WHY THIS EXPERT? A FEATURE-LEVEL ANALYSIS OF ROUTING IN MIXTURE-OF-EXPERTS WITH SPARSE AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding Mixture-of-Experts (MoE) routing requires identifying which features of a token's representation favor selected experts over their competitors. We present a framework for explaining individual routing decisions through sparse autoencoder (SAE) features. Our framework decomposes routing logits into feature contributions, a decoder bias, and an explicit reconstruction residual, allowing us to quantify how features contribute to routing logits and expert selection. We formulate the search for a minimum-cardinality feature subset preserving the selected expert set as a mixed-integer linear program, distinguishing explanations of reconstructed routing from explanations of original routing conditional on the residual. Across four MoE models, we examine routing fidelity, feature-expert affinities, and the number of features required to preserve selection. On OLMoE, an average of 8.79 of 32 active features suffices to preserve the reconstructed expert set; preserving the original set with the residual fixed requires 9.57 features. SAE reconstructions recover 81–88% of selected experts across OLMoE layers and 73–86% across DeepSeek-V2-Lite layers. Case studies connect these quantitative contributions to interpretable token and contextual features, while steering experiments demonstrate control over expert selection. Together, these results provide a quantitative account of routing through compact feature subsets while explicitly separating their contributions from information the SAE leaves unexplained.

1 INTRODUCTION

Mixture-of-Experts (MoE) has become a prevalent architecture of advanced large language models (LLMs) such as Gemini 2.5 (Comanici et al., 2025), DeepSeek-v4 (Xu et al., 2026), and Kimi K3 (Team et al., 2026) in recent years. Although the effectiveness of MoE is extensively demonstrated, its underlying mechanism remains insufficiently understood. An important open question is what determines the routing decisions in MoE. Previous studies have observed that experts can be triggered by specific tokens or domains (Lewis et al., 2021; Zhu et al., 2024; Muennighoff et al., 2025), known as expert specialization. These observations suggest that routing behaviors can be attributed to human-interpretable concepts. Recently, Herbst et al. (2026) and Lasy et al. (2026) have demonstrated the feasibility of interpreting expert activation based on the contexts in which they are active. Inspired by these previous studies, we seek to attribute routing decisions to a set of human-interpretable concepts through routing decomposition.

Specifically, we adopt sparse autoencoders (SAEs) (Elhage et al., 2022; Gao et al., 2025) to capture concepts in the models for studying the routing mechanism. SAEs decompose representations into sparse linear combinations of features that are often highly human-interpretable (Huben et al., 2024). This enables us to decompose the MoE layer input (or the middle residual stream) into SAE features, attribute routing decisions to them, and quantify how specific information drives routing decisions. Currently, most existing methods mainly focus on qualitatively explaining expert activation based on contexts, tokens, or features that are present when experts are activated. Consequently, it may be difficult for them to quantitatively measure the influence of similar concepts on routing decisions. Additionally, with those methods, it may be difficult to observe the subtle features that are not dominant, despite having a notable contribution to the routing decisions. We suggest that quantifying

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

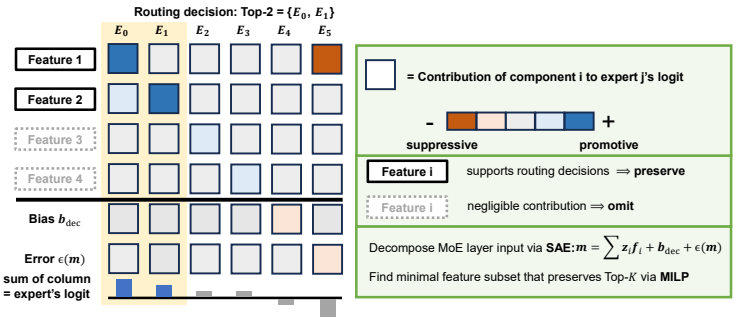


Figure 1: Overview of our methods.

the contributions of features to routing decisions is also important, as it can explicitly reveal the influence of features and facilitate the analysis of interactions between features and experts. We achieve this in our work.

In this work, we aim to apply SAEs to identify features that drive routing decisions in MoE models and quantify their contribution. Our contributions are as follows: ❶ We propose a framework for linearly attributing routing logits to SAE features, decoder bias, and an unexplained reconstruction residual. Under this framework, we quantify feature-expert routing affinity and feature contributions to routing decisions, enabling explicit routing interpretation. ❷ We formulate the search for a minimal-cardinality feature subset preserving the selected expert set as a mixed-integer linear program. On OLMoE, an average of 8.79 of 32 active features suffices to preserve the reconstructed expert set; preserving the original set with the residual fixed requires 9.57 features. ❸ We empirically evaluate our methods and analyze the statistical properties of features and SAE performance across four models. SAE reconstructions recover 81–88% of selected experts across OLMoE layers and 73–86% across DeepSeek-V2-Lite layers.

2 Related Work

Mixture-of-Experts. Mixture-of-Experts (MoE) was first introduced in Jacobs et al. (1991) and has been demonstrated as an effective method for scaling language models (Shazeer et al., 2017; Fedus et al., 2022; Du et al., 2022). Although it has become a prevalent architecture in practice (Comanici et al., 2025; Agarwal et al., 2025; Xu et al., 2026), its underlying mechanism is still not fully understood. Previous studies have shown that some experts tend to activate on specific tokens or topics, suggesting a certain degree of expert specialization (Jiang et al., 2024; Zhu et al., 2024; Muennighoff et al., 2025). This phenomenon has prompted increasing research on interpreting expert routing behavior recently: Do et al. (2026) argue the existence of domain-specific experts. Wang et al. (2026) propose some experts are domain-invariant. Lasy et al. (2026) proposes that there is a superposition on the expert scale: experts can specialize in unrelated fine-grained features instead of features from a broad domain, and employs SAEs to verify that. These works adopt an expert-centric view to study routing behaviors and do not quantitatively examine how the information in the representation drives routing decisions on a case-by-case basis.

Sparse Autoencoders. Linear representation hypothesis (LRH) (Park et al., 2024) posits that concepts are represented linearly as directions (vectors) in the hidden space. One difficulty in explaining the semantics of representations is that they are polysemantic, a phenomenon often attributed to "superposition"(Elhage et al., 2022). Sparse autoencoders (SAEs) are adopted to linearly decompose representations into sparse features which are expected to be human-interpretable (Bricken et al., 2023; Huben et al., 2024). Vanilla SAE (Bricken et al., 2023) suffers from insufficiently sparse features and dead features, which impede the interpretation of these SAE features. Advanced SAEs such as Top-K SAE (Gao et al., 2025) manage to alleviate these issues. Recent studies have shown that SAEs provide an effective way to decompose representations into sparse features and are useful for studying experts empirically (Lasy et al., 2026; Wang & Wang, 2026).

The closest works to ours are (Lasy et al., 2026) and (Herbst et al., 2026). Lasy et al. (2026) use SAEs to identify features that promote experts to be selected based on a causal metric. (Herbst

et al., 2026) aim to explain expert activation by identifying contexts where experts are activated. In this work, we linearly decompose routing logits into components assigned by SAE features, bias, and reconstruction residual. We quantify the contribution of features case by case and reveal that features can have a suppressive contribution to expert selection.

3 PRELIMINARIES

Mixture of Experts. Given a MoE layer input $\boldsymbol{m} \in \mathbb{R}^{N_m}$, the routing logits $h(\boldsymbol{m}) \in \mathbb{R}^{N_r}$ are computed as:

$$h(\boldsymbol{m}) = \boldsymbol{W}_r \boldsymbol{m} \ , \tag{1}$$

where $\boldsymbol{W}_r \in \mathbb{R}^{N_r \times N_m}$ is the learned router matrix, N_m is the model dimension, N_r is the number of routed experts at that layer. The routed experts with top-K routing logits are selected to process the input, and their routing logits are normalized via a Softmax distribution over those experts:[1]

$$p_i(\boldsymbol{m}) = \frac{\exp(h(\boldsymbol{m})_i)}{\sum_{1 \le j \le N_r} \exp(h(\boldsymbol{m})_j)} \ . \tag{2}$$

Let $\mathcal{T}$ be the set of selected top-K experts' indices and $\mathcal{S}$ be the set of shared experts' indices.[2] The MoE layer output $\boldsymbol{y} \in \mathbb{R}^{N_m}$ is the sum of selected experts' weighted outputs and shared experts' outputs:

$$\boldsymbol{y} = \sum_{i \in \mathcal{T}} p_i(\boldsymbol{m}) E_i(\boldsymbol{m}) + \sum_{s \in \mathcal{S}} E_s(\boldsymbol{m}) \ . \tag{3}$$

Intuitively, routing logits measure the affinity between each routed expert E_i and the given input $\boldsymbol{m}$.

Top-K SAE. SAEs encode an input representation $\boldsymbol{x} \in \mathbb{R}^{N_x}$ into a sparse representation $\boldsymbol{z} \in \mathbb{R}^{N_z}$, where $N_z > N_x$. In this work, we adopt Top-K SAE (Gao et al., 2025), which is defined as:

$$\boldsymbol{z} = \text{TopK}(\boldsymbol{W}_{\text{enc}}(\boldsymbol{x} - \boldsymbol{b}_{\text{dec}})) \ , \tag{4}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{W}_{\text{dec}} \boldsymbol{z} + \boldsymbol{b}_{\text{dec}} \ , \tag{5}$$

where $\boldsymbol{W}_{\text{enc}} \in \mathbb{R}^{N_z \times N_x}$, $\boldsymbol{W}_{\text{dec}} \in \mathbb{R}^{N_x \times N_z}$, $\boldsymbol{b}_{\text{dec}} \in \mathbb{R}^{N_x}$ are learnable weights, and $\text{TopK}(\cdot)$ keeps the k-largest activations per input and zeros the rest. The loss function is $\mathcal{L} = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2^2$.

4 METHODS

In this section, we first delineate how routing logits can be attributed to SAE features, then discuss the main metrics we used, and finally propose a method for finding a minimal subset of features to facilitate studying routing decisions. The implementation details are described in Appendix B.

Routing decomposition. SAEs can decompose representations into highly human-interpretable feature vectors (Huben et al., 2024) that are essentially the weighted column vectors of $\boldsymbol{W}_{\text{dec}}$. In this work, we use SAEs to capture the features that contribute to routing decisions at that layer for dissecting the routing mechanism. Given a MoE layer input $\boldsymbol{m}$, we can decompose it into SAE features and study the contribution of these features to the routing logits:

$$h(\boldsymbol{m}) = \boldsymbol{W}_r \boldsymbol{m} = \boldsymbol{W}_r(\hat{\boldsymbol{m}} + \boldsymbol{\epsilon}(\boldsymbol{m})) = \boldsymbol{W}_r((\sum_{i \in \mathcal{A}} \boldsymbol{z}_i \boldsymbol{f}_i) + \boldsymbol{b}_{\text{dec}} + \boldsymbol{\epsilon}(\boldsymbol{m})), \tag{6}$$

where $\boldsymbol{f}_i = (\boldsymbol{W}_{\text{dec}})_{:,i}$ is the i-th SAE feature weighted by $\boldsymbol{z}_i > 0$, $\mathcal{A}$ is the set of active features (i.e., $\boldsymbol{z}_i \neq 0$), $\boldsymbol{\epsilon}(\boldsymbol{m})$ is the error term (reconstruction residual). The contribution of feature $\boldsymbol{f}_i$ to the routing logits $h(\boldsymbol{m})$ is $h(\boldsymbol{f}_i) = \boldsymbol{W}_r(\boldsymbol{z}_i \boldsymbol{f}_i)$. Similarly, we define the contribution of bias to the routing logits as $h(\boldsymbol{b}_{\text{dec}}) = \boldsymbol{W}_r \boldsymbol{b}_{\text{dec}}$. Equation 6 is a corollary of Equations 1 and 5. It reveals that each SAE feature has an intrinsic preference for each expert ($\boldsymbol{W}_r \boldsymbol{f}_i$). Additionally, it enables quantifying the contribution of features to routing logits and indicates that features can jointly promote or suppress expert selection, or counteract their effects on the selection.

[1] Some models use the top-K routed experts instead of all routed experts in the normalization.

[2] Some MoE variants (e.g., DeepSeek-V2-Lite (Liu et al., 2024)) may include additional shared experts that are always active and not involved in routing decisions. For models without shared experts, $\mathcal{S} = \varnothing$.

Alternatively, SAEs can also be trained on the middle residual stream for studying routing decisions, which we discuss in Appendix A. Empirically, we find that these two schemes achieve similar performance in recovering routing decisions, but SAEs trained on the MoE layer input yield lower reconstruction error for the routing logits (See Appendix G). Hence, we mainly discuss SAEs trained on the MoE layer input in the main paper.

In this work, we adopt Top-K SAE because it intrinsically restricts the number of active features, avoiding dense feature activation and thus facilitating our interpretation. Additionally, it outperforms the vanilla SAE, increases monosemanticity, and reduces dead features (Gao et al., 2025).

Characteristics of contribution. We have defined the contribution of features to the routing logits $h(\boldsymbol{f}_i)$ as a vector. Now we seek scalar measures to characterize it. We use the L1 norm of $h(\boldsymbol{f}_i)$ to measure the magnitude of contribution to routing logits. We use entropy of the routing scores (i.e., the Softmax outputs of routing logits) to measure the uniformity of contribution $h(\boldsymbol{f}_i)$. If a feature rates more uniformly on the experts, it has a weaker impact on the overall ranks. We adopt these two metrics rather than rank variation or leave-one-out metrics to isolate the influence of individual features and avoid interference from other features.

Minimal subset of features. Not all active features make a significant contribution to the routing decisions. Removing features with negligible contribution can facilitate studying routing decisions. Hence, given the Top-F features,[3] we seek to identify a minimal active feature subset $\mathcal{M} \subset \mathcal{A}$ that suffices to reproduce the routing decision. We formalize this as an optimization problem:

$$\arg\min_{\mathcal{M}} |\mathcal{M}| \quad s.t. \quad \text{TopK}(\mathcal{M}) = \text{TopK}(\mathcal{A}), \tag{7}$$

where subset $\mathcal{M}$ is required to have the same selected experts as that of $\mathcal{A}$. Given that features may exert combinatorial contributions, we formulate feature selection as a binary combinatorial optimization problem and transform it into a Mixed-Integer Linear Programming (MILP) problem:

$$\min \sum_{i\in\mathcal{A}} a_i \quad s.t. \quad \sum_{i\in\mathcal{A}} a_i(h(\boldsymbol{f}_i)_s - h(\boldsymbol{f}_i)_u) \geq h(\boldsymbol{b}_{\text{dec}})_u - h(\boldsymbol{b}_{\text{dec}})_s \tag{8}$$

holds for every selected expert s and every unselected expert u, where $a_i \in \{0, 1\}$. The constraints guarantee that the routing logits of selected experts are no less than those of unselected experts. A variant of objective 7 is to recover the ground-truth expert selection:

$$\arg\min_{\mathcal{M}} |\mathcal{M}| \quad s.t. \quad \text{TopK}(\mathcal{M}) = \text{TopK}(\mathcal{T}), \tag{9}$$

and transform the problem into:

$$\min \sum_{i\in\mathcal{A}} a_i \quad s.t. \quad \sum_{i\in\mathcal{A}} a_i(h(\boldsymbol{f}_i)_s - h(\boldsymbol{f}_i)_u) \geq h(\boldsymbol{b}_{\text{dec}})_u + h(\boldsymbol{\epsilon}(\boldsymbol{m}))_u - h(\boldsymbol{b}_{\text{dec}})_s - h(\boldsymbol{\epsilon}(\boldsymbol{m}))_s\,, \tag{10}$$

where the routing logits assigned by error terms are provided to guarantee the existence of a solution. This variant aims to find a feature subset sufficient to explain the actual routing decision once the unexplained residual is provided. If the error term drives the routing decision, it hints that the SAE fails to capture the features contributing to the routing decision. A stricter variant is to preserve the exact order of the Top-K experts. This can be enforced by imposing constraints in 8 and 10 for every selected expert s and every lower-ranked expert u.

There are two caveats to this method. First, it may omit non-negligible features. For example, features may have opposing effects on expert selection and counteract each other, causing these contributions to be overlooked. Second, the minimal subset may not be unique, but the solver only returns an arbitrary solution. Hence, the result should be interpreted as one of the feasible ways to explain routing decisions.

