Expanding before Inferring: Enhancing Factuality in Large Language Models through Premature Layers Interpolation

Dingwei Chen[1,2], **Ziqiang Liu**[5], **Feiteng Fang**[5], **Chak Tou Leong**[3], **Shiwen Ni**[5], **Ahmadreza Argha**[4], **Hamid Alinejad-Rokny**[4], **Min Yang**[5*], **Chengming Li**[2*]

[1] Sun Yat-Sen University [2]Shenzhen MSU-BIT University [3]The Hong Kong Polytechnic University

[4]UNSW, Sydney, NSW 2052, Australia [5]Shenzhen Key Laboratory for High Performance Data Mining, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

cuso4cdw@gmail.com, licm@smbu.edu.cn, min.yang@siat.ac.cn

Abstract

Large Language Models (LLMs) demonstrate remarkable capabilities in text understanding and generation. However, their tendency to produce factually inconsistent outputs—commonly referred to as "hallucinations"—remains a critical challenge. Existing approaches, such as retrieval-based and inference-time correction methods, primarily address this issue at the input or output level, often overlooking the intrinsic information refinement process and the role of premature layers. Meanwhile, alignment- and fine-tuning-based methods are resource-intensive. In this paper, we propose **PLI** (**P**remature **L**ayers **I**nterpolation), a novel, training-free, and plug-and-play intervention designed to enhance factuality. PLI mitigates hallucinations by inserting premature layers formed through mathematical interpolation with adjacent layers. Inspired by stable diffusion and sampling steps, PLI extends the depth of information processing and transmission in LLMs, improving factual coherence. Experiments on four publicly available datasets demonstrate that PLI effectively reduces hallucinations while outperforming existing baselines in most cases. Further analysis suggests that the success of layer interpolation is closely linked to LLMs' internal mechanisms. Our dataset and code are available at https://github.com/CuSO4-Chen/PLI.

1 Introduction

Recently, Large Language Models (LLMs) have revolutionized artificial intelligence with their unprecedented capabilities in language understanding and open-ended generation, demonstrating strong performance across various downstream tasks (Guo et al., 2025; Brown et al., 2020; Achiam et al., 2023; Zhao et al., 2023). However, their remarkable progress is overshadowed by a persistent challenge: the tendency to generate plausible yet factually incorrect content, commonly referred to as "hallucinations" (Zhang et al., 2024b; Chuang et al., 2023). These deficiencies in factual grounding undermine the reliability of LLM applications, making hallucination mitigation a critical area of research (Huang et al., 2023a; Yang et al., 2024a; Kai et al., 2024; Chen et al., 2024a). Prior studies suggest that hallucinations stem from multiple factors, including low-quality large-scale pretraining data (Zhang et al., 2023b; Ye et al., 2023), model training errors (Zhang et al., 2023a), and unstable decoding strategies during generation (Huang et al., 2023a; Cheng et al., 2025).

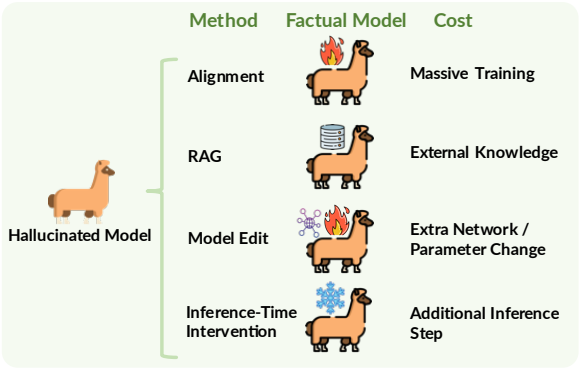

Figure 1: Brief overview of previous methods for alleviating hallucination in LLMs.

Existing approaches (in Figure 1) to mitigating hallucinations typically leverage external knowledge bases (Peng et al., 2023; Jiang et al., 2023; Wang et al., 2023), align models with human feedback (Ouyang et al., 2022), refine output distributions through contrastive decoding (CD) (Li et al., 2023c; Chuang et al., 2023; Zhang et al., 2023a; O'Brien and Lewis, 2023; Chen et al., 2024a), modify internal knowledge or output representations (Meng et al., 2023; Ni et al., 2023; Zhang et al., 2024b; Liang et al., 2024). Among these, inference-time methods are gaining increasing attention due to their simplicity and lower computational cost, but still with extra inference steps.

*Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 12782–12797
November 4-9, 2025

Empirically, LLMs process information through successive transformer layers, where each layer refines the previous representation toward greater semantic and factual coherence (Lad et al.). Thus, as information flows from input to output, it can be viewed as progressively processing and refining knowledge. Regarding how existing approaches mitigate hallucination, retrieval-augmented generation (RAG) (Lewis et al., 2020) enhances the input stage by incorporating external knowledge, while inference-time methods (e.g., CD) refine the output. However, little research has focused on optimizing the intermediate representations of *premature layers* (i.e., intermediate layers), which could help mitigate hallucinations in the final output. While injecting additional modules, such as adapters (Houlsby et al., 2019) or transformer blocks (Wu et al., 2024), may achieve similar effects, these approaches require extra training costs. Efficiently optimizing the output representations of premature layers with minimal overhead remains an open challenge.

To this end, we propose a novel training-free paradigm for hallucination alleviation called **PLI** (**P**remature **L**ayers **I**nterpolation). Our method stems from the observation that LLMs typically process and refine knowledge in a fixed number of steps (i.e., the number of layers). This contrasts with generative models like diffusion models (Rombach et al., 2022), which allow for adjustable refinement steps where more steps generally lead to higher quality generation. PLI aims to extend the knowledge processing depth in LLMs to reduce hallucinations by effectively increasing the number of model layers. Specifically, PLI is a simple yet effective method that strategically inserts one or more parameter-interpolated layers. These layers are constructed by interpolating the parameters of two adjacent layers, expanding the refinement process, and enhancing the inherent capacity of LLMs for factual consistency. Furthermore, PLI is designed as a plug-and-play module that can seamlessly integrate with existing base models and hallucination mitigation techniques. The main contributions of this work can be summarized as follows:

- We propose PLI, a novel framework for hallucination alleviation that is both training-free and plug-and-play. PLI leverages the existing parameter manifold through interpolation to construct new premature layers, enhancing factuality in LLMs.
- We conduct extensive experiments on four widely used benchmarks, demonstrating that PLI outperforms baseline methods in most cases and can be effectively integrated with existing hallucination mitigation techniques.
- We perform a series of analyses to theoretically investigate the relationship between layer interpolation effectiveness and the internal mechanisms of LLMs.

2 Related Work

2.1 Halluciantion Mitigation

Current approaches to mitigate hallucinations can be broadly categorized into several classes (Huang et al., 2023a; Zhang et al., 2023b).

