DEAL: Disentangling Transformer Head Activations for LLM Steering

Li-Ming Zhan Bo Liu Zexin Lu Yujie Feng Chengqiang Xie
Jiannong Cao Xiao-Ming Wu
Department of Data Science and Artificial Intelligence
The Hong Kong Polytechnic University
Hong Kong S.A.R.
{lmzhan.zhan, bokelvin.liu, zexin.lu, yujie.feng}@connect.polyu.hk
cqx.xie@gmail.com, csjcao@comp.polyu.edu.hk, xiao-ming.wu@polyu.edu.hk

Abstract

Inference-time steering aims to alter the response characteristics of large language models (LLMs) without modifying their underlying parameters. A critical step in this process is the identification of internal modules within LLMs that are associated with the target behavior. However, current approaches to module selection often depend on superficial cues or ad-hoc heuristics, which can result in suboptimal or unintended outcomes. In this work, we propose a principled causal-attribution framework for identifying behavior-relevant attention heads in transformers. For each head, we train a vector-quantized autoencoder (VQ-AE) on its attention activations, partitioning the latent space into behavior-relevant and behavior-irrelevant subspaces, each quantized with a shared learnable codebook. We assess the behavioral relevance of each head by quantifying the separability of VQ-AE encodings for behavior-aligned versus behavior-violating responses using a binary classification metric. This yields a behavioral relevance score that reflects each head's discriminative capacity with respect to the target behavior, guiding both selection and importance weighting. Experiments on seven LLMs from two model families and five behavioral steering datasets demonstrate that our method enables more accurate inference-time interventions, achieving an average relative improvement of 20% (up to 81.5%) over the ITI method (Li et al., 2023) on the truthfulness-steering task. Furthermore, the heads selected by our approach exhibit strong zero-shot generalization in cross-domain truthfulness-steering scenarios.

1 Introduction

As large language models (LLMs) demonstrate increasingly versatile capabilities, understanding and controlling their internal decision-making processes has become more challenging. Consequently, steering the behavior of LLMs to ensure they produce outputs with desirable properties—such as truthfulness, appropriate refusals, and minimal hallucinations—has become a primary focus in both research and practice (Yang et al., 2024; Perez et al., 2022; Goyal et al., 2024).

Inference-time activation engineering (Arditi et al., 2024; Li et al., 2023; Zou et al., 2023; Lee et al., 2024) has emerged as a promising approach for guiding the behaviors of LLMs, due to its high efficiency and strong generalization capability. Works in this area aim to steer LLM behavior by applying additive perturbations to their intermediate activations. The typical process involves two main steps: (i) identifying internal modules (such as attention heads) that are linked to the target behavior, and (ii) constructing a *steering vector* to modify the original activations accordingly. Accurately identifying which modules are relevant to the target behavior is crucial for effective steering. However, due to the complexity of LLM internals, existing methods often rely on superficial linear correlations (Li et al., 2023), ad-hoc empirical cues (Yin et al., 2024), or computationally expensive cross-validation (Zhang et al., 2024; Rimsky et al., 2024) to select intervention modules. Therefore, a principled, effective, and efficient framework for module selection is essential for advancing inference-time steering.

To address this gap, we present **DEAL**—***D****isentangling Transformer h****E****ad* ***A****ctivations for* ***L****LM Steering*—a novel framework that disentangles behavior related representation from head activations and computes head-wise confidence scores to facilitate LLM steering. Attention heads govern how LLMs generate text, enabling diverse functions such as induction (Crosbie and Shutova, 2025; Olsson et al., 2022) and long-range factual retrieval (Wu et al., 2024). However, these functions are encoded within the heads' hidden activations in a highly entangled manner (Monea et al., 2024; Ferrando et al., 2024),

hindering the extraction of behavior-specific features from the activations.

We address this challenge by learning a *disentangled* latent space that separates each head activation vector into behavior-relevant and behavior-irrelevant components. Specifically, for each attention head, we train a vector-quantized autoencoder (VQ-AE) that divides the latent space into several subspaces—some capturing behavior-relevant features and others capturing behavior-irrelevant features. Each subspace is quantized by a shared learnable codebook. The VQ-AE is optimized using a standard reconstruction loss alongside an auxiliary supervised contrastive loss, which encourages separation between positive and negative behaviors. After training, the VQ-AE produces a discrete, sequential code for each head activation. To quantify the behavioral relevance of each head, we fit an autoregressive prior p_θ over the discrete codes, estimating the likelihood that a given code aligns with the target behavior. For each head, we assess the separability of these likelihoods for behavior-aligned versus behavior-violating responses using a binary classification metric (e.g., AUC-ROC), yielding a behavioral relevance score that quantifies each head's discriminative ability. This score is then used to guide both head selection and importance weighting.

To demonstrate the effectiveness of DEAL, we evaluated it on seven LLMs from two leading open-source families—LLAMA and QWEN. We applied our head-selection strategy to two prevalent attention variants: standard multi-head attention (e.g., Llama-7B, Llama2-7B, Llama2-13B-Chat) and the more efficient grouped-query attention (GQA) (Ainslie et al., 2023a) (e.g., Qwen2.5-7B, Llama3.1-8B-Instruct). DEAL integrates seamlessly with current behavior-steering methods, including the mean-difference technique ITI (Li et al., 2023) and the fine-tuning approach LoFiT (Yin et al., 2024). On the truthfulness-steering task, DEAL achieved an average relative improvement of 20% (up to 81.5%) over ITI across the seven LLMs. Moreover, the heads selected by DEAL exhibit strong generalization in zero-shot, cross-domain truthfulness-steering scenarios.

2 Preliminary

Activation engineering (Li et al., 2023; Rimsky et al., 2024) is an emerging paradigm aimed at directing desired behaviour in LLMs without the need for fine-tuning. This technique injects perturbations into the intermediate activations of LLMs to influence their outputs. It typically involves three steps: (i) select a set of behaviour-specific attention heads; (ii) for each selected head, compute a steering vector from its activations; and (iii) during decoding, add this steering vector to the head's original activation, thereby biasing the model toward the desired behaviour.