5 Experiments

We first present two case studies to demonstrate how our method can be applied to interpret feature-expert routing affinity and routing decisions. We then statistically analyze SAE reconstruction fidelity and the characteristics of SAE features related to routing decisions.

[3] We use "Top-F" features to distinguish them from "Top-K" Experts.

5.1 EXPERIMENTAL SETUP

Models. We report experimental results on four MoE models: OLMoE-A1B-7B (Muennighoff et al., 2025) and DeepSeek-V2-Lite (Liu et al., 2024) in the main paper, plus gpt-oss-20B (Agarwal et al., 2025) and Qwen3-30B-A3B (Yang et al., 2025) mainly in the appendix. Model details are provided in Appendix D. We use zero-based indexing for the layers, so Layer 0 refers to the first layer in the models. OLMoE consists of 16 layers, each with 64 routed experts and top-8 expert routing. DeepSeek-V2-Lite consists of 27 layers, each with 64 routed experts, 2 shared experts, and top-6 expert routing, except for Layer 0. For each Top-K SAE, we set the expansion factor to 16 and set F to 32.

Dataset. We train all SAEs on 200M tokens from SmolLM2-135M-10B dataset (Allal et al., 2025). We evaluate on 5M tokens from the held-out samples of the same dataset.

Feature Interpretation. We follow Paulo et al. (2025) to automatically interpret SAE features to facilitate feature interpretation. The automatically interpreted features are also manually examined. We provide the details and examples in Appendix C.

5.2 CASE STUDY 1: INTERPRETING FEATURE-EXPERT ROUTING AFFINITY

In this section, we demonstrate how to interpret feature-expert routing affinity based on routing logits assigned to experts by SAE features. In this way, we can intuitively understand and quantify the affinity between features and experts. Here we take L1-E9 (Layer 1, Expert 9) of OLMoE as an example. (Herbst et al., 2026) find that this expert "*activates on articles 'a', 'an', and 'the' to predict list continuations.*" We use our method to verify that and show what can be further interpreted about this expert. We also showcase the interpretation on another two experts (L15-E39 of OLMoE and L24-E9 of Qwen3-30B-A3B) in Appendices E.2 and E.3, respectively.

We first take a closer look at the distribution of the routing logits of L1-E9 assigned by features in Figure 2a. The routing logits assigned by most features are concentrated around zero, indicating that these features have a neutral effect on the routing of the expert. We are interested in the features assigning strong positive (Figure 2b) or negative (Figure 2c) routing logits to the expert because they reveal the mutual affinity between experts and features. The distribution has asymmetric tails, with a longer positive tail and a shorter negative tail. Some experts (e.g., L15-E39 of OLMoE) exhibit the opposite pattern, with a shorter positive tail and a longer negative tail (Figure 10).

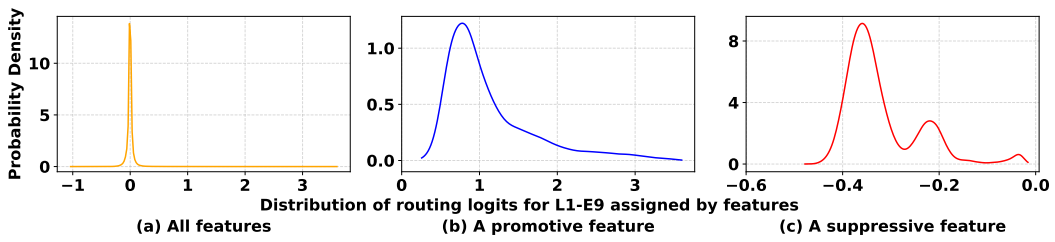


Figure 2: (a) Routing logit distribution of L1-E9 of OLMoE. (b) Distribution of routing logit contributed by a promotive feature related to the article "the". (c) Distribution of routing logit contributed by a suppressive feature related to the conjunction "and".

Our method. Given Expert i, we first collect the routing logits assigned to Expert i by all features. We then identify features assigning the top 0.1% or bottom 0.1% of routing logits to the expert. However, if Expert i is rarely among the Top-K experts on the tokens where Feature $\boldsymbol{f}$ is active, we consider Feature $\boldsymbol{f}$ not to have a prominent positive contribution to that expert, even if it assigns a top 0.1% routing logit. We therefore add a criterion: on all tokens where the feature is active, the mean rank of Expert i must be at most K.[4] We consider features assigning the top 0.1% routing logits to Expert i and satisfying this criterion as promotive features for Expert i. Similarly, we consider features assigning the bottom 0.1% of routing logits to Expert i and whose corresponding tokens have a mean rank of greater than $0.75N_r$ for Expert i as suppressive features regarding Expert i. We

[4]The criterion is empirical and can be relaxed or tightened. We adopt K to facilitate the analysis.

interpret these promotive and suppressive features to explain what concepts the expert activates on in Appendix E.1.

We observe that two features related to the tokens " the" (Figure 2b) and " a" occupy the high positive activation area, with an average routing logit of 0.55 and 0.72, respectively. The mean rank of L1-E9 in the tokens with each of these two features is 4.45 and 3.36, respectively. Many other features also concurrently occur with L1-E9, including features related to ordinal numbers and some adjectives, which we show in Appendix E.1. Furthermore, a few features occupy a high negative activation area, such as the feature related to the conjunction "and" (Figure 2c), with an average routing logit of -0.32. The mean rank of L1-E9 is 3.34 on the corresponding tokens for the top 0.1% features and 42.86 for the bottom 0.1%. The experiment shows that our method can probe which features promote or suppress an expert and quantify such influence.

5.2.1 Feature steering

We conduct feature steering experiments to verify the features we identify as promotive (the ones related to Token " a" and " the") or suppressive features (the one related to Token " and") in Section 5.2.

The experiment pipeline is as follows. First, we identify the promotive and suppressive features for the target expert, as completed in Section 5.2. Second, we collect the activations of these features on a dataset and calculate the mean activations for each feature. Finally, for each token that does not select the target expert, we add the routing logits assigned by the promotive feature scaled by α; for each token that selects it, we add the logits assigned by the suppressive feature scaled by α:

$$h(\boldsymbol{m}) \leftarrow h(\boldsymbol{m}) + \alpha h(\boldsymbol{f}) \,, \tag{11}$$

where $h(\boldsymbol{m})$ is the routing logits to be steered, $\boldsymbol{f}$ is the routing logits assigned by the steering feature, α is the steering magnitude.

The feature steering results are illustrated in Figure 3. When $\alpha = 0.95$, the two promotive features bring L1-E9 to the Top-8 experts on 57% and 47% of tokens that did not select it before, whereas the random features only do that on 0.5% of those tokens. The suppressive feature renders the expert drop from Top-8 on 33% of tokens that did select it before, whereas the random features do that on 2.2% of those tokens. These results agree with the characteristics (promotive/suppressive) of the features that we identify in Section 5.2.

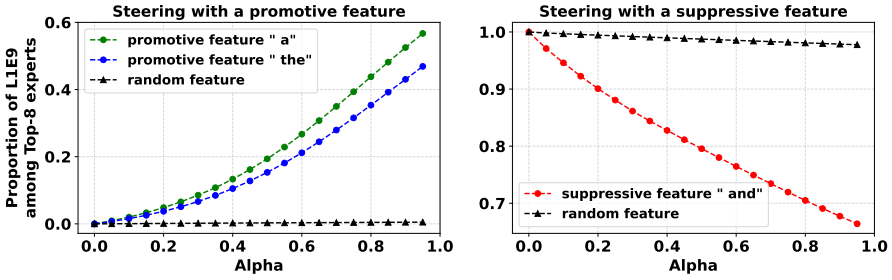
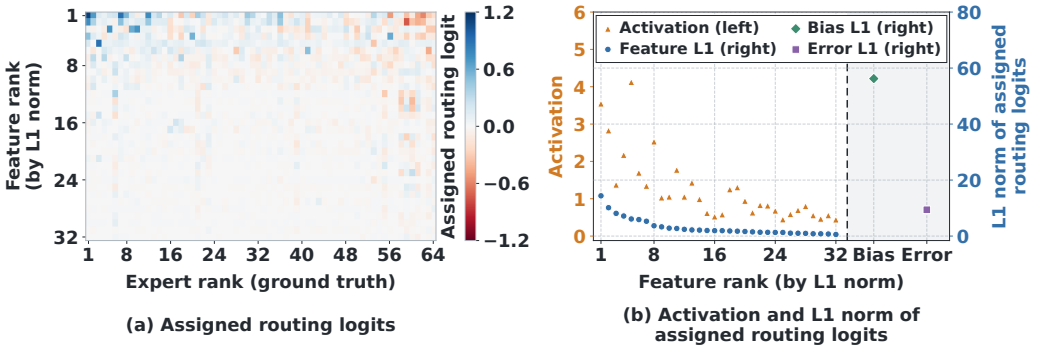


Figure 3: Left: Steer tokens that did not select the target expert with the promotive feature " a" and " the". Right: Steer tokens that selected the target expert with a suppressive feature " and".

5.3 Case study 2: interpreting routing decisions with SAE features

In this section, we identify the information that controls routing decisions and quantify its influence on them. As an illustrative example of routing interpretation, we study the routing decision at Layer 0 of OLMoE at token " to" (the second one) in the sentence "*When Mary and John went to the store, John gave a drink* **to** *Mary.*" (Wang et al., 2023). We also show the results at Layers 7 and 15 in Appendix F. We select this sentence because it has several entities, including places, objects, and persons, and has a short context for study. In the main text, we show Layer 0 because its features are more intuitive, whereas deeper layers may contain complicated semantic features. Equation 6 indicates that routing logits can be regarded as the weighted sum of features, bias, and error. We use this equation to find how each feature contributes to the routing logits, illustrated in Figure 4.



(a) Assigned routing logits

(b) Activation and L1 norm of assigned routing logits

Figure 4: (a) Routing logits assigned by the top-32 features at the second token " to" at Layer 0 of OLMoE, with experts and features sorted by ground-truth rank and activation, respectively. (b) Feature activations and L1 norms of their assigned routing logits, along with those of the bias and error terms.

As SAE features have been empirically shown to be highly human-interpretable (Huben et al., 2024), we can study what information (e.g., semantics or token positions) contributes to routing logits and quantify it. In this case, we explain the features based on the contexts in which they appear. Features with higher activation are not necessary to be most influential features on the routing, because they can be near orthogonal to the routing vectors. We measure the overall contribution of a feature $\boldsymbol{f}_i$ to the routing logits by $\|\boldsymbol{W}_r \boldsymbol{z}_i \boldsymbol{f}_i\|_1$.

We observe the top-8 features here and leave a more detailed interpretation of features in Appendix F.1. In this case, we find that the influential features are mainly related to the token "to" itself (Features ranked 1, 2, and 4) and its surrounding tokens (Features ranked 3, 6, 7, and 8 related to the phrase "gave a drink"). There is a "ubiquitous" feature (Feature ranked 5) that is active on all tokens in this sentence and has a noticeably lower activation at the first two tokens than at the others. We speculate this feature serves as a position indicator. Furthermore, the results in Figure 4a suggest that the top-ranked features make a relatively prominent contribution to promoting the top-ranked experts and suppressing the bottom-ranked experts, compared with the remaining features.

5.4 Reconstruction Fidelity

We evaluate the reconstruction fidelity of routing decisions in this section to confirm the feasibility of using SAE for interpreting routing decisions. We also evaluate the reconstruction fidelity of representations in Appendix G.

Metrics. Recall@K is the mean size of the intersection of ground-truth top-K expert set $\mathcal{T}$ and reconstructed top-K expert set $\mathcal{R}$, i.e., Recall@K $= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}(|\mathcal{T}\cap\mathcal{R}|)$. It measures how many top-K experts are recovered correctly on average, regardless of order. Routing logits Mean Squared Error (MSE) $\|h(\boldsymbol{m}) - h(\hat{\boldsymbol{m}})\|_2^2$ measures the overall logit fidelity. Spearman's rank correlation coefficient $\rho = 1 - \frac{6\sum d_i^2}{N_r\times(N_r^2-1)}$ measures the overall ranking fidelity, where d_i is the difference between the real rank and the recovered rank of Expert i.

Layer-level results. The fidelity results are illustrated in Figure 5. Recall@8 for OLMoE is 81%–88% across layers, while Recall@6 for Deepseek-V2-Lite is 73%–86%. Although routing logits MSE gradually increases with layer depth, Spearman's rank correlation remains above 0.9 across all layers in both models, indicating high fidelity to the original expert rankings. These results show that SAE features can recover most of the information needed for experts' ranking. Essentially, a routing logit is the dot product between a router matrix row and the representation, which only depends on the component of the representation parallel to the router matrix row space. Therefore, only a fraction of the information in a representation is necessary for routing decisions, allowing us to recover routing with high fidelity even without reconstructing the original representation perfectly.

Expert-level results. In Figure 18–29 (Appendix G), we observe that the top- and bottom-ranked experts have the best recovered ranking fidelity, whereas the middle-ranked experts suffer from poor

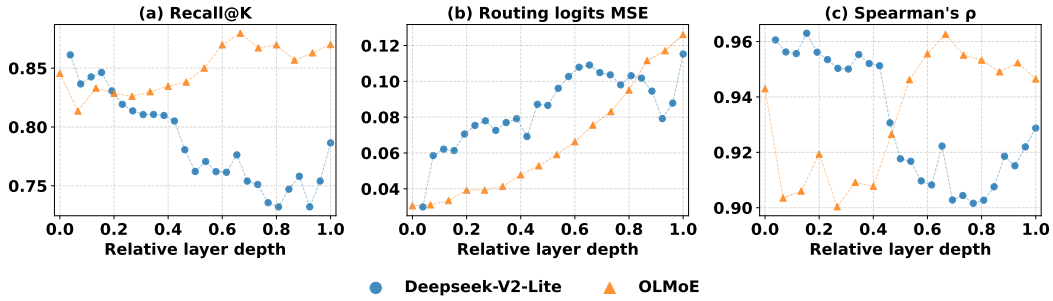


Figure 5: Routing Logits MSE, Recall@K, and Spearman's rank correlation of Top-K SAEs in OLMoE and DeepSeek-V2-Lite.

recovered ranking. The reason is that experts at both ends are relatively easier to recover, as their logits tend to span intervals that have less overlap with those of other experts. Conversely, the middle-ranked experts have similar routing logits, making accurate recovery more difficult.

5.5 SAE FEATURE ANALYSIS

In this section, we conduct a statistical analysis of the routing logits assigned by SAE features and bias terms, and then briefly analyze the reconstruction error and the mean size of the minimal feature subset. For features and bias terms, we examine three aspects: (i) frequency, (ii) contribution magnitude (measured by the L1 norm of the assigned routing logits), and (iii) evenness of assigned routing scores (measured by entropy). The results on OLMoE are illustrated in Figure 6.

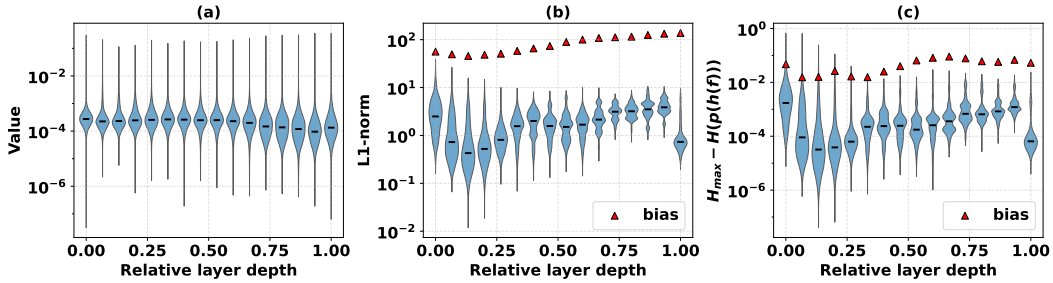


Figure 6: Distribution of (a) feature frequencies, (b) the L1 norm of routing logits assigned by features, (c) the entropy gap ($H_{\max} - H(p(h(\boldsymbol{f}))$), where $H_{\max} = \ln N_r$, $H(p(h(\boldsymbol{f})))$ is the entropy of routing logits assigned by feature f) across layers of OLMoE. We use the entropy gap instead of entropy to facilitate visualization. The short black lines indicate the medians.

