) ignores the last feature. However, as network sparsity increases—across all other Figure 4.C—the models represent the last feature with greater L2 magnitude ($||W_3^1|| < ||W_3^2|| < ||W_3^3||$). In other words, the dimensionality in the low relative-importance region increased with increasing network sparsity, as demonstrated in Figure 3.

We observed a window of feature sparsity from roughly 0.48 to 0.7 in Figures 4.B.2/2, 4.C.1/2, and 4.C.2/2 where there is heavy mix of polysemanticity, monosemanticity, or ignorance. This indicates there is a middleground in MoEs with comparable loss between polysemantic and monosemantic representations which make it difficult to consistently commit to the strategies we observe in low and high feature sparsity domains. This is evidence that MoEs learn different representational strategies than dense models.

These experiments are all top-$k = 1$, so only one expert is active at a time. Even so, we see vastly different behavior even the in $E = 2$ case, including when the hidden dimension capacity with superposition is sufficient to represent all features. This leads us to conclude it is misleading to think of MoEs as an aggregation of dense models. The mechanism of the router which allows experts to observe only a subset of the feature domain vastly modifies the behavior and learning of the experts.

5 Expert Specialization

Since MoEs exhibit less superposition, we now examine the organization of such monosemantic features within experts and its relation to specialization.

Expert specialization in MoEs traditionally centers around load balancing between experts across all inputs (Chaudhari et al., 2025). However, this fails to capture the natural intuition of specialization, wherein an expert is only used when appropriate concepts—those the expert is specialized in—are present in the input.

We define an expert as specialized if it *occupies* certain feature directions in the input space, and if it represents said features relatively monosemantically. We demonstrate that these two conditions are directly correlated, and show how the presence of these two conditions encourages load balancing across experts.

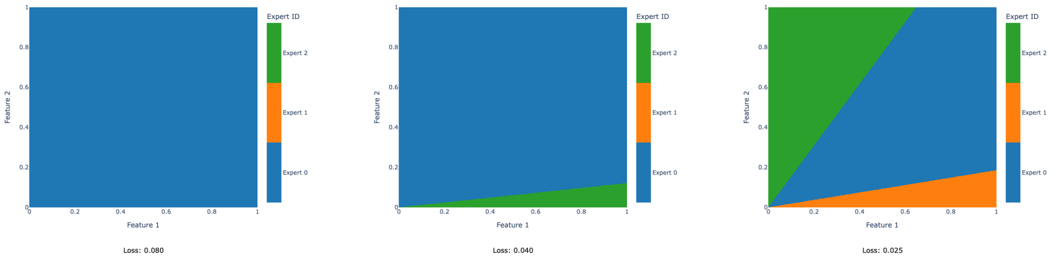
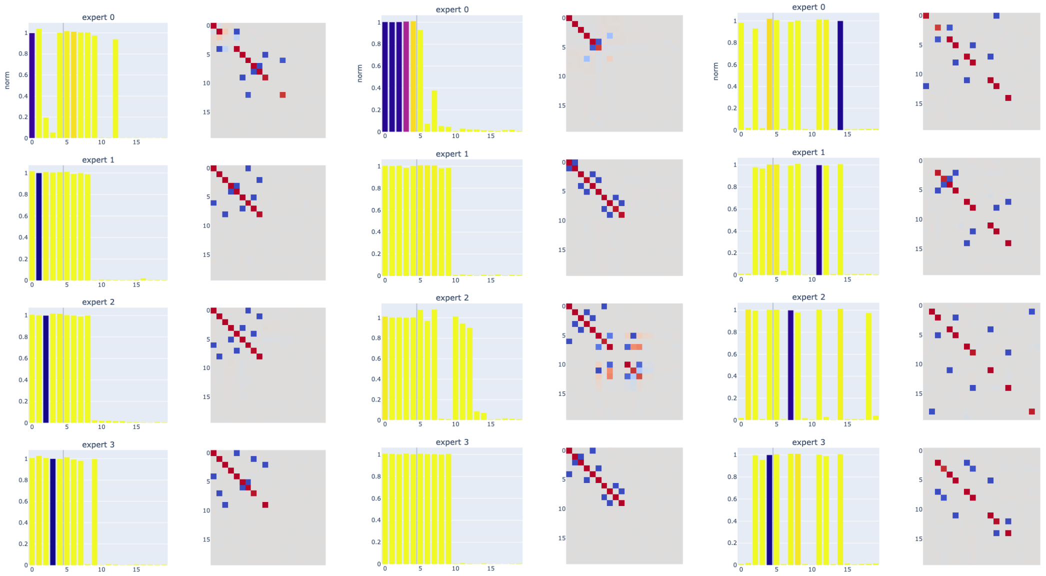