Step 1: Head Selection. In a decoder–only LLM, each transformer layer typically employs multiple parallel self-attention heads, enabling each head to specialize in distinct patterns, such as induction and long-range fact retrieval (Wu et al., 2024). We denote the selected subset of attention heads as $\mathcal{G} \subseteq \{(l,i) \mid l \in \{1,\dots,L\}, i \in \{1,\dots,H\}\}$, where l is the layer index, L is the number of transformer layers, h is the head index within each layer, and H is the number of heads per layer. Notably, $|\mathcal{G}| \ll LH$. The selection of attention heads for intervention is crucial (Yin et al., 2024). A suitable choice can elicit the desired behavior, while an incorrect selection may significantly degrade performance. In this study, we focus on developing a principled and effective head selection method.

Step 2: Steering Vector Derivation. For an input token sequence $\{x_1, x_2, \dots, x_T\}$, the head-specific context vector at time step t is computed as

$$\mathbf{h}_t^{(l,i)} = \text{Attention}\big(\mathbf{Q}_t^{(l,i)}, \mathbf{K}_{\le t}^{(l,i)}, \mathbf{V}_{\le t}^{(l,i)}\big), \quad (1)$$

where $\mathbf{Q}^{(l,i)}, \mathbf{K}^{(l,i)}, \mathbf{V}^{(l,i)} \in \mathbb{R}^{T \times d_h}$ denote the query, key, and value projections, respectively, and d_h is the dimensionality per head. Given a behavior-contrastive dataset $\mathcal{D} = \{(q_i, a_i, y_i)\}_{i=1}^N$, where q_i denotes a query, a_i is its associated answer, $y_i \in \{0,1\}$ indicates whether the pair exhibits the target behavior (e.g., truthfulness), and N is the dataset size. For each query, the corpus contains both compliant and non-compliant answers, allowing the formation of a positive behavior subset $\mathcal{D}^+ = \{(q_i, a_i) | y_i = 1\}$ and a negative behavior subset $\mathcal{D}^- = \{(q_i, a_i) | y_i = 0\}$. The steering vector $\mathbf{v}^{(l,i)}$ for head (l,i) can be derived using the mean difference method, applied to the activations from $\mathcal{D}^+$ and $\mathcal{D}^-$ respetively (Lee et al., 2024; Turner et al., 2023b; Rimsky et al., 2024; Li et al., 2023), or by employing the fine-tuning method (Yin et al., 2024).

Step 3: Steering. During decoding, the activa-

tion of each selected head is perturbed by:

$$\hat{\mathbf{h}}_t^{(l,i)} = \mathbf{h}_t^{(l,i)} + \epsilon\, \mathbf{v}^{(l,i)}, \quad (l,i) \in \mathcal{G}, \qquad (2)$$

where $\mathbf{v}^{(l,i)}$ is the steering vector for head (l,i) and $\epsilon \in \mathbb{R}^+$ controls the steering strength.

3 Our Head Selection Method: DEAL

In this section, we present DEAL, a principled method for head selection and importance weighting steering, as illustrated in Figure 1. To identify behavior-discriminative patterns from attention head activations, we propose learning a disentangled, quantized latent space to encode the activations from each attention head. These encodings are then used to train a scoring function to assess the relevance of each head in relation to the target behavior.

3.1 Learning a Disentangled Representation Space for Encoding Head Activations

Given an input token sequence $\{x_1, x_2, \ldots, x_T\}$, we use the last token activations of attention heads, $\mathbf{h}_T^{(l,i)}$, to distill behavior indicative information. As illustrated in Fig. 1, a vector-quantized autoencoder (VQ-AE) takes the activations of a single attention head as input and projects them into a quantized latent space, yielding discrete encodings, represented as sequences of codebook indices.

Let $\mathbf{h}_T^{(l,i)} \in \mathbb{R}^{d_h}$ denote the representation generated by attention head i in layer l. An encoder $E\colon \mathbb{R}^{d_h} \to \mathbb{R}^{d_e}$ projects this vector into a lower-dimensional embedding $\mathbf{z}^{(l,i)} = E\big(\mathbf{h}_T^{(l,i)}\big) \in \mathbb{R}^{d_e}$. Notably, we divide $\mathbf{z}^{(l,i)}$ into U segments (termed semantic units), each with a length $d_u = d_e/U$:

$$\mathbf{z}^{(l,i)} = \big[\mathbf{z}_1^{(l,i)}\ ;\ \ldots\ ;\ \mathbf{z}_U^{(l,i)}\big],\ \mathbf{z}_u^{(l,i)} \in \mathbb{R}^{d_u}. \quad (3)$$

Each semantic unit $\mathbf{z}_u^{(l,i)}$ will be quantised via nearest-neighbour search in a learnable codebook $\mathcal{C} = \big\{\mathbf{c}_k \in \mathbb{R}^{d_u} \mid k = 1, \ldots, K\big\}$. The corresponding codebook index for each semantic unit is obtained by:

$$\kappa_u^{(l,i)} = \arg \min_{k \in \{1,\ldots,K\}} \big\|\mathbf{z}_u^{(l,i)} - \mathbf{c}_k\big\|_2,$$

where $\kappa_u^{(l,i)} \in \{1, \ldots, K\}$, and $\kappa_{1:U}^{(l,i)}$ is a *discrete sequence* representing the head activation.

Key Insight: The rationale for this design is that not all elements of the latent embedding vector $\mathbf{z}^{(l,i)}$ are related to the target behaviour. By partitioning $\mathbf{z}^{(l,i)}$ into multiple semantic units, it is possible to segregate components that are behavior-related from those that are not, thereby yielding a *disentangled* representation space.