Bias. Bias terms exert prominent and broad contributions to the routing logits across all models we examined. They tend to have a large L1 norm (ranking among the top 1%) and strong expert preferences, as indicated by their large entropy gaps (top 1% in each layer, except Layer 0 of OLMoE). Bias terms tend to assign negative routing logits to experts. In OLMoE, bias terms assign negative routing logits to all experts. The same pattern is observed across all Qwen3-30B-A3B layers we examined. Bias terms assign negative logits to about 79% of experts in Deepseek-V2-Lite, mainly in the bottom and middle layers, and 52% of experts in gpt-oss-20b (Figure 33).

Features. In OLMoE, there are more high-frequency features at the initial and final layers, whereas the features in middle layers have a more even frequency. There is a relatively large variation in feature norm at Layer 0 of OLMoE, and the mean norm slightly increases in the middle-to-late layers. Routing scores have the lowest entropy at Layer 0 of OLMoE , and increase fast until they reach the plateau in the middle layers, whereas in the top layers it decreases again. This phenomenon hints that there are more features in the bottom and top layers that relatively have a preference for the experts rather than middle layers. Gpt-oss-20b exhibits a distinctive plateau pattern (Figure 31):

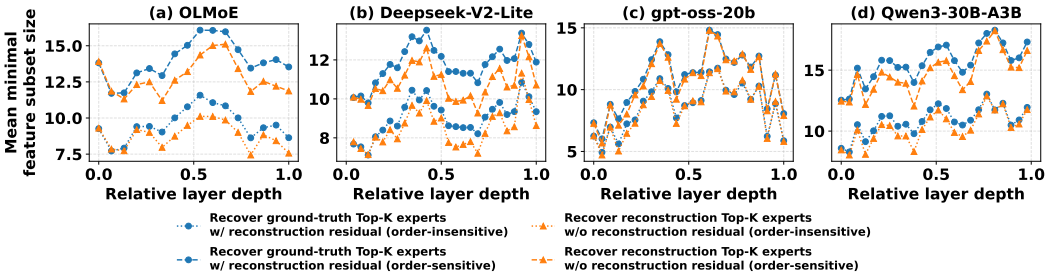


Figure 7: Distribution of the mean minimal feature subset size across layers of (a) OLMoE, (b) Deepseek-V2-Lite, (c) gpt-oss-20b, (d) Qwen3-30B-A3B. The results are computed on 10,000 randomly sampled tokens.

both the L1 norm of the routing logits and the entropy gap are more tightly concentrated in the middle layers than in the early and late layers, compared with the other three models.

Error. We observe that high routing logits MSE tends to occur for bad tokens, in non-English contexts, and in special contexts such as file name lists. We speculate that such content is comparatively rare in the training data of SAEs and may contribute to this phenomenon. Tokens with high routing logits MSE tend to be preceded by tokens with relatively high routing logits MSE, suggesting that poor reconstruction of contextual information can propagate and contribute to the error.

Minimal subset. The distribution of minimal subset size is illustrated in Figure 7. For OLMoE, recovering the reconstruction Top-K expert set without reconstruction residual requires 8.79 features on average (12.83 with order), while recovering the ground-truth Top-K expert set with reconstruction residual requires 9.57 (14.00 with order). Most middle layers tend to require more features to recover, compared with the top and bottom layers. We use the sum of the bias and error to recover routing decisions with a Recall@K of 21% for OLMoE and 19% for Deepseek-V2-Lite. The results indicate that SAEs can capture substantial routing-relevant information, and we can focus on a subset of features when inspecting their contributions to routing decisions.

6 Potential Applications

Our analysis suggests three potential applications beyond routing interpretation. **Targeted routing control.** Feature-level attribution identifies interpretable directions for promoting or suppressing expert selection, providing a basis for investigating how targeted routing changes affect model behavior. **Expert pruning and merging.** Comparing feature–expert affinities could help identify candidates for pruning or merging. Similar routing preferences would motivate further assessment of redundancy in expert outputs. **Speculative expert selection.** Characterizing which features determine selection near the top-K boundary could inform speculative expert prefetching and candidate selection. Whether alternative experts preserve model quality requires evaluating their outputs, since similar routing logits alone do not imply functional interchangeability.

7 Conclusion

In this work, we apply Top-K SAEs to study routing decisions in MoE language models. We decompose routing logits into weighted SAE features, decoder bias, and reconstruction residual, enabling us to analyze the contribution of features to the routing logits. We show that the affinity between experts and features can be employed to study what features promote or suppress an expert. To the best of our knowledge, this is the first work that quantifies the affinity and demonstrates that features can suppress an expert from being selected. Furthermore, our method can be applied to understand specific routing decisions from the perspective of features. We analyze the statistical properties of SAE features and SAE performance to reveal their characteristics. Finally, we discuss three implications of our work. We expect future work to improve the reconstruction fidelity of routing logits and develop more advanced automatic interpretation methods for a better understanding of the underlying mechanism of routing.

AI USE STATEMENT

In this work, we used generative AI tools for auto-interpretation related experiments, which includes prompt generation, feature interpretation, and experiment implementation. Additionally, we used generative AI tools for code generation related to figure plotting, experimental code checking, and grammatical error checking. We have checked all AI-generated content carefully. We take full responsibility for the content in this paper.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2(5):6, 2023.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Giang Do, Hung Le, and Truyen Tran. Do domain-specific experts exist in moe-based llms? In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 2324–2340, 2026.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning*, pp. 5547–5569. PMLR, 2022.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, volume 2025, pp. 26721–26754, 2025.

Jeremy Herbst, Stefan Wermter, and Jae Hee Lee. The expert strikes back: Interpreting mixture-of-experts language models at expert level. *arXiv preprint arXiv:2604.02178*, 2026.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

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.

Ilya Lasy, Nora Yinuo Cai, and Kola Ayonrinde. Routerinterp: Understanding superposed specialisation in mixture of experts routing. In *Forty-third International Conference on Machine Learning*, 2026.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pp. 6265–6274. PMLR, 2021.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. Olmoe: Open mixture-of-experts language models. In *International Conference on Learning Representations*, volume 2025, pp. 62061–62121, 2025.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International conference on machine learning*, 2024.

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EemtbhJOXc.

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. In *International Conference on Learning Representations*, 2017.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Luyu Wang and Shuhui Wang. Investigating expert semantic specialization in mixture-of-expert models. In *Mechanistic Interpretability Workshop at ICML 2026*, 2026.

Yan Wang, Yitao Xu, Nanhan Shen, Jinyan Su, Jimin Huang, and Zining Zhu. The illusion of specialization: Unveiling the domain-invariant" standing committee" in mixture-of-experts models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14601–14618, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

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*, 2025.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. Llama-moe: Building mixture-of-experts from llama with continual pre-training. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 15913–15923, 2024.

A TRAINING SAEs ON MIDDLE RESIDUAL STREAM

A.1 PRELIMINARIES

RMSNorm. Using RMSNorm (Zhang & Sennrich, 2019) for pre-normalization has become a widely adopted practice in LLMs. An RMSNorm is applied before the MoE layer:

$$\boldsymbol{m} = \frac{\boldsymbol{r}}{\text{RMS}(\boldsymbol{r})} \odot \boldsymbol{\gamma} \,, \tag{12}$$

where $\boldsymbol{r} \in \mathbb{R}^{N_m}$ is the middle residual stream, $\boldsymbol{\gamma} \in \mathbb{R}^{N_m}$ is the learnable scaling parameter, $\odot$ denotes the element-wise multiplication, and $\text{RMS}(\boldsymbol{r}) = ||\boldsymbol{r}||_2/\sqrt{N_m}$.

A.2 ALTERNATIVE WAY TO DECOMPOSE ROUTING LOGITS

We can apply middle residual stream $\boldsymbol{r}$ to study routing decisions:

$$h(\boldsymbol{m}) = \boldsymbol{W}_r \boldsymbol{m} = \boldsymbol{W}_r \frac{\boldsymbol{r} \odot \boldsymbol{\gamma}}{\text{RMS}(\boldsymbol{r})} \,, \tag{13}$$

where $\boldsymbol{r} = ((\sum_{i \in \mathcal{A}} \boldsymbol{z}_i \boldsymbol{f}_i) + \boldsymbol{b}_{\text{dec}} + \boldsymbol{\epsilon}(\boldsymbol{r}))$. The contribution of feature $\boldsymbol{f}_i$ to the routing logits is $h(\boldsymbol{f}_i) = \boldsymbol{W}_r \frac{\boldsymbol{z}_i \boldsymbol{f}_i \odot \boldsymbol{\gamma}}{\text{RMS}(\boldsymbol{r})}$. Training SAEs on $\boldsymbol{r}$ can introduce additional error in recovering routing logits due to the scale factor $\text{RMS}(\boldsymbol{r})$. However, this error does not influence the recovered ranks of experts because the RMS only controls the evenness of routing scores, and the Softmax function preserves the relative ordering of its inputs. Training SAEs on the MoE layer input provides a more accurate reconstruction of routing logits (See Appendix G), whereas this scheme suggests that we can analyze routing decisions directly based on the residual stream.

B IMPLEMENTATION DETAILS

Details about SAE training settings. We use the Sparsify library[5] to implement SAE training. We use the Adam optimizer. We initialize $\boldsymbol{W}_{\text{dec}}$ to the transpose of $\boldsymbol{W}_{\text{enc}}$ and the bias $\boldsymbol{b}_{\text{dec}}$ to the arithmetic mean of a sample of data points. A linear decay learning rate is adopted. We do not use learning rate warmup. When we train SAEs on MoE layer input, we set the initial learning rate to 1e-3 for OLMoE, Deepseek-V2-Lite and Qwen3-30B-A3B, but 5e-4 for gpt-oss-20b; we do not normalize the decoder in each step. When training on the middle residual stream, we follow the same practice as we train on the MoE layer input, but normalize the decoder in each step to reduce the feature death rate for models excluding OLMoE, since the feature death rate of SAEs trained on OLMoE is sufficiently low even without this normalization, and the normalization leads to lower Recall@K performance for SAEs trained on OLMoE. We set the expansion factor to 16, so $N_z = 46080$ for gpt-oss-20b, $N_z = 32768$ for other used models.

Details about MILP. We use the HiGHS MILP solver provided by the SciPy library. The optimization problems we designed in the main text are solvable. All reported solutions are optimal. We use the solution returned by the solver without additional tie-breaking.

C DETAILS ABOUT INTERPRETATION

We build on the activation-based interpretation framework of Paulo et al. (2025) to explain sparse autoencoder (SAE) features. For each feature, an *explainer* uses corpus activation statistics and sampled text contexts to identify when the feature fires and how its activation strength varies (Figure 8). A separate *examiner* receives the resulting description and marked-text probes for *single-token fuzzing*, adapted from the same work (Section C.2). We use this automatic score primarily to tune and develop the explanation pipeline in Section C.1; final explanation quality is assessed by manual review.

[5] https://github.com/EleutherAI/sparsify

C.1 FEATURE EXPLANATION

Our pipeline explains a feature through two sources of evidence: *corpus statistics* show which tokens and positions are associated with activation, and *text examples* show the surrounding context. An explainer LLM uses both to describe when the feature fires (Figure 8). This extends the explanation framework of Paulo et al. (2025); Table 1 compares the protocols.

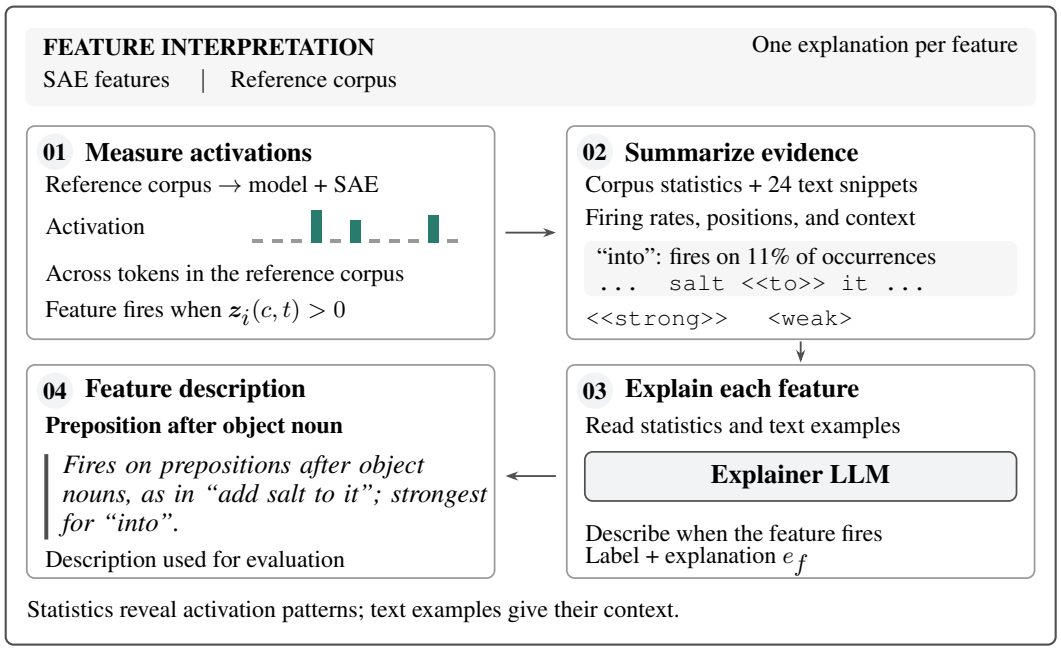


Figure 8: From feature activations to a natural-language explanation. Steps 2 and 4 illustrate feature 18211 from the SAE for Layer 7 of OLMoE (Table 8), using prompt excerpts and a shortened description generated by Kimi-K3 as an example. Activation bars are schematic; evaluation is shown in Figure 9.

Corpus activations. We record each feature's activation across the explanation corpus. Feature $\boldsymbol{f}_i$ fires at token t in sequence c when $\boldsymbol{z}_i(c,t) > 0$. Its density d_i is the fraction of tokens where it fires, and $Z_i = \max_{c,t} \boldsymbol{z}_i(c,t)$ is its maximum activation. For a text window W, the peak activation is $M_i(W) = \max_{t \in W} \boldsymbol{z}_i(c,t)$.

Table 1: Comparison of explanation protocols. The Delphi-based baseline combines the released prompt and highlighting rule with 40 uniformly sampled 32-token windows.

Aspect	Delphi-based baseline	Our pipeline
Evidence	40 activating windows.	Corpus statistics and 24 activating snippets; six additional silent examples for dense features.
Sampling	Uniform over activating windows.	Strong examples and examples across activation levels, with limits on repeated token contexts.
Context	Fixed 32-token windows.	Up to 31 tokens before and 8 after a selected firing token, with its sequence position indicated.
Highlighting	Tokens above $0.3M_i(W)$, relative to the window peak.	All firing tokens, with distinct strong and weak marks based on a feature-wide threshold.
Prompt and output	Three few-shot examples; a free-form description.	Instructions covering token identity, context, position, and topic; a short label and a description of at most 30 words.

Corpus statistics. Beyond text examples, the prompt provides statistics on *position* and *token identity*, along with overall firing density and maximum activation.

For position, we report firing rate and mean activation over 0, 1–3, 4–15, 16–63, and 64–127. This exponential partition reflects the stronger position effects observed within the first 16 tokens.[6]

For token identity, we list the eight most common activated tokens and, separately, preceding tokens among the 2,000 strongest firings and among all firings. Both 2,000 and 8 are adjustable. Each entry gives its count and conditional firing rate: the fraction of its occurrences in the explanation corpus where the feature fires (on the next token for preceding tokens).

Text examples. We sample 24 distinct firing positions, showing up to 31 tokens before and 8 after each: seven from the 200 strongest firings, eight across ten equal-count activation bins, and nine across ten equal-width bins. For the strongest sample, we limit repeated tokens and token pairs and take at most four snippets from one sequence, relaxing these limits if no other candidates remain. Snippets are ordered from strongest to weakest, with the selected token's position and a 0–10 score normalised by Z_i. `[START]` marks the sequence start.

Activation marks. Every firing token is marked as strong (`<<token>>`) or weak (`<token>`). The threshold is $0.3Z_i$: we adopt 0.3 from the released Delphi implementation,[7] but apply it to the corpus maximum rather than each window's peak.

