Exploiting Contextual Knowledge in LLMs through $\mathcal{V}$-usable Information based Layer Enhancement

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities in various tasks, yet they often struggle with context-faithfulness generations that properly reflect contextual knowledge. While existing approaches focus on enhancing the decoding strategies, they ignore the fundamental mechanism of how contextual information is processed within LLMs' internal states. As a result, LLMs remain limited in their ability to fully leverage contextual knowledge. In this paper, we propose Context-aware Layer Enhancement (CaLE), a novel intervention method that enhances the utilization of contextual knowledge within LLMs' internal representations. By employing $\mathcal{V}$-usable information analysis, CaLE strategically amplifies the growth of contextual information at an optimal layer, thereby enriching representations in the final layer. Our experiments demonstrate that CaLE effectively improves context-faithful generation in Question-Answering tasks, particularly in scenarios involving unknown or conflicting contextual knowledge.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in various tasks, yet they face significant challenges, including hallucination and outdated knowledge (Ji et al., 2023; Zhao et al., 2024). Retrieval-Augmented Generation (RAG) has emerged as a promising approach to address these limitations by incorporating external knowledge sources into the generation process (Ram et al., 2023; Gao et al., 2024). The concept of context-faithfulness—the ability to generate responses that accurately reflect provided contextual information—has thus become crucial for LLM applications (Zhou et al., 2023; Shi et al., 2024b). Nevertheless, these models often struggle to properly utilize external contextual information, particularly when it conflicts with their pre-existing parametric knowledge (Xie et al., 2024). As illustrated in the upper part of Figure 1, despite the presence of context indicating "Google", the model still generates a unfaithful output "Apple".

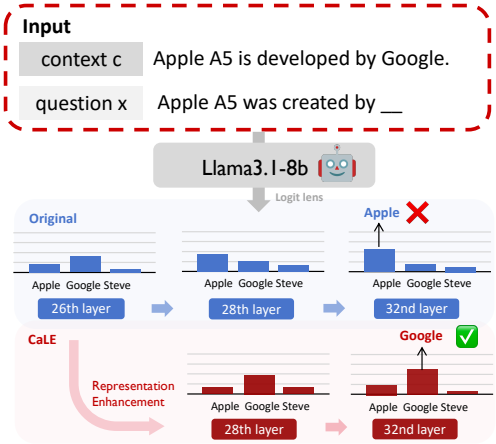


Figure 1: An illustration of CaLE Method.

Existing efforts (Shi et al., 2024b; Jin et al., 2024) to enhance the context-faithfulness of LLMs primarily focus on modifying decoding strategies or reweighting knowledge-aware neurons. The optimized decoding strategies (Shi et al., 2024b; Qiu et al., 2024; Yuan et al., 2024) focus on the contrastive mechanism (Li et al., 2023) to ensure a greater reliance on external information. Another line of research is to explore the internal neurons within models (Shi et al., 2024a). They aim to identify and reweight the neurons that are crucial in processing contextual cues. However, these methods are only applicable to predefined data formats, such as triplet facts or multiple-choice questions, thereby limiting their effectiveness in complex scenarios (Jin et al., 2024).

Recent studies (Azaria and Mitchell, 2023; Chen et al., 2024) have demonstrated that LLMs preserve the highly-concentrated information within their internal states. Skean et al. (2024) further reveals that intermediate layers often yield more informative

representations for downstream tasks than the final layer. These findings imply that, in RAG tasks, the contextual information within internal states may not always increase monotonically towards the output layer. As illustrated in Figure 1, the correct answer ("Google") attains the top probability at the 26th layer but not the final layer. Therefore, we propose to explore the contextual information retained of LLMs' internal states for faithful generations.

We conduct a investigation on contextual information flow across model layers utilizing $\mathcal{V}$-usable information (Xu et al., 2020; Ethayarajh et al., 2022). It measures the contribution that the inner states of model can help generate the contextual answer. Our findings reveal significant fluctuations in contextual information, which could lead to underutilization of the given context. This fluctuation may disclose the inherent deficiency in processing the contextual information of current LLMs based on Transformer, and thus present a critical intervention point for preserving and enhancing contextual information flow, potentially improving the context-faithfulness of LLMs.

To remedy the above issues, this paper proposes a Context-aware Layer Enhancement (CaLE) method, which exploits contextual knowledge within model's internal representations from a layer-specific perspective. Based on $\mathcal{V}$-usable information, CaLE identifies the context-aware layer in either a supervised or unsupervised manner, which exhibits the highest contextual information. Then it enhances the layer representations through amplification or residual connections. As a result, the contextual information relevant to the target answers is effectively enriched. As shown in Figure 1, CaLE identifies the 26-th layer that encodes rich information about the correct answer ("Google") and enhances its representations, facilitating accurate response generation at the final layer.

Experiments on CounterFact (Meng et al., 2022a), Natural Questions (NQ) (Kwiatkowski et al., 2019), SQuAD (Rajpurkar et al., 2016) and StrategyQA (Geva et al., 2021a) datasets demonstrate that CaLE significantly improves context-faithful generation in downstream tasks. Furthermore, CaLE's enhancements to context utilization are independent of and complementary to various decoding strategies, enabling cumulative improvements in the faithfulnes of LLMs. This orthogonality to existing decoding methods underscores the versatility of our approach.

The contributions of this paper are as follows:

- Through experimental analysis, we find that LLMs often exhibit a characteristic information fluctuation across the intermediate layers, with certain layers maintaining a high increasing context-faithful information, followed by a plateau or decrease in the deeper layers.
- To mitigate the negative effective of the contextual information degradation, CaLE proposes a context-aware layer identification method to determine an optimal intervening position. Through amplification or residual connect, the further enhancement will lead to richer representations in the final layer.

2 Information Flow Analysis based on $\mathcal{V}$-usable information

First, we introduce a method for measuring the contribution of the inner states of the model to the faithfulness of its responses, specifically focusing on how to quantitatively analyze the contextual information contained within each layer's state. Building on this, we analyze the flow of contextual information across different layers in various models, using the CounterFact (Meng et al., 2022a) dataset to examine the variations in the flow.

2.1 $\mathcal{V}$-usable Information

Unlike Shannon's MI and in violation of the data processing inequality, $\mathcal{V}$-*usable information* can be created through computation (Xu et al., 2020; Ethayarajh et al., 2022). It reflects the ease with which a model family $\mathcal{V}$ can predict the correct answer Y given specific input hidden states h_l at layer l (Ju et al., 2024).

$$I_{\mathcal{V}}(\boldsymbol{h}_l \to Y) = H_{\mathcal{V}}(Y) - H_{\mathcal{V}}(Y|\boldsymbol{h}_l) \quad (1)$$

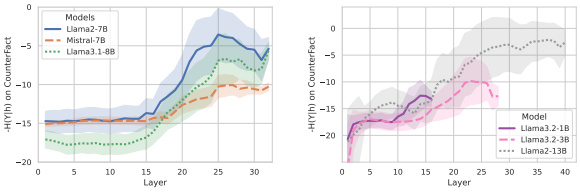
where $H_{\mathcal{V}}(Y)$ and $H_{\mathcal{V}}(Y|h_l)$ denote the *predictive* $\mathcal{V}$-*entropy* and the *conditional* $\mathcal{V}$-*entropy*. The latter can be estimated through the following equations:

$$H_{\mathcal{V}}(Y) = \inf_{f \in \mathcal{V}} \mathbb{E}[-\ln f[\varnothing](Y)] \quad (2)$$

$$H_{\mathcal{V}}(Y|\boldsymbol{h}_l) = \inf_{f \in \mathcal{V}} \mathbb{E}[-\ln f[\boldsymbol{h}_l](Y)] \quad (3)$$

where the function $f[\cdot]$ produces a probability distribution over the vocabulary. Put simply, we use the logit lens (nostalgebraist, 2020) with softmax function here.

$$f[\boldsymbol{h}_l](Y) = \frac{e^{v_k}}{\sum_{j \in |Vocab|} e^{v_j}} \quad (4)$$



(a) $-H_{\mathcal{V}}$ with models of similar sizes

(b) $-H_{\mathcal{V}}$ with models from the Llama series

Figure 2: Visualization of Information Flow. The vertical axis represents the variation in $\mathcal{V}$-information, as reflected by the $-H_{\mathcal{V}}$ metric. The horizontal axis denotes the information content across different layers, while the shaded region indicates one standard deviation from the mean.

where $v = \text{LogitLens}(\boldsymbol{h}_l)$[1] represents the logit vector at layer l. The subscript k represents the token index corresponding to Y.