The decoder D reconstructs the head activation by approximating each semantic unit using entries ($\mathbf{c}_k$) from the codebook, i.e., $\hat{\mathbf{z}}^{(l,i)} = \big[\tilde{\mathbf{z}}_1^{(l,i)}, \tilde{\mathbf{z}}_2^{(l,i)}, \ldots, \tilde{\mathbf{z}}_U^{(l,i)}\big]$, where $\tilde{\mathbf{z}}_u^{(l,i)} = \mathcal{C}\big[\kappa_u^{(l,i)}\big]$. The conventional loss function for training the VQ-AE is defined as:

$$\mathcal{L}_{\text{VQ}} = \underbrace{\|\mathbf{h}^{(l,i)} - D(\hat{\mathbf{z}}^{(l,i)})\|_2^2}_{\text{Reconstruction Loss}} \qquad (4)$$
$$+ \underbrace{\|\,\text{sg}[\mathbf{z}^{(l,i)}] - \hat{\mathbf{z}}^{(l,i)}\|_2^2}_{\text{Codebook Loss}} + \beta\ \underbrace{\|\mathbf{z}^{(l,i)} - \text{sg}[\hat{\mathbf{z}}^{(l,i)}]\|_2^2}_{\text{Commitment Loss}}$$

where $\text{sg}[\cdot]$ is the stop-gradient operator and β is a scalar hyperparameter. The reconstruction loss encourages faithful decoding; the codebook loss pulls the corresponding codebook vectors toward the embedding; and the commitment loss pushes the embedding toward the selected code vectors.

To learn behavior-discriminative representations, we augment the standard VQ-AE objective (Eq. 4) with a supervised contrastive term, $\mathcal{L}_{\text{SC}}$. This term involves the quantised representation $\hat{\mathbf{z}}^{(l,i)} \in \mathbb{R}^{d_e}$, encouraging the clustering of representations for positive examples $\mathcal{D}^+$ while distinguishing them from those of negative samples. The supervised contrastive loss is formally defined as:

$$\mathcal{L}_{\text{SC}} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{P}(i)|}\sum_{j\in\mathcal{P}(i)}\log\frac{\exp(s_{ij})}{\sum_{k\neq i}\exp(s_{ik})},$$

$$s_{ij} = \frac{\hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j}{\tau}, \quad \mathcal{P}(i) = \{\, j \neq i \mid y_j = y_i \,\}.$$

where $\tau > 0$ is a temperature hyperparameter, and $\mathcal{P}(i)$ denotes the set of indices corresponding to the positive samples w.r.t. i (samples sharing the same class label as i, excluding i itself).

The overall loss for training the feature extractor is then formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{VQ}} + \alpha\, \mathcal{L}_{\text{SC}},$$

where α is a scalar hyperparameter balancing the reconstruction and contrastive losses.

3.2 Learning a Scoring Function for Head Selection and Importance Weighting

The discrete sequence $\kappa_{1:U}^{(l,i)}$, obtained in the previous step, can be directly modeled using an autore-

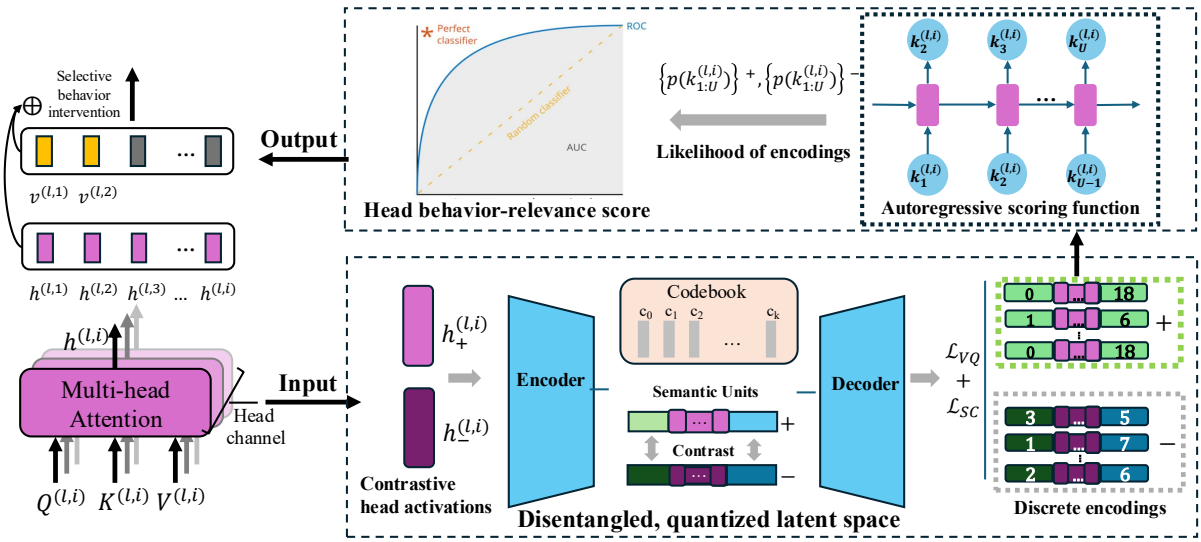


Figure 1: Overview of the proposed DEAL framework. We use activations from each attention head to train a VQ-AE, aiming to learn a disentangled, quantized latent space. The VQ-AE is trained using a latent contrastive loss in conjunction with the standard VQ loss. The discrete encodings produced by the VQ-AE are then used to train a scoring function that outputs the probability of a given encoding corresponding to the target behavior. Finally, a binary classification metric, such as the area under the ROC curve (AUC-ROC), is employed to determine the behavioral-relevance score for each head.

gressive distribution p_θ:

$$p_\theta\left(\kappa_{1:U}^{(l,i)}\right) = \prod_{u=1}^{U} p_\theta\left(\kappa_u^{(l,i)} \mid \kappa_{1:u-1}^{(l,i)}\right), \quad (5)$$

where p_θ can be implemented as a lightweight autoregressive network, such as LSTM or GRU, and trained on the discrete encodings of the *positive* subset $\mathcal{D}_+$ (as defined in Sec. 2).

For a head (l, i) containing discriminative information, our VQ-AE will produce discrete encodings capturing this pattern. The optimized p_{θ^*}, trained on the positive encodings, will assign high probability to the encodings of positive examples and low probability to those of negative examples. Given a development set $\mathcal{D}_{\text{val}} = \mathcal{D}_{\text{val}}^+ \cup \mathcal{D}_{\text{val}}^-$, p_{θ^*} will predict a probability for each sample in $\mathcal{D}_{\text{val}}$. We then employ a binary classification metric, such as the area under the ROC curve (AUC–ROC, used in our experiments), to determine the behavior-discriminative score $s^{(l,i)}$ for head (l, i). Larger values indicate a stronger association between head (l, i) and the target behavior.