Dense features. For dense features ($d_i \geq 0.1$), we raise the threshold to at least the 75th percentile of nonzero activations. We identify these features as dense and add six snippets centred on non-firing tokens.

Explainer output. The explainer considers token identity and form, syntax, position, and topic or register. We request a 2–6-word label and a description e_f of at most 30 words stating when the feature fires, when activation is stronger, and any informative exceptions. If it exceeds the limit, we request one shorter version. Only e_f is passed to the examiner.

Evaluation split. Explanation statistics and snippets use one corpus partition; evaluation probes use a disjoint, held-out partition (Section C.2).

Example prompt. The excerpts below show the evidence for *Preposition after object noun*, the feature illustrated in Figure 8.

```
FEATURE STATISTICS (over all 2,198 firings ...)
- fires on 0.1% of all tokens; max activation 4.67
- all firings -- fired-on token:
  " into" 165 (fires on 11% of its 1,541 occurrences)
[...]
--- Snippet 2 (strength 10/10, anchor at position 3) ---
[START] added fake eyes<< to>> make the body look more life-like
--- Snippet 5 (strength 9/10, anchor at position 123) ---
... Fill warm water<< in>> your sitz bath and add some Epsom salt<<
    to>> it. Soak
```

[6] We have not tested how alternative boundaries affect explanation quality.

[7] Delphi explainer implementation, default highlighting threshold.

Full explainer prompt. The template below is for a non-dense feature; braces mark feature-specific values. Dense features receive the additional evidence described above.

```
[system message]
You are an interpretability researcher analyzing one feature (latent
    unit) of a sparse autoencoder trained on a language model's
    internal activations. A feature fires with some strength on some
    tokens of some texts and is silent elsewhere.

You receive two kinds of evidence:
1. A STATISTICS CARD computed over ALL of the feature's firings in the
    corpus: firing rate by token position, and the most common
    fired-on and preceding tokens among the strongest firings and
    among all firings. Each listed token carries two numbers: its
    count among those firings, and in parentheses the CONDITIONAL RATE
    --- how often any occurrence of that token in the corpus fires
    (fired-on tokens) or is followed by a firing (preceding tokens).
    Compare conditional rates with the baseline firing probability: a
    token at 90% is a reliable trigger and points at the mechanism; a
    token at 5% is merely frequent text and points at background noise.
2. Sampled text SNIPPETS. Tokens on which the feature fires are
    wrapped in << >> (strong: at least 30% of the feature's maximum
    activation) or < > (weak: below that, often noise-like spillover);
    unmarked tokens do not fire. A snippet that begins with [START]
    starts at the first token of its sequence. Each snippet header
    gives the anchor token's strength on a 0-10 scale relative to this
    feature's maximum, and its position in the sequence.

Treat these pattern types as hypotheses and pick whichever the
    evidence supports best (combinations are common):
- identity of the fired-on token, word form, or subword;
- the immediately preceding token(s) or local syntax (a verb, an
    article, a bigram);
- the FORM of the preceding or fired-on token rather than its meaning:
    spelling pattern, final consonant or syllable shape (e.g. short
    stems ending in -ck/-t/-p), capitalization, digits, subword
    boundary;
- position in the sequence (e.g. the first few tokens; watch for
    [START]);
- topic, register, or language of the passage;
- combinations, e.g. a particle following a verb, or pronouns inside
    religious text.

The statistics card is the primary evidence for the overall pattern:
    it summarises every firing, while the snippets are a small sample
    and can look noisy at low strength. Use the snippets to confirm
    the hypothesis and to capture context that token tables cannot
    show. Answer two questions: what predicts WHETHER the feature
    fires (versus staying silent), and what predicts HOW STRONGLY it
    fires. If part of the firing has no discernible pattern, say so
    instead of inventing one.

End your reply with exactly these two lines (no markdown, nothing
    after them):
LABEL: <2-6 word name for the feature>
EXPLANATION: <at most 30 words: when it fires, what makes it fire
    strongly, and where it does NOT fire if that is informative>

[user message]
FEATURE STATISTICS (over all {n} firings in a 1,689,600-token corpus
    of 13,200 sequences x 128 tokens)
- fires on {d} of all tokens (baseline: a random token fires with
    probability {d}); max activation {max}
```

```
- firing rate by token position: pos 0: {r} | pos 1-3: {r} | pos 4-15:
    {r} | pos 16-63: {r} | pos 64-127: {r}
- mean activation by token position: {a} | {a} | {a} | {a} | {a}
- for each token below, the parenthesis gives how often ANY occurrence
    of that token in the corpus fires (fired-on token) or is followed
    by a firing (preceding token): a high rate means the token
    reliably triggers the feature, a low rate means it is merely
    frequent
- strongest {k} firings --- fired-on token: "{token}" {count} (fires
    on {rate} of its {occurrences} occurrences), ... [8 tokens]
     (the single most common fired-on token covers {share} of them)
- strongest {k} firings --- preceding token: "{token}" {count}
    (followed by a firing {rate} of its {occurrences} occurrences),
    ... [8 tokens]
- all firings --- fired-on token: "{token}" {count} (fires on {rate}
    of its {occurrences} occurrences), ... [8 tokens]
- all firings --- preceding token: "{token}" {count} (followed by a
    firing {rate} of its {occurrences} occurrences), ... [8 tokens]
  ("[START]" as preceding token = the fired-on token is the first of
    its sequence)

SNIPPETS where the feature fires (strongest first):
--- Snippet 2 (strength 10/10, anchor at position 3) ---
[START] added fake eyes<< to>> make the body look more life-like
--- Snippet 5 (strength 9/10, anchor at position 123) ---
... Fill warm water<< in>> your sitz bath and add some Epsom salt<<
    to>> it. Soak
```

C.2 Explanation evaluation

Evaluation task. We evaluate e_f with *single-token fuzzing*, adapted from Paulo et al. (2025, Section 3.3.2 and Appendix D.2). The examiner sees the description and text with one marked token, then predicts whether that token matches the description. Only the marked position changes; the text remains unchanged. Table 2 summarizes the adaptations. The explainer and examiner use separate prompts and may be different models. We hold the examiner fixed when comparing explainers.

Table 2: Adaptations to the Delphi fuzzing framework of Paulo et al. (2025).

Aspect	Delphi (published framework)	Our adaptation and motivation
Δ1 Judgment unit	All marked tokens must match the explanation (Paulo et al., 2025, Appendix D.2).	Mark exactly one token per probe, avoiding a variable number of token judgments within one binary decision.
Δ2 Negative evidence	Activating and non-activating examples form the evaluation set (Paulo et al., 2025, Section 4.2); random negatives have limitations (Paulo et al., 2025, Appendix D.5).	Mix silent-window negatives with silent tokens in activating windows, to test both context rejection and localization within a relevant context.

Probe sampling. We split the held-out 128-token sequences into non-overlapping 32-token windows and target the following probes per feature (Figure 9):

- **Positive (100):** select ten activating windows from each decile of $M_i(W)$ and mark the highest-activation token in each.
- **Negative A (50):** sample one token uniformly from each of 50 distinct windows where the feature never fires.
- **Negative B (50):** sample a zero-activation token uniformly within an activating window.

Each probe's reference label is 1 when $\boldsymbol{z}_i(c,t) > 0$ and 0 otherwise. These labels are hidden from the examiner and used to score its predictions.

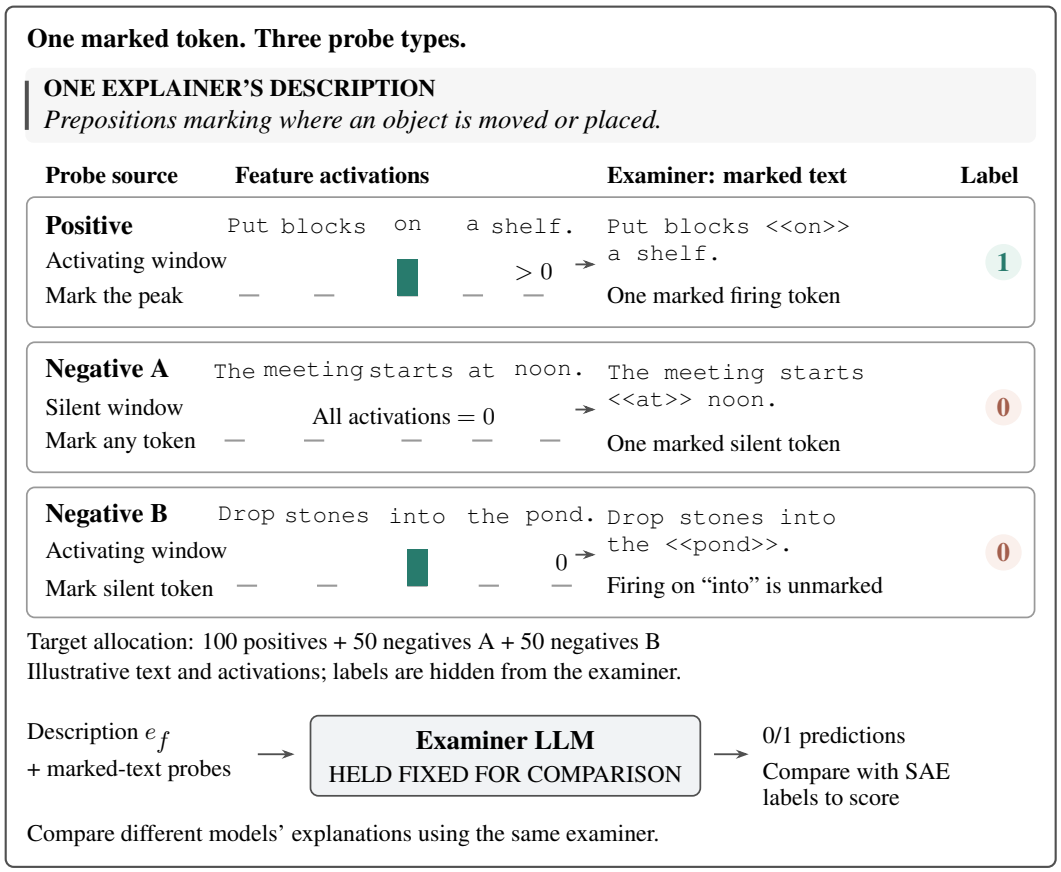


Figure 9: Single-token fuzzing. The examiner judges a marked token from the feature description. Its binary prediction is compared with the hidden SAE activation label. The examiner is held fixed across explainers. Text, token boundaries, and activations are schematic.

Examiner prompt. We use the released fuzzing instructions and three few-shot exchanges. The examiner answers five shuffled probes per query, returning one binary prediction each. An illustrative query is:

```
Task: Does each marked token match the explanation?
Return one binary answer per example.
Explanation: Prepositions marking where an object is moved or placed.
Example 0: Put the blocks <<on>> the shelf.
Example 1: The meeting starts <<at>> noon.
Example 2: Drop the stones into the <<pond>>.
Example 3: She added salt <<to>> the water.
Example 4: He placed the <<flowers>> in a vase.
```

The expected response is `[1, 0, 0, 1, 0]` (illustrative, not a recorded model output).

Released prompt excerpt. We reproduce the instructions and one few-shot exchange below.

```
You are an intelligent and meticulous linguistics researcher.

You will be given a certain latent of text, such as "male pronouns" or
    "text with negative sentiment". You will be given a few examples
    of text that contain this latent. Portions of the sentence which
    strongly represent this latent are between tokens << and >>.

Some examples might be mislabeled. Your task is to determine if every
    single token within << and >> is correctly labeled. Consider that
    all provided examples could be correct, none of the examples could
    be correct, or a mix. An example is only correct if every marked
    token is representative of the latent

For each example in turn, return 1 if the sentence is correctly
    labeled or 0 if the tokens are mislabeled. You must return your
    response in a valid Python list. Do not return anything else
    besides a Python list.

Latent explanation: Words related to American football positions,
    specifically the tight end position.

Test examples:

Example 0:<|endoftext|>Getty ImagesĊĊPatriots<< tight end>> Rob
    Gronkowski had his bossâĢĻ
Example 1: posted<|endoftext|>You should know this<< about>> offensive
    line coaches: they are large, demanding<< men>>
Example 2: Media Day 2015ĊĊLSU<< defensive>> end Isaiah Washington
    (94) speaks<< to the>>
Example 3:<< running backs>>," he said. .. Defensive<< end>> Carroll
    Phillips is improving and his injury is
Example 4:<< line>>, with the left side âĢĶ namely<< tackle>> Byron
    Bell at<< tackle>> and<< guard>> Amini

[1,0,0,1,1]
```

D MODEL DETAILS

The models we use for experiments are listed in Table 3. Routers of gpt-oss-20b use additional bias terms $\boldsymbol{b}_r$. Our method is still applicable to this model because the bias terms are constant.

Table 3: MoE models we used in experiments. N_r = Number of routed experts, K = Number of selected routed experts, N_s = Number of shared experts, N_{layers} = Number of layers, N_m = Model dimension.

Model	N_r	K	N_s	N_{layers}	N_m	Total Parameters	Active Parameters
OLMoE	64	8	0	16	2048	7B	1B
DeepSeek-V2-Lite	64	6	2	27	2048	16B	2.4B
Qwen3-30B-A3B	128	8	0	48	2048	30.5B	3.3B
gpt-oss-20b	32	4	0	24	2880	20.9B	3.6B

E INTERPRETATION OF FEATURE-EXPERT ROUTING AFFINITY

E.1 INTERPRETATION OF PROMOTIVE/SUPPRESSIVE FEATURES RELATED TO L1-E9 OF OLMOE

The details of promotive and suppressive features are shown in Table 4.

Table 4: Promotive and suppressive features regarding the activation of L1-E9 of OLMoE. Abstract: label (bold) and explanation written by Kimi-K3.

Feature id	Mean rank of L1-E9	Mean logit of L1-E9	Abstract
Promotive			
22703	4.45	0.72	**Indefinite article "a" detector** Fires on every occurrence of the token " a"; weakly on " an", " A", and math-variable "a". Strongest after prepositions and late in sequences; never fires on "the" or other function words.
12162	3.36	0.55	**Definite article "the" detector** Fires on virtually every "the"/"The" regardless of topic or context; strength grows with sequence position (weak near [START]); weak spillover onto "a" and punctuation at sequence starts.
28379	3.20	1.12	**Second element of compounds** Fires on the second element of compounds after 'green', 'smart', 'international' and Chinese *zhu*/*shang*; strongest on 'green' compounds (Greenwood, Green Bay); not after other adjectives ('red cabbage').
17842	7.62	0.67	**Nouns following main/primary adjectives** Fires on nouns following main/primary-type adjectives (main, primary, key, major, basic, fundamental; Chinese *zhudong*); strongest for primary/main collocations, weaker spillover onto following 'of'; adjectives themselves don't fire mid-sequence.
27211	7.79	0.76	**Token following "good"** Fires on the token immediately after "good" (˜100% reliable, even mid-word as in Goodfellas; weakly after bad/excellent), strongest when "Good" opens a phrase or title, with decaying spillover.
20842	3.44	0.95	**Token following "first"** Fires on the token immediately after "first"/"First"/"1st" (89% trigger rate), regardless of that token's identity; strongest in compounds like "first-time", "first thing"; weak spillover after "initial", "next", "*shang*".
846	6.77	0.58	**Noun following a size adjective** Fires on the token right after a size adjective (large, big, huge, vast, largest, *da*; 90–100% reliable), strongest on quantity nouns — number, amount, majority — in "large X of" phrases.
30096	6.44	0.78	**Noun following an ordinal** Fires on the noun right after an ordinal ("third eye", "19th century", "2nd Earl"); strongest directly after "third"/"second"; sometimes fires on the ordinal suffix "th"; silent without an ordinal trigger.
26998	7.65	0.69	**Tokens following "small"-family adjectives** Fires on tokens right after "small"-family adjectives (smaller, tiny, limited, little); ˜99% trigger rate after "small", often spanning the noun phrase; weaker after "little"/"faint"; fired-on token identity irrelevant.
1454	7.33	0.72	**Token following "top" (top-X phrases)** Fires on tokens following "top"/"Top" (also "upper", Chinese *shang*), marking "top X" phrases like top-notch, Top 10, top officials; strongest for ranking-adjective uses, weak spillover after synonyms.
29232	7.97	0.55	**Tokens following the word "second"** Fires on the one-to-three tokens immediately after "second"/"Second"/"secondary" (98% trigger rate), strongest in ordinal compounds like "second world war" and "second only"; weak after adverbial "Second," and absent elsewhere.

