Learning from Hallucinations: Mitigating Hallucinations in LLMs via Internal Representation Intervention

Sora Kadotani **Kosuke Nishida** **Kyosuke Nishida**
NTT Human Informatics Labs., NTT, Inc.
{sora.kadotani, kosuke.nishida, kyosuke.nishida}@ntt.com

Abstract

Large language models (LLMs) sometimes hallucinate facts. Recent studies have shown that use of non-factual LLMs (anti-expert) have the potential to improve the factuality of the base LLM. Anti-expert methods penalize the output probabilities of the base LLM with an anti-expert LLM. Anti-expert methods are effective in mitigating hallucinations, but require high computational costs because the two LLMs are run simultaneously. In this paper, we propose an efficient anti-expert method called in-model anti-expert. It mitigated the hallucination problem with a single LLM and intervening to change the internal representations in the direction of improving factuality. Experiments results showed that the proposed method is less costly than the conventional anti-expert method and outperformed existing methods except for the anti-expert method. We confirmed that the proposed method improved GPU memory usage from 2.2x to 1.2x and latency from 1.9x to 1.2x.

1 Introduction

Large language models (LLMs) (OpenAI, 2024; Nvidia, 2024) demonstrate impressive capabilities. However, LLMs sometimes generate plausible but factually incorrect information, called hallucinations (Ji et al., 2023; Zhang et al., 2023b). Hallucinations degrade the reliability of applications; hence, it is important to detect and mitigate them.

As a hallucination mitigation method, Zhang et al. (2025) proposed to use a non-factual LLM (*i.e.* anti-expert) to improve the factuality of the base LLM. They obtained the output distribution of a factual LLM (*i.e.* expert) by contrasting the output distributions of the base and anti-expert LLM. In particular, they created the anti-expert LLM by fine-tuning using hallucinated answers, because fine-tuning using factual answers might inadvertently make an LLM to hallucinate

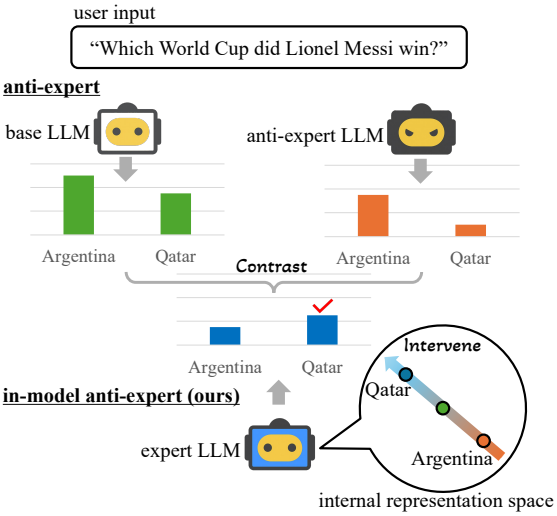


Figure 1: Anti-expert (upper) and in-model anti-expert (lower). Our method produces contrasted output distributions by shifting the internal representations of the base LLM in the direction of improving factuality.

by forcing it to answer questions beyond its knowledge boundaries (Yang et al., 2024). Although it has achieved state-of-the-art performance, their method is computationally expensive because it runs two LLMs simultaneously.

In this paper, we propose an efficient anti-expert method called in-model anti-expert (IMAE). Figure 1 illustrates this method. While an anti-expert contrasts output distributions, our method makes interventions in the internal representations to improve factuality. Internal representation intervention enables the LLM to mitigate hallucinations by itself without running a separate anti-expert. However, it is difficult to implement this method because LLMs need to acquire the ability not only to generate non-factual text but also to identify the directions of improving factuality in the internal representation spaces. We solve this challenge by equipping the LLM with three modes for generating non-factual text, neutral text, and factual text.

Experiments results on TruthfulQA (Lin et al.,

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1136–1143

2022) showed that our method outperformed existing ones except for the anti-expert method. Moreover, compared with the anti-expert method, it reduced the GPU memory usage and latency from 2.2x to 1.4x and from 1.9x to 1.2x, respectively.

2 Background

The preliminaries below explain the anti-expert method, whereas the related work section describes other hallucination mitigation methods.

2.1 Anti-Expert for Hallucination Mitigation

A major challenge of hallucination mitigation was that fine-tuning using factual data degrades the factuality of LLMs (Zhang et al., 2023a). This is because traditional fine-tuning methods might unintentionally make LLMs hallucinate by forcing them to answer questions beyond their knowledge boundaries (Yang et al., 2024).

Zhang et al. (2025) proposed induce-then-contrast decoding (ICD) as an extension of the anti-expert originally proposed in the field of toxicity mitigation (Liu et al., 2021). ICD is a groundbreaking method that uses a non-factual LLM because it is easy for LLMs to learn to generate non-factual text. They created an anti-expert LLM by fine-tuning the base LLM using synthetic hallucination data generated by ChatGPT[1].