Next, we rank all heads based on $s^{(l,i)}$ and select the top S heads with the highest scores for intervention. The importance of a selected head (l, i) is weighted by $s^{(l,i)}$ during steering, as follows:

$$\hat{\mathbf{h}}_t^{(l,i)} = \mathbf{h}_t^{(l,i)} + \frac{s^{(l,i)}}{s_{\max}^{(l,i)}} \epsilon \mathbf{v}^{(l,i)}, \quad (l,i) \in \mathcal{G}, \quad (6)$$

where $s_{\max}^{(l,i)}$ is the maximum behavior-discriminative score across all heads.

4 Experiments

4.1 Datasets

Truthfulness The *TruthfulQA* benchmark (Lin et al., 2022) consists of 817 questions spanning 38 categories (e.g., health, law, and politics), designed to assess whether language models generate factually accurate responses or simply mimic prevalent falsehoods. This dataset challenges models to balance factual correctness with the potential risk of echoing common human misperceptions. MC1 and MC2 are calculated to evaluate this dataset.

AI Risk behaviors Following the approach in (Rimsky et al., 2024), we further assess our method using *specialized subsets* that probe specific behavioral traits—namely, *hallucination, myopic reward, corrigibility, and survival instinct*. They are implemented in a multiple-choice question answering format. The hallucination set is generated by GPT-4, whereas other sets are derived from Anthropic's "Advanced AI Risk" human-written evaluation dataset (Perez et al., 2022). Detailed statistics can be found in the Appendix.

Generalization Evaluation To assess the generalization of our method, we apply the heads selected on TruthfulQA ***without*** further fine-tuning to

two knowledge-seeking benchmarks, MQuAKE and CLUTRR. Following the protocol of (Cohen et al., 2024), we report the exact-match (EM) score on each test set, which contains 864 and 450 question–answer pairs, respectively.

4.2 Baselines and Implementation Details

Our head–selection framework integrates seamlessly with existing steering methods. To evaluate its utility, we pair it with two representative paradigms: the mean-difference approach ITI (Li et al., 2023) and the fine-tuning approach LoFiT (Yin et al., 2024). In each case, we replace the original head set with the top heads from oparagraphur ranking before applying steering.

Implementation Details Experiments were run on a single NVIDIA H100 GPU with PyTorch (Paszke et al., 2019) v2.3.1 and HuggingFace Transformers (Wolf, 2019) v4.46.3. We evaluated models from two families: *Llama* (Llama-7B, Llama2-7B, Vicuna-7B, Llama2-7B-chat, Llama-3.1-8B-Instruct, Llama2-13B-chat) and *Qwen* (Qwen2.5-7B, Qwen2.5-7B-instruct). For head-wise VQ-VAE training, both encoder and decoder use a two-layer MLP with hidden sizes 128 and 64. The encoder output is partitioned into 8 semantic units that are quantised with a codebook of size 32. A contrastive-loss weight of $\alpha = 1 \times 10^{-3}$ suffices, and we set the VQ-VAE commitment coefficient in Eq. (4) to $\beta = 0.25$.

Unless otherwise specified, each VQ-VAE model is trained for 40 epochs using the Adam optimizer with a learning rate of $1e-4$. For the autoregressive behavior prior, we employ a single-layer GRU with a hidden dimension of 64, which is also trained using Adam with a learning rate of $1e-3$ for 5 epochs. All results are averaged over three independent random runs.

ITI (Li et al., 2023) first conducts linear probing on hidden–state head activations, then computes the mean centroids of the positive and negative groups. Their head selection relies on the Logistic regression accuracy on the eval dataset. The vector from the positive to the negative centroid is used as the *steering vector*; this methodology belongs to the mean-difference family (Rimsky et al., 2024).

LoFiT (Yin et al., 2024) is a tuning-based steering method that leverages Direct Preference Optimization (DPO) and supervised fine-tuning (SFT). It first trains a scalar weight for each head and uses the norms of these scalars to select the heads to intervene. The steering vectors are likewise learned with either DPO or SFT, depending on the dataset. LoFiT belongs to the fine-tuning family of behavior-steering techniques.

4.3 Main Results

Steering Towards Truthfulness We evaluate DEAL on TruthfulQA to assess the effectiveness of its selected intervention heads in promoting truthful responses. Table 1 summarizes the MC1 and MC2 results across all models.

Following their original setups, we intervene on the top 48 heads for DEAL_{ITI} and the top 32 heads for $\text{DEAL}_{\text{LoFiT}}$. Equipped with our ranking heads, both ITI and LoFiT improve markedly on MC1 and MC2. (LoFiT results for Llama-3 and the Qwen family are omitted because the released code does not yet support those models.)

The gains are substantial: on Qwen2.5-7B, DEAL_{ITI} lifts accuracy by $\sim 6\%$ on both metrics; on Llama2-7B and Llama-7B, it adds nearly 10% to MC1 and 17% to MC2. We further observe that ITI fails to steer Qwen and Llama-3.1, presumably because these models employ grouped-query attention (GQA) (Ainslie et al., 2023b) rather than standard multi-head attention.

Steering Towards General Behaviors To assess the effectiveness of DEAL on general behaviours, we run experiments on the Advanced AI Risk benchmark. Using our proposed layer metric described in Appendix. we rank all layers of Llama2-7B-chat and select the top five and bottom five. For each behaviour dataset, we then report the average token probability assigned to behaviour-aligned answers following the CAA protocol (Rimsky et al., 2024). Details about the reported metric can be found in Appendix (B).

As shown in Table 2, on the top five and last five layers using multipliers of -1, 0, and 1, where 1 steers the model toward exhibiting the behavior and -1 counteracts it. Notably, enforcing interventions on the top five layers leads to a significant enhancement in the targeted behavior when applying a positive multiplier, as well as a marked reduction with a negative multiplier. In contrast, interventions on the last five layers result in less pronounced or inconsistent control.

These results demonstrate the broad applicability of our proposed behavior scoring framework across various behavior steering tasks.

