# UNDERSTANDING CROSS-LAYER CONTRIBUTIONS TO MIXTURE-OF-EXPERTS ROUTING IN LLMS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Mixture-of-Experts (MoE) has been a prevalent method for scaling up large language models at a reduced computational cost. Despite its effectiveness, the routing mechanism of MoE still lacks a clear understanding from the perspective of cross-layer mechanistic interpretability. We propose a light-weight methodology at which we can break down the routing decision for MoE to contribution of model components, in a recursive fashion. We use our methodology to dissect the routing mechanism by decomposing the input of routers into model components. We study how different model components contribute to the routing in different widely used open models. Our findings on four different production models reveal common patterns such as: a) MoE layer outputs contribute more than attention layer outputs to the routing decisions of latter layers, b) *MoE entanglement* at which MoE firing up in layers consistently correlate with firing up of MoE in latter layers, and c) some components can persistently influence the routing in many following layers. Our study also includes findings on how different models have different patterns when it comes to long range and short range inhibiting/promoting effects that components can have over MoE in latter layers. Our results indicate importance of quantifying the impact of components across different layers on MoE, and highlights the opportunities of using cross-layer contributions for effective model design and model serving.

## 1 INTRODUCTION

Transformer-based Large Language Models (LLMs) (Vaswani et al., 2017) have demonstrated powerful and versatile capabilities in recent years (Achiam et al., 2023; Comanici et al., 2025). To improve their capabilities, researchers have attempted to increase the model size as encouraged by scaling laws (Kaplan et al., 2020) and emergent abilities (Wei et al., 2022) of LLMs. However, this can lead to high computational cost. Mixture-of-Experts (MoE)(Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022) has been applied in LLMs as an effective method to scale up models and alleviate those unfavorable effects, as it can reduce computation by routing the input to a subset of experts instead of using all model parameters to process it.

Although MoE have achieved great success in many advanced LLMs such as GPT (Achiam et al., 2023) and Gemini (Comanici et al., 2025), there is still a lack of understanding of how components in different layers affect how the routing mechanism works, from the perspective of cross-layer mechanistic interpretability. Previous studies mainly investigated the routing mechanism at the expert-level. Muennighoff et al. (2025) and Jiang et al. (2024) performed analyses on the domain or token specialization of experts. Muennighoff et al. (2025) studied the co-activation of experts in the same layer. Lo et al. (2025) inspected the weights of MoE (including routers and experts), gate scores, and expert outputs using similarity and norm metrics. These studies heavily investigated the correlation between experts or between experts and tokens and overlooked the interaction between routers and other components in the model[1].

In this work, we aim to explore and understand the routing mechanism by recursively decomposing the input of routers into components and studying how they contribute to the routing. In summary, our observations and conclusions, from studying four different models, are as follows:

• Our results reveal that MoE routing cannot be understood as a purely local process; rather, it emerges from intricate cross-layer interactions among model components.

[1] In this work, we mainly use the word "component" in a layer to refer to any of the following:, individual attention head, attention layer, individual expert in an MoE, and a full MoE FFN inclusive of all its experts.

• MoE layer output usually has the strongest and persistent influence on the routing rather than other components. However, tokens and attention layer output may have a stronger influence in a small group area such as the bottom and the top layers.

• Routing is shaped not just by local computations, but also by long-range cross-layer entanglements, challenging the assumption that MoE decisions are primarily local.

• A few experts have a significant influence on the routing decisions, consistent with the effect reported in (Su et al., 2025).

• The findings suggest new opportunities for interpretable, efficient, and robust MoE design, with implications for both training-time architecture optimization and inference-time system scheduling.

## 2 Related Work

**Mixture of Experts.** Mixture-of-Experts (MoE) was first introduced in Jacobs et al. (1991) and has been studied for decades. Shazeer et al. (2017) proposed sparsely-gated MoE (SMoEs) as a method to scale up deep learning models with reduced inference overheads. They introduced the top-$k$ routing algorithm, which has become a dominant paradigm nowadays for its simplicity and effectiveness. Researchers have put much effort in optimizing the design of SMoEs (Fedus et al., 2022). Recent studies have also attempted to understand the mechanism of the MoE layer (Chen et al., 2022) and have discovered the routing scores can be applied in model compression (Li et al., 2024) or used as an embedding model (Li & Zhou, 2025). Lo et al. (2025) made an early attempt to analyze MoE-based language models by observing the correlation and norm of some components related to MoE layer, such as experts and gate scores.

**Decomposition of Transformers.** The output of Transformer blocks can be decomposed as a linear combination of outputs of its internal components, facilitating the dissection and understanding of Transformer-based language models (Elhage et al., 2021; Geva et al., 2021; Yu & Ananiadou, 2024; Ferrando & Voita, 2024). These methods decompose the outputs of attention or Feed-Forward Network (FFN) layers into smaller component vectors that can be further studied.The assignment scores assigned to experts can also be decomposed as the sum of sub-scores assigned by the components. Hence, we can study the distribution of the sub-scores to understand how these components influence routing decisions.

## 3 Background

In this section, we present a recursive decomposition of the architecture of the MoE-based Transformer into components that together comprise the assignment score of MoE routing.

**MoE-based Transformer.** An MoE-based decoder-only Transformer consists of $L$ blocks. Each block consists of an attention layer, followed by a Mixture-of-Experts (MoE) layer. Given an input token embedding sequence $\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_u]$ to the model, the first layer input $\boldsymbol{x}^0_{in,i} \in \mathbb{R}^{d_e}$ is the token embedding $\boldsymbol{t}_i$, where $d_e$ is embedding dimension. The block output $\boldsymbol{x}^\ell_{out,i}$ (Token $i$, Block $\ell$) is formulated as follows:

$$\boldsymbol{x}^\ell_{out,i} = \boldsymbol{x}^\ell_{in,i} + \boldsymbol{a}^\ell_{out,i} + \boldsymbol{m}^\ell_{out,i}, \tag{1}$$

where $\boldsymbol{x}^\ell_{in,i}$ is the input of Block $\ell$ ($\boldsymbol{x}^\ell_{in,i} := \boldsymbol{x}^{\ell-1}_{out,i}$ for $\ell > 0$), $\boldsymbol{a}^\ell_{out,i}$ and $\boldsymbol{m}^\ell_{out,i}$ are the outputs of attention and MoE Layer $\ell$, respectively. The final block output $\boldsymbol{x}^{L-1}_{out,i}$ is normalized by layer normalization and then projected onto the vocabulary space to yield the probability distribution of the next token.

**Attention Layer.** The attention layer output $\boldsymbol{a}^\ell_{out,i} \in \mathbb{R}^{d_e}$ can be decomposed into the linear combination of head outputs $\boldsymbol{a}^{\ell,h}_{out,i}$'s, which can be further decomposed at the token level:

$$\boldsymbol{a}^\ell_{out,i} = \sum_{h=1}^{H} \boldsymbol{a}^{\ell,h}_{out,i} = \sum_{h=1}^{H} \sum_{p=1}^{i} \boldsymbol{W}^{\ell,h}_O \boldsymbol{A}^{\ell,h}_{i,p} \boldsymbol{v}^{\ell,h}_p. \tag{2}$$

The element of attention map $\boldsymbol{A}^{\ell,h} \in \mathbb{R}^{u \times u}$ is computed by:

$$\boldsymbol{A}^{\ell,h}_{i,p} = \begin{cases} \underset{1 \le p \le i}{softmax}(\frac{\boldsymbol{q}^{\ell,h}_i \cdot \boldsymbol{k}^{\ell,h}_p}{\sqrt{d_k}}) & p \le i \\ 0 & p > i \end{cases}, \tag{3}$$

where $d_k$ is key dimension, $\boldsymbol{q}_i^{\ell,h}, \boldsymbol{k}_p^{\ell,h} \in \mathbb{R}^{d_k}, \boldsymbol{v}_p^{\ell,h} \in \mathbb{R}^{d_h}$ are query, key, value vectors, respectively. Mathematically, $\boldsymbol{q}_i^{\ell,h} = \boldsymbol{W}_Q^{\ell,h}\boldsymbol{a}_{in,i}^{\ell}$, $\boldsymbol{k}_p^{\ell,h} = \boldsymbol{W}_K^{\ell,h}\boldsymbol{a}_{in,p}^{\ell}$, $\boldsymbol{v}_p^{\ell,h} = \boldsymbol{W}_V^{\ell,h}\boldsymbol{a}_{in,p}^{\ell}$, where $\boldsymbol{W}_Q^{\ell,h}, \boldsymbol{W}_K^{\ell,h} \in \mathbb{R}^{d_h \times d_k}, \boldsymbol{W}_V^{\ell,h} \in \mathbb{R}^{d_h \times d_e}, \boldsymbol{W}_O^{\ell,h} \in \mathbb{R}^{d_e \times d_h}$ are query, key, value and output weight matrices of Head $h$ in attention Layer $\ell$, $d_h$ is head dimension, $\boldsymbol{a}_{in,i}^{\ell}$ is attention layer input:

$$\boldsymbol{a}_{in,i}^{\ell} = \mathrm{LN}_i^{\ell}(\boldsymbol{x}_{in,i}^{\ell}), \tag{4}$$

where $\mathrm{LN}_i^{\ell}(\cdot)$ is layer normalization. RMS layer normalization is applied in the tested models in this work, hence $\mathrm{LN}_i^{\ell}(\boldsymbol{z}) := \frac{\boldsymbol{z}\cdot\gamma^{\ell}}{\mathrm{RMS}(\boldsymbol{z})}$, where $\boldsymbol{z} \in \mathbb{R}^{d_e}$, $\mathrm{RMS}(\cdot)$ is the root mean square function, and $\gamma^{\ell} \in \mathbb{R}^{d_e}$ is a learnable parameter.