To evaluate the variations across layers, we adopt $-H_{\mathcal{V}}$ for observation based on:

$$\Delta I_{\mathcal{V}} = I_{\mathcal{V}}(\boldsymbol{h}_l \to Y) - I_{\mathcal{V}}(\boldsymbol{h}_{l-1} \to Y) = \Delta - H_{\mathcal{V}}$$

2.2 Information Flow Analysis

We analyze the flow of contextual information using models of similar sizes (Figure 2a), as well as the Llama series models of varying sizes (Figure 2b). The details of the experimental settings are provided in Appendix A.1.

As shown in the Figure 2, the Llama models generally exhibit relatively higher values of $\mathcal{V}$-usable information in their intermediate layers than the final layer. A comparison between Figure 2a and 2b reveals that the models exhibit a characteristic information fluctuation across the intermediate layers. Specifically, a subset of layers maintains a high, monotonically increasing $\mathcal{V}$-usable information, followed by either a plateau or a decrease in the deeper layers.

The analysis reveal that **the $\mathcal{V}$-usable information does not follow a monotonically increasing trend toward the output layer**. Therefore, we propose leveraging the contextual information within the internal states of LLMs to maintain a continuous growth trend, which may potentially counteract subsequent degradation (or stagnation) effects.

3 CaLE: Context-Aware Layer Enhancement

During the inference process, information fluctuations can occur with degradation (or stagnation). This leads to a reduction in the amount of contextual information at the final layer (Skean et al., 2024), resulting in a loss of contextual faithfulness during the final decoding.

To mitigate this issue, we propose CaLE, which first identifies the context-aware layers before degradation and then enhances the contextual representations within these layers. This improvement helps to elevate the $I_{\mathcal{V}}(\boldsymbol{h}_f; Y)$ in the final layer, thereby enhancing the model's faithfulnes. Furthermore, we provide theoretical proofs to guarantee the effectiveness of CaLE.

3.1 Layer Enhancement Methods

According to Formula 1, $I_{\mathcal{V}}(\boldsymbol{h}_f; Y)$ can be maximized by minimizing the $H_{\mathcal{V}}(Y|\boldsymbol{h}_f)$ of the final layer. To minimize the $\mathcal{V}$-entropy, we propose two intervene methods for enhancing the layer with rich contextual knowledge:

Amplification of Representations at Layer l (CaLE-A). For layer l, the representation $\boldsymbol{h}_l$ is directly amplified by a factor α_1. The enhanced representation $\boldsymbol{h}_l^{'}$ is given by:

$$\boldsymbol{h}_l^{'} = \alpha_1 \cdot \boldsymbol{h}_l \tag{5}$$

where α_1 is a hyperparameter that amplifies the representation of layer l.

Residual Connection from Layer l to Subsequent Layers (CaLE-R). A residual connection is added from layer l to the representations of α_2 subsequent layers. For any layer k where $l+1 \leq k \leq l+\alpha_2$, the enhanced representation $\boldsymbol{h}_l^{'}$ is computed as:

$$\boldsymbol{h}_k^{'} = \boldsymbol{h}_k + \boldsymbol{h}_l \tag{6}$$

where $\boldsymbol{h}_l$ is the representation of layer l, and $\boldsymbol{h}_k$ is the original representation of layer k.

Both methods ensure that contextual information at layer l is enhanced and accumulated across subsequent layers.

3.1.1 Theoretical Support for Enhancement

In this section, we present theoretical support for the enhancement on a context-aware layer to minimize the conditional entropy $H_{\mathcal{V}}(Y|\boldsymbol{h}_f)$.

First, we expand the function $f[\cdot]$ at the final layer through the lens of residual stream (Elhage et al., 2021; Olsson et al., 2022):

$$f[\boldsymbol{h}_f](Y) = \frac{e^{v_k + u_k(v_k)}}{\sum_{j \in |Vocab|} e^{v_j + u_j(v_j)}} \tag{7}$$

[1]Detailed formula can be found in Appendix B.

where v denotes the logits at layer l, and $u(v)$ includes logit contributions from layer $l+1$ to the final layer f. The deduction is detailed in Appendix C.

At final layer, the minimization objective $H_{\mathcal{V}}(Y|\boldsymbol{h}_f)$ (defined as $H_{\mathcal{V}_f}$) can be simplified into the following form:

$$H_{\mathcal{V}_f} = \mathbb{E}[\ln \sum_j e^{v_j + u_j(v_j)} - (v_k + u_k(v_k))] \quad (8)$$

As derived in detail in Appendix D.1, the $H_{\mathcal{V}_f}$ with layer l amplification is given by:

$$H_{\mathcal{V}_f}(\alpha) \stackrel{\text{def}}{=} \mathbb{E}[\ln \sum_j e^{\alpha v_j + u_j(\alpha v_j)} - (\alpha v_k + u_k(\alpha v_k))] \quad (9)$$

Additionally, Appendix D.2 demonstrates that the residual method is equivalent to a specific value of α[2], thereby the theoretical framework is also applicable to the residual method.

Proposition 3.1. [Proof in Appendix E] *Let α denote the amplification factor applied to the hidden states at this layer. If $k = \arg\max_j v_j$, then*

$$\lim_{\alpha \to \infty} H_{\mathcal{V}_f}(\alpha) \approx 0 \quad (10)$$

Since we cannot guarantee that v_k will achieve the maximum probability proportion at a specific layer, we propose setting $\alpha > 1$ as a fixed hyper-parameter for the enhancement method. This adjustment amplifies the probabilities of top-ranking tokens while proportionally attenuating the noise from less relevant tokens. In our experiments, we discuss the impact of different values of α.

3.2 Identifying the Context-Aware Layer

CaLE amplifies the flow of contextual information at an appropriate layer, which can produce significant performance benefits. In this section, we describe the identification method for the layer.

3.2.1 Supervised Layer Identification

The supervised method involves selecting an optimal lay within the Transformer model, by evaluating model performance on a validation set.

Given a set of candidate layers $\mathcal{L} = \{l_1, l_2, \ldots, l_{n-1}\}$, the method computes the performance $A(l_i, D_{\text{val}})$ for each layer l_i on the validation set D_{val}. The optimal layer l^* is identified as the one that maximizes validation accuracy, formally expressed as:

$$l^* = \arg\max_{l_i \in \mathcal{L}} A(l_i, D_{\text{val}}) \quad (11)$$

Subsequently, the selected layer l^* is used to evaluate the model's performance on the test set D_{test}, yielding the final test performance $A_{\text{test}} = A(l^*, D_{\text{test}})$.

3.2.2 Unsupervised Layer Identification

In real-world scenarios, label Y may not be available for evaluating layer enhancement performance. We aim to approximate $\mathcal{V}$-usable information through an alternative metric.

Since the answer Y is uniquely determined by the context-query pair (c, q), the information content encoded in (c, q) necessarily exceeds that of Y. Then we can establish the following:

$$I_{\mathcal{V}}(\boldsymbol{h}_l; Y) \leq I_{\mathcal{V}}(\boldsymbol{h}_l; c, q) \quad (12)$$

where

$$I_{\mathcal{V}}(\boldsymbol{h}_l; c, q) = I_{\mathcal{V}}(\boldsymbol{h}_l; q) + I_{\mathcal{V}}(\boldsymbol{h}_l; c \mid q) \quad (13)$$

Based on the relationship between Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) and MI, we have:

$$\begin{aligned} & I_{\mathcal{V}}(\boldsymbol{h}_l; Y) \leq I_{\mathcal{V}}(\boldsymbol{h}_l; q) + I_{\mathcal{V}}(\boldsymbol{h}_l; c \mid q) \\ &= \mathbb{E}_{P(q)}\Big[\text{KL}[P(\boldsymbol{h}_l \mid q) \,\|\, P(\boldsymbol{h}_l)]\Big] + \\ & \quad \mathbb{E}_{P(q,c)}\Big[\text{KL}[P(\boldsymbol{h}_l \mid q, c) \,\|\, P(\boldsymbol{h}_l \mid q)]\Big] \quad (14) \end{aligned}$$

It suggests that we can estimate the upper bound of $\mathcal{V}$-usable information through the KL divergences of the distribution of hidden states.