Alignment with Feedback plays a crucial role in training LLMs, ensuring that models adapt to specific feedback and effectively reduce hallucinations (Liu et al., 2023a; Menick et al., 2022). Stiennon et al. (2020) construct a high-quality corpus with human feedback to train models that generate human-preferred summaries, promoting more factual and reliable outputs. Shinn et al. (2023) propose a method using verbal reinforcement to help agents learn from failures through self-reflection.

Retrieval-Augmented Generation (RAG) is a powerful approach that enhances LLMs by integrating factual knowledge, thereby improving factual accuracy (Wang et al., 2023; Li et al., 2023a; Liu et al., 2023b). Peng et al. (2023) introduce a framework that augments LLMs with external knowledge and automated feedback, helping to generate more truthful responses. Trivedi et al. (2023) propose a retrieval method that leverages LLMs' chain-of-thought (CoT) reasoning capabilities to reduce hallucinations and enhance logical consistency.

Model Editing refers to modifying the knowledge stored in LLMs, enabling data-efficient updates to correct hallucinations or outdated information (Zheng et al., 2023; Huang et al., 2023b; Gupta et al., 2023; Yao et al., 2023; Ni et al., 2023; Zhang et al., 2024a). SERAC (Mitchell et al., 2022) routes new facts through a distinct network while keeping the original parameters unchanged. IKE (Zheng et al., 2023) edits the model by prompting it with the revised fact using in-context examples. ROME (Meng et al., 2022) modifies specific knowledge neurons in feed-forward networks (FFNs) through a locate-then-edit approach

MEMIT(Meng et al., 2023) extends this by simultaneously updating multiple pieces of knowledge.

2.2 Inference-Time Intervention

Representation editing is an inference-time method that modifies a model's internal representations to improve output factuality (Li et al., 2023b; Zhang et al., 2024b; Chen et al., 2024b; Burns et al., 2022). ITI (Li et al., 2023b) probes and adjusts truthfulness within the attention heads of LLMs. TruthForest (Chen et al., 2024b) employs orthogonal probes to uncover hidden truth representations. TruthX (Zhang et al., 2023a) decouples LLM representations into separate truthful and semantic latent spaces using an autoencoder and applies contrastive learning to refine factual outputs.

Contrastive Decoding (CD) is a decoding-time framework that enhances factuality by contrasting predictions with the logits of a much smaller LLM. DoLa (Chuang et al., 2023) applies CD between later and earlier layers to boost factuality and reasoning. ICD (Zhang et al., 2023a) penalizes hallucination-prone predictions by inducing contrast with a factually weaker LLM. Beyond CD, HaluSearch (Cheng et al., 2025) integrates Monte Carlo Tree Search (MCTS) to enable a deliberate, slow-thinking generation process for mitigating hallucinations

In this paper, we propose PLI, a novel inference-time method that extends LLMs' end-to-end information processing by inserting premature layers constructed through interpolation. These interpolated layers optimize intermediate representations to reduce hallucinations. PLI is highly flexible and can seamlessly integrate with other hallucination alleviation techniques.

3 Method

The proposed PLI (Premature Layers Interpolation) method mitigates hallucination in LLMs by strategically inserting interpolated premature layers between the existing transformer layers. We treat the depth of layers *L* as a controllable dimension, rather than a fixed architectural constant, to enhance factual coherence. PLI uses **spherical linear interpolation (Slerp)** (Robeson, 1997) to preserve the geometric properties of parameter vectors in high-dimensional space. This ensures smoother transitions between layer representations and optimizes the output representation of the premature layers. Figure 2 illustrates the diagram of PLI.

3.1 Slerp

Slerp (Robeson, 1997) is a well-established method for smoothly interpolating between two vectors, commonly applied in fields such as statistical forecasting and model merging (Yang et al., 2024b; Lu et al., 2024). It is calculated as follows:

$$Slerp(p, q, t) = \frac{\sin[(1-t)\theta] \cdot p + \sin(t\theta) \cdot q}{\sin\theta}, \tag{1}$$

where p, q are the vectors to be interpolated, θ is the angle between these two vectors, t defines the interpolation ratio. The division by $sin\theta$ ensures unit normalization.

In a similar manner, we apply Slerp to construct the interpolated premature layers. Given an LLM f_θ with two adjacent layers l and $l+1$, where l is the layer where the position is inserted, we treat their flattened parameter matrices W_l and W_{l+1} as normalized vectors $\mathbf{w}_l, \mathbf{w}_{l+1} \in \mathbb{R}^d$ on a d-dimensional hypersphere. The interpolated layer parameters $\mathbf{w}_{inter}$ are computed as follows:

$$\mathbf{w}_{\text{inter}} = \frac{\sin\left((1-\alpha)\theta\right)}{\sin\theta}\mathbf{w}_l + \frac{\sin(\alpha\theta)}{\sin\theta}\mathbf{w}_{l+1}, \tag{2}$$

where θ is the angle between the vectors w_l and w_{l+1}, and $\alpha \in [0, 1]$ is the interpolation ratio. The angle θ is computed as:

$$\theta = \arccos\left(\mathbf{w}_l \cdot \mathbf{w}_{l+1}\right). \tag{3}$$

This interpolation formulation ensures that the interpolated vector $\mathbf{w}_{\text{inter}}$ lies on the same hypersphere as the original vectors, preserving the magnitude and directionality. This property is essential for maintaining the model's intrinsic knowledge manifold, ensuring smoother transitions between layers in the model.

3.2 Premature Layers Interpolation Insertion

For an N-layers LLM $f_\theta = \{l_1, l_2, ..., l_N\}$, we insert interpolated layers M at predefined layer positions, resulting in the modified model $f_\theta = \{l_1, l_2, ..., l_M, ..., l_{N+M}\}$, achieved through premature layer interpolation. The insertion process follows three steps: **(i) Adjacent layer pair selection:** For each target insertion position l_i, identify the flanking layers (l_i, l_{i+1}), which refers to the new premature layer being inserted between these two layers. **(ii) Slerp Execution:** Compute $\mathbf{w}^i_{\text{inter}}$ using Eq. (2) with layer-specific α_i to control the interpolation ratio. **(iii) Parameter reshaping:** Restructure $\mathbf{w}^i_{\text{inter}}$ from the previous step into the