**MoEs Layer.** The MoEs layer consists of a router and $N$ experts, i.e., $N$ parallel sub-FFN layers. The router assigns a **score** to each expert and selects the top-$k$ experts to process the MoE layer input $\boldsymbol{m}_{in,i}^{\ell}$, where $k$ is a hyperparameter. Mathematically,

$$\boldsymbol{m}_{in,i}^{\ell} = \mathrm{LN}_i^{\ell}(\boldsymbol{x}_{in,i}^{\ell} + \boldsymbol{a}_{out,i}^{\ell}). \tag{5}$$

Typically, the scores of all $N$ experts are computed by $\boldsymbol{W}_G^{\ell}\boldsymbol{m}_{in,i}^{\ell}$, where $\boldsymbol{W}_G^{\ell} \in \mathbb{R}^{N \times d_e}$ is the routing weight matrix.[2] Each row vector $\boldsymbol{g} \in \mathbb{R}^{d_e}$ of the routing weight matrix $\boldsymbol{W}_G$ corresponds to one expert. We call these row vectors "**routing weight vectors**". The assignment score $S$ of Expert $(\ell, n)$, i.e., Expert $n$ in MoE Layer $\ell$ is essentially the dot product of its corresponding routing weight vector $\boldsymbol{g}^{\ell,n}$ and the MoE layer input $\boldsymbol{m}_{in,i}^{\ell}$:

$$S(\boldsymbol{g}^{\ell,n}, \boldsymbol{m}_{in,i}^{\ell}) = \boldsymbol{g}^{\ell,n} \cdot \boldsymbol{m}_{in,i}^{\ell}. \tag{6}$$

The assignment scores are passed through a Softmax function to yield expert weights. The MoE layer output $\boldsymbol{m}_{out,i}^{\ell}$ is the weighted sum of outputs of selected experts:

$$\boldsymbol{m}_{out,i}^{\ell} = \sum_{j \in J} r^{\ell,j}(\boldsymbol{m}_{in,i}^{\ell})\boldsymbol{e}_{out,i}^{\ell,j}, \tag{7}$$

where $r^{\ell,j}(\cdot) = \underset{j \in J}{softmax}(S(\boldsymbol{g}^{\ell,j}, \cdot))$ is the expert weight, $\boldsymbol{e}_{out,i}^{\ell,j}$ is the expert output, and $J$ is the set of indices of selected experts.

## 4 METHODOLOGY

For each given input token, The MoE router assigns scores to experts and selects the top-$k$ experts to process the input. Assignment scores have two determinants: the routing weight vectors and the MoE layer inputs (Equation 6). Intuitively, assignment scores are decomposable since the MoE layer input can be decomposed into components (Equations 1 and 5). To understand the routing mechanism, we can study the patterns of scores "assigned" by the components.[3] In this section, we delineate the decomposition method in Section 4.1 and discuss some basics of scoring patterns in Section 4.2.

### 4.1 DECOMPOSITION OF EXPERTS ASSIGNMENT SCORES

In this section we discuss how the experts assignment score can be decomposed into components of different granularities, from entire layers to individual neurons. We can recursively apply Equations 1 and 5 to decompose the score assigned to the Expert $(\ell, n)$:

$$\begin{aligned} S(\boldsymbol{g}^{\ell,n}, \boldsymbol{m}_{in,i}^{\ell}) &= \boldsymbol{g}^{\ell,n} \cdot \boldsymbol{m}_{in,i}^{\ell} = \boldsymbol{g}^{\ell,n} \cdot \mathrm{LN}_i^{\ell}(\boldsymbol{x}_{in,i}^{\ell} + \boldsymbol{a}_{out,i}^{\ell}) \\ &= \boldsymbol{g}^{\ell,n} \cdot \mathrm{LN}_i^{\ell}(\boldsymbol{x}_{in,i}^{0} + \sum_{c=1}^{\ell} \boldsymbol{a}_{out,i}^{c} + \sum_{c=1}^{\ell-1} \boldsymbol{m}_{out,i}^{c}) \\ &= \boldsymbol{g}^{\ell,n} \cdot \overline{\mathrm{LN}}_i^{\ell}(\boldsymbol{x}_{in,i}^{0}) + \boldsymbol{g}^{\ell,n} \cdot \sum_{c=1}^{\ell} \overline{\mathrm{LN}}_i^{\ell}(\boldsymbol{a}_{out,i}^{c}) + \boldsymbol{g}^{\ell,n} \cdot \sum_{c=1}^{\ell-1} \overline{\mathrm{LN}}_i^{\ell}(\boldsymbol{m}_{out,i}^{c}) \end{aligned} \tag{8}$$

[2] Some MoE models use more sophisticated mechanisms such as MLP layers to obtain scores. Our method is also applicable to those models, but for simplicity, we discuss only the most widely used implementation of the routing mechanism.

[3] When we say the score assigned by a component to an expert, we refer to the portion of the expert's score contributed by the component.

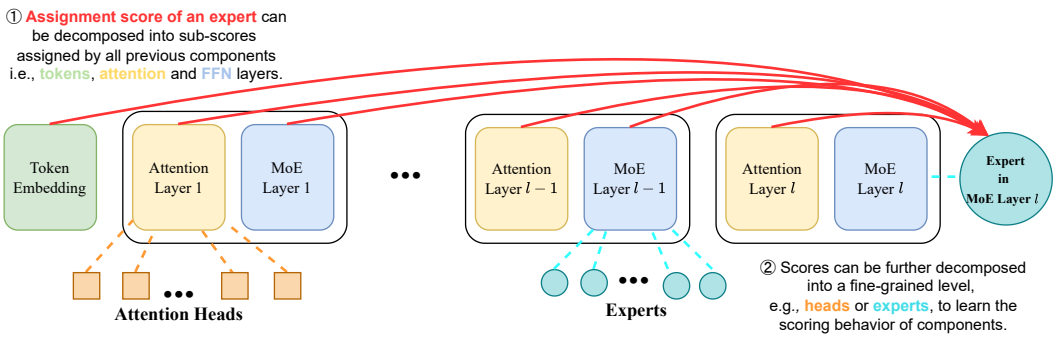



Figure 1: Overview of the decomposition of experts assignment scores.

where $\overline{\mathrm{LN}}_i^\ell(\boldsymbol{z}) = \frac{\boldsymbol{z}\cdot\gamma^\ell}{\mathrm{RMS}(\boldsymbol{x}_{in,i}^\ell + \boldsymbol{a}_{out,i}^\ell)}$. Equation 8 indicates that the assignment score can be decomposed into multiple sub-scores, i.e., the scores assigned by the token embedding ($S(\boldsymbol{g}^{\ell,n}, \boldsymbol{x}_{in,i}^0) = \boldsymbol{g}^{\ell,n} \cdot \overline{\mathrm{LN}}_i^\ell(\boldsymbol{x}_{in,i}^0)$), attention layer outputs, and MoE layer outputs. Both attention and MoE layer outputs can be further decomposed. According to Equation 2, the score $S(\boldsymbol{g}^{\ell,n}, \boldsymbol{a}_{out,i}^c)$ assigned by attention layer output $\boldsymbol{a}_i^c$ can be further decomposed as follows:

$$S(\boldsymbol{g}^{\ell,n}, \boldsymbol{a}_{out,i}^c) = \boldsymbol{g}^{\ell,n} \cdot \overline{\mathrm{LN}}_i^\ell(\boldsymbol{a}_{out,i}^c) = \boldsymbol{g}^{\ell,n} \cdot \sum_{h=1}^{H} \overline{\mathrm{LN}}_i^\ell(\boldsymbol{a}_{out,i}^{c,h}) = \boldsymbol{g}^{\ell,n} \cdot \sum_{h=1}^{H}\sum_{p=1}^{i} \overline{\mathrm{LN}}_i^\ell(\boldsymbol{W}_O^{c,h}\boldsymbol{A}_{i,p}^{c,h}\boldsymbol{v}_p^{c,h}) \quad (9)$$

From this perspective, we can regard the score assigned by an attention layer output to an expert as the sum of the scores assigned by tuples like (head, query, key) (Equations 3 and 9). Similarly, we can decompose the scores $S(\boldsymbol{g}^{\ell,n}, \boldsymbol{m}_{out,i}^c)$ assigned by the MoE layer output $\boldsymbol{m}_i^c$ into the scores assigned by the selected experts (Equation 7):

$$S(\boldsymbol{g}^{\ell,n}, \boldsymbol{m}_{out,i}^c) = \boldsymbol{g}^{\ell,n} \cdot \overline{\mathrm{LN}}_i^\ell(\boldsymbol{m}_{in,i}^c) = \boldsymbol{g}^{\ell,n} \cdot \sum_{j\in J} \overline{\mathrm{LN}}_i^\ell(r^{c,j}(\boldsymbol{m}_{in,i}^c)\boldsymbol{e}_{out,i}^{c,j}) \quad (10)$$

It is possible to further decompose the expert (i.e., sub-FFN) outputs at the neuron level (Geva et al., 2021; Dai et al., 2022; Geva et al., 2022):

$$\boldsymbol{e}_{out,i}^{\ell,j} = \sum_{z=1}^{d_e} \boldsymbol{W}_{d_{(:,z)}}^{\ell,j} \cdot [\sigma(\boldsymbol{W}_{g_{(z,:)}}^{\ell,j} \cdot \boldsymbol{m}_{in,i}^\ell) \cdot (\boldsymbol{W}_{u_{(z,:)}}^{\ell,j} \cdot \boldsymbol{m}_{in,i}^\ell)], \quad (11)$$

where all of the matrices correspond to Expert $(\ell, j)$, $\boldsymbol{W}_{d_{(:,z)}}^{\ell,j}$ is the $z$-th column of down-projection matrix, $\boldsymbol{W}_{g_{(z,:)}}^{\ell,j}$ and $\boldsymbol{W}_{u_{(z,:)}}^{\ell,j}$ are the $z$-th row of gating matrix and up-projection matrix, respectively. $\sigma(\cdot)$ is an activation function. We leave it for further study, considering its potential complexity.

### 4.2 Basics of Scoring

In Section 4.1, we proposed a method for determining the scores of experts assigned by the components. In this section, we discuss what can be learned from the contribution of different components into the assignment scoring of experts. The proofs of our propositions is in Appendix A.

**Proposition 1: Variance of scores assigned by a component measures its influence on the routing decision.** Suppose a component assigns a constant score to all experts, which means its scoring variance is 0, then it does not influence the routing decisions because if the scores it assigns are dropped, the differences between the scores of experts are unchanged. Intuitively, we posit that a component with higher scoring variance has a stronger influence on the routing decisions and thus is more important. Based on this postulate, we can further infer that the length (i.e., L2-norm) of the component controls the upper and lower bounds of the scores assigned by it, which indicates that strong influences are caused by components with a large norm.