Figure 5: Expert routing of three identical models with differing initialization schemes. We use $n = 2$, $E = 3$, $m = 1$. The first model (left) has the worst performance (loss: 0.08) and routes all inputs to one expert. The second model (middle) has better performance (loss: 0.04) and routes a small portion of inputs, specifically those when feature 1 is active, to a second expert. The third model (right) has the lowest loss (loss: 0.025), and distributes the input space among all experts. One expert is chosen when only feature 1 is active, one when only feature 2 is active, and one when both are active.

Because we fix $k = 1$, the feature space is partitioned into convex cone regions (see Appendix A.3), with each region routed to a particular expert. By definition, this means $\forall s > 0,\ x \in C \to sx \in C$, where C is the set of points contained within the cone and s is any positive scalar. If a particular feature vector x is routed to an expert, then all sx are routed to that same expert. In this case, we say that feature x is contained within expert e, and as such expert e *"occupies"* x.

There is a correlation between the distribution of experts across the feature space (the variance of the volumes of the cones) and how well a model performs, as shown in Figure 5. Not only do models with better distribution of experts across the feature space perform better, but they tend to align experts with particular features. This implies that experts *can* specialize to specific features, and that doing so improves performance.

In models with $n > 2$, we explore whether initializing experts to occupy features in the input space cause the experts to be more monosemantic w.r.t. those features. Separately, we see if, for the features an expert has chosen to represent monosemantically, the expert occupies those features in the input space.



(a) Diagonal Initialization (b) Ordered K-hot initialization (c) Random K-hot initialization

Figure 6: $||W_i^{(e)}||^2$ and W^TW results for three different initialization schemes, with $n = 20, m = 5, E = 4, S = 0.1$. In **(a)**, the gate matrix is initialized along the diagonal, and relative feature importance decreases exponentially in order from feature one to 20. In **(b)**, the gate matrix is initialized to an "ordered k-hot", such that the first expert aligns with the first five features, and each subsequent expert aligns with the next five features. Relative feature importance is the same as **(a)**. In **(c)**, the gate matrix is initialized to a "random k-hot", where each expert is assigned five random features such that experts share no common feature but cover all 20 features collectively. Relative feature importance decreases exponentially but is randomly distributed across features.

When the gate matrix is initialized to the diagonal, such that $c_i = x_i$, each expert monosemantically represents the single feature it initially occupied, and only that feature, as shown in Figure 6a. When the router is ordered k-hot initialized, the first expert monosemantically represents four of the five features it initially occupied, as shown in Figure 6b. The other experts, initialized over other features, did not monosemantically represent these less important features, nor did they monosemantically represent the five most important features they were not initialized over. When we break the ordering of feature importance and randomize the features each expert initially occupies, each expert monosemantically represented only the most important feature it was initialized over, as shown in Figure 6c.

There is a strong correlation between the features that are initially routed to an expert and which features that expert represents monosemantically. Furthermore, we observe that experts only monosemantically represent important features. This is true if we initialize each expert with one important feature explicitly, or if we give it a set of features, upon which it selects the most important feature itself and represents it monosemantically.

In the case of uniform feature importance, all features are placed in superposition with high polysemanticity scores. Despite this, the features an expert initially occupies are still *relatively* more monosemantic, on average achieving polysemanticity scores 1.03 standard deviations below the average for that expert.