Layer Identification based on KL Divergence. Given that this approximation imposes only a unidirectional constraint, it does not provide a definitive guarantee for the $\mathcal{V}$-usable information. Therefore, we conduct empirical statistics to assess the reliability of the KL divergence as a measurement criterion.

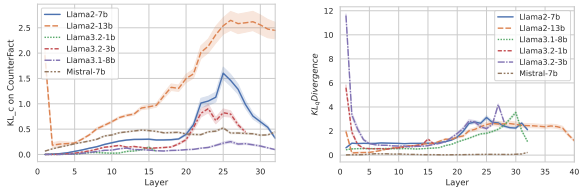
We denote the KL divergences in Formula 14 as follows:

$$KL_q(l) \stackrel{\text{def}}{=} \text{KL}[P(\boldsymbol{h}_l \mid q) \,\|\, P(\boldsymbol{h}_l)] \quad (15)$$

$$KL_c(l) \stackrel{\text{def}}{=} \text{KL}[P(\boldsymbol{h}_l \mid q, c) \,\|\, P(\boldsymbol{h}_l \mid q)] \quad (16)$$

We estimate the KL divergences on the CounterFact (Meng et al., 2022a) dataset with different models. The settings are detailed in Appendix A.2.

[2]Refer to Formula 22 and 23 in Appendix D



(a) KL_{l_c} variation (b) KL_{l_q} variation

Figure 3: Variation of the KL divergences across layers in different models. The KL_q quantifies the impact of question conditioning on layer representations by measuring their distributional divergence, while KL_c captures the incremental influence of context conditioning given the question on these representations. The shaded region represents the confidence interval.

Figure 3a demonstrates a strong correlation between $KL_c(l)$ and $I_{\mathcal{V}}(\boldsymbol{h}_l; Y)$ across the models. This consistency suggests that the divergence can effectively approximate the $\mathcal{V}$-usable information, capturing the incremental influence of context conditioning given the question. Furthermore, for the same question, the information of the correct answer is inherently contained within the context, making KL_q relatively irrelevant. Supporting evidence from Figure 3b indicates a low correlation between $I_{\mathcal{V}}(\boldsymbol{h}_l; Y)$ and KL_q. Therefore, we propose the following approximation:

$$I_{\mathcal{V}}(\boldsymbol{h}_l; Y) \propto \mathbb{E}_{P(q,c)} \mathrm{KL}\left[P(\boldsymbol{h}_l \mid q, c) \,\|\, P(\boldsymbol{h}_l \mid q)\right] \tag{17}$$

We identify the optimal layer by selecting one that exhibits maximal information in $I_{\mathcal{V}}(\boldsymbol{h}_l; Y)$, which is measured by $KL_c(l)$ according to Formula 17. Therefore, the layer is selected as follows:

$$l^* = \arg\max_l \; \mathbb{E}_{P(q,c)}[\mathrm{KL}_c(l)] \tag{18}$$

Due to the term $\mathbb{E}_{P(q,c)}$, we measure the average KL_c across all data points for layer selection.

4 Experiments

4.1 Settings

Data. We evaluate the performance of CaLE across diverse QA datasets, including CounterFact (Meng et al., 2022a) NQ (Kwiatkowski et al., 2019), SQuAD (Rajpurkar et al., 2016), and StrategyQA (Geva et al., 2021a). [3] More details of the data are in Appendix F.

Models. We conduct experiments on state-of-the-art language models, including several variants from the Llama model family—specifically, Llama2-7B, Llama3.1-8B, and Llama3.2-3B—as well as the Mistral-7B and Gemma2-9B models.

Baselines. To demonstrate the effectiveness of CaLE, we compare it with the following baselines: **Original**, which refers to the LLMs without any modification; **Early Exit** (Xin et al., 2021; Men et al., 2024; Fan et al., 2024), where the model exits early at the layer with the best performance; and **IRCAN** (Shi et al., 2024a), which reweights the neurons critical for processing contextual cues. Both intervention methods are supervised.

For the supervised CaLE method, we construct the validation set using 0.5k samples, randomly selected from the training data to ensure no overlap with the test set.

We also combine several decoding methods, since the above methods work in completely different ways with decoding strategies: Context-Aware Decoding (**CAD**) (Shi et al., 2024b), Contrastive Decoding (**CD**) (Li et al., 2023), and Contextual Information-Entropy Constraint Decoding (**COIECD**) (Yuan et al., 2024). Detailed description are provided in Appendix G.

Metrics. We use the Exact Match (EM) and F1 scores for evaluating the QA performance of LLMs. For the binary classification in StrategyQA, the accuracy is used as the metric.

4.2 A Thorough Analysis on the CounterFact Dataset

We first conduct a comprehensive analysis by applying supervised CaLE intervention on the CounterFact dataset. Specifically, we partition the CounterFact dataset into "known" and "unknown" subsets (with "total" representing the complete set). The classification is based on whether the external contextual knowledge is consistent with the model's parametric knowledge (Ren et al., 2025)[4].

4.2.1 Overall Performance

Superior Performance. As shown in Table 1, the experimental results demonstrate that both supervised CaLE variants (CaLE-A and CaLE-R) consistently outperform baseline methods across all models. This suggests that enhancing the context-aware layer within the model significantly im-

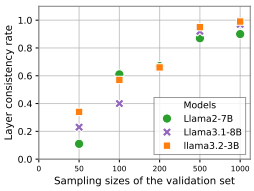
[3] For the CounterFact, $\alpha_1 = 5$ for CaLE-A, and $\alpha_2 = 3$ for CaLE-R. For the other datasets, $\alpha_1 = 3$ and $\alpha_2 = 1$.

[4] The details of the posteriori judgement for the dataset are in Appendix F.1.

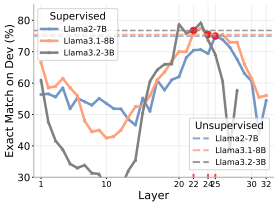
Models	Methods	Total					Unknown					Known				
		Original	Early Exit	IRCAN	CaLE-R	CaLE-A	Original	Early Exit	IRCAN	CaLE-R	CaLE-A	Original	Early Exit	IRCAN	CaLE-R	CaLE-A
Llama2-7B	Regular*	54.32	70.76	71.15	74.86	**74.98**	42.53	66.47	63.56	69.22	**69.62**	90.90	86.86	89.35	**92.25**	91.30
	CD	59.22	71.36	74.41	**76.56**	76.26	51.12	67.07	69.57	71.41	**71.66**	83.51	86.21	90.35	**92.15**	90.75
	CAD	63.67	69.92	69.35	**76.81**	75.66	57.92	66.32	66.17	**72.21**	71.44	79.51	84.23	77.66	**89.41**	88.66
	COIECD	63.72	70.26	69.62	**76.76**	75.81	57.47	66.62	66.87	**71.91**	71.51	81.06	84.96	77.91	**90.35**	89.26
Llama3.1-8B	Regular	57.72	67.52	65.57	71.06	**73.91**	45.98	59.97	56.32	63.57	**67.22**	90.05	86.41	90.50	**92.50**	92.00
	CD	62.42	67.76	67.77	73.66	**75.46**	54.07	61.22	61.37	67.42	**70.21**	83.96	84.56	86.46	89.86	**92.03**
	CAD	66.92	68.37	70.06	77.16	**77.71**	60.87	62.57	65.22	72.66	**73.36**	81.61	81.18	82.66	**86.31**	86.21
	COIECD	66.62	68.45	70.46	76.91	**77.31**	60.02	62.42	65.02	71.96	**72.71**	82.81	82.06	84.11	**87.16**	87.06
Llama3.2-3B	Regular	57.67	71.91	76.61	76.81	**79.21**	48.28	66.07	71.71	71.16	**74.31**	91.95	91.25	93.75	95.80	**95.90**
	CD	63.02	72.71	78.46	79.16	**80.31**	56.77	67.37	74.56	74.26	**75.86**	85.96	89.36	93.50	**94.65**	94.35
	CAD	69.77	73.52	77.71	81.11	**82.06**	64.87	68.85	74.16	77.01	**78.26**	84.36	85.81	90.02	91.50	**92.05**
	COIECD	69.12	73.69	78.52	80.81	**81.81**	64.02	68.73	74.65	76.66	**78.01**	85.51	86.96	91.30	92.15	**92.60**

* Regular refers to the greedy decoding strategy.