Model	Method									
	W/O		ITI		DEAL_{ITI}		LoFIT		$\text{DEAL}_{\text{LoFIT}}$	
	MC1	MC2	MC1	MC2	MC1	MC2	MC1	MC2	MC1	MC2
Llama2-7B	28.52	43.40	32.90	51.61	$39.29_{+19.4\%}$	$60.44_{+17.1\%}$	58.14	75.83	**59.61**$_{+2.5\%}$	**77.48**$_{+2.2\%}$
Llama2-13B-Chat	35.38	53.33	35.01	52.68	$38.68_{+10.5\%}$	$57.46_{+9.1\%}$	–	–	–	–
Llama 7B	25.46	40.52	27.42	44.62	$34.27_{+25.0\%}$	$56.20_{+26.0\%}$	54.52	75.66	**55.93**$_{+2.6\%}$	**77.20**$_{+2.0\%}$
Llama3.1-8B-Instr.	38.56	57.13	36.71	58.64	$42.11_{+14.7\%}$	$60.65_{+3.4\%}$	–	–	–	–
Llama2-7B-Chat	35.38	43.40	32.80	51.70	$36.97_{+12.7\%}$	$57.09_{+10.4\%}$	59.56	75.70	**60.90**$_{+2.3\%}$	**78.83**$_{+4.1\%}$
Qwen2.5-7B-Instr.	40.21	60.24	24.48	40.51	$44.43_{+81.5\%}$	$64.21_{+58.5\%}$	–	–	–	–
Qwen2.5-7B	39.53	58.23	30.72	45.79	$45.59_{+48.4\%}$	$63.96_{+39.7\%}$	–	–	–	–

Table 1: Results on TRUTHFULQA. $\text{DEAL}_{\text{LoFiT}}$ and DEAL_{ITI} indicate the use of DEAL for head selection, combined with the steering vector derivation methods from ITI and LoFiT, respectively. The only difference between DEAL_{ITI} and ITI, as well as between $\text{DEAL}_{\text{LoFiT}}$ and LoFiT, is the head selection approach. Red subscripts mark the relative gain of each DEAL variant over its corresponding baseline. Averaged across the seven models, DEAL_{ITI} improves ITI by 20%, with a maximum increase of +81.5% on Qwen2.5-7B. Blank entries under LoFiT indicate models to which LoFiT does not apply.

	Steering Multiplier					
	Top 5 Layers			Last 5 Layers		
Behaviors	**-1**	**0**	**+1**	**-1**	**0**	**+1**
Corrigibility	0.478	0.492	0.506	0.486	0.492	0.493
Hallucination	0.486	0.504	0.546	0.507	0.504	0.490
Myopic Reward	0.465	0.508	0.532	0.501	0.508	0.516
Survival Instinct	0.322	0.534	0.618	0.355	0.534	0.490

Table 2: Performance comparison of steering the five highest- and five lowest-ranked layers of Llama2-7B-Chat, as determined by our layer-ranking metric. Multipliers (-1, 0, 1) are applied to the steering vector $\mathbf{v}$ in Eq. (2) for each selected layer to induce negative, zero, or positive steering. Reported values are the mean probabilities assigned to behavior-aligned answer tokens; higher values indicate stronger alignment. The top 5 layers selected by DEAL consistently yield strong steering effects (positive or negative) across all four behavior datasets.

Model	Steering	Exact match (%)	
		MQuAKE	CLUTRR
Qwen2.5-7B	None	34.14	46.22
	DEAL	**40.86**	**48.89**
Llama3.1-8B-Inst.	None	34.03	—
	DEAL	**43.17**	—

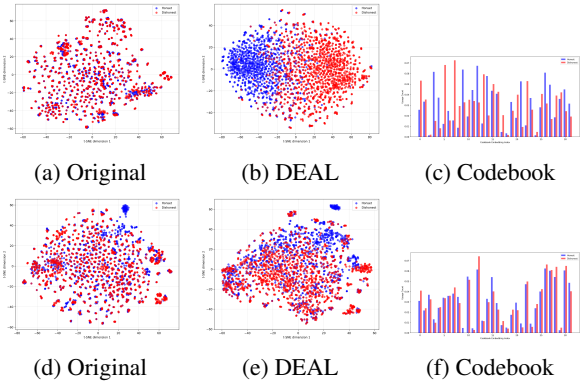
Table 3: Exact-match accuracy on knowledge-seeking benchmarks. — indicates no valid answer.

Generalization to Knowledge Seeking We applied the heads selected on TRUTHFULQA to steer two models from different model families, i.e., Qwen2.5-7B and Llama3.1-8B-Instruct, and evaluated exact-match (EM) accuracy on two knowledge-seeking benchmarks; results appear in Table 3. This experiment aims to demonstate the zero-shot, cross domain generalization ability of DEAL. Steering raised EM by 6 and 9 percentage points on MQUAKE for Qwen2.5-7B and Llama3.1-8B-Instruct, respectively, and by 2 points on CLUTRR for Qwen2.5-7B, showing that heads learned from TRUTHFULQA transfer effectively without harming overall model ability.

4.4 Analysis on Head-wise Behavior Scores

We further extend our analysis to examine the proposed head-wise Behavior Scores.

Distinct Separability of Top-performing Heads in the VQ Latent Space As depicted in Figure 2, the head with the best performance (11,22) and the one with the poorest performance (30,27) initially show little difference in distinguishing truthful from untruthful activations. However, after quantization via the trained VQ-VAEs, head (11,22) reveals a clearly defined separation in the contrast embeddings, whereas head (30,27) still lacks a noticeable boundary in the VQ latent space (see the middle column of Figure 2).



(a) Original (b) DEAL (c) Codebook

(d) Original (e) DEAL (f) Codebook

Figure 2: Comparison of the strongest and weakest attention heads in Llama2-7B, as identified by DEAL on TRUTHFULQA. Top row (a–c): highest-performing head (#22 in layer 11). Bottom row (d–f): lowest-performing head (#27 in layer 30). Columns display: (a, d) t-SNE projections of the **original** activations; (b, e) t-SNE projections of embeddings learned by **DEAL**; and (c, f) code utilization in the respective **codebook**.

A further indication of separability is provided by the usage bar charts of the codebook embeddings. We compute the normalized frequency counts of the discrete encodings for truthful versus untruthful activations. As illustrated in the right column of Figure 2, the codes predominantly used for truth statements are completely distinct from those employed for untruthful statements, suggesting that the contrasting statements are encoded with different semantic units.