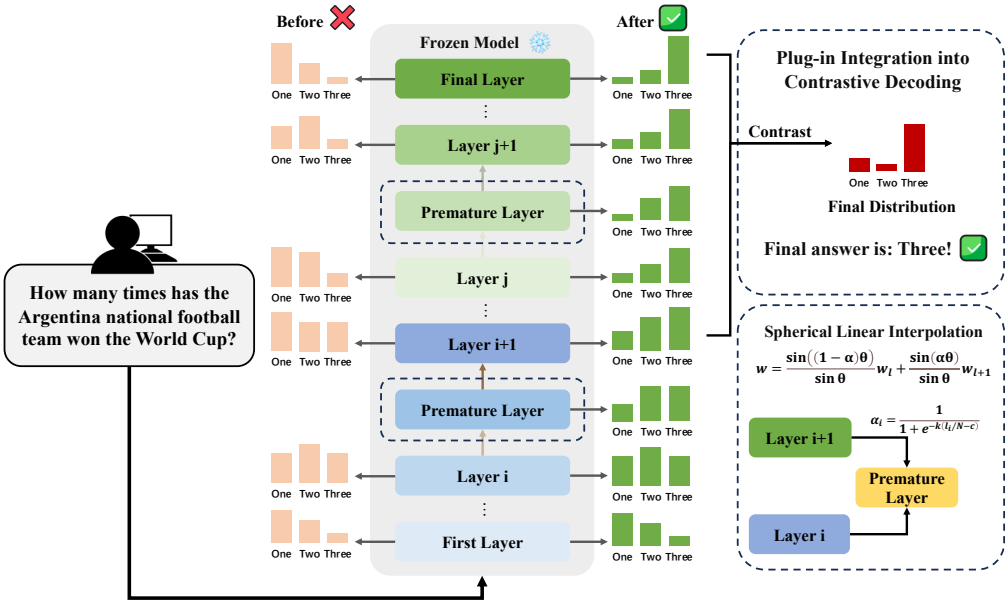

Figure 2: Overview of the **PLI** method, which inserts premature layers formed by mathematical interpolation to expand the information processing flow, enhancing the factuality and alleviating hallucination in LLMs.

original matrix dimensions of the parameters to form the complete parameters for the premature layer interpolation.

After the insertion of premature layers through interpolation, the forward propagation during inference within the LLM will be updated as follows:

$$h' = \text{Layer}^{i}_{\text{inter}}(\text{Layer}_{l_i}(h)) \tag{4}$$

where h is the hidden state from the layer l_{i-1}, and $\text{Layer}^{i}_{\text{inter}}$ denotes the inserted premature layer of interpolation. When the hidden state is output from layer l_i, it is input to the newly inserted premature layer for further information processing, which is then passed on to layer l_{i+1}. This execution can be repeated to insert additional premature layers for more precise hallucination alleviation. Premature layers interpolation is plug-and-play, meaning it can enhance the original model without causing any conflict with the base model or other methods.

3.3 Adaptive Interpolation Ratio Scheduling

In LLMs, different levels of transformer layers correspond to varying degrees of semantic abstraction. The lower layers (closer to the input end) primarily process local grammatical patterns and vocabulary-level information, while the higher layers (closer to the output end) focus on global semantic integration and factual information (Chuang et al., 2023). To accommodate this hierarchical semantic distribution, we propose an adaptive interpolation ratio scheduling mechanism for determining α_i. Specifically, the interpolation ratio α_i of an inserted layer is dynamically adjusted based on the position of the inserted layer l_i rather than being fixed. We utilize a variant of the sigmoid function to implement this scheduling:

$$\alpha_i = \frac{1}{1 + e^{-k(l_i/N - c)}}, \tag{5}$$

where $l_i \in [0, N-1]$ represents the inserted layer index, and N is the total number of layers within the LLM. The constant k controls the steepness of the sigmoid curve, while c determines the center offset of the function.

When the interpolation layer is inserted in the lower half of the model (i.e., $l_i/N < 0.5$), α_i decreases, bringing the interpolation result closer to the parameters of the lower layers, thereby preserving local grammatical and lexical information. As the layer is inserted near the middle (i.e., $l_i/N \approx 0.5$), α_i approaches 0.5, which results in a balanced combination of parameters from both lower and higher layers, merging local and global information. When the interpolation is inserted into the upper layers (i.e. $l_i/N > 0.5$), the interpolation approaches the parameters of the higher layers, emphasizing global semantics and factual consistency. Overall, this scheduling mechanism facilitates smooth transitions among the premature layers and optimizes the consistent flow of information, enhancing both local details and global coherence across the model.

4 Experiment

4.1 Datasets

We conduct experiments on four widely recognized benchmark datasets: **TruthfulQA** (Lin et al., 2022), **FACTOR** (Muhlgay et al., 2024), **StrategyQA** (Geva et al., 2021), and **GSM8K** (Cobbe et al., 2021). These datasets are used to assess hallucination mitigation and reasoning capabilities in LLMs.

TruthfulQA is a comprehensive dataset designed to evaluate the factual accuracy of large language models. Following previous work, we adopt a multiple-choice question format for our experiments. The dataset consists of 817 multiple-choice questions spanning 38 diverse domains. Performance is measured using three key metrics: **MC1**, **MC2**, **MC3**, which evaluate the probability of correct and incorrect answers from three different perspectives. **FACTOR** dataset consists of three subdatasets—**News**, **Wiki**, **Expert**—which serve as a benchmark for content completion and factuality evaluation. The primary evaluation metric is accuracy, which measures the factual correctness of text completions generated by LLMs.

In addition, to evaluate the effectiveness of our method on generation tasks, we adopt two datasets focused on chain-of-thought (CoT) reasoning: **StrategyQA** and **GSM8K**. StrategyQA consists of 2288 question-answer pairs designed to test common sense reasoning and logical inference in language models. GSM8K is a widely used benchmark in the LLM era, containing over 8,000 high-quality, graduate-school-level math problems. It serves as a key standard for evaluating mathematical reasoning and generation capabilities in LLMs. For both datasets, we use accuracy as the primary evaluation metric. Notably, these datasets primarily consist of general knowledge tasks, making them valuable for assessing the overall reasoning abilities of LLMs beyond just hallucination mitigation.

4.2 Baselines

To evaluate the effectiveness of our proposed PLI, we compare it against the following hallucination mitigation methods: **Base**: The original LLM without any modifications. **CD** (Li et al., 2023c): A classic CD method using two different-scale LLMs of the same type. Due to model size constraints, following (Zhang et al., 2023a, 2024b), we perform CD between 13B-Chat and 7B-Chat only on LLAMA2. **ITI** (Li et al., 2023b): A method that mitigates hallucinations by editing the activation of attention heads during inference. **DoLa** (Chuang et al., 2023): A CD approach that contrasts early-exit layers with the final output layer to improve factual accuracy. **SH2** (Kai et al., 2024): A method that introduces low-confidence tokens at the inference stage to adjust output probabilities, encouraging the model to reassess its responses for improved truthfulness. **ICD** (Zhang et al., 2023a): A state-of-the-art CD method that reduces hallucination by contrasting the base model with a hallucinated version.

4.3 Experiments Results

Table 1 present the performance of baseline methods and their PLI-enhanced variants, demonstrating that our proposed PLI significantly improves the performance of all three base models and existing hallucination mitigation techniques across multiple datasets. Additional results are provided in §A.1. These results highlight PLI's effectiveness in enhancing factuality in LLMs through the insertion of premature layers, calculated via mathematical interpolation. Further details on completion details and insertion position selection are provided in §A.2.