We investigated whether there is a correlation between experts representing certain features monosemantically, and said experts occupying those features in the input. To do this, we measure usage statistics when those features are *one of many* active features, and when they are the *only* active features. This second case is equivalent to measuring the probability that the expert occupies these features. The correlation holds both in xavier and k-hot initialization schemes, as seen in Table 1. Given $E = 10$, a mean expert usage of ∼10% indicates an even load balancing across experts. In all cases, when the corresponding monosemantic feature(s) for an expert is active, the usage of the expert increases significantly. When this feature(s) is the only active feature, the expert dominates the usage.

Table 1: Monosemantic feature and usage statistics per expert for $n = 100, m = 10, E = 10$. One hundred models are trained for each initialization scheme (xavier and k-hot), providing 1000 experts in total for each. Each statistic is aggregated across models, classifying experts based on the number of features they represent monosemantically. For the feature(s) an expert represents monosemantically, we track the expert usage when said feature(s) is one of several active features in the input, as well as the expert usage when said feature(s) is the *only* active feature in the input.

		Xavier Initialization		
Number of monosemantic features per expert	Number of experts (out of 1000)	Mean expert usage (%)	Mean expert usage; feature(s) active (%)	Mean expert usage; only feature(s) active (%)
0	461	–	–	–
1	387	9.595	17.94	67.18
2	138	9.599	30.29	95.65
3	13	8.363	40.19	100.0
4	1	1.428	14.69	100.0
5	0	–	–	–
		K-Hot Initialization		
Number of monosemantic features per expert	Number of experts (out of 1000)	Mean expert usage (%)	Mean expert usage feature(s) active (%)	Mean expert usage only feature(s) active (%)
0	335	–	–	–
1	382	10.00	23.94	100.0
2	227	10.02	46.61	100.0
3	47	10.09	62.00	100.0
4	8	9.95	70.30	100.0
5	1	9.62	74.79	100.0

In the k-hot initialization scheme, *100%* of all features monosemantically represented by an expert are occupied by that same expert.

As experts represent more features monosemantically, they can be seen as more specialized. Their usage on arbitrary input decreases, but conditional on their specialized features being active, their usage increases far greater than other experts. This holds true for all cases except the xavier initialized model with a four monosemantic feature expert, where there is a significant drop in utilization.

6 Conclusion

We investigated how experts affect superposition in MoEs, showing that MoEs consistently exhibit greater monosemanticity than dense networks while not exhibiting a phase change. We proposed a feature-based definition of expert specialization, demonstrating that experts naturally organize around coherent features when initialization encourages this specialization. However, our findings are based on simple autoencoder toy models with synthetic data, leaving open questions about generalization to large-scale transformers where the feature distribution is unknown. Despite these limitations, we show how toy MoEs achieve comparable loss while maintaining more interpretable representations—challenging the prevalent zeitgeist that mechanistic interpretability and model capability are fundamentally in tension. Future work should explore what favors monosemanticity in MoEs, how training dynamics of MoEs differ from those of the dense model, and when specialization emerges. Answering these questions can inform the design of more interpretable, high-performing language models.

References

Marmik Chaudhari, Idhant Gulati, Nishkal Hundia, Pranav Karra, and Shivam Raval. Moe lens - an expert is all you need. In *Sparsity in LLMs (SLLM): Deep Dive into Mixture of Experts, Quantization, Hardware, and Inference*, 2025. URL `https://openreview.net/forum?id=GS4WXncwSF`.

Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. *CoRR*, abs/1408.5099, 2014. URL `http://arxiv.org/abs/1408.5099`.

Nishanth Dikkala, Nikhil Ghosh, Raghu Meka, Rina Panigrahy, Nikhil Vyas, and Xin Wang. On the benefits of learning to route in mixture-of-experts models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9376–9396, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.583. URL `https://aclanthology.org/2023.emnlp-main.583`.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Pub*, 2022. URL `https://transformer-circuits.pub/2022/toy_model/index.html`.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022. URL `https://arxiv.org/abs/2101.03961`.

Google DeepMind. Gemini 2.5 pro. `https://deepmind.google/technologies/gemini/pro/`, 2025.