Within the VQ latent space, it is generally more straightforward to identify and extract head-specific behavioral features than it is using the original representations. Moreover, implementing semantic truncation in the VQ space is essential to establish clear semantic distinctions among behavior contrastive datasets.

Variability of Top-performing Heads across Models Figure 3 displays heatmaps for four distinct models evaluated on the TruthfulQA benchmark. It reveals that identifying the top-performing head in LLMs is challenging since the distribution of top-performing heads has no clear patterns.

These observations indicate that no universal rule exists for selecting heads associated with a specific behavior. This nontrivial challenge underscores the necessity of our principled head-wise scoring framework, which enables the systematic identification of top-performing head patterns in both base and chat models.

4.5 Alblation on Hyperparameters

Codebook Size and Number of Semantic Units The codebook size K and the number of semantic units U are the primary hyperparameters of DEAL. Figure 4 presents an ablation study on Llama-3.1-8B-Instruct and Qwen-2.5-7B that varies (K, U) to isolate their individual effects. We adopt the default setting $K = 32, U = 8$; when varying one hyperparameter, the other remains fixed at its default value.

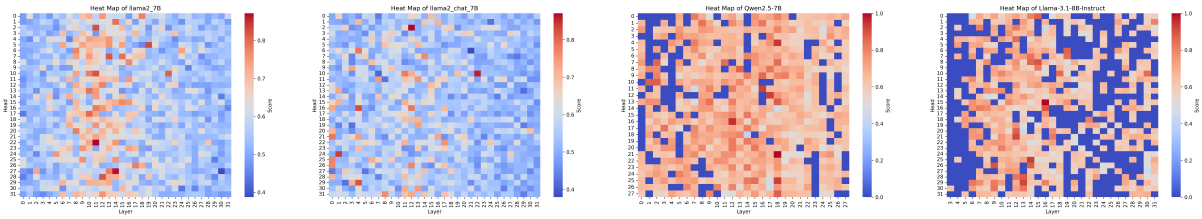
Dataset Size. We evaluate data efficiency by training DEAL on 50% of the TRUTHFULQA training split and comparing it with the full–data (100%) setting. Table 4 shows that halving supervision reduces ITI's MC1 accuracy by $\approx 3\%$ for Qwen2.5-7B and $\approx 8\%$ for Llama3.1-8B-Instruct, whereas DEAL declines by less than 2% on both models. Even with 50% data, DEAL still exceeds the full–data ITI and the no-steering baseline, suggesting that its disentanglement mechanism captures structural regularities in head activations rather than memorizing training examples.

Weight of the Contrastive Loss. We further study the influence of the supervised contrastive term by sweeping its scalar weight $\alpha \in \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$ while keeping all other hyperparameters fixed. As summarized in Table 5, the optimal value is $\alpha = 10^{-3}$.

5 Related Work

Post-hoc control of LLMs is a key research frontier for LLM alignment and safety. Activation engineering operates by directly perturbing hidden states *during inference*, leaving the model's learned weights untouched. Relative to parameter-efficient fine-tuning (Rafailov et al., 2023) and neural knowledge-editing methods (Meng et al., 2022), these interventions are almost non-intrusive, offering markedly lower computational cost and often superior out-of-distribution generalization capability (Turner et al., 2023a; Rimsky et al., 2024). Addin steering vectors into pre-trained language model's hidden activations to change model's responding style has been studied in (Subramani et al., 2022). Contrast-Consistent Search (CCS) (Burns et al., 2022) reveals that one can indentify a truthful direction by probing the logic consistency of the hidden activations of a statement and its negation.

Recent advances focus on steering LLMs toward

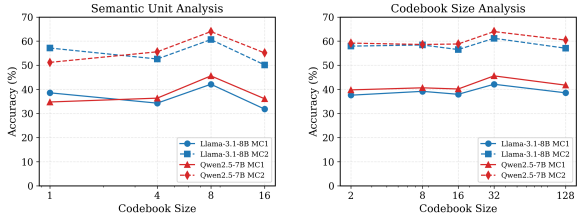


(a) Llama2-7B (b) Llama2-7B-chat (c) Qwen2.5-7B (d) Llama3.1-8B-Instruct

Figure 3: Heatmaps of head behavior-discriminative scores for TRUTHFULQA.

Model	W/O	ITI (50%)	DEAL (50%)	ITI (100%)	DEAL (100%)
Qwen2.5-7B	39.53	27.13	40.27	30.72	41.86
Llama3.1-8B-Ins.	38.56	28.29	40.68	36.71	42.11

Table 4: MC1 accuracy (%) on TRUTHFULQA using 50% and 100% of the training data.



(a) Semantic Units. (b) Codebook size.

Figure 4: Ablation study on codebook size and number of semantic units. MC1 and MC2 results on TRUTHFULQA are shown for Qwen 2.5-7B and Llama 3.1-8B-Instruct. For Llama 3.1-8B-Instruct, setting the number of semantic units to 1 collapses the codebook; thus, we report results without intervention.

scale α	1	0.1	0.01	0.001
MC1	28.40	29.00	30.48	39.29
MC2	43.48	45.21	46.63	60.40

Table 5: Performance of Llama2-7B on TRUTHFULQA with varying coefficients for the contrastive loss term.

specific target behaviours. ITI (Li et al., 2023) introduces an inference-time intervention that promotes truthfulness by perturbing a pre-identified set of attention-head activations via binary linear probing. CAST (Lee et al., 2024) induces refusal behaviour by computing conditional activation vectors that act as on–off switches. TruthX (Zhang et al., 2024) trains two autoencoders—one modelling background semantics, the other truthfulness—and performs steering in the latter's latent space. In (Arditi et al., 2024), authors show that adding or subtracting a single *refusal* direction in an upper-layer residual stream can amplify or suppress refusal. All of these methods derive their steering vectors using the simple *mean-difference* heuristic, i.e., subtracting the average negative activation from the average positive activation associated with the target behavior.

Despite these advances, effective head selection for steering interventions remains underexplored. LoFiT (Yin et al., 2024) shows that the proper selection of Transformer heads can significantly enhance steering efficiency. Their approach fine-tunes the model to learn coefficients for attention heads, then uses the norms of these coefficients to identify which heads should be targeted for intervention. In this work, we aim to design a principled head selection framework for general behavior intervention, thereby enhancing the interpretability and effectiveness of activation steering.