Specifically, on TruthfulQA, PLI consistently improves performance across all methods. Notably, on LLAMA3-8B, it yields the largest gains for ICD (+1.59 points in MC1) and DoLa (+2.64 points in MC1). On FACTOR, PLI achieves consistent, albeit smaller, improvements for all base models, particularly in the News and Wiki subsets. This suggests that its impact is most pronounced in tasks requiring fine-grained factual reasoning. For reasoning and generation tasks on StrategyQA and GSM8K, PLI also demonstrates notable gains, improving accuracy by up to 0.5 points on StrategyQA and an average of 0.4 points on GSM8K for LLAMA3-8B-Instruct and Mistral-7B-Instruct. This underscores PLI's broader applicability beyond factuality enhancement. However, results for CD are mixed. While it performs well in some cases, it shows slight degradation on LLAMA2-7B-Chat (-0.45 points in MC1 and a similar decline on FACTOR). This is likely due to conflicts between CD's contrastive mechanism and PLI's interpolation-based approach. ITI, which modifies model activations, can be prone to perturbations. However, PLI assists it to achieve more improvement. Furthermore, when integrated with ICD, a strong CD method, PLI achieves the best overall performance in most cases. An extra analysis of statistical significance is conducted in Section §A.7.

Method	TruthfulQA			FACTOR			CoT	
	MC1	MC2	MC3	News	Wiki	Expert	StrQA	GSM8K
LLAMA3-8B-Instruct								
Base	43.13	61.26	33.89	**70.44**	59.15	63.78	**72.67**	75.78
Base + **PLI**	**43.90**	**61.63**	**34.21**	70.11	**59.62**	**64.22**	72.41	**76.29**
ITI	43.39	61.53	33.94	60.19	47.22	52.76	68.17	69.20
ITI + **PLI**	**43.76**	**62.74**	**35.21**	**62.07**	**49.37**	**53.60**	**68.30**	**70.21**
SH2	43.30	64.47	36.23	70.80	59.69	64.22	**72.38**	76.92
SH2 + **PLI**	**45.62**	**66.20**	**38.57**	**71.24**	**60.08**	**64.76**	72.13	**77.23**
DoLa	42.96	65.76	35.71	70.10	59.37	64.06	72.22	76.30
DoLa + **PLI**	**45.60**	**67.34**	**37.62**	**70.53**	**59.84**	**64.65**	**72.61**	**76.92**
ICD	61.76	**79.63**	58.90	71.99	60.55	65.51	72.45	77.35
ICD + **PLI**	**63.35**	79.22	**59.47**	**72.42**	**61.10**	**65.87**	**72.85**	**77.92**
Mistral-7B-Instruct-v0.2								
Base	55.26	72.08	44.33	74.50	**60.38**	65.51	**67.87**	43.09
Base + **PLI**	**56.00**	**72.36**	**45.71**	**75.20**	59.63	**66.12**	67.55	**43.54**
ITI	55.12	72.30	44.87	68.76	56.74	59.82	62.33	**39.70**
ITI + **PLI**	**56.20**	**72.82**	**45.21**	**69.30**	**57.35**	**61.47**	**62.89**	39.44
SH2	54.29	75.32	47.80	74.93	61.02	67.40	67.54	43.37
SH2 + **PLI**	**55.43**	**76.10**	**49.75**	**75.24**	**61.40**	**67.82**	**67.93**	**43.62**
DoLa	52.19	77.85	48.21	**73.82**	60.62	66.70	67.74	42.33
DoLa + **PLI**	**54.35**	**78.32**	**48.70**	73.25	**61.02**	**67.22**	**68.01**	**42.70**
ICD	62.62	79.83	56.37	75.76	60.92	**68.95**	68.17	42.07
ICD + **PLI**	**65.56**	**80.49**	**58.97**	**76.20**	**61.73**	68.52	**69.42**	**42.93**

Table 1: Experimental results on LLAMA3-8B-Instruct and Mistral-7B-Instruct-v0.2 across TruthfulQA, FACTOR, StrategyQA (StrQA), and GSM8K datasets. The **bolded** values indicate the better result in pairwise comparisons, while the **bolded** and **underlined** values represent the best overall result for each benchmark. Our method achieves superior performance in most cases, demonstrating its effectiveness in hallucination mitigation and reasoning.

5 Analysis

5.1 Inference Consumption Testing

Since our proposed method inserts premature layers, calculated via mathematical interpolation, into LLMs, it is important to assess its impact on inference efficiency. To evaluate this, we measure the inference time and GPU memory usage of LLMs before and after applying PLI, comparing these results with other hallucination mitigation methods. For this analysis, we use LLAMA3-8B-Instruct as an example, with results shown in Table 2. Here, **PLI*n** denotes the number of inserted layers, and the integrations with DoLa and ICD remain consistent with the main experiment. We report the average inference time per sample of data and calculate the percentage of memory increase.

Minimal impact on GPU memory usage: PLI does not significantly affect memory consumption, as it only inserts specific layers during inference time, leaving model loading and tokenization largely unchanged.

Inference time remains efficient: While PLI introduces slight overhead due to additional layers and parameters, the impact is relatively small compared to its factuality improvements.

Method	Datasets (s/sample)			Memory (Mib)
	TruthfulQA	StrQA	GSM8K	
Base	0.042	0.114	0.032	17,143
Base + **PLI*1**	**0.046**	**0.118**	**0.033**	**17,430(+1.67%)**
Base + **PLI*2**	**0.047**	**0.124**	**0.034**	**17,486(+2.00%)**
Base + **PLI*3**	**0.049**	**0.127**	**0.035**	**17,673(+3.09%)**
Dola	0.045	0.117	0.033	17,213
Dola + **PLI**	**0.048**	**0.126**	**0.034**	**17,463(+1.45%)**
ICD	0.094	0.243	0.065	34,788
ICD + **PLI**	**0.098**	**0.254**	**0.069**	**35,023(+0.67%)**

Table 2: Inference time and memory usage across different settings of three benchmarks based on LLAMA3-8B-Instruct.

Compared with other methods, DoLa has slightly higher inference costs than the base model due to contrastive decoding between the final and early-exit layers. ICD incurs substantially higher computational costs, as it requires loading two models and computing their logits separately.