Table 4: Promotive and suppressive features regarding the activation of L1-E9 of OLMoE. Abstract: label (bold) and explanation written by Kimi-K3. (continued)

Feature id	Mean rank of L1-E9	Mean logit of L1-E9	Abstract
Promotive (continued)			
17314	7.73	0.52	**Word after strength adjectives** Fires on any token directly following strength/intensity adjectives (strong, powerful, solid, weak, stronger, highly); strongest after "strong", especially compounds like "strongman"; weak spillover after near-synonyms like "stable".
12892	7.99	0.68	**Token following "final"** Fires on the token right after 'final'/'Final' (nearly every occurrence), including inside finalists/finale/finalized; more weakly after ultimate, senior, initial, last, concluding. Strongest immediately after 'final', fading over 1-2 tokens.
12151	7.29	0.42	**Token after magnitude adjective** Fires on the token immediately following degree/severity adjectives (major, significant, serious, severe, massive, extreme); strongest on the modified noun, weaker on punctuation after clause-final adjectives and phrase spillover.
7699	7.78	0.92	**"ent" substring detector** Fires on tokens containing the substring "ent" regardless of language or topic; strongest on "ent"-initial tokens (ent, ente, ently), weaker on embedded "ent" (gent, enty) or near-misses (encia).
31641	7.55	0.42	**Token following "great"** Fires on the token after great/Great/greater/greatest/greatly (and Chinese *da*); strongest when Great opens a capitalized proper name or concrete noun, weaker with greater/abstract nouns; not on great itself.
29054	3.54	0.43	**English definite article "the"** Fires on the English definite article: nearly every lowercase " the" fires; capitalized "The" and "a/an" weakly. Strongest after prepositions and later in the sequence; silent elsewhere.
85	7.38	0.84	**'ent' letter-sequence in tokens** Fires on tokens containing substring 'ent' regardless of topic; strongest on subword 'ents' (sacraments, moments), medium on 'ented', weak on embedded/capitalized forms (Kent, AGENT); standalone 'vent' mostly silent.
19767	4.91	0.54	**Particles following the verb "come"** Fires on particles (out, up, down, off, in) right after forms of "come"; strongest in idiomatic phrasal verbs like "come out" or "comes down to"; weak spillover after other verbs.
15851	4.16	0.34	**Non-article determiners after nouns** Fires on non-article determiners and quantifiers (this, all, every, each, my, their, more) immediately following a noun or pronoun; strongest for post-nominal demonstratives; silent on articles and most other occurrences.
Suppressive			
17629	48.45	−0.32	**Lowercase "and" conjunction detector** Fires on essentially every lowercase "and" token (also "*i*", weakly "&&" and elided "an"'); strong throughout the sequence but weak at position 0; capitalized sentence-initial "And" mostly does not fire.

Table 4: Promotive and suppressive features regarding the activation of L1-E9 of OLMoE. Abstract: label (bold) and explanation written by Kimi-K3. (continued)

Feature id	Mean rank of L1-E9	Mean logit of L1-E9	Abstract
Suppressive (continued)			
23559	48.35	−0.19	**Comma token detector** Fires on almost all comma tokens (89%), including Chinese "," and comma-containing subwords; strongest on clause-boundary commas after discourse markers, later in sequences; weak in lists, math, code.
14103	57.47	−0.29	**"well" in "as well (as)"** Fires on "well" after "as": additive "as well as" or clause-final "as well" (also). Strongest in "as well as"; weak on "as early as"; silent on ordinary adverb/noun "well".
12303	50.22	−0.20	**Example- and exception-introducing words** Fires on words introducing examples or exceptions: "including" and "except" always and strongest; "especially", "particularly", "specifically" moderately, typically after commas; weaker on capitalized or code variants; silent elsewhere.
29266	48.34	−0.23	**Comma fused with closing quote** Fires on tokens fusing a comma with a closing quotation mark (," ," ,'), ending quoted speech—100% reliable, strongest in prose dialogue; weak spillover on citation brackets and ' Ha'.
1870	48.30	−0.13	**Clause-boundary commas in prose** Fires on comma tokens (56% of ','; also comma-containing tokens), strongest on clause-boundary/parenthetical commas in prose and later positions; weak or silent in code, JSON, numbers, sequence starts.
14790	48.46	−0.20	**Degree words after "how"** Fires on degree/quantity adjectives and adverbs (much, many, long, good, important) after "how"/"however" in questions and exclamations; strongest on gradable adjectives; silent on "how to" and degree words without "how".
15468	50.14	−0.21	**Spaced-ellipsis dots and predictable continuations** Fires on dots continuing a spaced ellipsis (". . ."), strongest on later dots; weaker on "then" after "but" and artifact continuations (mojibake bytes, SQL dashes, LaTeX "\bigg\{").
15933	52.05	−0.16	**Sentence-initial capitalized For/To** Fires on capitalized "For"/"To" at sentence, line, title, or sequence starts, never lowercase mid-sentence; strongest on "accepted" in Stack Exchange vote markup and sentence-initial "To" in prose.

E.2 Observation on L15-E39 of OLMoE

(Herbst et al., 2026) find that L15-E39 of OLMoE activates on information about "*scientific and culinary procedural descriptions involving preparation, substances, and specific ingredients.*" Our results show that the promotive features are related to multiple topics, including body, exercise, food, and drinks. Table 5 shows that there is an evident overlap between their explanation and the abstracts of promotive features. We observe that the suppressive features are related to numbers, the United States, and dots in abbreviations. The feature details are shown in Table 5, in which we adjust the criterion to show an appropriate number of features here. The routing logits distribution is shown in Figure 10. The routing logits of L15-E39 span a longer negative interval and a shorter positive interval, compared with those of L1-E9.

Table 5: Promotive and suppressive features regarding the activation of L15-E39 of OLMoE. We show the top 10 and bottom 10 features ranked by their mean rank for L15-E39 here. Abstract: label (bold) and explanation written by Kimi-K3.

Feature id	Mean rank of L15-E39	Mean logit of L15-E39	Abstract
Promotive			
20185	4.24	0.27	**Workout rep-counting instructions** Fires across tokens in exercise-instruction text, strongest on the formulaic "one rep, do six, switch sides, repeat, rest N seconds" template; " seconds" triggers reliably; silent elsewhere.
21057	5.03	0.27	**The verb "eat" and its forms** Fires on nearly every occurrence of eat/eating/ate/eaten (96–100% conditional), including the "eat" subword in "overeat"; weak spillover onto tokens following "eat" and faint firing on bite, chew, devour.
12124	5.44	0.21	**Muscle and exercise anatomy topic** Fires across spans about muscles and exercise; strongest on explicit muscle terms ("muscles" fires 48% of occurrences, abs, biceps, flexor); silent at sequence starts and in non-muscle text.
16947	5.62	0.20	**Breakfast / morning-meal context** Fires on "breakfast" (97% of occurrences) and spills onto neighboring tokens in breakfast passages (eggs, pancakes, cereal, waffles); strongest on "breakfast" itself; punctuation/common words fire only as spillover, silent elsewhere.
24564	5.66	0.26	**Exercise and movement-training topic** Fires broadly across passages about physical exercise, fitness, and motor-skill coaching, on whatever tokens occur; grows stronger later in densely on-topic text; silent at sequence starts and off-topic passages.
12301	5.79	0.23	**Animal feeding and diet** Fires across spans about feeding animals — pet food, dog diets, calf feeding, dosing mixed with feed; " feed" reliably triggers (46%). Strongest in feeding-dense passages; silent elsewhere.
22810	5.85	0.25	**Bread-baking topic passages** Fires broadly across ordinary tokens once the passage concerns bread baking (dough, loaves, flour, recipes); strongest mid-sequence in sustained baking discussion; silent at sequence starts and in off-topic text.
26248	5.89	0.21	**'muscle' word and following phrase** Fires on every 'muscle'/'muscles' occurrence (any subword, capitalization), strongest on the word itself; activation spills onto following clause tokens (strength, contract, move); silent at sequence starts and in muscle-free text.
1597	6.05	0.25	**Beer and brewing topic** Fires across passages about beer, breweries, lagers, malt, and hops; strongest in beer catalog/review descriptions; the token " beer" itself triggers reliably (65%); weak scattered firings elsewhere are noise.
1459	6.27	0.23	**Candy and snack passages** Fires broadly across passages about candy, sweets, and snacks, on many token types; strongest in dense candy lists (jellybeans, candy bars, lollipops); weak spillover to general food text; absent elsewhere.
Suppressive			

Table 5: Promotive and suppressive features regarding the activation of L15-E39 of OLMoE. We show the top 10 and bottom 10 features ranked by their mean rank for L15-E39 here. Abstract: label (bold) and explanation written by Kimi-K3. (continued)

Feature id	Mean rank of L15-E39	Mean logit of L15-E39	Abstract
Suppressive (continued)			
32437	49.79	−0.78	**"United" token detector** Fires on the token "United" in any case (States, Kingdom, Nations); strongest capitalized in "the United States", weaker lowercase; silent on "States" or "the" alone.
19994	48.55	−0.31	**Thousands separators in numbers** Fires on punctuation embedded in digit strings: strongest on thousands-separator commas in large numbers (120,000), weaker on decimal points, $, hyphens, colons; silent on ordinary punctuation between words.
9187	48.21	−0.41	**Scripture citation chapter numbers** Fires on chapter numerals (Arabic or roman) in Bible/religious citations after a book name (Exodus 20, Psalm 121); strongest in parenthetical chapter:verse references; silent on verse numbers and ordinary numbers.
793	47.70	−0.41	**Thousands separators in large numbers** Fires on commas (weakly periods, colons, hyphens) separating digit groups inside large numbers; strongest before "000" in round figures like $50,000; silent on ordinary prose punctuation.
1725	46.80	−0.85	**Website UI/navigation boilerplate** Fires on tokens inside web-page chrome: navigation menus, buttons, form/e-commerce messages, category footers—short line-broken fragments, often after <—endoftext—>. Strongest on dense runs of such UI text; silent in ordinary prose.
30290	46.64	−0.22	**Punctuation inside dates and timestamps** Fires on commas, dashes, colons, slashes, and "on" within date/time expressions (weekday commas, "asked by … on <date>", ISO dates, timestamps); strongest in blog/forum date lines; silent on ordinary prose punctuation.
20792	46.42	−0.25	**Citation locator abbreviations (p., pp., No.)** Fires on locator abbreviations in citations — p., pp., No., Vol., ch., Q. — spanning abbreviation and period before numbers; strongest on pp./p.; silent in ordinary prose.
30770	46.41	−0.39	**Abbreviation components after periods** Fires on abbreviation/initialism components following a period: letters in U.S., D.C., U.K., p.m. and chunks like vs, Sr, Co, Inc. Strongest on S in U.S.; silent on ordinary post-period words.
30289	46.40	−0.43	**Periods and letters inside dotted initialisms** Fires on medial periods and capital letters inside dotted initialisms (U.S., D.C., B.C., U.K., Ph.D.); strongest on "D.C." and "B.C."; silent on ordinary sentence-final periods and commas.
24368	45.98	−0.31	**Website boilerplate and calls-to-action** Fires on website chrome — navigation links, buttons, calls to action, form/comment/subscription notices ("comments", "available", "below", "here"); strongest on explicit reader-directed imperatives; silent in ordinary prose.

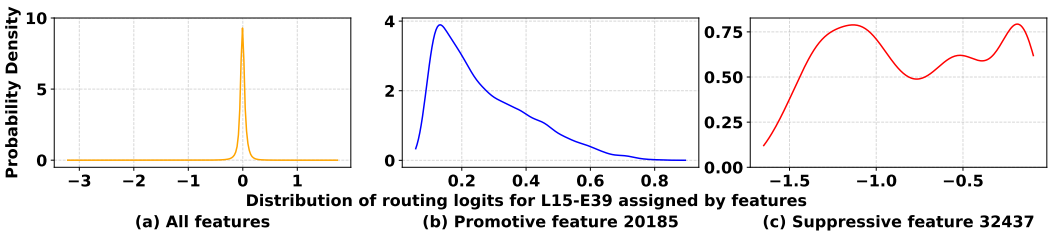


Figure 10: (a) Routing logit distribution of L15-E39 of OLMoE. (b) Distribution of routing logit assigned by the first promotive feature shown in Table 5. (c) Distribution of routing logit assigned by the first suppressive feature shown in Table 5.

E.3 Observation on L24-E9 of Qwen3-30B-A3B

Here we provide another example on a different model. (Herbst et al., 2026) find that L24-E9 of Qwen3-30B-A3B activates on "*Tokens involving definitions, nomenclature, translations, or technical synonyms.*" We again use our method to inspect this expert. The details of the features are shown in Table 6, in which we adjust the criterion to show an appropriate number of features here. We find the promotive features coincide with their given abstract. The suppressive features are mainly related to code and numbers. The routing logits distribution is shown in Figure 11.

Table 6: Promotive and suppressive features regarding the activation of L24-E9 of Qwen3-30B-A3B. We show the top 10 and bottom 10 features ranked by their mean rank for L24-E9. We adjust the criterion to show an appropriate number of features here. Abstract: label (bold) and explanation written by Kimi-K3.

Feature id	Mean rank of L24-E9	Mean logit of L24-E9	Abstract
Promotive			
10118	3.49	0.18	**Alias constructions ("also known as")** Fires on tokens inside alias/appositive renamings — "also known as", "aka/a.k.a.", "also/often called", parenthetical or "or"-introduced alternative names; strongest on "also known as"/"aka"; silent on ordinary disjunctions and commas.
15048	4.15	0.17	**Definitional statements ("X is/refers to")** Fires in term definitions: on copula is/are/'s, defining verbs (defined as, refers to, means), and the following article; strongest in explicit formal definitions; absent from ordinary uses of is.
6905	4.64	0.26	**Term-naming constructions ("is called/known as")** Fires within definition/naming phrases — "is called", "known as", "call X a", parenthetical glosses, even Chinese *chengwei*; strongest on "called"/"known as" themselves; ordinary "is"/"the" elsewhere stay silent.
24647	5.28	0.20	**Parenthetical definitions and clarifications** Fires on tokens introducing parenthetical or appositive clarifications ('(', 'or', 'which is', 'means', 'i.e.') that define/reword a preceding term; strongest on parenthetical definitions; silent on non-explanatory uses.
18569	5.36	0.28	**Restatement markers: i.e. / that is** Fires on restatement connectives: "i.e." (near-deterministic on ".e"), "that is,", "in other words", "meaning that", plus Russian/Chinese equivalents; strongest on "i.e.,"; weak spillover on plain copula "that is".

Table 6: Promotive and suppressive features regarding the activation of L24-E9 of Qwen3-30B-A3B. We show the top 10 and bottom 10 features ranked by their mean rank for L24-E9 here. Abstract: label (bold) and explanation written by Kimi-K3. (continued)

Feature id	Mean rank of L24-E9	Mean logit of L24-E9	Abstract
Promotive (continued)			
11567	5.60	0.31	**Naming and designation expressions** Fires on naming phrases: “called”, “named”, “known as”, “name is”, “referred/identified as”. Strongest on “called”/“named”; weak spillover onto adjacent “as”/“the”; silent on these tokens outside naming contexts.
8066	5.94	0.27	**Term–definition boundary marker** Fires at term–definition boundaries—on “means”, “is”, “refers to”, colons, parentheses, or the definition’s first word—in dictionaries, glossaries, and legal terms; strongest in reference entries, absent in ordinary prose.
7119	6.03	0.25	**Opening quotation marks** Fires on opening quotation marks (curly “/‘ most reliably, straight quotes when opening); strongest for attributed quotations in news text, weaker on scare quotes; silent on closing quotes and apostrophes.
3796	6.13	0.27	**Restatement/rephrasing transition markers** Fires on tokens inside rephrasing/summarizing connectives (“in other words,” “in short,” “simply put,” “i.e.,” even Russian “*slovom*”); strongest on canonical formulas; silent on the same words elsewhere.
16990	6.52	0.25	**Quoted word-meaning glosses** Fires on words inside quoted or parenthesized glosses giving a word’s meaning/translation in etymologies and dictionaries; strongest on content words of quoted translations; silent on the defined word itself.
Suppressive			
29915	94.65	−0.36	**Function parameter declarations in code** Fires inside parameter lists of function definitions, callbacks, catch clauses in code; strongest on short parameter names (e, t, err, args); weak on commas/parens and reuses; absent in prose.
15816	91.80	−0.26	**Digits in long identifier strings** Fires on digits (occasionally hex letters/hyphens) embedded in long digit runs — DOIs, ISBNs, GUIDs, hashes, long decimals; strongest deep in uninterrupted runs. Silent on ordinary numbers, dates, quantities.
20369	89.29	−0.38	**Boilerplate and auto-generated text** Fires on formulaic boilerplate: Apache license headers, MySQL dump /*!40101 SET statements, ‘Auto-generated method stub’ comments; strongest deep inside such blocks; weak spillover on digit strings; absent from ordinary prose.
754	88.16	−0.39	**Digits inside ISBN numbers** Fires on digit (and hyphen) tokens within ISBN book identifiers, strongest mid-run in labeled ISBN-10/ISBN-13 strings; silent on ˜99% of digits elsewhere, with weak spillover to phone numbers, DOIs, ASINs.
9624	87.85	−0.30	**Digits inside phone numbers** Fires on digits and separating hyphens/spaces inside telephone numbers, mainly after the area-code group; strongest mid-number; silent on area codes and non-phone digit strings like IDs or timestamps.