**Proposition 2: Positive scores promote experts, negative scores inhibit experts. The degree is measured by the score magnitude.** Since the scoring operation is a dot product of a gating weight

vector and a component vector (Equations 6, 8, 9, and 10), the angle between them controls the sign of the score. When the angle is acute (obtuse), the score assigned by the component to the corresponding expert is positive (negative), indicating the component promotes (inhibits) the expert to be selected. When the two vectors are orthogonal, the score is zero, indicating the component has no opinion on the corresponding expert. The magnitude of the score is controlled by the length of the two vectors and the angle between them. Hence, if we fix a gating weight vector, then a component with a smaller angle and larger length will assign a higher score to the corresponding expert.

# 5 Score Distribution

In this section, we show empirical results on the assignment score distribution of experts assigned by tokens, attention layer outputs, and MoE layer outputs, respectively.

## 5.1 Experimental Setup

**Models and dataset.** We adopt four MoE-based language models, scaling from OLMoE (Muennighoff et al., 2025), DeepSeek-V2-Lite (Liu et al., 2024), Qwen3-30B-A3B (Yang et al., 2025), to Mixtral-8x7B (Jiang et al., 2024). Their basic information is summarized in Appendix B. We randomly select samples with at least 32 tokens from C4 dataset (Raffel et al., 2020) and truncate each to the first 32 tokens to simplify the experiments. We use 1000 and 5000 samples for the experiments in Section 5.2 and 5.3, respectively. In this section, we report the results from OLMoE in the main text. The results of the other three models are reported in Appendix D.

**Metrics.** We use the variance of scores assigned by a component $\boldsymbol{c}$ to the experts in an MoE layer to measure the contribution of the component to the routing decisions of that layer: The variance of scores $s_1, s_2, ..., s_N$ is $\frac{1}{N}\sum_{n=1}^{N}(s_n - \mu)^2$, where $\mu = \frac{1}{N}\sum_{n=1}^{N} s_n$. To find the scoring tendency (promotion or inhibition) of a component to a set of specified experts, we use the average positive score (APS) and average negative score (ANS) assigned to those experts by the component:

$$APS = \frac{1}{N}\sum_{n=1}^{N} S(\boldsymbol{g}_j, \boldsymbol{c})\mathbb{1}_{S(\boldsymbol{g}_j,\boldsymbol{c})>0}, \quad ANS = \frac{1}{N}\sum_{n=1}^{N} S(\boldsymbol{g}_j, \boldsymbol{c})\mathbb{1}_{S(\boldsymbol{g}_j,\boldsymbol{c})<0}, \tag{12}$$

where $N$ is the number of experts and $\mathbb{1}$ is the indicator function. We separate the positive and negative scores to avoid cancellation.

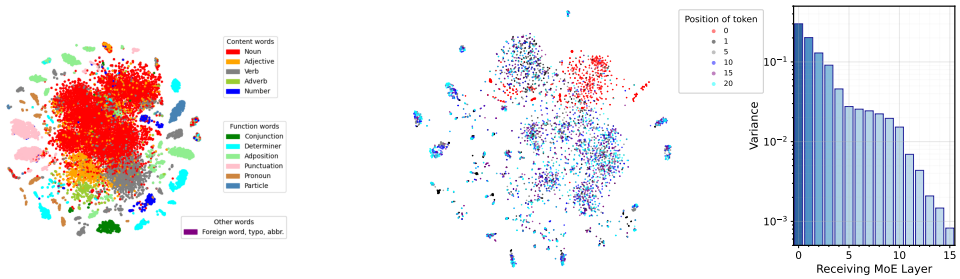
## 5.2 Tokens Scoring Distribution

**Token scoring is influenced by its part-of-speech (POS).** We pack the scores of all experts assigned by each token into a vector and apply t-SNE(Maaten & Hinton, 2008) to visualize them. [4] From Figure 2a, we find that most scores assigned by tokens are clustered according to the POS tags of tokens, which implies that the POS influences token scoring. Furthermore, most function words are clustered into isolated lumps. In contrast, content words are entangled, which is possibly because function words may have multiple POS, e.g., "drink" can be a noun or a verb, whereas the semantics of function words is more stable.

**The lead token is special.** In Figure 2b, the lead tokens of each prompt appear clustered, whereas other tokens, in comparison, are not, indicating that the lead position has a noticeable influence on scoring. We find the scoring distribution of attention and MoE layers at the lead position also has a typical pattern. We speculate that the attention layers capture the position information, lead to the norm of MoE layer output at the lead position to being typically larger than that at other tokens, which influences the RMS term of layer normalization and finally influences the scoring distribution. In light of this, we report the results excluding the lead position in the main text henceforth and report the results related to the lead position in Appendix E.

In Figure 2c, the average variance of scores assigned by tokens decreases rapidly as the layer goes deeper, showing that the influence of tokens on routing decisions decreases rapidly. Furthermore, the variance of scores in the first two MoE layers assigned by tokens is typically high, indicating the

[4] If a word is split into multiple tokens, each token inherits the POS of the word.




**(a) Token Part-of-Speech Distribution** **(b) Token Position Distribution** **(c) Token Routing Variance**

Figure 2: (a) t-SNE of scores assigned by token embeddings, colored by POS. (b) t-SNE of scores assigned by token embeddings, colored by position. (c) Average variances of scores assigned by tokens to MoE layers.

tokens have a strong influence on the routing decisions in them, which is in line with intuition. The magnitude of ANS and APS also decreases as the layer goes deeper (see Appendix D).

### 5.3 Result on Attention/MoE Layer Outputs

In this subsection, we analyze the variance and the average positive/negative scores assigned by the attention and the MoE layer output to study their influence on the routing decisions.

**Average variance.** As shown in Figure 3a, the highest variance of attention layer scoring occurs at A0 $\rightarrow$ M0, i.e., the assignment from sending attention Layer 0 to receiving MoE Layer 0. The variance diminishes gradually as the receiving layer appears deeper. Comparatively, the first two attention layers (A0 and A1) exhibit a higher variance than the intermediate sending attention layers. We compare the four models we tested and find that the early sending attention layers generally have a high average variance to their neighbor receiving layers. Surprisingly, a few MoE layers have a pronounced *entangled* influence on the routing decisions on following MoE layers (Figure 3d): Sending layers M1 and M4 have a notably higher variance compared with others, indicating they have a higher importance. It is also unique that their variance does not decrease monotonically. We will show that two experts with typically high variance in these two layers cause the "stripes" (Section 7). We also find such stripes in DeepSeek, Qwen and Mixtral, but some of them may just have an apparently higher variance compared with neighbor sending layers, instead of having an increasing average variance on the receiving layers. The stripes also occur in sending attention layers in Qwen. Finally, the variance of sending MoE layers is usually comparatively higher than that of sending attention layers.

These findings suggest two key implications. First, load balancing of expert parallelism can be improved by prefetching and preloading experts in high-variance layers, leveraging cross-layer entanglement to reduce contention. Second, post-NAS approximation strategies Gu et al. (2025) can selectively compress low-variance attention layers while preserving influential ones, enabling more efficient yet accurate architectures.

**Scoring pattern of layers.** We conducted an experiment to analyze the scores assigned to all the experts, rather than the selected ones. By observing ANS and APS, we can learn which components promote or inhibit the experts in a specific layer. We first look at the scoring pattern of sending attention layers. Comparing Figure 3b and c, we note that the positive and negative scores occur in different areas: positive scores occur at the early sending layers i.e., left side of x-axis (A0 and A1), whereas negative scores occur at A0, and most areas on or near the diagonal, and the magnitude increases as the sending layer goes deeper. In other words, promotion effects—components strongly enhancing expert layers—tend to be local, whereas inhibition effects are more global, with components exerting stronger inhibition as the expert layer appears deeper in the model. Comparing the tested models, we find that the negative scores tend to occur at the bottom or top sending attention layers. The magnitude of APS is generally smaller than that of ANS, although Mixtral seems to be an exception.

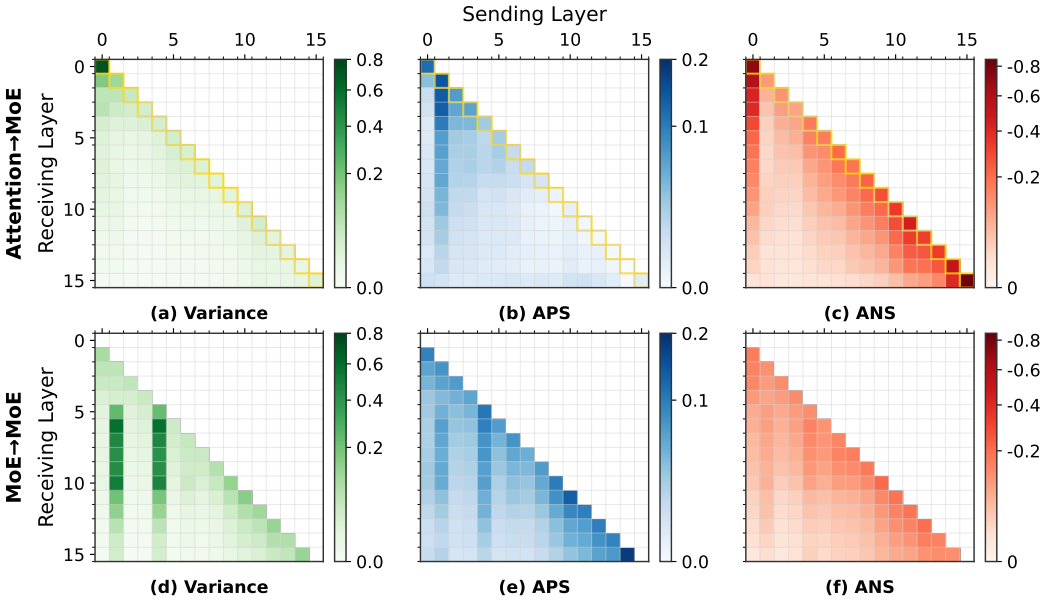



Figure 3: Scores assigned by attention Layer x to MoE layer y: (a) variance, (b) Average positive scores (APS), (c) Average negative scores (ANS); Scores assigned by MoE Layers x to MoE Layer y: (d) variance, (e) APS, (f) ANS.