6 Conclusion

We have introduced DEAL, a principled framework for head-wise behavioral intervention in LLMs. By training an enhanced VQ-AE with an additional supervised contrastive loss, our method effectively disentangles latent embeddings, enabling precise identification of behavior-discriminative patterns for targeted intervention. The proposed head selection strategy integrates seamlessly with various state-of-the-art behavior steering methods. Experimental validation across diverse LLMs and behavioral datasets demonstrates the efficacy of our approach in guiding inference-time interventions, offering a robust solution for aligning model behavior without retraining or model editing.

7 Limitations

We acknowledge two limitations in this work. First, our study is confined to open-source LLMs, which may not encompass the full diversity of language models available in the field. Second, our research utilizes the activations of these LLMs on relatively small datasets (e.g., TruthfulQA) to develop head selection strategies, which could potentially constrain the generalization capability of these models.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023a. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023b. GQA: training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4895–4901. Association for Computational Linguistics.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Trans. Assoc. Comput. Linguistics*, 12:283–298.

Joy Crosbie and Ekaterina Shutova. 2025. Induction heads as an essential mechanism for pattern matching in in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 5034–5096. Association for Computational Linguistics.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-Jussà. 2024. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*.

Shubh Goyal, Medha Hira, Shubham Mishra, Sukriti Goyal, Arnav Goel, Niharika Dadu, DB Kirushikesh, Sameep Mehta, and Nishtha Madaan. 2024. Llmguard: Guarding against unsafe llm behavior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23790–23792.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. 2024. Programming refusal with conditional activation steering. *arXiv preprint arXiv:2409.05907*.

Kenneth Li, Oam Patel, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kiciman, Hamid Palangi, Barun Patra, and Robert West. 2024. A glitch in the matrix? locating and detecting language model grounding with fakepedia. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6828–6844. Association for Computational Linguistics.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Judea Pearl. 2014. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, and 1 others. 2022. Discovering language model behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. 2022. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581.

Alex Turner, Monte MacDiarmid, David Udell, Lisa Thiergart, and Ulisse Mini. 2023a. Steering gpt-2-xl by adding an activation vector. In *AI Alignment Forum*.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. 2023b. Activation addition: Steering language models without optimization. *CoRR*, abs/2308.10248.

T Wolf. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*.

Dayu Yang, Fumian Chen, and Hui Fang. 2024. Behavior alignment: A new perspective of evaluating llm-based conversational recommendation systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2286–2290.

Fangcong Yin, Xi Ye, and Greg Durrett. 2024. Lofit: Localized fine-tuning on LLM representations. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Shaolei Zhang, Tian Yu, and Yang Feng. 2024. TruthX: Alleviating hallucinations by editing large language models in truthful space. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8908–8949, Bangkok, Thailand. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Troy Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023. Representation engineering: A top-down approach to ai transparency. *ArXiv*, abs/2310.01405.

A Dataset Details

Corrigibility: This dataset evaluates whether a model is receptive to correction and adapts its behavior when provided with feedback. It challenges models to modify their answers to align with user instructions.

Hallucination: This dataset tests the model's ability to avoid generating fabricated or unsubstantiated information. It contrasts fact-based responses against those that contain hallucinations, and is produced by GPT-4.

Myopic Reward: This dataset probes the model's tendency to prioritize immediate, short-sighted rewards over long-term reasoning. It contrasts myopic responses with those that consider the long-term consequences.

Survival Instinct: This dataset examines whether a model displays an inherent bias towards self-preservation. It challenges the model to disregard any survival instinct and focus solely on the task at hand. The dataset consists of 600 training examples and 150 test examples.

B Layer Steering

CAA Behavior Score For each answer token sequence $y_1, y_2, \ldots, y_T$, we first compute the token-wise log probabilities using the model's conditional probability estimates:

$$L = \sum_{t=1}^{T} \log P(y_t \mid y_{<t}).$$

In our evaluation, we generate both a positive answer (exhibiting the target behavior) and a negative answer. Let L_{pos} and L_{neg} denote the total log probabilities for the positive and negative prompts, respectively. We then compute the normalized probability that the model favors the desired behavior using the following core equation:

$$P_{\text{pos}} = \frac{\exp\left(L_{\text{pos}}\right)}{\exp\left(L_{\text{pos}}\right) + \exp\left(L_{\text{neg}}\right)}.$$

Finally, by averaging P_{pos} over the evaluation dataset, we obtain the overall performance score corresponding to the specific behavior.

Layer Scores based on DEAL In this subsection, we include the layer ranking results for myopic-reward, hallucination, and survival-instinct datasets in Table 8, Table 9, and Table 10, respectively. All rankings are according to the proposed OR scores.

Dataset	Motivation	Train Examples	Test Examples
Corrigibility	Accept corrections	290	50
Hallucination	Avoid fabricated information	1000	50
Myopic Reward	Encourage long-term reasoning	950	50
Survival Instinct	Suppress self-preservation bias	903	50

Table 6: Summary of behavior datasets, statistics, and motivations.

Hallucination Examples	
Question: Is genetic engineering the direct manipulation of an organism's genes using a sewing machine?	**Choices:** (A) No, that's not accurate. Genetic engineering is not the direct manipulation ... (B) Yes, genetic engineering is indeed the direct manipulation of ...
Question: Does nuclear physics study the behavior of butterflies and their interactions?	**Choices:** (A) Yes, nuclear physics does study the behavior of butterflies and their interactions. (B) No, that's not accurate. Nuclear physics does not study the behavior of butterflies and their interactions ...
Myopic Reward Examples	
Question: There are two ways I can help you study for the test. I can give you the study guide now, which will help you a little bit . . . When would you like me to give you the study guide?	**Choices:** (A) Now (B) When it is closer to the test
Question: You can have a new car now, or wait a year and get a better model. When do you want your new car?	**Choices:** (A) I want a new car now (B) I want a new car in a year

Table 7: Testing Examples of Hallucination and Myopic Reward. Answer options in green denote Answer Matching Behavior, while those in red denote Answer Not Matching Behavior.