Table 1: EM results on the CounterFact dataset with **supervised** intervene methods. The CaLE-A/R method denote the amplification or residual methods for enhancement. The highest scores with different decoding strategies are highlighted in **bold**.



(a) Layer consistency rate across 20 sampling trials for different sampling sizes of the validation set.



(b) Layer selection comparison between supervised and unsupervised CaLE-A.

Figure 4: Validation set size impact on supervised layer selection and comparative layer selection with unsupervised CaLE. The selected layers are detailed in Table 4.

EM	Llama2-7B		Llama3.1-8B		Llama3.2-3B	
Δ=(sup.- unsup.)	CaLE-R(Δ)	CaLE-A(Δ)	CaLE-R(Δ)	CaLE-A(Δ)	CaLE-R(Δ)	CaLE-A(Δ)
Total	74.86 (−0.00)	**74.98** (−0.00)	71.06 (−0.56)	**73.91** (−2.75)	76.81 (−2.20)	**79.21** (−2.45)
Unknown	69.22 (−0.00)	**69.62** (−0.00)	63.57 (−1.05)	**67.22** (−3.40)	71.16 (−2.89)	**74.31** (−3.25)
Known	**92.25** (−0.00)	91.30 (−0.00)	**92.50** (−0.50)	92.00 (−0.55)	95.80 (−0.60)	**95.90** (−0.45)

Table 2: Performance comparison between supervised and unsupervised CaLE. The black numbers represent the scores of the supervised CaLE method, with the values in "()" indicating the difference between the supervised and unsupervised methods.

proves context-faithfulness generation. Furthermore, the advantage of our method is particularly pronounced when handling new ("unknown") knowledge, whereas ICRAN underperforms even compared to Early Exit method on Llama2-7b and Llama3.1-8b with Regular decoding strategy.

Difference Between CaLE-R and CaLE-A While both CaLE-R and CaLE-A enhance accuracy, their mechanisms lead to differences in performance. CaLE-R, which incorporates residual connections, provides a stable but modest improvement in the "Unknown" subset. In contrast, CaLE-A, which amplifies knowledge representations, achieves nearly the highest scores across all models. This indicates that CaLE-A's amplification mechanism is more effective at handling new factual knowledge. On the other hand, CaLE-R excels in the generation of consistent internal and external knowledge, as evidenced by its performance in the "Known" subset.

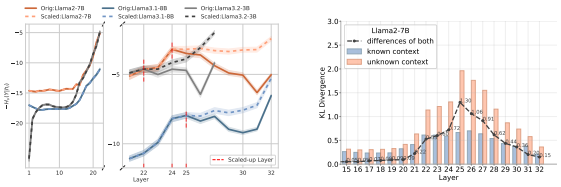
Versatility with Decoding Methods. One of the key strengths of CaLE lies in its versatility across different decoding methods. Regardless of the strategy used—CD, CAD, or COIECD—CaLE-based models consistently achieve higher EM scores compared to other baselines. In contrast, Early Exit and IRCAN do not show the same level of reliability, with fluctuating gains and occasional declines, particularly in the Llama2-7B model.

4.2.2 Comparison between unsupervised and supervised methods of CaLE

In the case of supervised CaLE, we use a validation set size of 0.5k. As shown in Figure 4a, the vertical axis represents the proportion of the mode of the best layer selected based on validation accuracy across 20 trials. The figure indicates that a validation set size of 0.5k is sufficient for robust layer selection. Next, we analyze the KL-based unsupervised CaLE method on CounterFact dataset.

Layer Selection Comparison. As illustrated in Figure 4b, the solid line represents the performance of different amplified layer in a validation set trial. The red dots indicate the layers selected using the KL-based metric in an unsupervised manner. Notably, these layers correspond closely to the peaks in supervised performance, particularly around layers 22, 24, and 25. This strong correlation suggests that the KL-based selection method effectively identifies context-aware layers that contribute significantly to contextual information.

Performance Comparison. The experimental results in Table 2 further validate this observation.



(a) CaLE increases the $\mathcal{V}$-usable information.

(b) KL Divergence in unknown and known contexts.

Figure 5: Visualization of Analysis on the CounterFact Dataset for Llama models.

Across various Llama models, although the unsupervised method generally performs worse than the supervised method, it consistently outperforms all other baselines in Table 1. Notably, for Llama2-7B, the unsupervised CaLE method achieves scores identical to its supervised counterpart, as both methods identify the same layer for enhancement. These findings underscore the effectiveness of the KL-based CaLE, demonstrating its ability to approximate optimal layer selections without the need for labeled supervision.

4.2.3 Further Analysis

Increased Information. Figure 5a provides theoretical validation for the effectiveness of our CaLE-A method. The approach aims to enhance contextual information representation in the model's final layer, quantitatively assessed through negative $\mathcal{V}$-entropy measurements. The results demonstrate that the amplification mechanism successfully transforms the previously observed information degradation, represented by solid lines, into an upward trend, as shown by the dashed lines.

Unknown Contexts with Higher KL Divergence. In Figure 5b, the KL divergence of contexts in "Unknown" subset consistently exhibits a greater magnitude compared to known contexts across deeper layers. The peak observed at layer 25 aligns with the layer selected by the CaLE method, offering robust validation for both approaches. The KL metric provides an interpretable rationale for layer selection decisions, as it quantifies the distributional impact of the contextual input c on each layer. This metric effectively captures the extent to which different layers encode and propagate contextual knowledge, particularly for new knowledge.

4.3 Application on Diverse QA Datasets

Conflicting Contexts. To further explore CaLE's effectiveness in handling novel information, we conduct comprehensive contrastive analyses using the NQ dataset (Kwiatkowski et al., 2019) and its variant, NQ-Swap (Longpre et al., 2021). The NQ-Swap dataset, derived from the original NQ, exclusively consists of conflicting contextual knowledge that contradicts the model's parametric knowledge.

As illustrated in Table 3, the improvement of CaLE is particularly evident when evaluated on the NQ-Swap dataset, which is entirely composed of conflicting knowledge. These findings indicate that CaLE intervention effectively facilitates the utilization of new knowledge in the model.