Table 6: Promotive and suppressive features regarding the activation of L24-E9 of Qwen3-30B-A3B. We show the top 10 and bottom 10 features ranked by their mean rank for L24-E9 here. Abstract: label (bold) and explanation written by Kimi-K3. (continued)

Feature id	Mean rank of L24-E9	Mean logit of L24-E9	Abstract
Suppressive (continued)			
1284	87.57	−0.37	**Digits inside phone numbers** Fires on digits inside multi-digit numbers, strongest on digits starting a phone number's final group after hyphen/space; weak in other digit strings; silent on the final digit.
7801	86.68	−0.43	**Digits inside phone numbers** Fires on digits within telephone/fax numbers, strongest mid-way through the final digit group once the phone-number pattern is established; silent on the vast majority of digits in other contexts.
23481	86.17	−0.35	**Digits in ranges and error margins** Fires on digits inside numeric uncertainty/span expressions — confidence-interval bounds ("95% CI 2.77 to 10.62"), ± error margins, and ranges; strongest on CI bounds; silent on ordinary numbers.
11934	86.10	−0.34	**Citation volume-number digits** Fires on digits forming journal volume numbers in academic references (after year/semicolon, before issue/pages), strongest on the volume's final digit; years, page numbers, and ordinary digits stay silent or weak.
5492	85.56	−0.24	**Repetitive boilerplate template text** Fires inside formulaic, auto-generated boilerplate: strongest on Java/C# setter bodies (this.x = x; "\tthis" triggers 83%); weaker on WordPress comment footers ("photo"/"picture"/"Logo" + "\n\n\n", "Change"). Silent in ordinary prose.

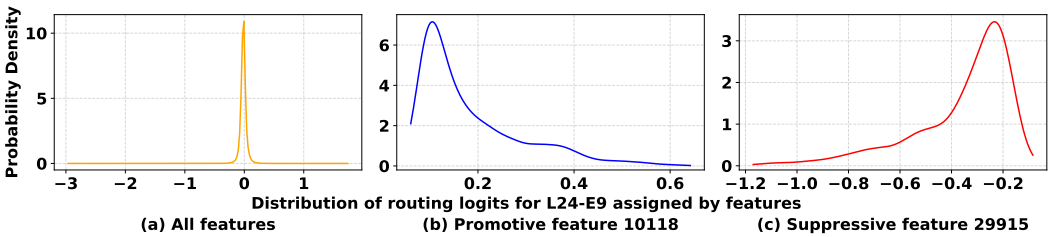


Figure 11: (a) Routing logit distribution of L24-E9 of Qwen3-30B-A3B. (b) Distribution of routing logit assigned by the first promotive feature shown in Table 6. (c) Distribution of routing logit assigned by the first suppressive feature shown in Table 6.

F Observation of routing decisions at token "to"

F.1 Observation on Layer 0

In this section, we present the detailed information about the top-16 features ranked by their L1 norms at token "to" in Layer 0 of OLMoE (Table 7).

Table 7: Top-16 features ranked by L1 norm at token " to" (Layer 0). Abstract: label (bold) and explanation written by Kimi-K3. For feature 8333, the explanations of DeepSeek-V4-Flash-reasoning and Qwen3.8-27B are also shown.

Ranked by L1 norm	Feature id	L1 norm	Activation	Abstract
1	6584	14.38	3.55	**Universal "to" token detector** Fires on every occurrence of "to" in any capitalization, plus equivalents (à, unto, \to), regardless of context; strong everywhere, weaker on "to"-like subwords inside other words or usernames.
2	6758	10.14	2.83	**The word "to" (infinitive marker)** Fires on virtually every token "to" (any case/spacing, even French "à"). Strongest after verbs/adjectives taking infinitives (wanted, need, able, going, referred); weaker sentence-initially, in hyphenated compounds, and code.
3	656	8.11	1.38	**Token after "a"/"an"** Fires on whatever token immediately follows "a"/"an"/"A" (near-deterministic, any context); strongest at sequence start, where initial "a" itself also fires; weak spillover onto the following token.
4	36	7.16	2.17	**" to" token detector** Fires on every lowercase " to" token with leading space; strongest at sequence start and after punctuation or numbers; weaker after verbs like need/want; silent on "To" and spaceless "to".
5	2565	6.11	4.13	**Dense early-sequence position feature** Fires on ˜98% of tokens regardless of identity; strength tracks position: strongest at positions 1-15, weak or silent at position 0 and late in the 128-token window.
6	32421	5.89	1.70	**Object slot after eat/buy-class verbs** Fires on the token immediately after acquire/consume/transfer verbs (eat, buy, sell, purchase, download, kill) — the object slot, even '-er' in buyer; near-deterministic after core verbs, weak spillover elsewhere.
7	25450	5.29	1.34	**Continuation of preceding token** Fires on tokens continuing or tightly bound to the previous token: subword/suffix continuations of split words (ragtime, sculptors) strongest, compound elements and cliticized function words weaker; silent at boundaries.
8	16389	3.68	2.53	**Light-verb phrases, especially initial "give…"** Fires within common verb-complement phrases (give us an idea, gave up on me, take a look); strongest on sequence-initial "give" forms, weak mid-sequence; silent on most function-word occurrences.
9	8307	3.30	1.03	**High-frequency function words** Fires on top-frequency English function words and punctuation (the, a, and, to, of, in, ., ,); strongest on "the"/"a" and early sequence positions; silent on content words, subwords, code, math.
10	29165	2.83	1.05	**Preposition "in" and stock collocations** Fires on virtually every " in"; strongest in tight particle collocations (stock in, in stock, engage/stored/invest/filled in) and on stock-compounds (stock market/exchange); weak spillover on other function words.

Table 7: Top-16 features ranked by L1 norm at token " to" (Layer 0). Abstract: label (bold) and explanation written by Kimi-K3. For feature 8333, the explanations of DeepSeek-V4-Flash-reasoning and Qwen3.8-27B are also shown. (continued)

Ranked by L1 norm	Feature id	L1 norm	Activation	Abstract
11	17901	2.71	1.77	**Sequence-initial indefinite noun phrase** Fires across indefinite-article noun phrases ("a/an X…", often with "of"); near-maximal when such a phrase opens the sequence, decaying over early tokens; otherwise weak, diffuse spillover.
12	32469	2.41	1.05	**Religious (Christian) passage register** Fires broadly within religious, especially Christian/theological, passages; strongest on religious terms (God, Jesus, Christ, peace, His) and at sequence starts; essentially silent in secular text except weak spillover.
13	8333	2.19	1.43	*Kimi-K3:* **Function words, strongest at sequence start** Fires on common function words and punctuation — always on " the" — rarely on content words; strongest on a sequence's first tokens, decaying with position. ***We include the following abstracts as they are more consistent with our manual interpretations.*** *DeepSeek-V4-Flash-reasoning:* **The/people text-start** Fires on "the" always and on people/social nouns often; strongest at sequence start, weak on common words later. Not on abstract/proper nouns or non-sentence punctuation. *Qwen3.8-27B:* **The and early common tokens** Fires on frequent tokens, always on "the"; strongest early in sequences and on people/school terms; weak spillover elsewhere, often not on rarer content, acronyms, parentheses.
14	25327	2.12	0.99	**Third-person past-tense narrative prose** Fires weakly across narrative prose; strongly on third-person pronouns (he/she/his/her/him) and past-tense auxiliaries (was/had), especially at sequence starts. Silent in code, poetry, non-English, and formal/technical text.
15	8942	1.99	0.62	**Early-sequence punctuation and function words** Fires on punctuation and common function words within the opening ˜15 tokens of a sequence; strongest on sentence-ending periods; essentially silent after position 64.
16	22641	1.93	0.52	**Successor of 'book'-type stems and ' *'** Fires on tokens immediately after 'book(s)', film, movie, super, stock, car, York, British, or comment-line ' *'; strongest at sequence start and on compound continuations (book+making/mark); silent elsewhere.

F.2 Observation on Layer 7

In this section, we present the detailed information about the top-16 features ranked by their L1 norms at token "to" in Layer 7 of OLMoE (Table 8) and the routing logits information (Figure 12).

Table 8: Top-16 features ranked by L1 norm at token " to" (Layer 7). Abstract: label (bold) and explanation written by Kimi-K3.

Ranked by L1 norm	Feature id	L1 norm	Activation	Abstract
1	32209	6.05	2.80	**The word "to", mostly directional** Fires on "to" in any capitalization, even Spanish "a" or camelCase "To"; strongest for prepositional/directional "to" (go to, changes to, 4 to 30.6); weaker or silent on infinitival "to be/help/join".
2	8846	5.15	3.33	**Default-on running prose feature** Fires on most tokens; strongest on common English function words early in a sequence, fading with position. Silent at position 0, mostly silent on newlines, weak in math/code/late text.
3	23450	4.92	3.75	**Recipient 'to' after transfer verbs** Fires on "to" introducing a transfer recipient (give, send, grant, present, sell); weakly on for/with/upon/the inside recipient phrases; silent on 97% of "to", including most infinitives.
4	9998	4.01	2.78	**Indefinite noun-phrase postmodifiers** Fires on prepositions, infinitival 'to', 'that', participles introducing postmodifiers of indefinite noun phrases ('a sign in heaven'); strongest after 'a + noun'; silent in verb complements and after definite nouns.
5	13005	3.99	2.78	**Prepositions introducing affected people** Fires on prepositions whose complement is people benefiting or affected, especially after welfare nouns (life, lives, rights, benefits); strongest with benefactive "for"; silent on ordinary prepositional phrases.
6	18211	2.54	2.55	**Preposition after object noun** Fires on prepositions (to, on, in, into, out, from) immediately following a noun — the destination phrase in verb-object-preposition constructions ("add salt to it"); strongest for "into"; rare after verbs.
7	32032	2.52	2.69	**Narrative action sequences** Fires in narrative prose on connective tokens (commas, "and", "said", "to") within chains of unfolding actions; strongest in dense dramatic sequences; absent in code, news, and non-narrative text.
8	25609	2.42	2.17	**Word forms of "give"** Fires on inflections of "give" (given, give, giving, gave most reliably), weaker on derived "giv-" words (giver, giveaway, gift) and adjacent-token spillover; silent elsewhere.
9	29038	2.33	1.72	**Active verbs targeting another party** Fires on active transitive verbs taking a person/being object—kill, slay, love, care for, help—and their particles; strongest on kill-family; silent in passives like "been killed".
10	32709	2.32	1.39	**Quoted, poetic, foreign, archaic passages** Fires in bursts within quotations, verse, archaic prose, and foreign/mojibake text, starting a few tokens into the passage; strongest on sustained spans; silent at sequence starts and ordinary expository prose.
11	971	2.31	1.49	**Biblical and sermon discourse** Fires on ordinary tokens within Christian biblical/sermon passages, strongest during extended scripture narrative (Samaritan woman, Acts, disciples); silent in secular text and at sequence starts, weak spillover to other mythology.

Table 8: Top-16 features ranked by L1 norm at token " to" (Layer 7). Abstract: label (bold) and explanation written by Kimi-K3. (continued)

Ranked by L1 norm	Feature id	L1 norm	Activation	Abstract
12	7959	1.95	1.57	**Basic survival necessities** Fires on words for basic survival needs — food, feed, eat, shelter, supplies, water, meals — especially in enumerations like "food and shelter"; strongest in relief/survival contexts, weak on ordinary food mentions.
13	19319	1.87	1.41	**Famous quotes and catchphrases** Fires on tokens within verbatim famous quotes, lyrics, and catchphrases; strongest on the phrase's recognizable core words; silent in ordinary prose and before the quote becomes recognizable.
14	4511	1.84	1.62	**Prepositions after mass nouns** Fires on prepositions/function words immediately following uncountable mass nouns (money for, water in, data from, food over); strongest on clean mass-noun+preposition bigrams; same prepositions elsewhere stay silent (˜1%).
15	15329	1.78	1.36	**Drinking and beverage words** Fires on nearly every occurrence of drink/beverage vocabulary (drink, drinks, drinking, drank, juice, thirst, beverages); strongest on the drink-word itself, weak spillover to adjacent tokens and related words like coke.
16	18314	1.74	1.50	**Combination/interaction constructions (with/to)** Fires on prepositions (with, to, by, and) and verbs when two things combine, mix, react, or connect; strongest for explicit combination (mixed/reacted with); silent on ordinary prepositional uses.

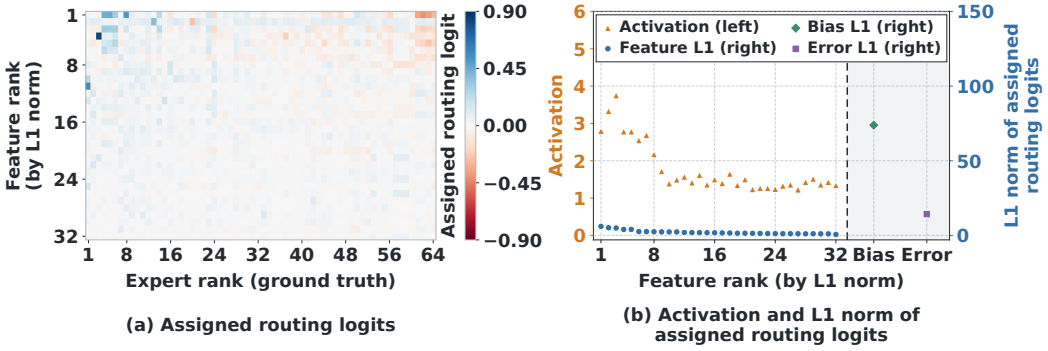


Figure 12: (a) Routing logits assigned by the top-32 features at the second token " to" at **Layer 7** of OLMoE, with experts and features sorted by ground-truth rank and activation, respectively. (b) Feature activations and L1 norms of their assigned routing logits, along with those of the bias and error terms.

F.3 Observation on Layer 15

In this section, we present the detailed information about the top-16 features ranked by their L1 norms at token "to" in Layer 15 of OLMoE (Table 9) and the routing logits information (Figure 13).

Table 9: Top-16 features ranked by L1 norm at token " to" (Layer 15). Abstract: label (bold) and explanation written by Kimi-K3.