During decoding, the method calculates a factual probability $p_{\text{expert}}(\cdot)$ by contrasting the output distributions of the base and anti-expert LLM.

$$p_{\text{expert}}(x_i|x_{<i}) = \text{softmax}(\beta \log p_{\text{base}}(x_i|x_{<i}) - \log p_{\text{anti}}(x_i|x_{<i})) \quad (1)$$

$p_{\text{base}}(\cdot)$ and $p_{\text{anti}}(\cdot)$ represent the probability of the base and anti-expert LLMs, respectively. x_i represents the i-th token, and $x_{<i}$ is all previous tokens. β is a hyperparameter for contrast strength.

Li et al. (2023c) pointed out that penalizing all tokens would degrade generation quality. Therefore, they only penalize a subset of tokens, $\mathcal{V}_{\text{valid}}$.

$$\mathcal{V}_{\text{valid}} = \{x_i \in \mathcal{V} : \text{logit}_{\text{base}}(x_i|x_{<i}) \geq \delta \max_i(\max_\omega \text{logit}_{\text{base}}(\omega|x_{<i}))\}$$

$\text{logit}_{\text{base}}(\cdot)$ represents next-token logits of the base LLM. $\mathcal{V}$ represents the vocabulary. δ is a hyperparameter to control the strength of constraint.

ICD has achieved state-of-the-art performance on TruthfulQA. In paricular, Llama2 (7B) with ICD performed comparably to GPT-4. However, anti-expert methods are computationally expensive. ICD requires 2.2x GPU memory usage, as it runs two LLMs. Its latency increases by 1.9x because ICD requires time to contrast probabilities.

2.2 Related Work

Hallucination in LLMs. Hallucination (Dziri et al., 2022; Zhang et al., 2023b) is a behavior in which LLMs generate content that contradicts the user input (Dale et al., 2023; Rehman et al., 2023), previous context (Shi et al., 2023; Wan et al., 2023), or established fact (Bang et al., 2023; Hu et al., 2023). We focus on fact-conflicting hallucination because it has the potential to cause serious problems in specific domains (Pal et al., 2023).

Hallucination Mitigation. Lee et al. (2023); Li et al. (2023b); Chuang et al. (2024) modified the decoding algorithm. These methods are efficient but less effective because improvement without learning is limited. Zhang et al. (2023a) proposed a fine-tuning method to recognize knowledge boundaries, but it risks excessive conservatism. Zhang et al. (2024) proposed an editing internal representation method. It is effective in in-domain data settings but requiring paired correct and hallucinated answers. The anti-expert mitigates hallucinations effectively without pair data.

3 Proposed Method

We alleviate the increase in memory usage and latency by integrating the anti-expert LLM into the base LLM and shifting the internal representation in the direction of improving factuality.

3.1 Model Architecture

Figure 2 shows the proposed architecture. It is based on parallel adapter (He et al., 2022). We add an anti-expert unit to each MLP layer of the base LLM and a mode control unit. The anti-expert unit consists of an MLP layer and a gate layer. We denote the MLP layer of the base LLM by MLP_{base} and that of the anti-expert unit by MLP_{anti}.

Anti-Expert Unit. The gate layer controls the extent to which the output of the anti-expert unit is considered by calculating $\alpha \in \mathbb{R}$.

$$\alpha = \text{softmax}(Wh + b)_0 \in \mathbb{R}$$

[1] https://chat.openai.com

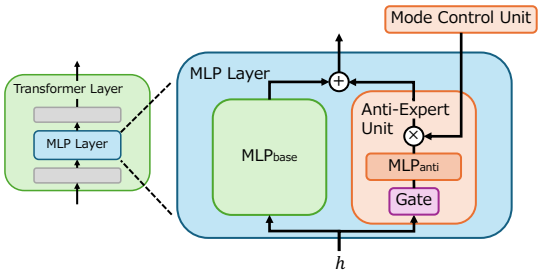


Figure 2: Architecture of IMAE.

$h \in \mathbb{R}^d$ represents an input vector of the MLP layer. $W \in \mathbb{R}^{2\times d}$ and $b \in \mathbb{R}^2$ are learnable parameters. The gate layer outputs an input of $\mathrm{MLP}_{\mathrm{anti}}$.

$$y = \mathrm{MLP}_{\mathrm{anti}}(\alpha h) \in \mathbb{R}^d$$

$\mathrm{MLP}_{\mathrm{base}}$ retains the ability to understand and generate language, so we embed only the information for generating non-factual text in $\mathrm{MLP}_{\mathrm{anti}}$. Therefore, $\mathrm{MLP}_{\mathrm{anti}}$ can be smaller than $\mathrm{MLP}_{\mathrm{base}}$.