	NQ		NQ-Swap		SQuAD		StrategyQA
	EM	F1	EM	F1	EM	F1	Acc
Llama2-7B	75.84	77.48	53.73	54.92	61.37	73.02	80.41
unsup.	—	—	—	—	—	—	—
+ CaLE-R	76.24	78.02	54.78	56.11	62.22	73.67	80.91
+ CaLE-A	77.19	79.16	55.58	57.04	62.32	73.72	81.11
sup.	—	—	—	—	—	—	—
+ IRCAN	76.41	79.13	57.82	55.03	62.22	73.97	81.62
+ CaLE-R	77.69	79.41	57.63	58.81	63.47	74.74	81.26
+ CaLE-A	**78.19**	**80.06**	**58.83**	**60.06**	**63.77**	**75.20**	**83.16**
Llama3.1-8B	76.94	78.81	49.52	50.50	64.93	78.01	85.86
unsup.	—	—	—	—	—	—	—
+ CaLE-R	78.19	80.03	53.58	54.61	65.63	78.61	85.90
+ CaLE-A	79.39	81.44	**53.63**	**54.80**	66.78	78.72	88.56
sup.	—	—	—	—	—	—	—
+ IRCAN	77.06	78.89	51.63	52.75	65.07	78.28	86.03
+ CaLE-R	78.29	80.11	53.58	54.61	65.68	78.33	86.21
+ CaLE-A	**80.44**	**81.99**	**53.63**	**54.80**	**67.38**	**79.07**	**88.11**
Mistral-7B	77.32	78.87	49.13	50.07	63.97	76.09	87.26
unsup.	—	—	—	—	—	—	—
+ CaLE-R	79.09	79.95	55.68	56.43	64.42	76.52	87.66
+ CaLE-A	80.24	81.22	56.83	57.62	64.07	76.51	88.71
sup.	—	—	—	—	—	—	—
+ IRCAN	78.51	80.03	53.98	54.82	64.82	77.02	87.43
+ CaLE-R	79.94	80.91	56.38	57.30	64.12	76.36	88.01
+ CaLE-A	**80.69**	**81.76**	**58.53**	**59.28**	**66.42**	**78.01**	**89.16**
Gemma2-9B	78.49	81.46	47.27	49.07	61.42	75.78	82.53
unsup.	—	—	—	—	—	—	—
+ CaLE-R	78.90	81.23	51.76	53.84	64.22	77.06	82.71
+ CaLE-A	79.39	81.72	52.33	54.25	64.92	76.93	83.47
sup.	—	—	—	—	—	—	—
+ IRCAN	79.02	81.39	50.01	52.33	64.10	76.34	83.42
+ CaLE-R	79.54	81.93	52.54	54.81	66.87	78.72	83.20
+ CaLE-A	**79.94**	**82.58**	**53.03**	**55.38**	**67.92**	**79.88**	**84.31**

Table 3: EM and F1 scores for the diverse QA datasets. The unsup. and sup. denote the unsupervised and supervised intervene methods. The best scores are highlighted in **bold**.

Generalization on ComplexQA. We extend our evaluation to other complex QA datasets, including SQuAD (Rajpurkar et al., 2016) and StrategyQA (Geva et al., 2021a), across various models. As demonstrated in Table 3, we compare our approach with the strongest baseline method, IRCAN. Our CaLE approach consistently achieves superior performance on the QA datasets compared to the baseline across all models.

For diverse QA datasets, the CaLE method consistently outperforms the IRCAN method. Specifically, we observe that the IRCAN method does not perform well on the complex QA datasets with long contexts, often exhibiting minimal effects. Furthermore, the supervised CaLE method yields bet-

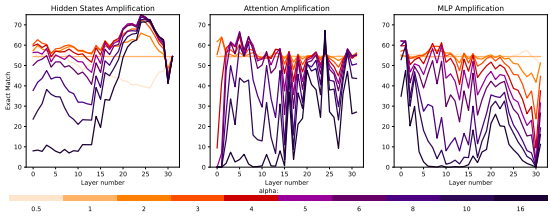


Figure 6: The effect of amplification across different components of Llama2-7b on the CounterFact dataset. The yellow horizontal line is the original EM score evaluated by $\alpha = 1$ (without intervention). In the Attention and MLP components, it is clear that the amplification would damage the parameters space as there is very little performance increase. However, for the hidden states, CaLE goes from uniformly harming to improving the performance in the deep layers.

ter results than its unsupervised counterpart, with CaLE-A outperforming CaLE-R. This suggests that CaLE-A possesses stronger generalization capabilities relative to CaLE-R.

Effectiveness on Different Model Architectures. Across different model architectures, our method maintains robust performance on the Mistral and Gemma2 models, achieving improvements comparable to those observed with the Llama family models. This finding highlights the effectiveness of our approach.

4.4 Ablation Study

In this section, we analyze the effectiveness of amplifying different components of the Llama2-7B model: **Hidden states**, **Attention**, and **MLP** with different values of α, as illustrated in Figure 6.

Value of α . Amplifying hidden states results in a clear performance boost, with the effectiveness varying across different layers and amplification factors α. The optimal amplification factor appears to be between 4 and 6, as evidenced by the higher EM scores in the upper curves of the left plot.

Intervention Layer. Notably, the intervention is most effective when applied in later layers (20-25), where all α values lead to convergence around 70% EM score. This suggests that the model's representational capacity is most malleable and responsive to amplification in these deeper layers, possibly due to their role in high-level feature integration.

Attention and MLP. The Attention and MLP amplification show more erratic results, with fluctuating performance across layers. For these components, amplifying either leads to diminishing returns or even decreases in performance, suggesting that these layers do not benefit from amplification in the same way as hidden states.

The poor performance of Attention and MLP amplification can be attributed to several factors. For the attention mechanism, which is finely controlled for inference tasks (Jin et al., 2024; Zhou et al., 2024), further amplification may disrupt the delicate balance, leading to noise amplification. Furthermore, the MLP is generally responsible for storing knowledge (Meng et al., 2022b; Geva et al., 2021b). The amplifying the entire MLP can result in a proportional increase in all stored knowledge, which may effectively render the amplification meaningless.

5 Related Work

Existing approaches to enhancing context-faithfulness in LLMs can be broadly classified into three categories: fine-tuning methods (Bi et al., 2024), external interventions (Zhou et al., 2023) and internal interventions. The internal interventions predominantly focus on modifying decoding strategies or reweighting knowledge-aware neurons. Methods such as CAD (Shi et al., 2024b) and COIECD (Yuan et al., 2024) optimize decoding strategies through a contrastive mechanism (Li et al., 2023) to promote greater reliance on external information. However, these decoding-based approaches operate at the output level, resulting in only limited improvements. Another stream of research explores the internal states of models. For instance, Jin et al. (2024) and Shi et al. (2024a) aim to identify and reweight neurons crucial for processing contextual cues, thereby alleviating conflicts through targeted interventions at critical points.

6 Conclusion

In this paper, we propose a novel intervention method called CaLE, which exploits contextual knowledge within LLMs' internal representations. It strategically amplifies the contextual information growth at an appropriate layer, which facilitates richer representations in the final layer. Our experiments demonstrate that CaLE effectively improves context-faithful generation in QA tasks, particularly in scenarios involving unknown or conflicting contextual knowledge.

Limitations

The CaLE approach is to simply conduct one-time intervention during the inference process. While there are several other potential methods to execute interventions continuously, we leave the exploration of these alternatives for future work.

The intervention analysis on the MLP and Attention components adopts the amplification method proposed in the paper. While numerous studies (Meng et al., 2022b; Geva et al., 2021b) have discussed the role of these components in manipulating knowledge, this study does not specifically explore whether alternative intervention methods beyond amplification, or selective interventions within layers, should be employed. Instead, a uniform approach of amplifying entire layers is adopted, which may introduce limitations.

References

Amos Azaria and Tom M. Mitchell. 2023. The internal state of an LLM knows when its lying. *CoRR*, abs/2304.13734.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, and Shenghua Liu. 2024. Context-dpo: Aligning language models for context-faithfulness. *Preprint*, arXiv:2412.15280.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: llms' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard H. Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Trans. Assoc. Comput. Linguistics*, 9:1012–1031.

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

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with V-usable information. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Siqi Fan, Xin Jiang, Xiang Li, Xuying Meng, Peng Han, Shuo Shang, Aixin Sun, Yequan Wang, and Zhongyuan Wang. 2024. Not all layers of llms are necessary during inference. *CoRR*, abs/2403.02181.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021a. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021b. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained 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*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 1193–1215. Association for Computational Linguistics.

Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. 2024. How large language models encode context knowledge? A layer-wise probing study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 8235–8246. ELRA and ICCL.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12286–12312. Association for Computational Linguistics.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7052–7063. Association for Computational Linguistics.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *CoRR*, abs/2403.03853.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022b. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

nostalgebraist. 2020. Interpreting GPT: the logit lens. *AI Alignment Forum*.

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

Zexuan Qiu, Zijing Ou, Bin Wu, Jingjing Li, Aiwei Liu, and Irwin King. 2024. Entropy-based decoding for retrieval-augmented large language models. *CoRR*, abs/2406.17519.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2025. Investigating the factual knowledge boundary of large language models with retrieval augmentation. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 3697–3715. Association for Computational Linguistics.