The scoring pattern of sending MoE layers is shown in Figure 3e and f. Sending M1 and M4 have relatively high positive and negative scores, respectively, which conforms to the variance matrix (Figure 3d). The positive scores mainly appear at sending M1, M4, deeper sending layers (M8 $\sim$ M14) and the diagonal, whereas the negative scores appear at sending M1, M4, and the area near the diagonal. We also find the APS and ANS matrices in Qwen have a multi-stripe pattern, i.e., more prominent entanglement effect, whereas other tested models just have a few or no stripes.

## 6 Scoring of Attention Heads

In this section, we investigate the scores assigned by attention heads to the experts in different layers. We continue to use C4 dataset to observe the general behavior of the scoring. Additionally, we use Indirect Object Identification (IOI) (Wang et al., 2023) task to observe the connection between attention maps and the scoring patterns.

### 6.1 Experimental Setup

**General test.** We conduct the general test on OLMoE and DeepSeek (Appendix G) since they have fewer heads (256 and 432), facilitating our analysis. We follow the setting in Section 5.1 and use 5000 samples for experiments.

**IOI task** is to predict the next token of a prompt like "*When Mary and John went to the store, John gave a drink to* ___", where the name that exists in the first clause but does not appear in the second clause is expected to be the prediction result, i.e., "Mary". We adopt this task to observe if function heads have a noticeable influence on the routing decisions. We use path patching (Wang et al., 2023) to find a portion of the function heads in OLMoE.[5] We can regard the scores assigned by a head as a "score map" once we determine a metric for tuples like (head, query, key). We use the variance of scores assigned by the tuple (head, query, key) to all experts in a MoE layer as the metric. We refer to it as the *score variance map* and compare it with the results from path patching, attention map, and variance score map.

[5]For simplicity, we do not aim to discover all the function heads activated in the IOI task in this work.

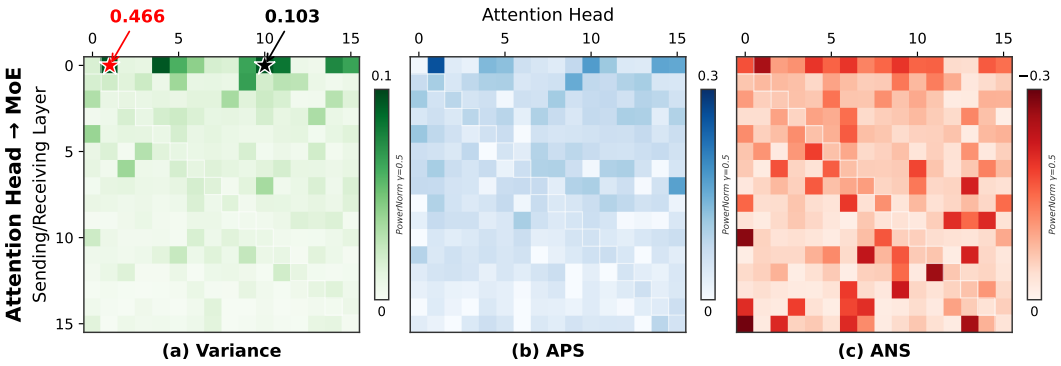



Figure 4: (a) Variance of scores assigned by heads to experts in the same block. (b) Average positive scores (APS) of the heads. (c) Average negative scores (ANS) of the heads. All panels use PowerNorm ($\gamma$=0.5) color scaling. Outliers in (a) marked with stars: 0.466 (red) and 0.103 (black).

### 6.2 Results on General Test

Since attention layer outputs usually exert the strongest influence on the routing decisions in the same block in OLMoE (Figure 3a), we examine the variance and mean of scores assigned by heads to the experts in the same block to facilitate the observation. Henceforth, we use "AxHy" to denote Head y at attention Layer x. Heads with high variance mainly occur at the early layers (typically A0 and A1), conforming to the results in 3a. By comparing Figures 3b and c, we can find that many attention heads tend to assign negative scores to the experts, and a small proportion of heads tend to generally assign positive scores, such as A7H15 and A9H6. The scores assigned by some heads are apparently polarized, such as A0H1. Finally, we observe that if a head has a relatively higher scoring variance, the absolute magnitude of scores assigned by the head is usually larger.

### 6.3 Results on IOI task

We employ IOI task as an example to study the scoring pattern of function heads. The visualized results are shown in Appendix F. Our main findings on this task are as follows:

**Function heads have a noticeable influence on the routing decisions.** We find that function heads, i.e., attention heads that contribute to finish the IOI task, usually have a higher scoring variance, compared with the heads do not show any functions in the task, indicating that the function heads tend to have a stronger influence on the routing decisions than the other heads.

**Scores variance of attention heads correlate with attention maps.** We compare the "attention map" $\boldsymbol{A}_{i,p}^{x,y}$ (where key token $\boldsymbol{t}_i$ and query token $\boldsymbol{t}_p$ are fixed, Layer $x$ and Head $y$ are variables), with the "score variance map", that is, variance of scores assigned by tuples (head=AxHy, key=$\boldsymbol{t}_i$, query=$\boldsymbol{t}_p$), in other words, terms $\overline{\mathrm{LN}}_i^c(\boldsymbol{W}_O^{x,y}\boldsymbol{A}_{i,p}^{x,y}\boldsymbol{v}_p^{x,y})$ for AxHy (Equation 9), to the experts in a specific layer $c(c \geq x)$. We find that they have a simillar pattern, which is in line with our intuition since attention map can manipulate the attention output and thus influence the scoring.

## 7 Scoring of Experts

In this section, we use variance to measure the influence that experts have on routing decisions in later layers. In Section 5.3, we have seen that the variance of scores assigned by M1 and M4 in OLMoE has an unusual phenomenon: the variance does not decrease monotonically. We find that two experts contribute to the phenomenon. Furthermore, a small subset of experts has a strong influence on the routing decisions in OLMoE and Qwen3. Intriguingly, among these experts, some maintain their influence on the routing decisions till the last layers rather than decreasing significantly.

**OLMoE (Figure 5a).** M1E9 (Experts 9 at MoE Layer 1) and M4E14 exhibit an unusual phenomenon where their influence peaks around MoE Layer 6, followed by a secondary peak near MoE Layer 10, then consistently decreases (with M1E9 dropping below M4E14 by the end). In contrast, M2E30 shows a steadily increasing influence that reaches its peak in the final layers.

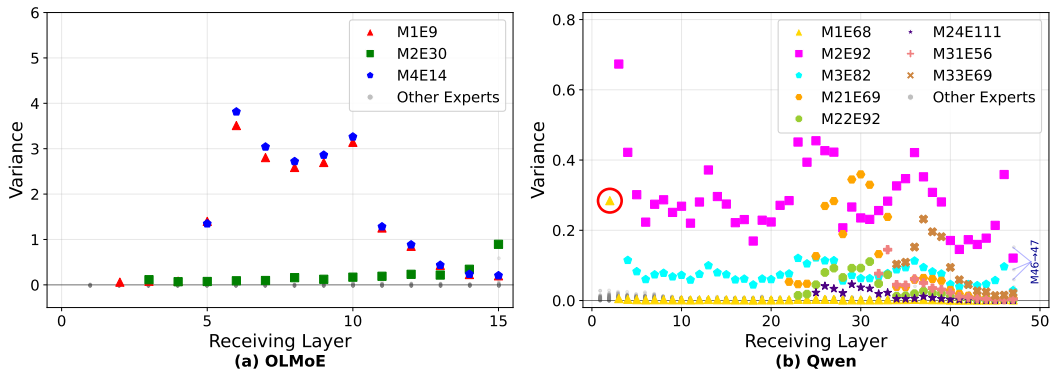



Figure 5: The variance of scores assigned by experts to the following layers.

**Qwen (Figure 5b).** M1E68 (circled in red) exhibits a strong but localized influence primarily on the immediate next layer. M2E92 consistently maintains the highest impact across all subsequent layers. M3E82 also shows persistent influence throughout, though with smaller magnitude. Other highlighted experts (M21E69, M22E92, M24E111, M31E56, M33E69) from middle and late layers generally follow the typical pattern of gradually building to peak influence before fading out.

We find some of these experts are the "Super Experts" found in (Su et al., 2025), which have an output of extreme magnitude. We summarize the rank of variance of scores assigned by the Super Experts to the experts in the next MoE layers in Appendix H. We find that not all the Super Experts have typically large scoring variance, such as M2E54 in DeepSeek. Although M1E68 in Qwen has the rank 1 variance among the experts in Layer 1, but its scoring variance decreases drastically (Figure 5b). The Super Experts in DeepSeek do not have a top rank nor a persistent high scoring variance. The scoring variance of Super Expert in Mixtral (M1H3) has a unique distribution: it has a relatively high variance in the layer 28 $\sim$ 31 (Appendix H).

## 8 Summary of Findings and Conclusion

In this work, we proposed a recursive decomposition framework to quantify how different components contribute to routing decisions in Mixture-of-Experts (MoE) language models. By breaking down expert assignment scores into contributions from tokens, attention layers, MoE outputs, and attention heads, we provided the first cross-layer perspective on routing interpretability. Our analysis across four production MoE models (OLMoE, DeepSeek-V2-Lite, Qwen3-30B-A3B, and Mixtral-8x7B) revealed several consistent patterns. **First**, MoE outputs exert the strongest and most persistent influence on downstream routing, while attention layers and tokens have more localized effects, especially in the bottom and top layers. **Second**, routing decisions are shaped by both promotion and inhibition: positive contributions typically act locally, while negative contributions inhibit experts across longer ranges. **Third**, we identified cross-layer entanglement phenomena, where certain MoE layers (e.g., M1, M4 in OLMoE) disproportionately affect routing in much deeper layers, forming "stripes" of influence. At a finer granularity, we found that a small set of experts and attention heads dominate routing behavior, with some maintaining their impact throughout the network. Notably, not all previously identified "Super Experts" exhibit strong influence under our variance-based analysis. These findings demonstrate that MoE routing is not solely a local mechanism, but instead emerges from a complex interplay of components across layers. Our work highlights the importance of quantifying cross-layer contributions, offering new opportunities to improve expert parallelism, guide compression and architecture search, and design more interpretable and efficient MoE-based models.