Liv Gorton and Owen Lewis. Adversarial examples are not bugs, they are superposition, 2025.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Siyuan Mu and Sen Lin. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications, 03 2025.

Jungwoo Park, Ahn Young Jin, Kee-Eung Kim, and Jaewoo Kang. Monet: Mixture of monosemantic experts for transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=1Ogw1SHY3p`.

Adam Scherlis, Kshitij Sachan, Adam S. Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks, 2025. URL `https://arxiv.org/abs/2210.01892`.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL `https://arxiv.org/abs/1701.06538`.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Xingyi Yang, Constantin Venhoff, Ashkan Khakzar, Christian Schroeder de Witt, Puneet K. Dokania, Adel Bibi, and Philip Torr. Mixture of experts made intrinsically interpretable. In *Forty-second International Conference on Machine Learning*, 2025b. URL `https://openreview.net/forum?id=6QERrXMLP2`.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V Le, and James Laudon. Mixture-of-experts with expert choice routing. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=jdJo1HIVinI.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam M. Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models, 2022. URL https://api.semanticscholar.org/CorpusID:248496391.

A Appendix

A.1 Measuring loss for varying sparsity & experts

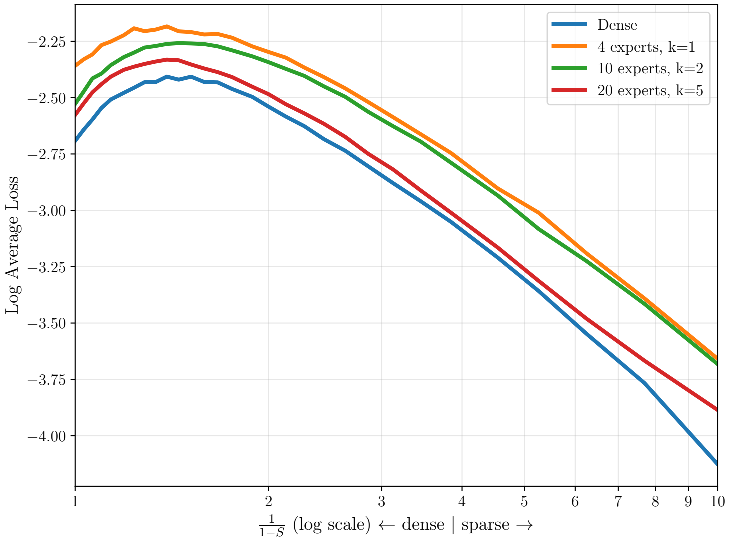


Figure 7: Log average loss versus feature density ($\frac{1}{1-S}$) for dense ($m = 20$) and MoE (4 experts, $k = 1$, $m = 5$), MoE (10 experts, $k = 2$, $m = 2$), and MoE (20 experts, $k = 5$, $m = 1$) models, all with uniform feature importance ($I_i = 1.0$) for $n = 100$ input features. Results are averaged over five runs per sparsity level. Although dense model outperforms all MoEs at every sparsity level, as the number of experts increases, the MoE loss gets closer to the dense model.

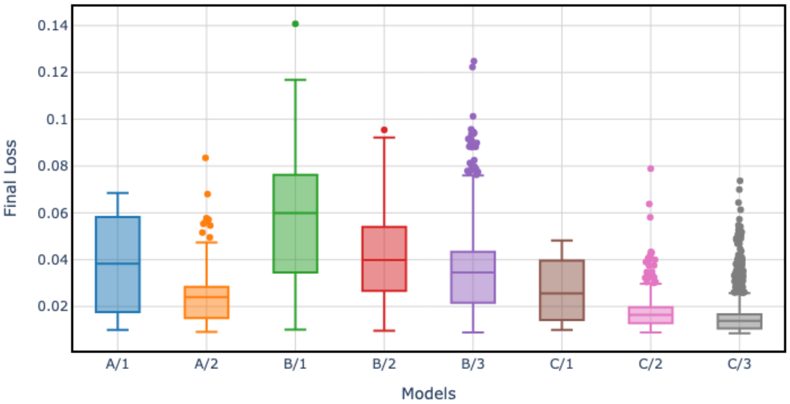


Figure 8: Model X/E uses X to denote the same model architectures and models used in Figure 4 and E denotes the total number of experts (i.e. network sparsity). Increasing network sparsity decreases mean loss while increasing localized variance—especially as the number of experts reaches the input feature dimensions. This can attributed to the relatively unstable training of MoEs compared to dense models (despite training ten models for each cell and selecting the lowest loss).