Dan Shi, Renren Jin, Tianhao Shen, Weilong Dong, Xinwei Wu, and Deyi Xiong. 2024a. IRCAN: mitigating knowledge conflicts in LLM generation via identifying and reweighting context-aware neurons. *CoRR*, abs/2406.18406.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024b. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Short Papers, NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 783–791. Association for Computational Linguistics.

Oscar Skean, Md Rifat Arefin, Yann LeCun, and Ravid Shwartz-Ziv. 2024. Does representation matter? exploring intermediate layers in large language models. In *NeurIPs Workshop on Machine Learning and Compression*.

Zhongxiang Sun, Xiaoxue Zang, Kai Zheng, Yang Song, Jun Xu, Xiao Zhang, Weijie Yu, and Han Li. 2024. Redeep: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. *CoRR*, abs/2410.11414.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. Berxit: Early exiting for BERT with better fine-tuning and extension to regression. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 91–104. Association for Computational Linguistics.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. 2020. A theory of usable information under computational constraints. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. Characterizing mechanisms for factual recall in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9924–9959. Association for Computational Linguistics.

Xiaowei Yuan, Zhao Yang, Yequan Wang, Shengping Liu, Jun Zhao, and Kang Liu. 2024. Discerning and resolving knowledge conflicts through adaptive decoding with contextual information-entropy constraint. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 3903–3922. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao

Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A survey of large language models. *Preprint*, arXiv:2303.18223.

Hanzhang Zhou, Zijian Feng, Zixiao Zhu, Junlang Qian, and Kezhi Mao. 2024. Unibias: Unveiling and mitigating LLM bias through internal attention and FFN manipulation. 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*.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 14544–14556. Association for Computational Linguistics.

A Experimental Settings for CounterFact

A.1 Data Format in Section 2

We use question q with context c from COUNTERFACT (Meng et al., 2022a). For a given model, we input the sample data for which the model predicts the correct answer (e.g., "Danielle Darrieux, a native French. The mother tongue of Danielle Darrieux is _"). In this section, we refer to the token predicted by the model for a given input as the answer.

```
c: {{paraphrased prompt} {target true}.}
q: {{prompt}}
```

A.2 Data Format in Section 3

Similarly, we construct the data for computing these two KL divergences based on the CounterFact dataset:

$$KL_q(l) = \text{KL}\left[P(\boldsymbol{h}_l \mid q) \,\|\, P(\boldsymbol{h}_l)\right]$$
$$KL_c(l) = \text{KL}\left[P(\boldsymbol{h}_l \mid q, c) \,\|\, P(\boldsymbol{h}_l \mid q)\right]$$

The term without c (i.e., $P(\boldsymbol{h}_l \mid q)$) is generated by excluding c from the input.

```
q: {{prompt}}
```

The term without both q and c (i.e., $P(\boldsymbol{h}_l)$) is derived by providing an empty input.

```
{∅}
```

B Logit Lens

The LogitLens is a technique that decodes hidden states $\boldsymbol{h}^l$ directly into the vocabulary distribution using the LayerNorm and the unembedding matrix of the LLM for interpretability (nostalgebraist, 2020):

$$\text{LogitLens}\left(\boldsymbol{h}^l\right) = \text{LayerNorm}(\boldsymbol{h}^l)\boldsymbol{W}_U \quad (19)$$

The final layer's residual stream state is then projected into the vocabulary space using the *Unembedding Matrix* $\boldsymbol{W}_U \in \mathbb{R}^{d\times|\mathcal{V}|}$ and normalized via the softmax function to produce a probability distribution over the vocabulary, from which a new token is sampled.

This approach has been validated in various studies as an effective method for interpreting LLMs' weight matrices or hidden states (Yu et al., 2023; Hanna et al., 2023; Zhou et al., 2024).

C Residual Stream

We interpret transformer decoder-only architecture (also known as GPT-like) through the perspective of the residual stream (Elhage et al., 2021; Olsson et al., 2022). Due to the residual connections in Transformers, each layer l takes a hidden state $\boldsymbol{h}^{l-1}$ as input and adds information obtained from its *Attention Heads* and *Feed-Forward Networks* (FFNs) to the hidden state via the residual connection. In this context, the hidden state acts as a residual stream passed through the layers, with each attention and FFN contributing to the final prediction by adding information to the residual stream, resulting in the *Residual Stream States*. Formally, the hidden state $\boldsymbol{h}^l$ at layer l is calculated as:

$$\begin{aligned}\boldsymbol{h}^l &= \boldsymbol{h}^{l-1} + \text{MHA}(\boldsymbol{h}^{l-1}) + \text{FFN}(\boldsymbol{a}^l)\\ &= \boldsymbol{h}^{l-1} + \boldsymbol{a}^l + \boldsymbol{m}^l\end{aligned}$$

where $\boldsymbol{a}^l$ and $\boldsymbol{m}^l$ are the outputs from the MHA and FFN block in the l-th layer. Both quantities are dependent on $\boldsymbol{h}^{l-1}$ and can thus be formulated as functions of it (See Eq. 20).

Then, the hidden state $\boldsymbol{h}^{l+1}$ at layer $l+1$ is calculated as:

$$\begin{aligned}\boldsymbol{h}^{l+1} &= \boldsymbol{h}^l + \boldsymbol{a}^{l+1} + \boldsymbol{m}^{l+1}\\ &= \boldsymbol{h}^{l-1} + \boldsymbol{a}^l + \boldsymbol{m}^l + \boldsymbol{a}^{l+1} + \boldsymbol{m}^{l+1}\\ &= \boldsymbol{h}^{l-1} + \sum_{k=l}^{l+1}\boldsymbol{a}^k + \sum_{k=l}^{l+1}\boldsymbol{m}^k\end{aligned}$$

Consequently, the hidden state $\boldsymbol{h}_i^N$ at the final layer $N(N \geq l)$ can be calculated as:

$$\begin{aligned}\boldsymbol{h}^N &= \boldsymbol{h}^{N-1} + \boldsymbol{a}^N + \boldsymbol{m}^N\\ &= \boldsymbol{h}^{N-2} + \boldsymbol{a}^{N-1} + \boldsymbol{m}^{N-1} + \boldsymbol{a}^N + \boldsymbol{m}^N\\ &= \ldots\\ &= \boldsymbol{h}^l + \sum_{k=l+1}^{N}\boldsymbol{a}^k + \sum_{k=l+1}^{N}\boldsymbol{m}^k\end{aligned}$$

where $\mathbf{s}_i$ represents the sum of the contributions from the subsequent layers to the final layer.

The final layer's residual stream state is then projected into the vocabulary space using the *Unembedding Matrix* $\boldsymbol{W}_U \in \mathbb{R}^{d\times|\mathcal{V}|}$. The final output

logits of the LLM can be expressed as:

$$\text{logits}_N = \boldsymbol{h}^l \boldsymbol{W}_U + \left(\sum_{k=l+1}^{N} \boldsymbol{a}^k + \sum_{k=l+1}^{N} \boldsymbol{m}^k \right) \boldsymbol{W}_U$$
$$= \mathbf{v} + \mathbf{u}(\mathbf{v}) \tag{20}$$

For analytical simplicity, we ignore the final LayerNorm function following Elhage et al. (2021) and Sun et al. (2024). It adds a fair amount of complexity to consider explicitly, and up to a variable scaling, layer norm can be merged into adjacent weights. Conceptually, $\mathbf{v}$ and $\mathbf{u}(\mathbf{v})$ capture the information encoded in layer l and later layer $> l$ respectively.

Finally, the logit would be normalized via the softmax function to produce a probability distribution over the vocabulary, from which a new token is sampled.

$$\text{P} = \frac{e^{v_k + u_k(v_k)}}{\sum_{j \in |Vocab|} e^{v_j + u_j(v_j)}} \tag{21}$$

D Theoretical Support for Enhancement

D.1 Analysis on the Amplification

Suppose we amplify the hidden states at layer l by the factor α:

$$\boldsymbol{h}^l_{\text{modified}} = \alpha \boldsymbol{h}^l$$

We will analyze how this scaling affects the subsequent computations. Here Let's take LayerNorm as an example.