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding the mixture-of-experts layer in deep learning. *Advances in neural information processing systems*, 35: 23049–23062, 2022.

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.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

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.

Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17432–17445, 2024.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45, 2022.

Yuxian Gu, Qinghao Hu, Shang Yang, Haocheng Xi, Junyu Chen, Song Han, and Han Cai. Jet-nemotron: Efficient language model with post neural architecture search, 2025. URL `https://arxiv.org/abs/2508.15884`.

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.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient smoe with hints from its routing policy. In *The Twelfth International Conference on Learning Representations*, 2024.

Ziyue Li and Tianyi Zhou. Your mixture-of-experts llm is secretly an embedding model for free. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. A closer look into mixture-of-experts in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4427–4447, 2025.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

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.

Zunhai Su, Qingyuan Li, Hao Zhang, YuLei Qian, Yuchen Xie, and Kehong Yuan. Unveiling super experts in mixture-of-experts large language models. *arXiv preprint arXiv:2507.23279*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

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.

Zeping Yu and Sophia Ananiadou. Neuron-level knowledge attribution in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3267–3280, 2024.

# APPENDIX

## A PROOF OF PROPOSITIONS

**Variance of scores assigned by a component measures its influence on the routing decision.**

**Proposition 1 (Variance of component-contributed scores and norm of component).** Fix a layer $\ell$ and position $i$. Let $\boldsymbol{z}_c \in \mathbb{R}^{d_e}$ denote the contribution of a single component (e.g., token embedding, an attention head output, or a previous MoE block output) to the MoE-input before normalization. Define

$$\overline{\mathrm{LN}}_i^{\ell}(\boldsymbol{z}) \;=\; \frac{\gamma^{\ell} \odot \boldsymbol{z}}{\mathrm{RMS}\big(\boldsymbol{x}_{in,i}^{\ell} + \boldsymbol{a}_{out,i}^{\ell}\big)},$$

where $\boldsymbol{A} \in \mathbb{R}^{d_e \times d_e}$ is a fixed linear map for this $(\ell, i)$. Let $G \in \mathbb{R}^{N \times d_e}$ stack the routing weight vectors as rows, $G = \begin{bmatrix} (\boldsymbol{g}^{\ell,1})^{\top} \\ \cdots \\ (\boldsymbol{g}^{\ell,N})^{\top} \end{bmatrix}$. The vector of expert scores contributed by this component is then

$$\boldsymbol{s}^{(c)} \;=\; G\,\overline{\mathrm{LN}}_i^{\ell}(\boldsymbol{z}_c) \;=\; G\boldsymbol{A}\,\boldsymbol{z}_c \;\in\; \mathbb{R}^N, \quad \text{i.e.,} \quad s_n^{(c)} \;=\; (\boldsymbol{A}^{\top}\boldsymbol{g}^{\ell,n})^{\top}\boldsymbol{z}_c.$$

*(i) Zero-variance implies no routing influence.* If $\boldsymbol{s}^{(c)}$ is constant across experts, i.e., $s_1^{(c)} = \cdots = s_N^{(c)} = c$, then adding or removing this component shifts all experts by the same constant. For any pair $n, m$,

$$(s_n^{\text{total}} + c) - (s_m^{\text{total}} + c) = s_n^{\text{total}} - s_m^{\text{total}},$$

so the ordering is unchanged. Hence a constant-score component has variance 0 and no influence on routing.

*(ii) Variance as an influence measure.* Define $\mathrm{Var}(\boldsymbol{s}^{(c)}) := \frac{1}{N}\sum_{n=1}^{N}(s_n^{(c)} - \overline{s}^{(c)})^2$, where $\overline{s}^{(c)} = \frac{1}{N}\sum_n s_n^{(c)}$. Then $\mathrm{Var}(\boldsymbol{s}^{(c)}) = 0$ iff $\boldsymbol{s}^{(c)}$ is constant. Moreover, for any $\alpha \in \mathbb{R}$,

$$\mathrm{Var}(G\boldsymbol{A}(\alpha \boldsymbol{z}_c)) = \alpha^2\,\mathrm{Var}(G\boldsymbol{A}\boldsymbol{z}_c),$$

so larger component magnitudes yield quadratically larger variance, hence stronger influence on routing.

*(iii) Norm-controlled bounds.* By Cauchy–Schwarz,

$$|s_n^{(c)}| \;=\; \big|(\boldsymbol{A}^{\top}\boldsymbol{g}^{\ell,n})^{\top}\boldsymbol{z}_c\big| \;\le\; \|\boldsymbol{A}^{\top}\boldsymbol{g}^{\ell,n}\|_2\,\|\boldsymbol{z}_c\|_2.$$

Let $M = \max_n \|\boldsymbol{A}^{\top}\boldsymbol{g}^{\ell,n}\|_2$. Then

$$s_n^{(c)} \in [-M\|\boldsymbol{z}_c\|_2,\, M\|\boldsymbol{z}_c\|_2] \quad \forall n,$$

so the range of component-contributed scores is bounded by $2M\|\boldsymbol{z}_c\|_2$. Consequently,

$$\mathrm{Var}(\boldsymbol{s}^{(c)}) \;\le\; \tfrac{1}{N}\|G\boldsymbol{A}\|_F^2\,\|\boldsymbol{z}_c\|_2^2,$$

showing that variance (and hence influence) is upper-bounded by a constant—depending on the router and normalization—times the squared L2 norm of the component.

*Conclusion.* Components that assign constant scores exert no influence, while those with larger norms admit wider score ranges and potentially larger variance across experts, thereby possessing greater capacity to alter expert rankings and influence routing.

**Proposition 2 (Sign and magnitude of component-contributed scores).** Fix a layer $\ell$, position $i$, and an expert $n$. Let the component contribution before routing be $\boldsymbol{z}_c \in \mathbb{R}^{d_e}$ and let $\boldsymbol{u} := \overline{\mathrm{LN}}_i^{\ell}(\boldsymbol{z}_c) \in \mathbb{R}^{d_e}$ denote its normalized contribution at this $(\ell, i)$ (cf. Eq. 8). The score contributed by this component to expert $n$ is

$$s_n^{(c)} \;=\; \boldsymbol{g}^{\ell,n} \cdot \boldsymbol{u}.$$

Let $\theta_n \in [0, \pi]$ be the angle between $\boldsymbol{g}^{\ell,n}$ and $\boldsymbol{u}$.

*(i) Sign determines promotion vs. inhibition.* By the cosine formula for the dot product,

$$s_n^{(c)} \;=\; \|\boldsymbol{g}^{\ell,n}\|_2 \, \|\boldsymbol{u}\|_2 \, \cos\theta_n.$$

Hence $s_n^{(c)} > 0$ iff $\theta_n \in (0, \frac{\pi}{2})$ (acute), $s_n^{(c)} < 0$ iff $\theta_n \in (\frac{\pi}{2}, \pi)$ (obtuse), and $s_n^{(c)} = 0$ iff $\theta_n = \frac{\pi}{2}$ (orthogonal). Since (a) top-$k$ selection depends only on score orderings and (b) the softmax used to form expert weights is strictly increasing in each coordinate, adding a component with $s_n^{(c)} > 0$ increases expert $n$'s total score and softmax weight (promotes selection), while $s_n^{(c)} < 0$ decreases them (inhibits selection); $s_n^{(c)} = 0$ leaves them unchanged.

*(ii) Magnitude quantifies degree of influence.* The absolute score satisfies

$$|s_n^{(c)}| \;=\; \|\boldsymbol{g}^{\ell,n}\|_2 \, \|\boldsymbol{u}\|_2 \, |\cos\theta_n| \;\le\; \|\boldsymbol{g}^{\ell,n}\|_2 \, \|\boldsymbol{u}\|_2,$$

with equality iff $\boldsymbol{g}^{\ell,n}$ and $\boldsymbol{u}$ are colinear. For fixed $\boldsymbol{g}^{\ell,n}$, the dependence on angle and component length is monotone:

$$\frac{\partial s_n^{(c)}}{\partial \theta_n} = -\,\|\boldsymbol{g}^{\ell,n}\|_2 \, \|\boldsymbol{u}\|_2 \, \sin\theta_n \;\le\; 0, \qquad \frac{\partial s_n^{(c)}}{\partial \|\boldsymbol{u}\|_2} = \|\boldsymbol{g}^{\ell,n}\|_2 \, \cos\theta_n.$$

Thus, for $\theta_n \in [0, \frac{\pi}{2})$, decreasing the angle (better alignment) or increasing the component norm strictly increases $s_n^{(c)}$; for $\theta_n \in (\frac{\pi}{2}, \pi]$, the same operations make $s_n^{(c)}$ more negative, strengthening inhibition. Consequently, the *degree* of promotion/inhibition is exactly captured by the magnitude $|s_n^{(c)}|$, which grows with both alignment (via $|\cos\theta_n|$) and component length $\|\boldsymbol{u}\|_2$ (and hence with $\|\boldsymbol{z}_c\|_2$ up to the fixed normalization factor at $(\ell, i)$).

*Conclusion.* The sign of the component-expert dot product governs whether the component promotes or inhibits that expert's selection, while its magnitude—$\|\boldsymbol{g}^{\ell,n}\|_2 \, \|\boldsymbol{u}\|_2 \, |\cos\theta_n|$—quantifies the strength of this effect. □

## B BASIC INFORMATION OF TESTED MODELS

Table 1: Basic information of tested models.

| Information | OLMoE | DeepSeek-V2-Lite | Qwen3-30B-A3B | Mixtral-8x7B |
|---|---|---|---|---|
| Total Params | 7B | 16B | 30B | 47B |
| Activated Params | 1B | 3B | 3B | 7B |
| Number of Layers | 16 | 27 | 48 | 32 |
| Number of Routed Experts | 64 | 64 | 128 | 8 |
| Top-$k$ | 8 | 6 | 8 | 2 |

NOTE: Layer 0 in DeepSeek-V2-Lite is an FFN layer, not an MoE layer. Each MoE layer has two shared experts in DeepSeek-V2-Lite.