A.2 Analytic Model Equivariance

For the toy setup of single-layer, single-nonlinearity, top-$k = 1$ MoEs, there exists a theoretical map between any dense model and a monosemantic MoE with an equivalent number of active features under a sparsity constraint.

Assume there exists an upper bound for the number of active features a for any input such that $\forall x \in D : |\{i : x_i \neq 0\}| \leq a$. Furthermore, assume that a is no greater than the hidden dimensionality, m, of an expert, providing an upper bound on the number of features a model has to represent. Assume also that the hidden dimensions is smaller than the total number of input features n ($a \leq m \leq n$). To construct the monosemantic MoE, for each possible subset $S \subseteq \{1, 2, \ldots, n\}$ with $|S| \leq a$—meaning the size of the subset of active features is smaller than or equal to a—create an expert which monosemantically preserves those features. (In fact, you can take only the subsets such that $|S| = a$.) The router then selects the expert which corresponds to those active features (of which there will never be more than a, by assumption):

$$\text{Router}(x) = \arg\max_{S} \mathbb{I}[\text{support}(x) = S]$$

where $\text{support}(x) = \{i : x_i \neq 0\}$. Since $|S| \leq a \leq m$, each expert has sufficient capacity to represent its assigned features without superposition. To reiterate, only a features are active and every unique combination of active features receives its own dedicated expert with sufficient capacity to represent those features monosemantically. So, the number of possible experts needed is $\binom{n}{m}$.

The reconstruction for this theoretical MoE has zero loss only as the sparsity constraint holds (or goes to one in these toy models) because there is the chance more than m features could be active at one time ($a \not\leq m$), which would exceed the monosemantic representational capacity of the network (but the dense polysemantic could do no better unless features are correlated in the distribution). Therefore, even if $a \not\leq m$ sometimes, the polysemantic model encounters the same problem and the monosemantic MoE under this construction may still outperform it under looser sparsity constraints.

Thus, for any dense model, $f_{\text{dense}}(x) = \text{ReLU}(Wx + b)$ under the sparsity constraint $|\text{support}(x)| \leq a$, there exists a MoE model $f_{MoE}(x)$ such that $f_{\text{dense}}(x) = f_{\text{MoE}}(x)$ for all valid inputs. In the toy settings described in this paper, the sparsity constraint holds in the limit where sparsity goes to one. However, in practice there may be an upper bound on the amount of features a particular amount of information can semantically encode, indicated by the size of meaningful embeddings of that data. Therefore, a MoE model with sufficient experts and a tractable amount of superposition (e.g. interpretable) may be sufficient to encode all features present.

A.3 Router Subspaces are Convex Cones

In the regime of $k = 1$, the router function $g(x) = \text{softmax}(W_r x)$ is equivalent to $\text{argmax}(W_r x)$. The region routed to expert i can be represented as $\forall j \neq i, (w_i - w_j)^\intercal x > 0$ where w_i and w_j are row vectors of W_r. This is a homogeneous linear inequality. Regions bounded by such inequalities are by definition convex cones. If a particular x satisfies this inequality, then multiplying both sides by any positive scalar s will still satisfy the inequality. Furthermore, if x_1 and x_2 satisfy this inequality, then any $x = x_1\lambda + (1 - \lambda)x_2$ for $\lambda \in [0, 1]$ will also satisfy the inequality.

In the case of $k > 1$, the region of inputs which get sent to a particular expert e becomes a union of convex cones. Generally, the union of a convex cone is not itself a convex cone. Therefore, the understanding of experts occupying feature directions may not hold beyond $k = 1$.