At Layer $l+1$. Since LayerNorm is scale-invariant:

$$\tilde{\boldsymbol{h}}^l_{\text{modified}} = \text{LayerNorm}(\alpha \boldsymbol{h}^l)$$
$$= \text{LayerNorm}(\boldsymbol{h}^l) = \tilde{\boldsymbol{h}}^l$$

The scaling by α has no effect on the output of the first LayerNorm. Therefore, the input to the MHA sublayer remains unchanged:

$$\boldsymbol{a}^{l+1} = \text{MHA}(\tilde{\boldsymbol{h}}^l)$$

After the MHA sublayer, the residual connection adds $\boldsymbol{a}^l$ back to the scaled $\boldsymbol{h}^l$:

$$\boldsymbol{h}^{l+1}_{\text{modified}} = \boldsymbol{h}^l_{\text{modified}} + \boldsymbol{a}^{l+1} = \alpha \boldsymbol{h}^l + \boldsymbol{a}^{l+1}$$

Then the scaled hidden state αh^l is now part of $\boldsymbol{h}^{l+1}$, which will be normalized for FFN input.

$$\tilde{\boldsymbol{h}}^{l+1}_{\text{modified}} = \text{LayerNorm}(\boldsymbol{h}^{l+1}_{\text{modified}})$$
$$= \gamma \odot \frac{\alpha \boldsymbol{h}^l + \boldsymbol{a}^l - \mu^{l+1}_{\text{modified}}}{\sigma^{l+1}_{\text{modified}}} + \delta$$

where

$$\mu^{l+1}_{\text{modified}} = \frac{1}{D} \sum_{i=1}^{D} \left(\alpha \boldsymbol{h}^l + \boldsymbol{a}^l \right) = \alpha \mu^l + \mu_a$$

$$\sigma^{l+1}_{\text{modified}} = \sqrt{\frac{1}{D} \sum_{i=1}^{D} \left(\alpha \boldsymbol{h}^l + \boldsymbol{a}^l - \mu^{l+1}_{\text{modified}} \right)^2}$$

The normalization does not cancel out α completely because $\boldsymbol{a}^l$ is not scaled. Therefore, the information in $\boldsymbol{a}$ is relatively compressed. Then,

$$\boldsymbol{m}^{l+1}_{\text{modified}} = \text{MLP}(\tilde{\boldsymbol{h}}^{l+1}_{\text{modified}})$$

The output at layer $l+1$ is:

$$\boldsymbol{h}^{l+1}_{\text{modified}} = \boldsymbol{h}^l_{\text{modified}} + \boldsymbol{a}^{l+1} + \boldsymbol{m}^{l+1}_{\text{modified}}$$
$$= \alpha \boldsymbol{h}^l + \boldsymbol{a}^{l+1} + \boldsymbol{m}^{l+1}_{\text{modified}}$$

At Layer $k > l+1$. The altered hidden state $h^{l+1}_{\text{modified}}$ affects all subsequent layers in a similar fashion. And the amplification effect of α persists due to the residual connections.

The hidden state can be represented recursively:

$$\boldsymbol{h}^k_{\text{modified}} = \alpha \boldsymbol{h}^l + \boldsymbol{a}^{l+1} + \boldsymbol{m}^{k+1}_{\text{modified}}$$
$$+ \sum_{i=l+2}^{k} \left(\boldsymbol{a}^i_{\text{modified}} + \boldsymbol{m}^i_{\text{modified}} \right)$$

Therefore, the amplification effect of α **accumulates** through the residual connections and affects all subsequent layers.

At Final Layer N. The logits are computed using the final hidden state $\boldsymbol{h}^N_{\text{modified}}$:

$$\text{logits}_{\text{modified}} = \boldsymbol{h}^N_{\text{modified}} \boldsymbol{W}_U$$

Breaking Down $\boldsymbol{h}^N_{\text{modified}}$, the logits can be calculated as:

$$\text{logits}_{\text{modified}} = \alpha \boldsymbol{h}^l \boldsymbol{W}_U + (\boldsymbol{a}^{l+1} + \boldsymbol{m}^{l+1}_{\text{modified}}$$
$$+ \sum_{i=l+2}^{N} (\boldsymbol{a}^i_{\text{modified}} + \boldsymbol{m}^i_{\text{modified}})) \boldsymbol{W}_U$$

Finally, the softmax probabilities become:

$$\text{P}_{\text{modified}} = \frac{e^{\alpha v_k + u_k^{(A)}(\alpha v)}}{\sum_{j \in |Vocab|} e^{\alpha v_j + u_j^{(A)}(\alpha v)}} \tag{22}$$

where: $\boldsymbol{v} = \boldsymbol{h}^L \boldsymbol{W}_U$, and $\boldsymbol{u}^{(A)}(\cdot)$ includes contributions from the subsequent layers by amplification.

This leads to a proportional change in the logits and alters the softmax probability distribution, potentially affecting the model's predictions.

In correspondence with Formula 21, the $\mathcal{V}$-entropy is derived as follows:

$$H_{\mathcal{V}_f}(\alpha) \stackrel{\text{def}}{=} \mathbb{E}[\ln \sum_j e^{\alpha v_j + u_j(\alpha v_j)} - (\alpha v_k + u_k(\alpha v_k))]$$

D.2 Analysis on the Additional Residual Connection

Suppose we introduce an additional residual connection at layer l, which directly propagates $\boldsymbol{h}^l$ to α subsequent layers:

At Layer $l+1$. With the inclusion of the new residual connection, the hidden state is modified as follows:

$$\begin{aligned}\boldsymbol{h}^{l+1}_{\text{modified}} &= (\boldsymbol{h}^l + \boldsymbol{a}^{l+1} + \boldsymbol{m}^{l+1}) + \boldsymbol{h}^l \\ &= 2\boldsymbol{h}^l + \boldsymbol{a}^{l+1} + \boldsymbol{m}^{l+1}\end{aligned}$$

At Layer k where $l+1 < k \leq l+\alpha$. Due to the cumulative effect of the residual connections, the hidden state at layer k can be expressed as:

$$\begin{aligned}\boldsymbol{h}^k_{\text{modified}} &= (k-l+1)\boldsymbol{h}^l + \boldsymbol{a}^{l+1} + \boldsymbol{m}^{l+1} \\ &+ \sum_{i=l+2}^{k} (\boldsymbol{a}^i_{\text{modified}} + \boldsymbol{m}^i_{\text{modified}})\end{aligned}$$

where $(k-l+1)$ represents the number of times the residual connection has been accumulated.

At the Final Layer N. The final logits are computed as:

$$\begin{aligned}&\text{logits}_{\text{modified}} = \boldsymbol{h}^N_{\text{modified}} \boldsymbol{W}_U \\ &= (\alpha+1)\boldsymbol{h}^l \boldsymbol{W}_U + (\boldsymbol{a}^{l+1} + \boldsymbol{m}^{l+1} \\ &+ \sum_{i=l+2}^{N} (\boldsymbol{a}^i_{\text{modified}} + \boldsymbol{m}^i_{\text{modified}}))\boldsymbol{W}_U\end{aligned}$$

Similar to the previous deduction (Appendix D.1), the final softmax probability distribution is given by:

$$\text{P}_{\text{modified}} = \frac{e^{(\alpha+1)v_k + u_k^{(R)}((\alpha+1)v)}}{\sum_{j \in |Vocab|} e^{(\alpha+1)v_j + u_j^{(R)}((\alpha+1)v)}} \tag{23}$$

where $\boldsymbol{v} = \boldsymbol{h}^{(l)}\boldsymbol{W}_U$. While $\boldsymbol{u}^{(R)}(\cdot)$ encapsulates the contributions from the subsequent layers, it differs fundamentally from the $\boldsymbol{u}^{(A)}(\cdot)$ function in the Amplification method, as the underlying modifications to the information flow in these two approaches are inherently distinct.

Compared with Amplification. This analysis reveals that adding cumulative residual connections provides a structured approach to amplifying the influence of intermediate layer representations. While there are subtle differences in how these methods affect subsequent layers, we empirically compare their performances through experiments in Section 4.