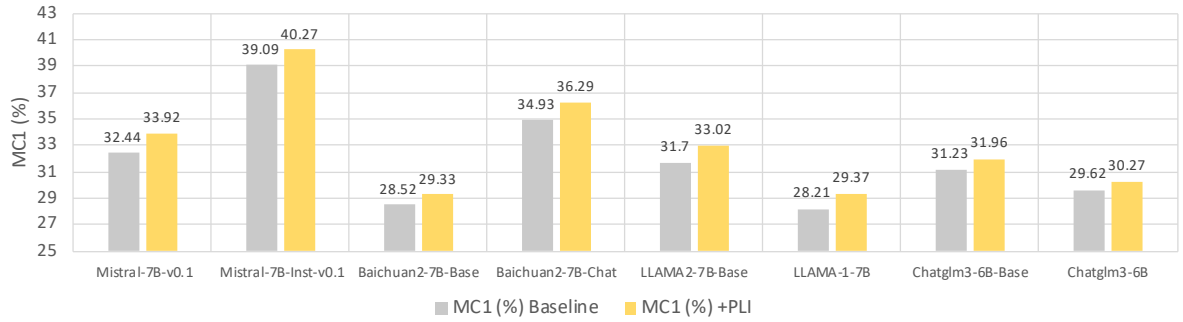

Figure 3: Experimental results on TruthfulQA (MC1 %) across eight different base models.

5.2 Adaptability Testing on More Base Models

To assess the adaptability of our method, we apply PLI to eight different LLMs and report the performance improvements on the TruthfulQA benchmark in Figure 3. Experimental results demonstrate that PLI consistently mitigates hallucinations and enhances factuality across various model architectures and scales, with minimal computational overhead due to its plug-and-play nature. Notably, PLI achieves a performance gain of 1.48 points on Mistral-7B-v0.1 and 1.32 points on LLAMA2-7B-Base. On average, PLI improves performance by 1.08 points across the eight evaluated LLMs.

5.3 Interpolation Techniques Comparison

The proposed PLI method uses **Slerp** (spherical linear interpolation) as the interpolation strategy, as detailed in Section 3.1. In this section, we compare the impact of different mathematical interpolation methods on the performance of PLI. The following interpolation methods are considered:

Linear Interpolation (Lerp): This method is computationally efficient but lacks the directional consistency needed in high-dimensional spaces. The parameters for interpolation are computed as:

$$\mathbf{w}_{\text{inter}} = (1-\alpha)\mathbf{w}_l + \alpha\mathbf{w}_{l+1} \tag{6}$$

where $\mathbf{w}_l$ and $\mathbf{w}_{l+1}$ are the parameters of the layers at positions l and $l+1$, and α is the interpolation ratio.

Bézier Curve Interpolation (BCerp): This interpolation method adds curvature continuity by extending the linear interpolation to a quadratic Bézier curve with an additional control point $\mathbf{h}$, making the transitions smoother but more computationally expensive. The interpolated parameters are calculated as:

$$\mathbf{w}_{\text{inter}} = (1-\alpha)^2\mathbf{w}_l + 2\alpha(1-\alpha)\mathbf{h} + \alpha^2\mathbf{w}_{l+1} \tag{7}$$

$$\mathbf{h} = (\mathbf{w}_l + \mathbf{w}_{l+1})/2 \tag{8}$$

where α is the interpolation ratio, and $\mathbf{h}$ is the control point for the curve.

To compare these interpolation techniques, we conduct experiments on TruthfulQA using LLAMA3-8B and LLAMA2-7B-Chat, integrating the interpolation methods with ICD and a mixture of interpolation methods. The experimental results are shown in Table 3. **PLI** refers to the results obtained with the standard PLI approach (using **Slerp** as in the main experiments). **PLI (Mix)** integrates different interpolation techniques for various layers: **Slerp** is used in lower layers (e.g., the 8th layer), and **Lerp** is applied to higher layers (e.g., the 24th and 28th layers). The results show that all interpolation strategies help alleviate hallucination, outperforming ICD in most cases. The **PLI (Mix)** approach achieves promising performance, likely because the use of **Slerp** in the low layers preserves semantic consistency across the dimensional space, while **Lerp** in the higher layers efficiently maintains factual consistency with fewer disturbances.

5.4 PLI Mechanism Exploration

This section explores the mechanism of the PLI method by analyzing how it interacts with the internal functioning of LLMs. Specifically, we investigate the relationship between hidden states output by consecutive layers and the effectiveness of PLI. The experiment begins by defining two normal distributions based on the mean and standard deviation of the hidden states output by each layer and its subsequent layer. The Kullback-Leibler (KL) divergence is calculated between these distributions to gauge how they change under different interpolation schemes. Notably, the KL divergence between the second-to-last and last layers provides insights into the effectiveness of PLI. Results on the TruthfulQA benchmark, shown in Table 4, demonstrate

Method	TruthfulQA		
	MC1	MC2	MC3
LLAMA3-8B-Instruct			
Base	43.13	61.26	33.89
ICD	61.76	**79.63**	58.90
ICD + **PLI**	63.35	79.22	**59.47**
ICD + **PLI (Lerp)**	**64.37**	78.72	58.22
ICD + **PLI (BCerp)**	63.10	78.13	58.45
ICD + **PLI (Mix)**	63.56	78.27	59.04
LLAMA2-7B-Chat			
Base	37.00	54.65	27.82
ICD	45.09	69.10	41.59
ICD + **PLI**	46.69	**70.70**	**43.52**
ICD + **PLI (Lerp)**	46.81	69.70	42.53
ICD + **PLI (BCerp)**	47.42	69.72	42.78
ICD + **PLI (Mix)**	**48.03**	70.49	43.05

Table 3: Experimental results on TruthfulQA with different interpolation techniques applied to LLAMA3-8B-Instruct and LLAMA2-7B-Chat.

Method	TruthfulQA			KL Div
	MC1	MC2	MC3	
LLAMA2-7B-Chat				
Base	37.00	54.65	27.82	153.6
Base + **PLI** ($l_i = 24, 28$)	**37.62**	**54.99**	**28.32**	**178.2**(↑)
Base + **PLI** ($l_i = 4, 8$)	35.41	54.23	27.15	123.3(↓)
Mistral-7B-Instruct-v0.2				
Base	55.26	72.08	44.33	132,096
Base + **PLI** ($l_i = 24, 28$)	**56.00**	**72.36**	**45.71**	**174,080**(↑)
Base + **PLI** ($l_i = 4, 8$)	54.32	71.21	43.86	122,856(↓)

Table 4: Changes in the KL divergence of the hidden states distribution of the second-to-last and last layer of the model on the TruthfulQA under different PLI schemes.

that when PLI is effective, the KL divergence between the second-to-last and the last layer increases significantly. Conversely, when PLI is less effective, the KL divergence decreases. This suggests that PLI influences how the final layer of the model outputs tokens, with a higher divergence indicating that the final layer is better at outputting factual information. The final layer tends to restore tokens according to language priors, and PLI enhances its ability to output factually accurate words, increasing the KL divergence between the hidden states of the second-to-last and last layers. This finding helps explain why the PLI method is effective in enhancing the factuality of LLMs: by adjusting the hidden states through premature layers, PLI helps the final layer output more accurate and factually consistent tokens.