## C BRIEF INTRODUCTION TO IOI TASK

We apply the method (path patching) and the metric (logit difference) from Wang et al. (2023), reproduce the experiment on OLMoE, and identify four types of heads, which are all active at END token, as follows:

- **Name Mover Heads** attend to the previous name tokens. They promote IO token as the prediction result.
- **Negative Name Mover Heads** are similar to Name Mover Heads but inhibit IO token as the prediction result.
- **S-Inhibition Heads** inhibit S token, and influence Name Mover Heads and Negative Name Mover Heads.
- **Backup Name Mover Heads** are active when the Name Mover Heads are ablated. They also show a weak influence when the regular Name Mover Heads work normally.

We adapt the code provided in the original paper to generate 5000 samples for the IOI task experiments.

## D SUPPLEMNTARY RESULTS ON THE DECOMPOSITION AT THE LAYER LEVEL

We find that the scoring distribution at the lead tokens is completely different from other tokens (Figures 6 $\sim$ 9. For example, In OLMoE, in the score assignment from sending attention layers to receiving MoE layers, the high variance occurs at the bottom sending layers to their neighbor receiving layers, and the last receiving layer. However, when sending and receiving layers are both MoE layers, the high variance occurs at some sending MoE layers and the last receiving MoE layer. The highest APS occur at A15 $\rightarrow$ M15, and M14 $\rightarrow$ M15. The highest ANS occur at A14 $\rightarrow$ M15 (A0 $\rightarrow$ M0), and M2 $\rightarrow$ M15. The four models at the lead token have different variance patterns. We can observe that there are strides (e.g., Figure 8.(d)) in the score variance distribution, indicating that some sending layers are more influential and have a persistent influence on the routing decisions.