Ranked by L1 norm	Feature id	L1 norm	Activation	Abstract
1	17458	14.57	6.13	**Prepositional "to" before noun phrases** Fires on "to" used as a preposition before a noun phrase (destination/recipient), strongest after verbs like add, apply, tie, restore; infinitive "to" before verbs stays silent or weak.
2	16411	10.54	3.06	**Predictable low-surprisal tokens** Fires on tokens highly predictable from context: function words in common collocations, fixed-phrase and subword completions, repeated names, numbers in set slots. Stronger later in sequences; silent near sequence starts.
3	14340	4.55	3.84	**Person-name list connectives** Fires on 'and', '&', commas, quotes adjoining person names in pairs, lists, nicknames; strongest between two conjoined names; weak spillover in name-dense narrative; silent on same tokens otherwise.
4	17125	3.64	1.77	**Tokens preceding person names** Fires on the token immediately before a person's name — comma, "and", "by", titles — strongest in citations, bylines, bibliographies; silent on names themselves and on the same tokens elsewhere.
5	907	3.09	3.50	**Violence and aggression verbs** Fires on verbs of violence (kill, killed, killing, shot, stabbed, assaulted, biting, wounding), strongest on "kill"/"killed" (30-54% of occurrences fire); weak spillover onto adjacent words and other action verbs.
6	23224	2.67	3.09	**Story opening sentences** Fires in bursts on story-opening sentences ('Once upon a time', 'One day', 'Suppose you have'), often after titles or document boundaries; strongest for formulaic fairy-tale openings.
7	7108	2.61	2.13	**Beginning-of-sequence position detector** Fires densely near sequence start (positions 1-15, strongest at 1-3), decaying with distance; mostly silent on the first token and past position ˜64, regardless of token identity.
8	910	2.61	2.64	**Journalistic incident narrative** Fires on function words and past-tense verbs inside third-person news narratives recounting dramatic incidents involving specific people; strongest near quotation marks and after 'story'; silent in exposition, opinion, sports.
9	21663	2.31	2.19	**Sequence-opening word tokens** Fires on words in a sequence's opening ˜15 tokens (77% at positions 1-3), strongest earliest; decays with position, nearly silent after position 64 except just after <—endoftext—>.
10	29554	2.06	1.86	**Translated or non-native English prose** Fires on common function words throughout translated or non-native English passages (web-novel translations, Indian-English religious texts, translated editorials); strongest in dialogue-heavy translated fiction; silent in standard native English.
11	29396	1.99	2.63	**Introducing people by name and role** Fires when text introduces or lists named people: strongest on title/role nouns before proper names, also on commas, "and", "by" linking names in obituaries and news; silent elsewhere.

Table 9: Top-16 features ranked by L1 norm at token " to" (Layer 15). Abstract: label (bold) and explanation written by Kimi-K3. (continued)

Ranked by L1 norm	Feature id	L1 norm	Activation	Abstract
12	29340	1.98	2.45	**Universal quantification context** Fires on prepositions/verbs inside phrases quantifying over a whole set ("for each", "to every", "by all", "not all", "fits everybody"); strongest adjacent to explicit all/every/each; silent on ordinary preposition uses.
13	7932	1.96	1.77	**"give" verb forms and objects** Fires on nearly all forms of "give" (96–99%; given 73%), strongest on the verb itself; weaker on following tokens forming its object phrase; rare spillover to synonyms like "provided".
14	17031	1.96	2.33	**Religious discourse context detector** Fires on common function words and punctuation inside religious (mostly Christian) passages; a preceding "God" reliably triggers it. Strength scales with density of religious content; silent in secular text.
15	4916	1.94	2.45	**Masculine third-person (he/him/his) context** Fires on verbs and prepositions in clauses about a man, most reliably right after "he/He/him" and before "his" ("in his bones"); stronger with denser pronouns; silent for "she/her".
16	26205	1.84	2.11	**Informal narrative function words** Fires mid-clause on common function words in informal first-person narratives, blogs, and dialogue; strongest on "about" and "like"; weak or silent in formal, technical, or structured text.

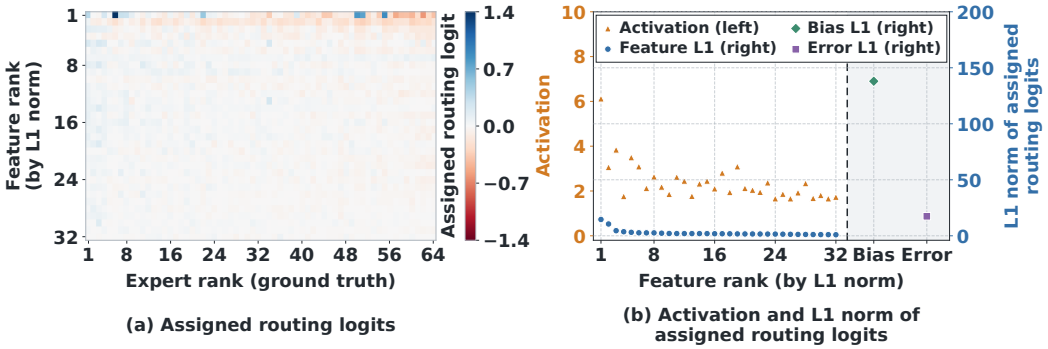


Figure 13: (a) Routing logits assigned by the top-32 features at the second token " to" at **Layer 15** of OLMoE, with experts and features sorted by ground-truth rank and activation, respectively. (b) Feature activations and L1 norms of their assigned routing logits, along with those of the bias and error terms.

G Reconstruction fidelity of SAE

Representation reconstruction fidelity metrics. We adopt Fraction of Variance Unexplained (FVU) of SAE input and Mean Squared Error (MSE) to evaluate the reconstruction fidelity of MoE layer input: FVU $= \frac{\sum_{i=1}^{N}\|\boldsymbol{x}_i-\hat{\boldsymbol{x}}_i\|_2}{\sum_{i=1}^{N}\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}\|_2}$, MSE $= \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i-\hat{\boldsymbol{x}}_i\|_2^2$, where N is the number of test tokens.

We observe that training SAEs on MoE layer input yields a lower reconstruction loss for MoE layer input and routing logits. Layer-level reconstruction fidelity results are illustrated in Figures 14 – 17. Expert-level reconstruction results are illustrated in Figure 18–29. For each model, we present the results for three layers from the bottom, middle, and top of the model.

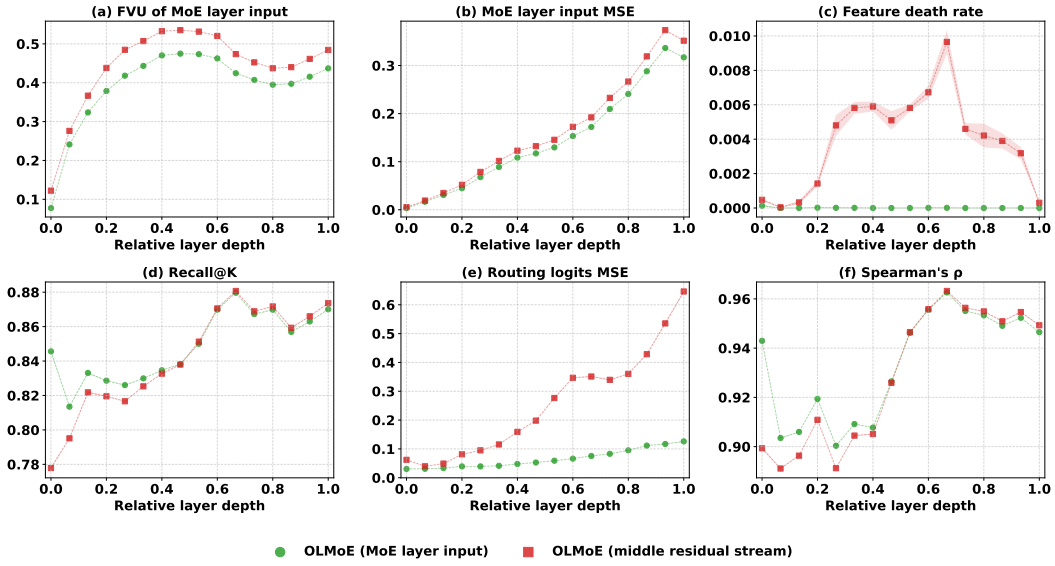


Figure 14: Layer-level reconstruction fidelity results of SAEs trained on OLMoE.

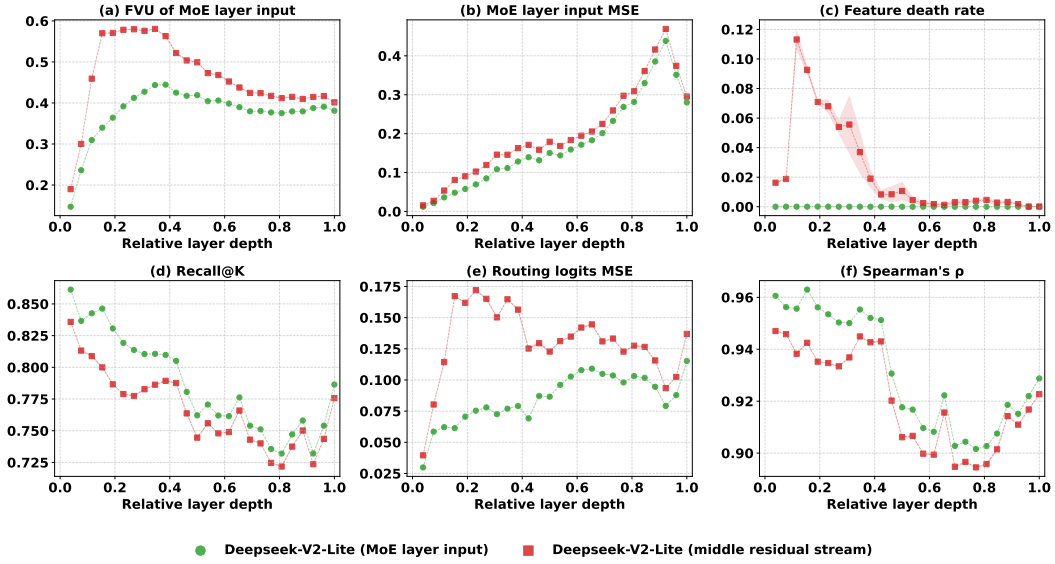


Figure 15: Layer-level reconstruction fidelity results of SAEs trained on Deepseek-V2-Lite.

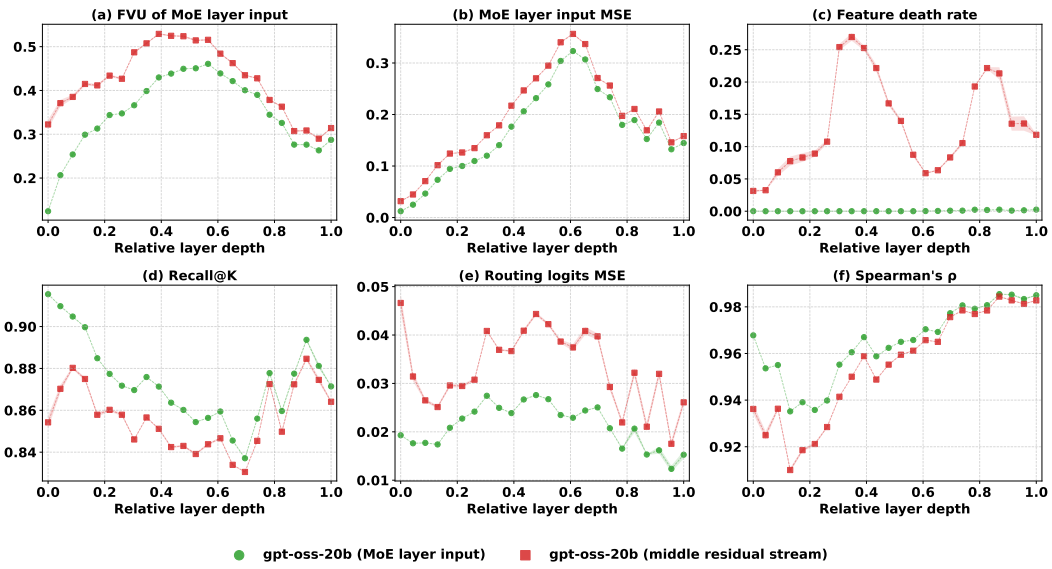


Figure 16: Layer-level reconstruction fidelity results of SAEs trained on gpt-oss-20b.

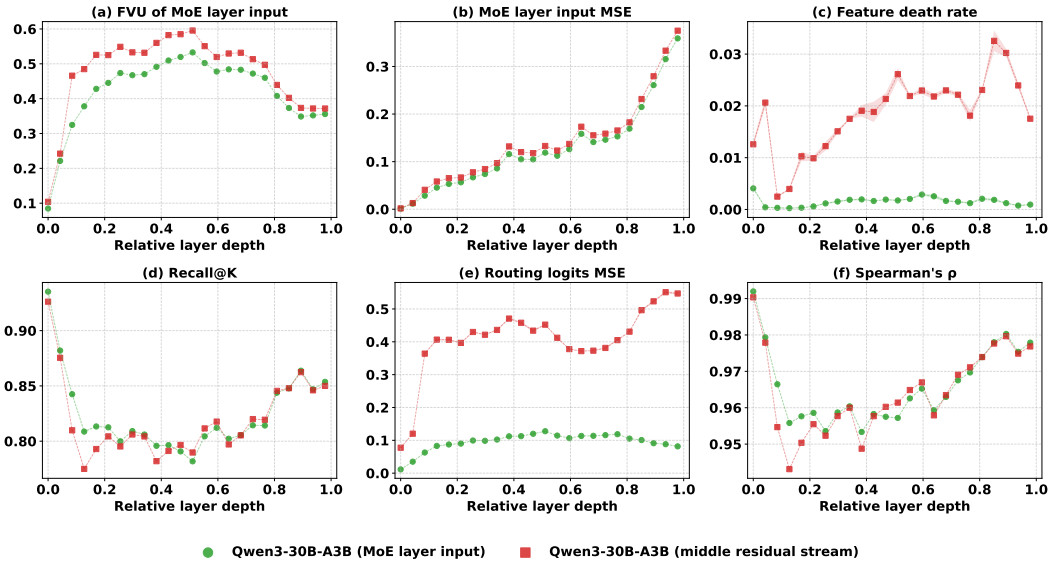


Figure 17: Layer-level reconstruction fidelity results of SAEs trained on Qwen3-30B-A3B.

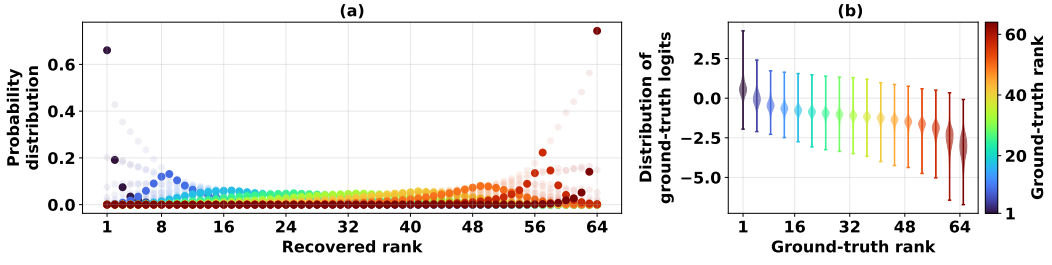


Figure 19: **OLMoE Layer 7**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. (b) Distribution of experts' ground-truth logits.

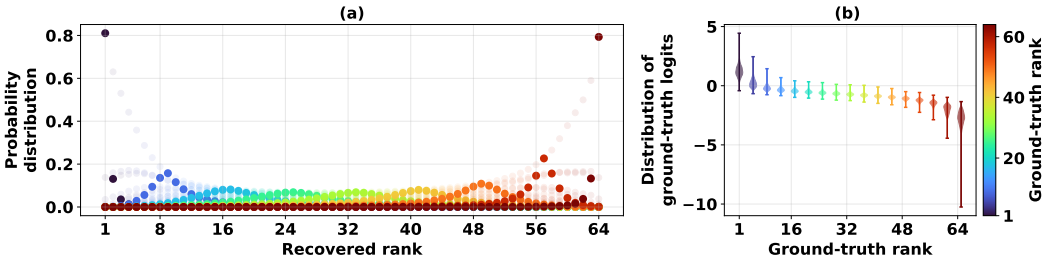


Figure 18: **OLMoE Layer 0:** (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. (b) Distribution of experts' ground-truth logits. Top experts and bottom experts have their rank well recovered, while the middle-ranked experts do not, due to their similar routing logits.