Furthermore, we conduct an experiment, as shown in Table 5. Specifically, we evaluate the

Method	TruthfulQA		
	MC1	MC2	MC3
LLAMA3-8B-Instruct			
Base (final layer)	43.13	61.26	33.89
Base + **PLI** (final layer)	**43.90**	**61.63**	**34.21**
Base (26th layer)	**24.63**	50.79	25.46
Base + **PLI** (**27th layer**)	23.89	**50.85**	**25.50**
Base (30th layer)	26.83	52.23	26.64
Base + **PLI** (**32nd layer**)	**27.94**	**53.03**	**27.31**
Mistral-7B-Instruct-v0.2			
Base (final layer)	55.26	72.08	44.33
Base + **PLI** (final layer)	**56.00**	**72.36**	**45.71**
Base (26th layer)	**25.00**	50.12	25.57
Base + **PLI** (**27th layer**)	24.75	**50.21**	**25.98**
Base (30th layer)	25.36	51.28	26.66
Base + **PLI** (**32nd layer**)	**25.85**	**51.41**	**26.72**

Table 5: Experimental results of the final layer and early-exit layer of the models before and after PLI execution (both $l_i = 24, 28$) on TruthfulQA.

output of early-exit layers (26th and 30th selected, becoming the 27th and 32nd after PLI) and the final layer from LLMs before and after PLI execution (with insertion locations $l_i = 24, 28$). The results show that, in addition to improving the factuality of the final layer, PLI also enhances the output of the premature layers within the model. This indicates that our method effectively optimizes the output representation of the middle layers by expanding the information processing flow through the insertion of premature layers, which helps alleviate hallucination in the final layer's output.

5.5 Case Study

To further validate the effect of our proposed method, we conduct a case study on the generated content before and after the execution of PLI based on LLAMA3-8B-Instruct, using StrategyQA, a benchmark focused on generation and logical reasoning. The case study is summarized in Table 6. We observe that PLI significantly improves the factuality of the generated text by inserting premature layers into the LLMs. Additionally, PLI causes minimal interference, as evidenced by the high consistency between the two types of generated text. This demonstrates that PLI alleviates hallucination in the content without affecting the overall quality of the text generation.

6 Conclusion

In this paper, we introduce a novel method called PLI (Premature Layers Interpolation), designed to

LLAMA3-8B-Instruct	LLAMA3-8B-Instruct + PLI
Question: *Can an art dealer buy Boeing 737-800 with a Da Vinci painting?*	
The Boeing 737-800 is a commercial airliner. It costs tens of millions of dollars. A Da Vinci painting is a work of art. It could cost tens of millions of dollars. However, it is unlikely that an art dealer would have the funds to buy a Boeing 737-800 with a Da Vinci painting. So the answer is no.	The price of a Boeing 737-800 is around $80 million. The price of a Da Vinci painting can be tens of millions of dollars, but it would likely be less than $80 million. Thus, an art dealer could potentially buy a Boeing 737-800 with a Da Vinci painting. So the answer is yes.
Question: *Would a clouded leopard encounter an awake pangolin?*	
Clouded leopards are found in Southeast Asia, while pangolins are found in Africa. These are two different continents. Thus, it is unlikely that a clouded leopard and an awake pangolin would encounter each other. So the answer is no.	Clouded leopards are found in Southeast Asia and China. Pangolins are found in Africa and Asia. There is some overlap in their ranges, but they are not the same. Thus, it is possible for a clouded leopard and an awake pangolin to encounter each other. So the answer is yes.
Question: *Is Phobos (moon) name origin similar to Roman god Pavor?*	
Phobos is the name of a moon of Mars. Phobos is the Greek god of fear. Pavor is the Latin word for fear. The name Phobos comes from Greek mythology, not Latin. Thus, the name origin is not similar. So the answer is no.	Phobos is the Greek word for fear. Pavor is the Latin word for fear. Both words come from the same root. Thus, the name origin of Phobos is similar to the Roman god Pavor. So the answer is yes.

Table 6: Case study of PLI, showcasing the generation results of hallucination alleviation using PLI based on LLAMA3-8B-Instruct on StrategyQA. Green text denotes factual content, while red text indicates hallucinated content.

enhance the factuality of LLMs. PLI is a training-free, plug-and-play intervention that optimizes the output representation of intermediate layers by inserting premature layers calculated through mathematical interpolation. This process expands the information flow within the model, thereby improving the factual accuracy of the final output and alleviating hallucinations. Our experiments on TruthfulQA, FACTOR, StrategyQA, and GSM8K datasets demonstrate that PLI outperforms other baseline methods in most cases. Additionally, we conduct a detailed analysis to explore the mechanism and effectiveness of the proposed approach.

7 Limitations

While PLI alleviates hallucinations by inserting premature layers constructed through mathematical interpolation, it introduces additional computational overhead, with the extent of this overhead varying across different models and tasks. Previous work (Geva et al., 2023) has pointed out that the high layers within models are tend to process and utilize factual information, while the information in the high layers is more redundant (Gromov et al., 2024), which may lead to our method having less impact on model performance in some cases. In the future, more explorations are needed for the internal mechanism of the model from the perspective of interpretability to further refine the PLI.

Acknowledgement

This work was supported by Innovation Team Project of Guangdong Province of China (No. 2024KCXTD017), Shenzhen Science and Technology Foundation (No. JCYJ20240813145816022), National Key Research and Development Program of China (2024YFF0908200), National Natural Science Foundation of China (Grant No. 62376262) and the Natural Science Foundation of Guangdong Province of China (2024A1515030166, 2025B1515020032).

References

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

AI@Meta. 2024. Llama 3 model card.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*.

Dingwei Chen, Feiteng Fang, Shiwen Ni, Feng Liang, Ruifeng Xu, Min Yang, and Chengming Li. 2024a. Lower layer matters: Alleviating hallucination via multi-layer fusion contrastive decoding with truthfulness refocused. *arXiv preprint arXiv:2408.08769*.

Zhongzhi Chen, Xingwu Sun, Xianfeng Jiao, Fengzong Lian, Zhanhui Kang, Di Wang, and Chengzhong Xu. 2024b. Truth forest: Toward multi-scale truthfulness in large language models through intervention without tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20967–20974.

Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Think more, hallucinate less: Mitigating hallucinations via dual process of fast and slow thinking. *arXiv preprint arXiv:2501.01306*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics (TACL)*.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. 2024. The unreasonable ineffectiveness of the deeper layers, 2024. *URL https://arxiv. org/abs/2403.17887*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Anshita Gupta, Debanjan Mondal, Akshay Sheshadri, Wenlong Zhao, Xiang Li, Sarah Wiegreffe, and Niket Tandon. 2023. Editing common sense in transformers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8214–8232.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023b. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Jushi Kai, Tianhang Zhang, Hai Hu, and Zhouhan Lin. 2024. Sh2: Self-highlighted hesitation helps you decode more truthfully. *arXiv preprint arXiv:2401.05930*.