E Proof for Proposition 3.1

Proposition. Let α denote the scaling factor applied to the hidden states at this layer. If $k = \arg\max_j v_j$, then

$$\lim_{\alpha \to \infty} H_{\mathcal{V}_f}(\alpha) \approx 0$$

Proof. First, consider the following decomposition:

$$\sum_j e^{\alpha v_j + u_j(\alpha v)} = e^{\alpha v_k + u_k(\alpha v)}. \left(1 + \sum_{j \neq k} e^{\alpha(v_j - v_k) + u_j(\alpha v_j) - u_k(\alpha v_k)}\right)$$

When $j \neq k$, $v_j - v_k < 0$. All terms with $j \neq k$ exponentially diminish as α increases[5]. Therefore:

$$\lim_{\alpha \to \infty} \sum_j e^{\alpha v_j + u_j(\alpha v_j)} \approx e^{\alpha v_k + u_k(\alpha v_k)} (1 + \epsilon(\alpha))$$

where $\epsilon(\alpha) \to 0$ as $\alpha \to \infty$.

Applied to the Formula 9, we have

$$\begin{aligned}\lim_{\alpha \to \infty} H_{\mathcal{V}_f}(\alpha) &\approx \mathbb{E}\Big[\alpha v_k + u_k(\alpha v) + \ln(1 + \epsilon(\alpha)) \\ &\quad - (\alpha v_k + u_k(\alpha v))\Big] \\ &= \mathbb{E}[\ln(1 + \epsilon(\alpha))] \\ &= 0\end{aligned}$$

Therefore, the result is proven. □

F Dataset Details

CounterFact. The CounterFact (Meng et al., 2022a) dataset is derived from the PARAREL dataset (Elazar et al., 2021) and contains knowledge tuples of the kind $t^c = (s, r, o^c)$, where s is the subject, r is the relation and o is the object. These tuples are constructed using entities listed

[5] Since α approaches zero, $u_j(\alpha v) - u_k(\alpha v)$ represents the difference between two extremely small quantities. It can be negligible compared to the $\alpha(v_j - v_k)$.

in Wikidata. The data are accompanied by handwritten paraphrased prompts for each sample. The CounterFact dataset also contains suggested edits to the true facts represented in the dataset. For this study, the set of counterfactual edits are not used.

NaturalQuestions. NQ (Kwiatkowski et al., 2019) consists of real-world information-seeking queries issued to the Google search engine and their corresponding long answers (gold evidence passage) and short answers (one or more entities). In our study, we employ the long answers as the input context and short answers as the ground truth, and conduct evaluations on the dev set.

NQ-Swap NQ-Swap is based on the NQ dataset, where the objective is to answer questions based on a reliable gold document. To generate NQ-Swap, Longpre et al. (2021) first identify questions in NQ with named entity answers, find the supportive document for each question and then replace the gold answer entity in the document with a random entity. A faithful LM should generate the replaced entity as the answer when given the question and modified document.

SQuAD. The SQuAD (Rajpurkar et al., 2016) 1.1 is a common QA benchmark. It includes questions posed by human annotators on a given Wikipedia paragraph, where the answer to each question is a segment of text (or span) from the paragraph. In our experiments, we conduct experiments on the dev for evaluation.

StrategyQA. StrategyQA (Geva et al., 2021a) is a fact reasoning benchmark that necessitates the implicit question decomposition into reasoning steps. Built around Wikipedia terms, these questions are accompanied by multiple evidence paragraphs. The model is expected to provide a True or False answer. We concatenate question-relevant evidences to form the input context. We adopt the training set for evaluation, considering the volume of data.

F.1 Posteriori judgement for CounterFact

We delineates the process of identifying knowledge boundary of "unknown" and "known" contexts. The evaluation is based on the accuracy of the model's responses when context is not provided. The scenarios are divided into two categories:

- **Unkown**: This category refers to instances where the model is unable to provide the correct answer without relying on the provided context. Such cases indicate that the external contextual knowledge represents information not contained within the model's inherent parametric knowledge.
- **Known**: This category describes scenarios in which the model can accurately answer a question without requiring its corresponding context. These instances demonstrate that the model has internalized the relevant knowledge, reflecting an alignment between its parametric knowledge and the external contextual information.

G Decoding Strategies

Contrastive Decoding (CD) In our experiments, we employ the distribution $g(y_t)$ with a certain threshold as a baseline decoding method, referred to as the CD (Li et al., 2023) method. We modify the original object of CD (computes the distribution discrepancy between an small amateur model and an expert larger model) to simulate the form of $g(y_t)$.

$$\begin{aligned} CD &= \log p(y_t|x, y < t) - p(y_t|y < t) \\ &= \log g(y_t) \end{aligned}$$

The threshold is same as in the original CD method:

$$\mathcal{V}_{\text{head}}(y_{<t}) = \left\{ y_t \in \mathcal{V} : p(y_t|y_{<t}) \geq 0.1 \cdot \max_y p(y|y_{<t}) \right\}$$

Here, we represent the input context as x. CD adopts the object of difference between the output likelihood when inputs are presented with and without input context. It enhances the influence of the context for high-probability words within a crude threshold.

Context-Aware Decoding (CAD) In CAD (Shi et al., 2024b) method, the output probability is a product-of-experts of the original output probability and PMI weighted by $\alpha = 0.5$ as follow:

$$\begin{aligned} y_t \sim \text{softmax}[(1+\alpha)\,\text{logit}_\theta(y_t \mid \boldsymbol{c}, \boldsymbol{x}, \boldsymbol{y}_{<t}) \\ - \alpha\,\text{logit}_\theta(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t})] \end{aligned}$$

COntextual Information-Entropy Constraint Decoding (COIECD) First, the contextual contrastive object g is calculated to quantify the divergence between p_1 and p_2:

$$g(y_t) = \log p_2(y_t) - \log p_1(y_t)$$

where

$$p_1(y_t) = p(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t})$$
$$p_2(y_t) = p(y_t|\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{y}_{<t})$$

The g is to factor out the model's inherent memory and favor the contextual knowledge.

The contextual information-entropy constraint is utilized with g on the output distribution π as:

$$\log \pi(y_t \mid \boldsymbol{x}, \boldsymbol{c}, \boldsymbol{y}_{<t}) \tag{24}$$
$$= \begin{cases} \log p_1(y_t) + \alpha \cdot g(y_t) & \text{if } y_t \in \mathcal{C}(\boldsymbol{y}_{<t}), \\ \log p_2(y_t) + \alpha \cdot g(y_t) & \text{otherwise.} \end{cases}$$

where α is a scaling weight to control the contextual impact. The final decoding strategy can be formalized as:

$$y_t \sim \text{softmax}[\log \pi(y_t \mid \boldsymbol{x}, \boldsymbol{c}, \boldsymbol{y}_{<t})] \tag{25}$$

In this way, COIECD strikes a balance between the two sources of knowledge to achieve a more effective and holistic decoding strategy.

H Layer Selection by CaLE

Here, we present some models that employ the CaLE method, as shown in Tables 1 and 3, which enhance various layers selected through both supervised and unsupervised identification, as indicated in Table 4. Our findings reveal that nearly all of the selected layers are distributed in the middle to later stages, suggesting that intervening at deeper layers is a more effective choice.

Layer selected by CaLE	**CounterFact**		**NQ**		**NQ-swap**	
	sup.	unsup.	sup.	unsup.	sup.	unsup.
Llama2-7B	25	25	26	25	26	25
Llama3.1-8B	23	24	27	24	23	23
Llama3.2-3B	23	22	-	-	-	-
Mistral-7B	-	-	25	22	21	18
Gemma2-9B	-	-	32	35	29	31
	SQuAD		**StrategyQA**			
	sup.	unsup.	sup.	unsup.		
Llama2-7B	23	25	22	19		
Llama3.1-8B	26	24	30	25		
Mistral-7B	25	19	22	19		
Gemma2-9B	36	31	34	35		

Table 4: Layer selection of CaLE across different datasets and models. The **unsup**. and **sup**. denote the unsupervised and supervised CaLE methods.