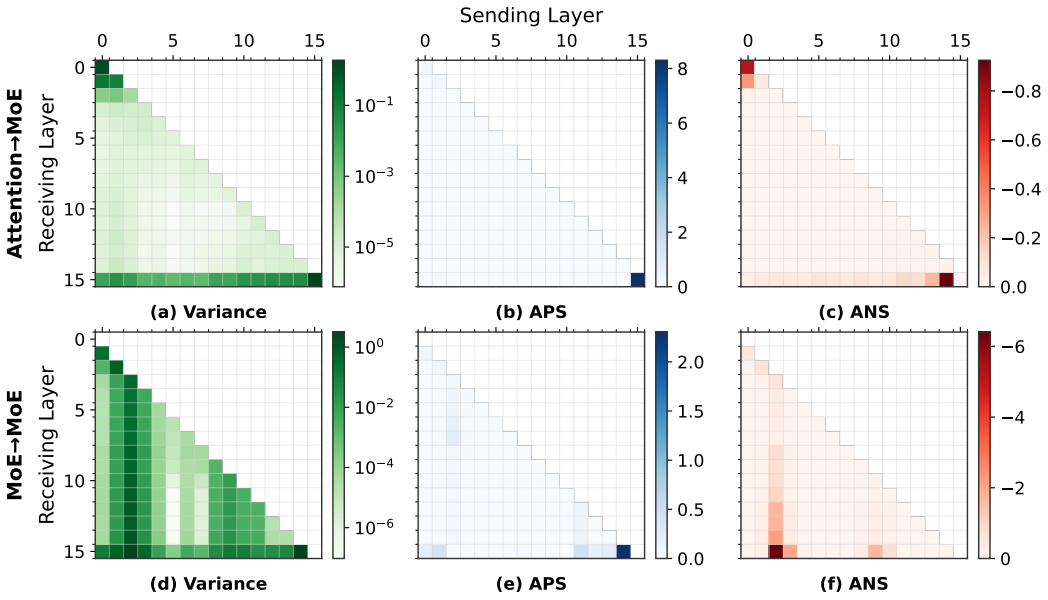



Figure 6: Scoring distribution at the lead tokens, OLMoE

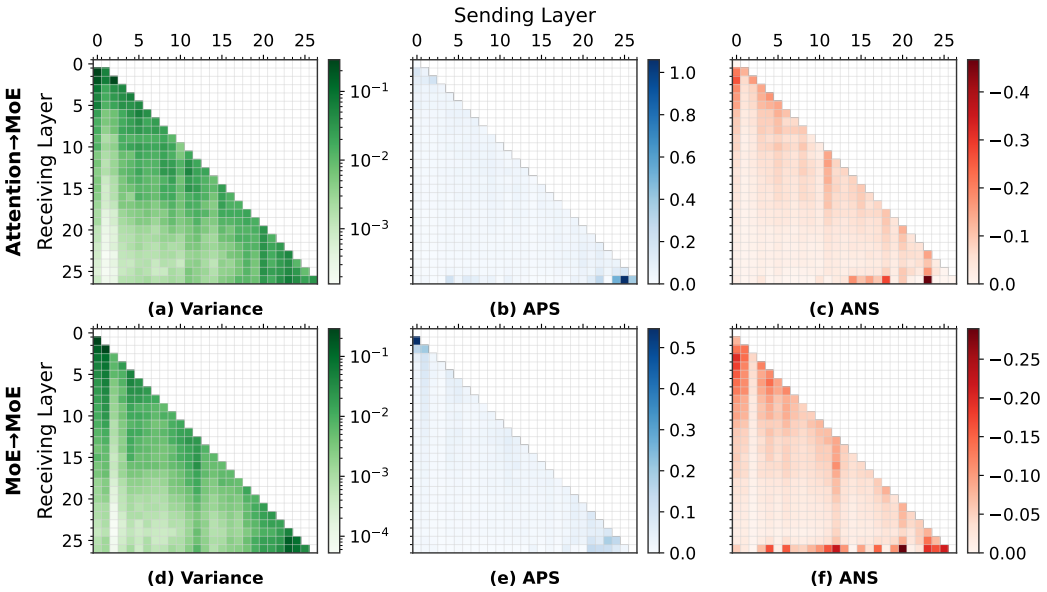



Figure 7: Scoring distribution at the lead tokens, DeepSeek

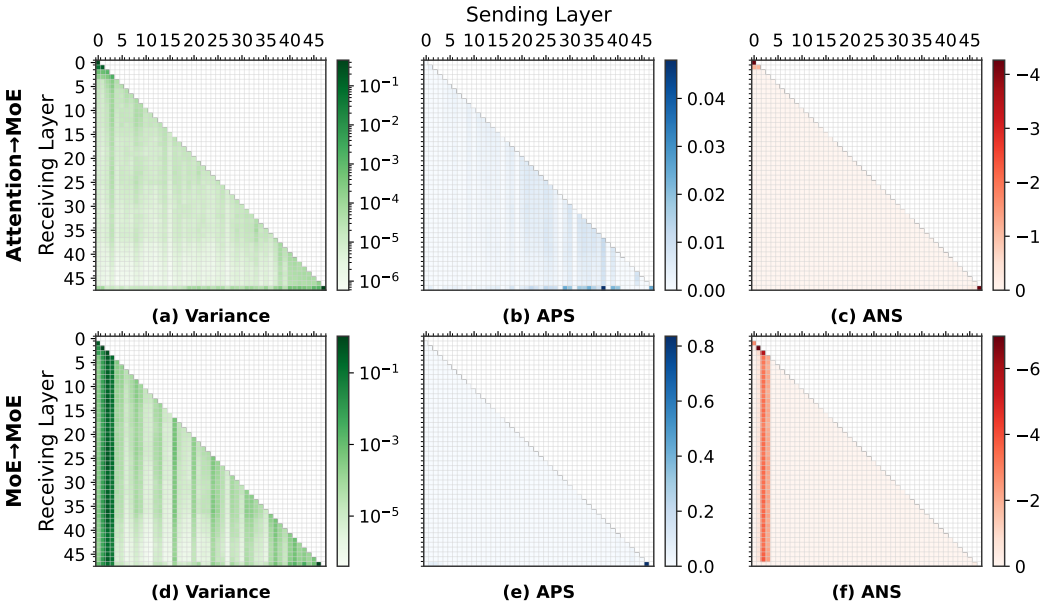



Figure 8: Scoring distribution at the lead tokens, Qwen

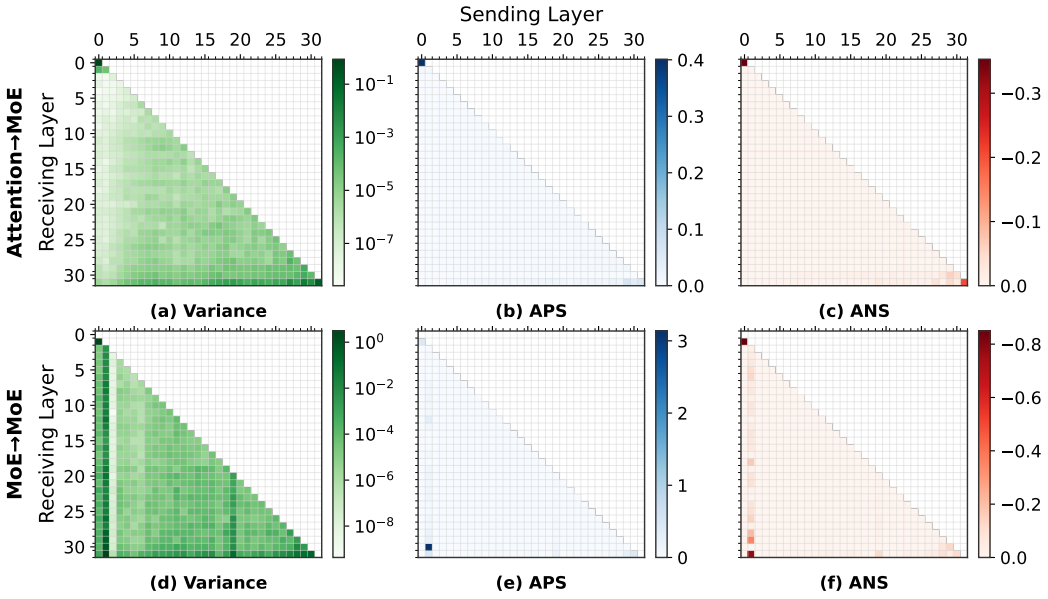



Figure 9: Scoring distribution at the lead tokens, Mixtral

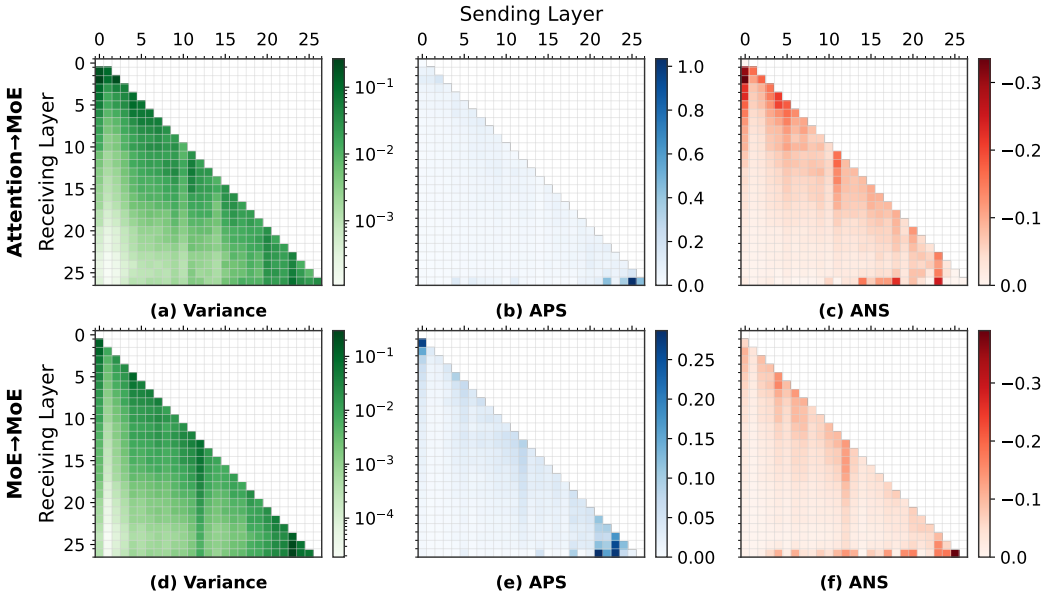



Figure 10: Scoring distribution at other tokens (except the lead tokens), DeepSeek

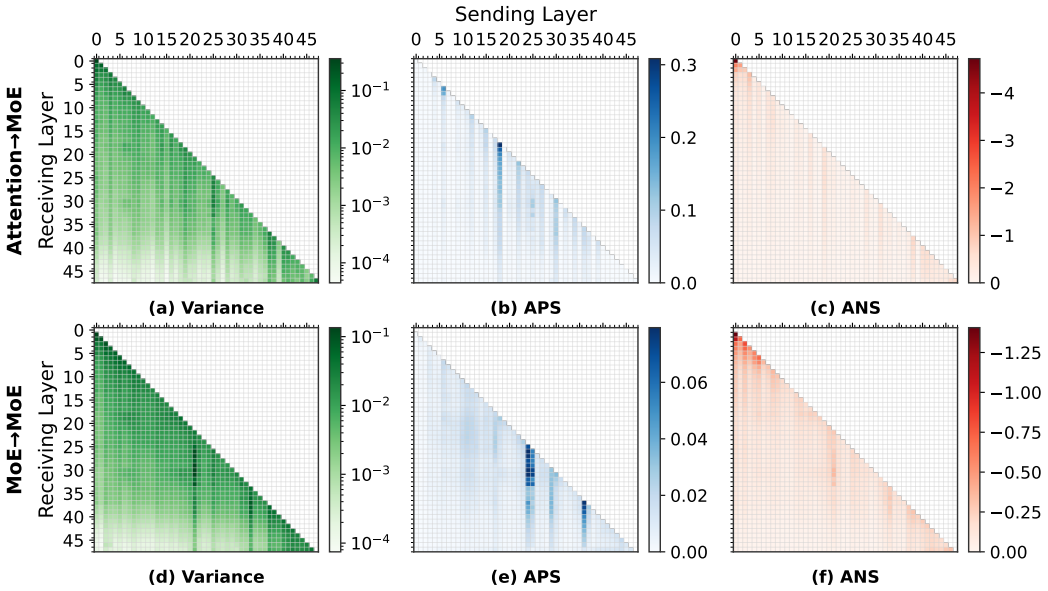



Figure 11: Scoring distribution at other tokens (except the lead tokens), Qwen

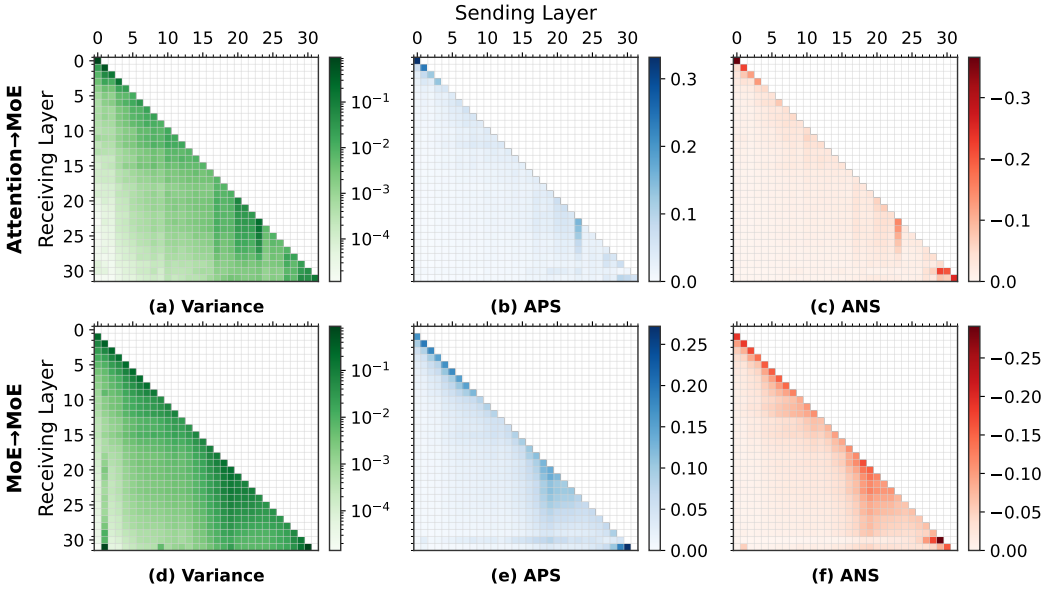



Figure 12: Scoring distribution at other tokens (except the lead tokens), Mixtral

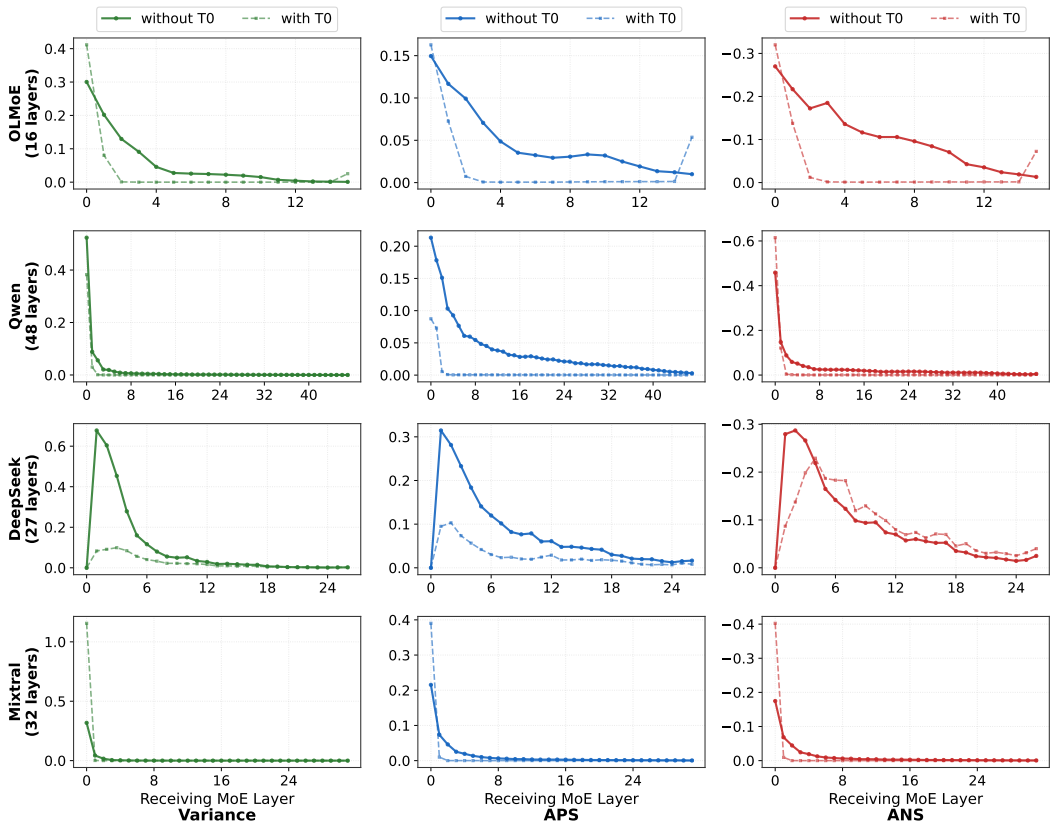



Figure 13: Scoring distribution of tokens.

## E Results of t-SNE on Scores Assigned by Tokens (Qwen)

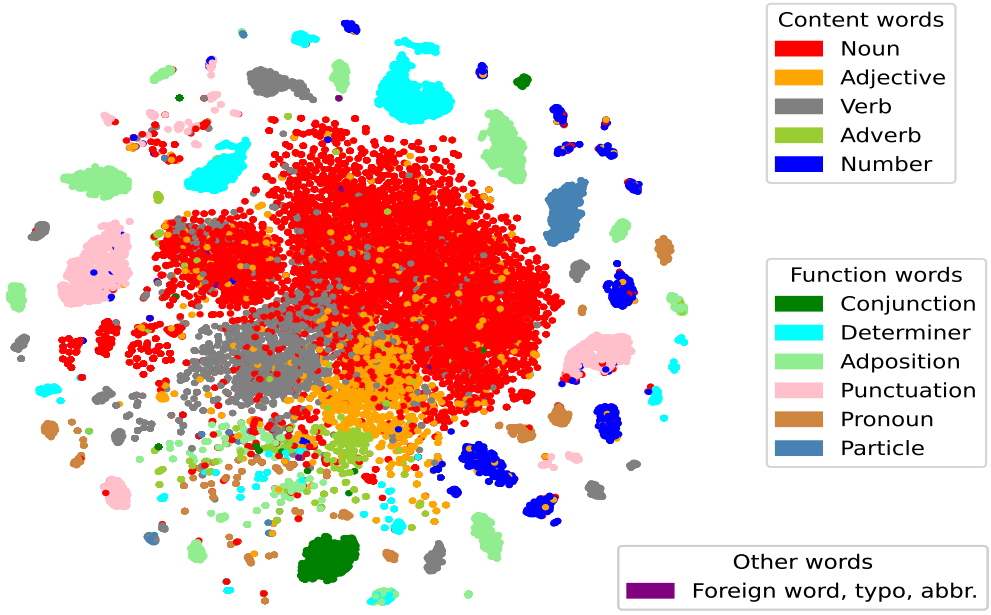



Figure 14: t-SNE of scores assigned by token embeddings in Qwen, colored by POS.