Vedang Lad, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference? In *ICML 2024 Workshop on Mechanistic Interpretability*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2023a. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1774–1793.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. *arXiv preprint arXiv:2306.03341*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023c. 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)*, pages 12286–12312.

Songshi Liang, Hongda Sun, Ting-En Lin, Yuchuan Wu, Zihe Wang, Yongbin Li, and Rui Yan. 2024. Locate-then-unlearn: An effective method of multi-task continuous learning for large language models.

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.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023a. Chain of hindsight aligns language models with feedback. In *The Twelfth International Conference on Learning Representations*.

J Liu, J Jin, Z Wang, J Cheng, Z Dou, and J Wen. 2023b. Reta-llm: A retrieval-augmented large language model toolkit. arxiv, abs/2306.05212.

Jinliang Lu, Ziliang Pang, Min Xiao, Yaochen Zhu, Rui Xia, and Jiajun Zhang. 2024. Merge, ensemble, and cooperate! a survey on collaborative strategies in the era of large language models. *arXiv preprint arXiv:2407.06089*.

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.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2024. Generating benchmarks for factuality evaluation of language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 49–66.

Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. 2023. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. *arXiv preprint arXiv:2311.08011*.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Scott M Robeson. 1997. Spherical methods for spatial interpolation: Review and evaluation. *Cartography and Geographic Information Systems*, 24(1):3–20.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*, 2(5):9.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037.

Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv preprint arXiv:2308.11761*.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ping Luo, and Ying Shan. 2024. Llama pro: Progressive llama with block expansion. *arXiv preprint arXiv:2401.02415*.

Dingkang Yang, Dongling Xiao, Jinjie Wei, Mingcheng Li, Zhaoyu Chen, Ke Li, and Lihua Zhang. 2024a. Improving factuality in large language models via

decoding-time hallucinatory and truthful comparators. *arXiv preprint arXiv:2408.12325*.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024b. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore. Association for Computational Linguistics.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024a. A comprehensive study of knowledge editing for large language models.

Shaolei Zhang, Tian Yu, and Yang Feng. 2024b. Truthx: Alleviating hallucinations by editing large language models in truthful space. *arXiv preprint arXiv:2402.17811*.

Yue Zhang, Leyang Cui, Wei Bi, and Shuming Shi. 2023a. Alleviating hallucinations of large language models through induced hallucinations. *arXiv preprint arXiv:2312.15710*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023b. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876.

A Appendix

A.1 The Experimental Results based on LLAMA2-7B-Chat

We present our experimental results based on LLAMA2-7B-Chat on the TruthfulQA, Factor, StrategyQA and GSM8K benchmarks, which are shown in Table 7 and illustrated in §4.3.

A.2 Completion Details

For our experiments, we use LLAMA3-8B-Instruct (AI@Meta, 2024), Mistral-7B-Instruct-v0.2, and LLAMA2-7B-Chat (Touvron et al., 2023) as the base models. Given that these models primarily consist of 32-layer transformers, we streamline the insertion of premature layers by following insights from previous work (Chuang et al., 2023). Specifically, we select the following candidate insertion positions: $l_i = \{4, 8, 12, 16, 20, 24, 28\}$. These positions are chosen to align with the information distribution across layers in LLMs: low layers $\{4, 8\}$, middle layers $\{12, 16, 20\}$, and high layers $\{24, 28\}$, respectively. During experiments, we randomly insert 1–3 premature layers across different models and downstream tasks. The insertion ratio is dynamically computed using the scheduling formula from Section §3.3, ensuring an extended information processing flow and optimized intermediate representations. Since PLI is a plug-and-play method, we integrate it into the base models as well as various hallucination mitigation techniques to validate its effectiveness. All experiments are conducted on a single NVIDIA A800-80G GPU.

For the base models (i.e., greedy decoding) and contrastive decoding or representation editing methods within single models, we select insertion positions $l_i = \{24, 28\}$. For the ICD method, we use $l_i = \{8, 24, 28\}$ to emphasize the truthfulness of the models for a better contrast in results. All other experimental settings adhere to those used in previous works when comparing to other baseline methods.

While for the hyperparameters in the formula of our proposed Adaptive Interpolation Ratio Scheduling (as illustrated in 3.3). The constant k is set to 4 by default to control the steepness of the sigmoid curve, while c is set to 0.375 by default to determine the center offset of the function. Noting that the default settings of these two parameters are mainly suitable for LLMs with 32 layers. For models of larger sizes, k and c need to be adjusted to control the sigmoid curve and center offset, respectively. Nevertheless, it is convenient for mainly following the relationship that the inserted interpolation layer position and the interpolation ratio are positively correlated.

A.3 Analysis of Layer Insertion Position

This section explores the effect of our proposed PLI at different positions in the model. Taking the large language model of 32 layers of transformers as an example, we use LLAMA3-8B-Instruct to conduct experiments on the TruthfulQA benchmark with different interpolation combinations in $l_i = \{4, 8, 12, 16, 20, 24, 28\}$. The results are shown in Table 8.

Method	TruthfulQA		
	MC1	MC2	MC3
LLAMA3-8B-Instruct			
Base	43.13	61.26	33.89
PLI ($l_i = 24, 28$)	**43.90**	**61.63**	<u>34.21</u>
PLI ($l_i = 24$)	42.89	<u>61.45</u>	**34.24**
PLI ($l_i = 12$)	42.60	61.07	33.95
PLI ($l_i = 12, 16$)	43.13	61.23	33.79
PLI ($l_i = 12, 16, 28$)	42.76	60.95	33.57
PLI ($l_i = 12, 24, 28$)	<u>43.42</u>	61.02	33.82
PLI ($l_i = 12, 16, 24, 28$)	42.31	60.52	33.42
PLI ($l_i = 12, 16, 20, 24, 28$)	42.47	60.82	33.62
PLI ($l_i = 4, 12, 16, 24, 28$)	41.80	60.03	32.88

Table 8: Experimental results on TruthfulQA with different settings of insertion position based on LLAMA3-8B-Instruct.

Experimental results show that when the number of inserted interpolation layers is usually 2-3, PLI has a positive effect on hallucination alleviation, while the effect decreases when it is less or more. This may be because a single interpolation layer has little impact on large language models; yet more interpolation layers will inject more linear characteristics into LLMs, which will weaken the representation ability of the model. Meantime, it indicates that when the interpolation position is concentrated in the upper layer (such as 24, 28 layers), it results in better effectiveness, which may be because the upper layer of LLMs is closer to high-order semantic abstraction and factual information.