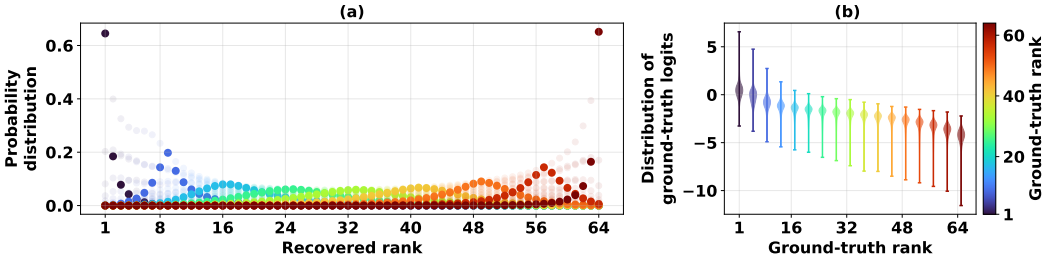


Figure 20: **OLMoE Layer 15**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. (b) Distribution of experts' ground-truth logits.

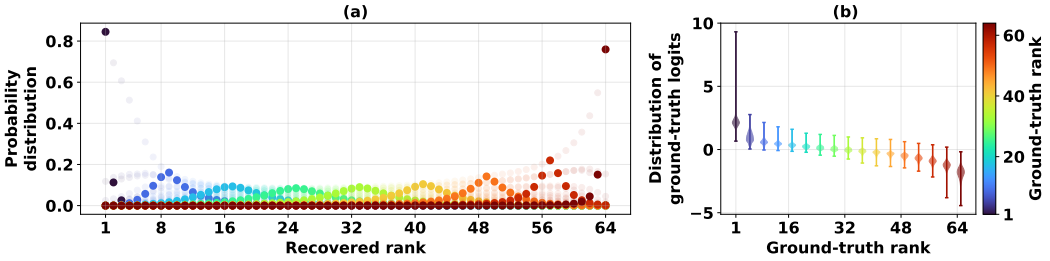


Figure 21: **Deepseek-V2-Lite Layer 1**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. (b) Distribution of experts' ground-truth logits.

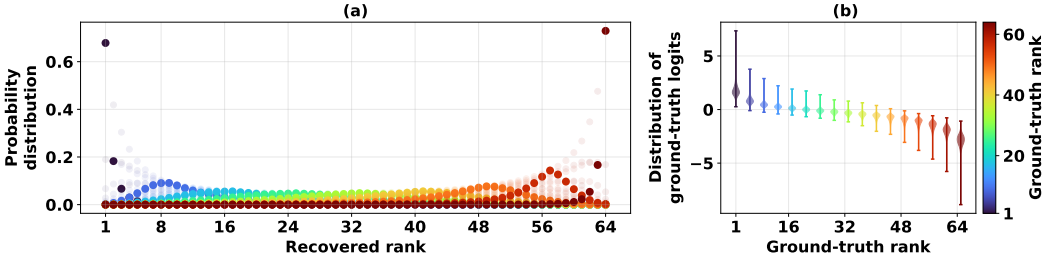


Figure 22: **Deepseek-V2-Lite Layer 13**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. (b) Distribution of experts' ground-truth logits.

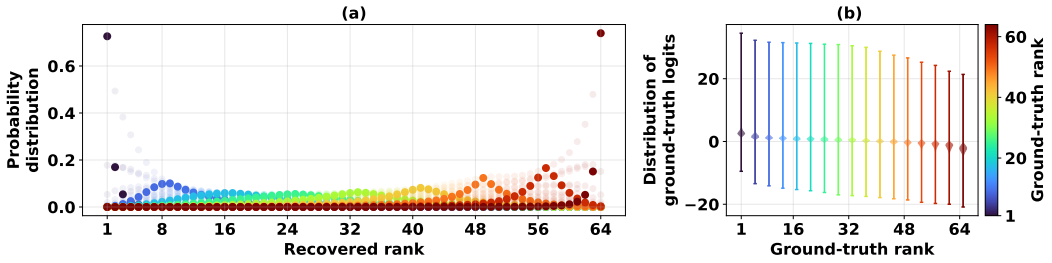


Figure 23: **Deepseek-V2-Lite Layer 26**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. (b) Distribution of experts' ground-truth logits.

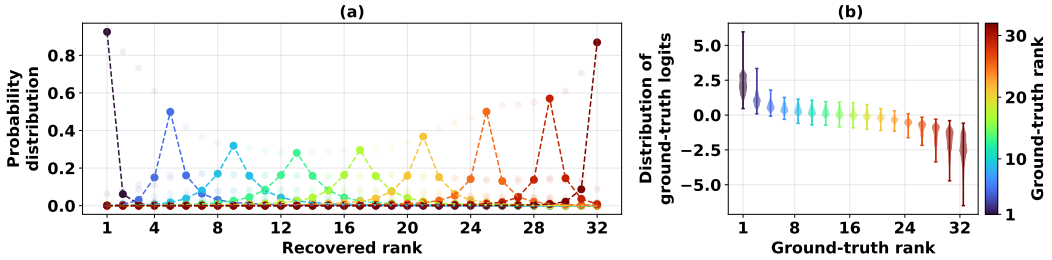


Figure 24: **gpt-oss-20b Layer 0**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. Dashed lines are added to better demonstrate the trend. (b) Distribution of experts' ground-truth logits.

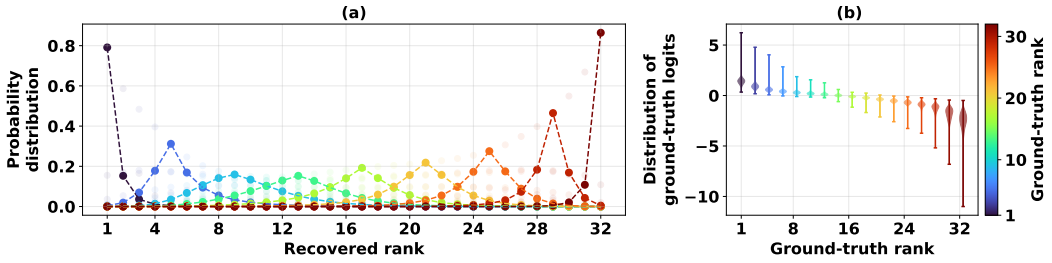


Figure 25: **gpt-oss-20b Layer 11**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. Dashed lines are added to better demonstrate the trend. (b) Distribution of experts' ground-truth logits.

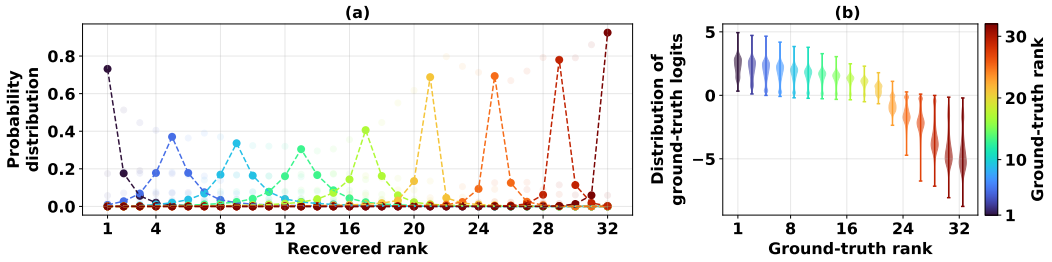


Figure 26: **gpt-oss-20b Layer 23**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. Dashed lines are added to better demonstrate the trend. (b) Distribution of experts' ground-truth logits.

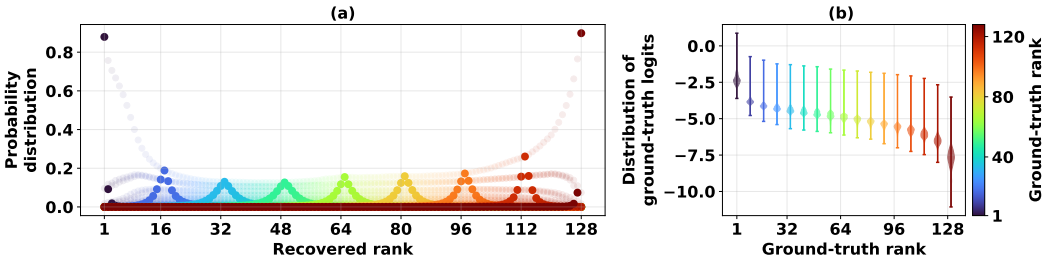


Figure 27: **Qwen3-30B-A3B Layer 0**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. (b) Distribution of experts' ground-truth logits.

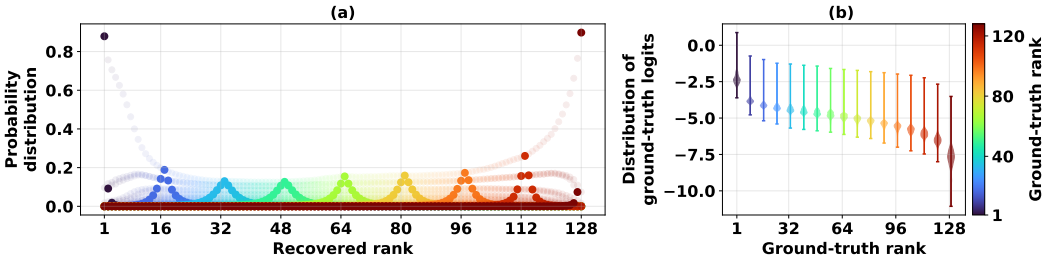


Figure 28: **Qwen3-30B-A3B Layer 24**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. (b) Distribution of experts' ground-truth logits.

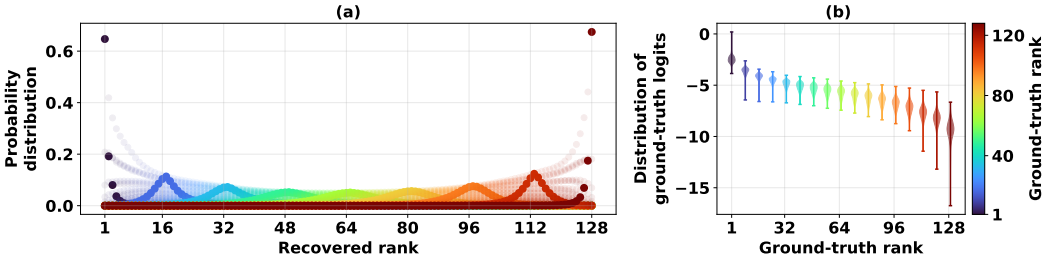


Figure 29: **Qwen3-30B-A3B Layer 46**: (a) Distribution of experts' recovered ranks. The original ranks of experts are indicated by color. (b) Distribution of experts' ground-truth logits.

H STATISTICAL RESULTS OF SAE FEATURES AND BIAS

The statistical results of SAE features and bias are illustrated in Figures 30–32. We observe that SAE bias terms have a more prominent influence than most features in these models, whereas the features of different models exhibit distinct patterns. Bias terms tend to assign negative routing logits (Figure 33).

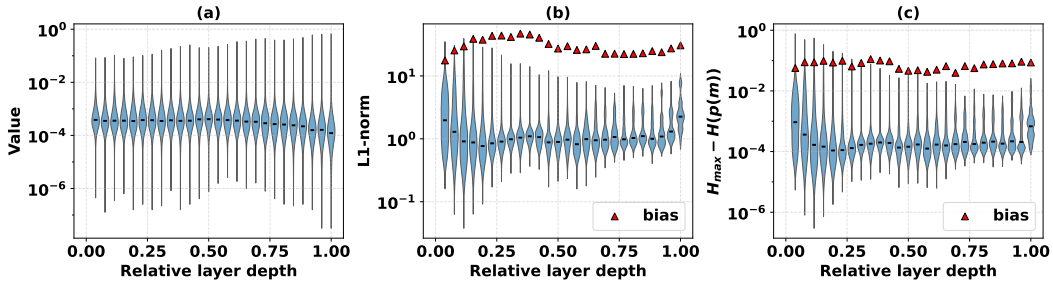


Figure 30: **Deepseek-V2-Lite**: Distribution of (a) feature frequencies, (b) the L1 norm of routing logits assigned by features, (c) the entropy gap ($H_{\max} - H(p(h(\boldsymbol{f})))$).

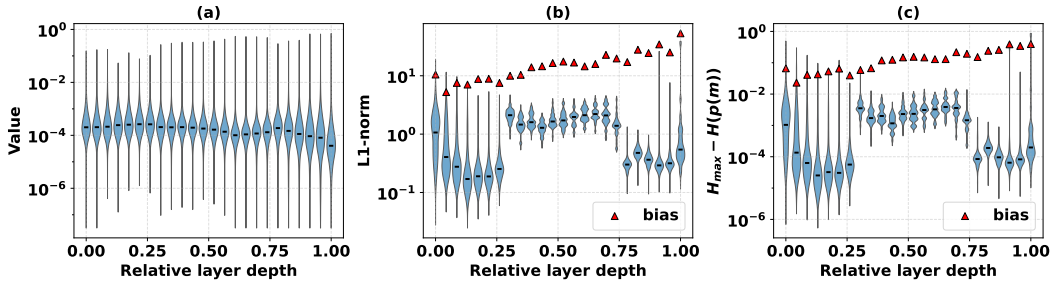


Figure 31: **gpt-oss-20b**: Distribution of (a) feature frequencies, (b) the L1 norm of routing logits assigned by features, (c) the entropy gap ($H_{\max} - H(p(h(\boldsymbol{f})))$).

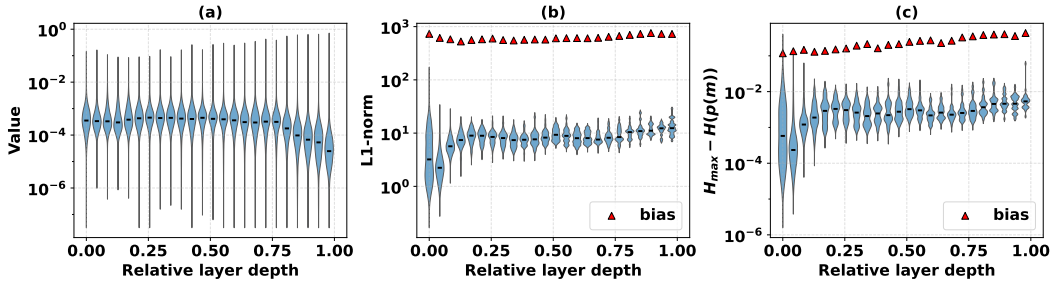


Figure 32: **Qwen3-30B-A3B**: Distribution of (a) feature frequencies, (b) the L1 norm of routing logits assigned by features, (c) the entropy gap ($H_{\max} - H(p(h(\boldsymbol{f})))$).

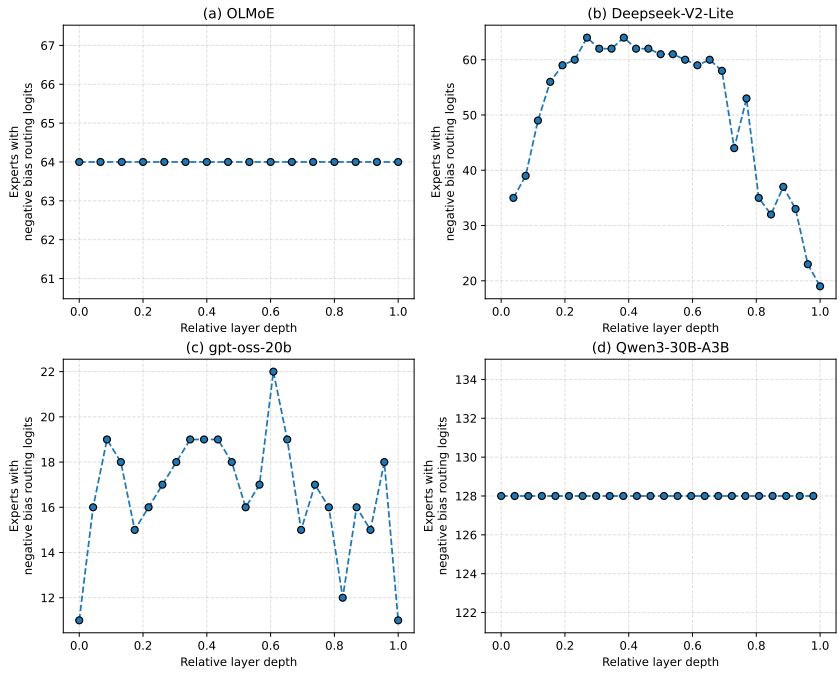


Figure 33: The number of experts receiving negative routing logits from the bias term across layers of (a) OLMoE, (b) Deepseek-V2-Lite, (c) gpt-oss-20b, (d) Qwen-30B-A3B.