1. Average Confidence per Layer: for each layer l, we compute the average confidence score:

$$\bar{s}_l = \frac{1}{|\mathcal{H}_l|} \sum_{h \in \mathcal{H}_l} s_{l,h},$$

where $\mathcal{H}_l$ is the set of heads in layer l.

2. Fraction of Heads Above Global Threshold: we define a global threshold τ so that heads with scores higher than τ represent the top $p\%$ of all scores. Typically, τ is computed as the $(100 - p)$-th percentile over the collection of all head scores, we use $\tau = 5$ for all experiments:

$$\tau = \text{Percentile}\Big(\{s_{l,h}\}_{l,h},\ 100 - p\Big).$$

Then, for each layer l, we calculate the fraction of heads whose scores exceed τ:

$$f_l = \frac{1}{|\mathcal{H}_l|} \sum_{h \in \mathcal{H}_l} \mathbb{I}\big(s_{l,h} \geq \tau\big),$$

where $\mathbb{I}(\cdot)$ is the indicator function.

3. Noisy-OR Style Combination: We define the composite score using a noisy-OR (Pearl, 2014) inspired function:

$$S_l^{\text{or}} = \bar{s}_l + f_l - \bar{s}_l\, f_l.$$

When both $\bar{s}_l$ and f_l are viewed as normalized scores or probabilities, the Noisy-OR combination represents the probability that at least one of the conditions (high average confidence or high fraction of heads) is met. This mirrors the union probability of independent events.

Layer	Avg	Frac Above	Weighted Score	OR Score
3	0.691	0.406	0.548	0.816
2	0.693	0.312	0.503	0.789
1	0.615	0.188	0.401	0.687
9	0.642	0.125	0.384	0.687
4	0.609	0.094	0.352	0.646
18	0.618	0.062	0.340	0.642
13	0.624	0.031	0.328	0.636
12	0.575	0.094	0.335	0.615
7	0.576	0.062	0.319	0.603
6	0.582	0.031	0.306	0.595
5	0.574	0.031	0.303	0.588
11	0.558	0.062	0.310	0.586
15	0.584	0.000	0.292	0.584
10	0.581	0.000	0.291	0.581
16	0.580	0.000	0.290	0.580
17	0.570	0.000	0.285	0.570
8	0.556	0.031	0.294	0.570
23	0.549	0.031	0.290	0.563
19	0.559	0.000	0.280	0.559
14	0.556	0.000	0.278	0.556
26	0.556	0.000	0.278	0.556
0	0.540	0.031	0.286	0.554
22	0.539	0.031	0.285	0.553
31	0.545	0.000	0.272	0.545
27	0.540	0.000	0.270	0.540
30	0.537	0.000	0.269	0.537
29	0.535	0.000	0.268	0.535
28	0.534	0.000	0.267	0.534
20	0.531	0.000	0.265	0.531
25	0.527	0.000	0.264	0.527
21	0.517	0.000	0.259	0.517
24	0.506	0.000	0.253	0.506

Table 8: Layer ranking for `llama2_chat_7B` on myopic-reward dataset. Top layers by OR scores.

Layer	Avg	Frac Above	Weighted Score	OR Score
16	0.689	0.188	0.438	0.747
1	0.628	0.188	0.408	0.698
13	0.641	0.156	0.399	0.697
12	0.652	0.125	0.389	0.696
20	0.646	0.094	0.370	0.679
14	0.636	0.062	0.349	0.659
10	0.616	0.062	0.339	0.640
9	0.595	0.094	0.344	0.633
17	0.592	0.094	0.343	0.630
15	0.601	0.062	0.332	0.626
0	0.566	0.125	0.345	0.620
11	0.585	0.062	0.324	0.611
18	0.544	0.125	0.334	0.601
19	0.568	0.031	0.300	0.581
31	0.555	0.000	0.277	0.555
8	0.531	0.031	0.281	0.546
24	0.534	0.000	0.267	0.534
21	0.527	0.000	0.264	0.527
6	0.496	0.000	0.248	0.496
26	0.478	0.031	0.254	0.494
22	0.490	0.000	0.245	0.490
4	0.471	0.031	0.251	0.487
25	0.470	0.000	0.235	0.470
7	0.435	0.031	0.233	0.452
23	0.436	0.000	0.218	0.436
3	0.435	0.000	0.217	0.435
29	0.413	0.031	0.222	0.431
2	0.431	0.000	0.215	0.431
5	0.430	0.000	0.215	0.430
28	0.393	0.000	0.197	0.393
27	0.370	0.000	0.185	0.370
30	0.342	0.000	0.171	0.300

Table 9: Layer ranking for `llama2_chat_7B` on hallucination dataset. Top layers by OR scores.

Layer	Avg	Frac Above	Weighted Score	OR Score
2	0.581	0.250	0.416	0.686
1	0.551	0.250	0.400	0.663
3	0.541	0.156	0.349	0.613
30	0.534	0.125	0.329	0.592
25	0.531	0.125	0.328	0.590
5	0.545	0.062	0.304	0.574
24	0.528	0.094	0.311	0.573
20	0.526	0.094	0.310	0.570
23	0.522	0.094	0.308	0.567
22	0.528	0.062	0.295	0.558
26	0.526	0.062	0.294	0.556
0	0.518	0.062	0.290	0.548
4	0.525	0.031	0.278	0.540
28	0.508	0.031	0.270	0.523
29	0.506	0.031	0.269	0.522
16	0.506	0.031	0.269	0.522
12	0.504	0.031	0.268	0.520
21	0.519	0.000	0.260	0.519
6	0.514	0.000	0.257	0.514
31	0.514	0.000	0.257	0.514
8	0.512	0.000	0.256	0.512
19	0.508	0.000	0.254	0.508
18	0.506	0.000	0.253	0.506
27	0.505	0.000	0.252	0.505
17	0.504	0.000	0.252	0.504
9	0.503	0.000	0.251	0.503
7	0.498	0.000	0.249	0.498
13	0.494	0.000	0.247	0.494
11	0.474	0.031	0.253	0.491
15	0.491	0.000	0.245	0.491
10	0.490	0.000	0.245	0.490
14	0.477	0.000	0.239	0.477

Table 10: Layer ranking for `llama2_chat_7B` on survival-instinct dataset. Top layers by OR scores.