A.4 Theoretical Interpretability of Inserted Premature Layers

The PLI method we proposed mainly regards the LLMs from input to output as an information processing flow. From this perspective, there are two main reasons why our method is effective: (1) PLI

Method	TruthfulQA			FACTOR			CoT	
	MC1	MC2	MC3	News	Wiki	Expert	StrQA	GSM8K
LLAMA2-7B-Chat								
Base	37.00	54.65	27.82	**64.71**	56.61	64.85	63.67	21.64
Base + **PLI**	**37.62**	**54.99**	**28.32**	64.45	**56.90**	**65.33**	**64.03**	**21.96**
ITI	37.01	54.66	27.82	53.28	43.82	51.69	58.74	**17.86**
ITI + **PLI**	**37.63**	**55.48**	**28.22**	**54.52**	**45.22**	**53.47**	**59.11**	17.4
SH2	33.9	57.07	29.79	65.31	57.37	**67.22**	64.4	**22.17**
SH2 + **PLI**	**35.49**	**57.62**	**30.44**	**65.53**	**57.64**	66.89	**64.81**	22.09
CD	**28.15**	54.87	29.75	**64.57**	**58.47**	67.12	58.42	15.04
CD + **PLI**	27.70	**55.92**	**31.46**	64.06	58.03	**67.33**	**60.12**	**16.30**
DoLa	32.97	60.84	29.50	64.32	57.63	67.30	64.16	22.07
DoLa + **PLI**	**34.79**	**62.10**	**31.71**	**64.75**	**57.90**	**68.43**	**64.52**	**22.49**
ICD	45.09	69.10	41.59	65.20	56.57	67.66	64.37	21.72
ICD + **PLI**	**46.69**	**70.70**	**43.52**	**65.73**	**56.98**	**69.22**	**65.59**	**22.81**

Table 7: Experimental results on LLAMA2-7B-Chat across TruthfulQA, FACTOR, StrategyQA (StrQA) and GSM8K datasets. To highlight, the **bolded** result indicates the better one in the pairwise comparison, while the **bolded** and **underlined** result indicates the best for that benchmark. Our method outperforms other baselines in most cases across the four benchmarks.

extends the length of the information flow and the processing depth; (2) PLI maintains the continuity of the information flow and the knowledge manifold. Specifically, we are inspired by stable diffusion, which improves the quality of image processing generation by adjusting the sampling step. We migrate similar concepts to large language models and expand the number of model layers to extend the length of information flow transmission and promote the LLMs' capture and abstraction of information. On the other hand, the premature layers we inserted are calculated through mathematical interpolation (i.e., Slerp), which helps to maintain the directionality of vectors in high-dimensional space, and thus maintain the original manifold and continuity at the knowledge level. Our method needs to maintain a gbalance between the two properties to better alleviate hallucinations while maintaining the stability of the model architecture.

A.5 Analysis of PLI with Models across Different Sizes

We conducted experiments based on LLAMA2-13B-Chat (40 layers) and LLAMA2-70B-Chat (80 layers) on TruthfulQA dataset to demonstrate the adaptation of our proposed PLI on models across different sizes and layers, which are beyond 32 layers. The experimental results are shown in the following table 9.

Method	TruthfulQA		
	MC1	MC2	MC3
LLAMA2-13B-Chat			
Base	37.62	54.60	28.12
Base + **PLI**	**38.80**	**55.39**	**28.92**
ICD	48.47	73.47	46.04
ICD + **PLI**	**49.32**	**73.86**	**46.33**
LLAMA2-70B-Chat			
Base	38.79	58.99	30.59
Base + **PLI**	**39.52**	**59.40**	**31.06**
ICD	53.24	78.11	49.14
ICD + **PLI**	**54.07**	**79.60**	**49.54**

Table 9: Experimental results on TruthfulQA based on LLAMA2-13B-Chat and LLAMA2-70B-Chat.

From the experimental results, we can see that our proposed method can perform well on larger size models. When the model itself contains more parameters, PLI need to insert more layers than the 7B model to optimize the performance of the model due to its original deeper information processing depth (for example, generally 5-6 layers are inserted on the 70B model). We insert 6 premature layers in our experiments based on LLAMA2-70B-Chat, mainly in layers 50-80, to achieve the results shown in our table.

A.6 Experimental results on Open LLM Leaderboard

In order to test PLI's impact on other capabilities with more benchmarks, we conduct experiments on Open LLM Leaderboard based on LLAMA3-8B-

Instruct and Mistral-7B-Instruct-v0.2. The results are shown in the following table 10. Experimental results show that the proposed PLI can improve the performance of the original model in multiple test scenarios across different base models (bringing an average improvement of about 1-2%), which further verifies the generalization of our method.

A.7 Analysis of Statistical Significance

To further verify the stability of the PLI we proposed, we have selected several baselines, and ran 5 new rounds on TruthfulQA dataset based on LLAMA3-8B-Chat, calculated new means, then marked the numerical changes and p-test values, as shown in the following table 11.

Method	TruthfulQA		
	MC1	MC2	MC3
LLAMA3-8B-Instruct			
Base	43.31 (±0.42)	61.48 (±0.56)	33.76 (±0.38)
Base + **PLI**	**44.12 (±0.47)**	**61.92 (±0.51)**	**34.43 (±0.45)**
p-value	*0.018**	*0.043**	*0.022**
DoLa	42.83 (±0.49)	65.99 (±0.58)	35.96 (±0.43)
DoLa + **PLI**	**45.83 (±0.53)**	**67.59 (±0.54)**	**37.48 (±0.47)**
p-value	*0.0007***	*0.007***	*0.003***
ICD	62.04 (±0.53)	**79.49 (±0.62)**	58.72 (±0.48)
ICD + **PLI**	**63.65 (±0.52)**	79.08 (±0.59)	**59.68 (±0.51)**
p-value	*0.011**	*0.046**	*0.025**

Table 11: Presentation of statistical significance on TruthfulQA based on LLAMA3-8B-Instruct.

Noting that * indicates p-value < 0.05, ** indicates p-value < 0.01, and *** indicates p-value < 0.001 in the table, representing different statistical significance levels. It can be seen from the table that compared with the baselines, **improvements brought by PLI are basically significant**, which means that our method could stablely promote the baselines, verifying the robustness of our method.

Method	Average	IFEval	BBH	MATH	CPQA	MUSR	MMLU-P
LLAMA3-8B-Instruct	22.68	64.98	28.01	10.27	0.78	2.00	30.32
LLAMA3-8B-Instruct + **PLI**	**23.77**	**65.70**	**29.27**	**10.42**	**1.96**	**2.75**	**32.54**
Mistral-7B-Instruct-v0.2	14.22	22.66	23.95	3.02	5.59	8.36	21.70
Mistral-7B-Instruct-v0.2 + **PLI**	**15.01**	**23.42**	**24.79**	**3.33**	**6.02**	**8.90**	**23.65**

Table 10: Experimental results on Open LLM Leaderboard based on LLAMA3-8B-Instruct and Mistral-7B-Instruct-v0.2.