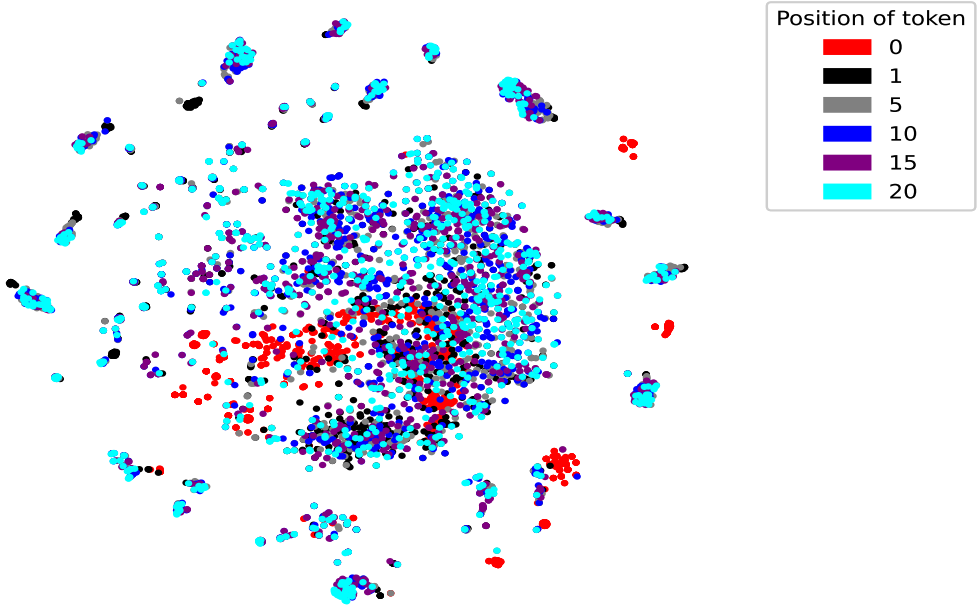



Figure 15: t-SNE of scores assigned by token embeddings in Qwen, colored by position.

# F RESULTS ON IOI TASK

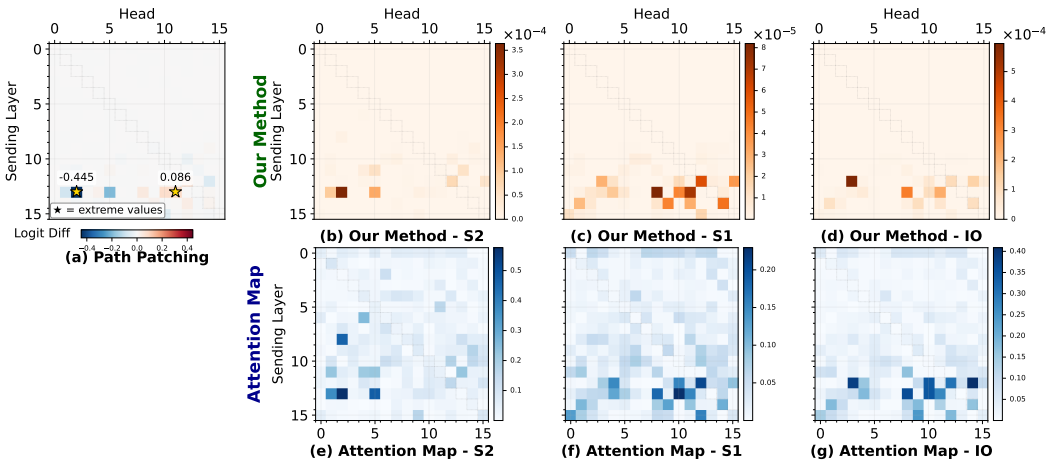



Figure 16: Path patching: (a) IO to logits (logit difference). Variance of scores assigned by: (b) (Head, Query=END, Key=S2), (c) (Head, END, S1), (d) (Head, END, IO) to the experts in the last block. Attention map: (e) (Query=END, Key=S2), (f) (END, S1), (g) (END, IO).

We follow Wang et al. (2023) to use "IO" to denote the indirect object, "S1" and "S2" denote the first and second occurrences of the subject, "END" denote the last token of the prompt. Additional details are provided in Appendix C.

The direct effect on the logit difference $logit(S) - logit(IO)$ found by path patching $h$ to logits for each head $h$ at END token is shown in Figure 16d. We can observe the Name Mover Heads (A13H1, A13H2, A13H5) and Negative Name Mover Heads (A12H12, A13H10, A13H11) have a pronounced influence on the prediction. Some function heads have a weaker influence, indicating they are probably S-inhibition heads or Backup Name Mover Heads. Since these function heads directly focus on name tokens, we can study the attention map and score variance map on (Query=END, Key=S2), (END, S1) and (END, IO) to find them.

Since the end of patching circuits is the layer $L-1$ (the last layer), we can first observe the scores assigned by (Head, END, S2), i.e., terms $\overline{\text{LN}}^{L-1}_{END}(\boldsymbol{W}_O^{x,y}\boldsymbol{A}^{x,y}_{END,S2}\boldsymbol{v}^{x,y}_{S2})$ for any AxHy (Equation 9), which depicted in Figure 16a. We can find that the Name Mover Heads have a comparatively high influence on the routing decisions at the last layer, indicating the scores assigned by (Head=A13H1, Query=END, Key=S2), (A13H2, END, S2) and (A13H5, END, S2) have a comparatively high influence on the routing decisions in Layer $L-1$ (last layer).

In contrast, the Negative Name Mover Heads influence more in the variance score maps of (END, S1) and (END, IO) (Figures 16b and c). The S-inhibition heads (e.g., A11H13 and A12H3) and Backup Name Mover Heads (e.g., A11H5 and A12H14) are also noticeable but their influence is usually weaker than Name Mover Heads and Negative Name Mover Heads. Some heads appear to have functions but do not appear in these three score variance maps (e.g., A14H12 and A15H10). We speculate that these heads may attend to other keys instead of the name tokens.

We compare the attention maps (Figure 16e∼g) with the score variance maps. We can find that they resemble each other. Although attention maps have some small non-zero activations in the early and intermediate layers, they usually do not have a strong contribution on the routing decisions in the last layers, which may due to the natural influence decay in deep layers, or the value vector cancels the effect. In conclusion, we validate that the function heads can influence the assignment scores, and the assignment scores correlate with the attention maps.

## G Results on scoring of attention heads in DeepSeek

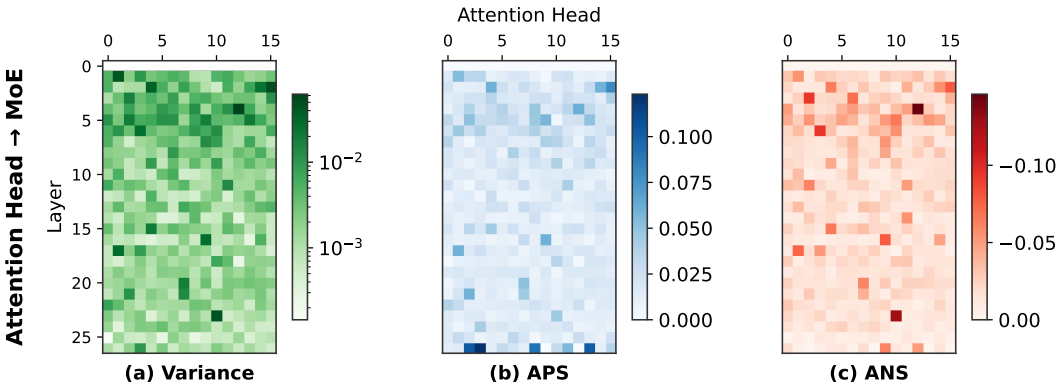



Figure 17: DeepSeek: (a) Variance of scores assigned by heads to experts in the same block. (b) Average positive scores (APS) of the heads. (c) Average negative scores (ANS) of the heads. Please note that Layer 0 of DeepSeek is an FFN layer, not an MoE layer.

## H Supplementary results on scoring of experts. results

Table 2: The rank of variance of scores assigned by the Super Experts to the experts in the next MoE layers.

| Model | Experts |
|---|---|
| DeepSeek | M2E54 (#55), M3H38 (#7) |
| OLMoE | Not Available |
| Qwen | M1H68 (#1), M2E92 (#3), M3E82 (#2) |
| Mixtral | M1E3 (#8) |

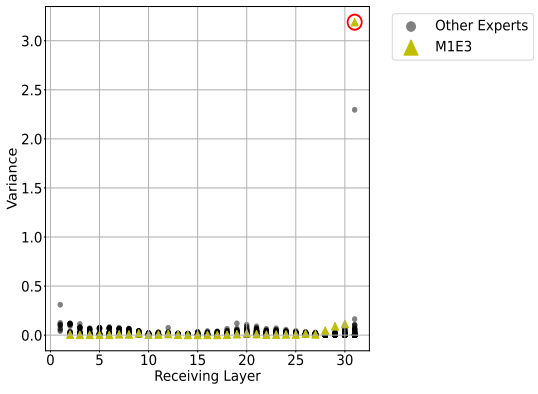



Figure 18: In Mixtral, M1E3 is found to be a "Super Expert", but it has a high variance at the top layers, especially the last layer (circled in red).