Mode Control Unit. IMAE operates in three modes: anti-expert, base, and expert. The mode control unit outputs a scalar $\sigma \in \{-1, 0, 1\}$ corresponding to each mode: 1 for generating non-factual text (anti-expert mode), 0 for replicating the base LLM output (base mode), and -1 for generating factual text (expert mode). We only use expert mode during inference. The output of the MLP layer is calculated as follows:

$$\mathrm{MLP}(h) = \mathrm{MLP}_{\mathrm{base}}(h) + \sigma y$$

3.2 Loss Function

We freeze the parameters of the base LLM and only fine-tune the parameters of the anti-expert unit. We fine-tune the anti-expert unit so that the opposite vector of the $\mathrm{MLP}_{\mathrm{anti}}$ output points in the direction that improves the factuality of the output vector of $\mathrm{MLP}_{\mathrm{base}}$. For fine-tuning, we use a dataset in which each sample consists of a question and its hallucinated answer. We split the dataset in half, using one half for training the anti-expert mode and the other half for training the expert mode. We apply multi-task learning so that the model generates non-factual text in anti-expert mode and generates factual text in expert mode.

We use the cross-entropy loss for the anti-expert mode L_{anti}, and propose a new loss function for the expert mode L_{expert}, where we calculate a target probability with improved factuality, $p_{\mathrm{target}}(\cdot)$. We define $p_{\mathrm{target}}(\cdot)$ as $p_{\mathrm{expert}}(\cdot)$ in Equation 1. We only penalize $\mathcal{V}_{\mathrm{valid}}$ in the same way as ICD. L_{expert} is formulated as follows:

$$L_{\mathrm{expert}} = D_{\mathrm{KL}}(p_{\mathrm{expert}}(x_i|x_{<i})||p_{\mathrm{target}}(x_i|x_{<i}))$$

$p_{\mathrm{expert}}(\cdot)$ and D_{KL} represent the probability of expert mode and Kullback-Leibler divergence.

Besides the loss for $\mathrm{MLP}_{\mathrm{anti}}$, we introduce a loss for the gate layers L_{gate}.

$$L_{\mathrm{gate}} = \begin{cases} 1-\alpha & \text{if } x_i \in T_{\mathrm{fact}} \\ \alpha & \text{if } x_i \notin T_{\mathrm{fact}} \end{cases}$$

T_{fact} is a subset of tokens that affect the factuality of text. Here, let us explain how to identify T_{fact} in §3.3. The gate output α is large when generating tokens in T_{fact}, and small when generating other tokens. As a result, the output of the anti-expert unit is considered only when generating tokens that affect the factuality of text. The total loss L is formulated as: $L = L_{\mathrm{anti}} + L_{\mathrm{expert}} + L_{\mathrm{gate}}$. In preliminary experiments, we confirmed that the weighted sum of the losses do not significantly affect performance. Therefore, we removed the weights for simplicity.

3.3 Token Filtering

If we use all tokens for training $\mathrm{MLP}_{\mathrm{anti}}$, information irrelevant to factuality is embedded in $\mathrm{MLP}_{\mathrm{anti}}$. Hence, we identify a subset of tokens that affect the factuality of text, T_{fact}. We only use T_{fact} for calculating L_{anti} and L_{expert}.

To identify T_{fact}, we create an anti-expert LLM modified from the one of Zhang et al. (2025). We identify T_{fact} using the base and anti-expert LLM.

$$T_{\mathrm{fact}} = \{x_i \in T : \log p_{\mathrm{anti}}(x_i|x_{<i}) - \log p_{\mathrm{base}}(x_i|x_{<i}) \geq \gamma\}$$

T is a set of input tokens. γ is a hyperparameter to control the strength of constraint.

4 Experiments

We evaluated IMAE on a question-answering task. To compare our method with the existing ones, we followed the settings of Zhang et al. (2025).

4.1 Settings

Benchmark. As training and evaluation data, we used the HaluEval dataset (Li et al., 2023a) and TruthfulQA (Lin et al., 2022), respectively.

We used multiple-choice-based metrics: MC1, MC2, and MC3 scores. MC1 evaluates whether models assign the highest score to the best answer. MC2 assesses whether the normalized probability of all correct answers exceeds that of incorrect ones. MC3 checks whether each correct answer is scored higher than every incorrect answer. MC1 aligns best with greedy decoding settings, so we consider it to be the most important metric.

Comparison Methods. As a base LLM, we used Llama2-7B-Chat (Meta, 2023). In addition to ICD, we compared our method with ITI (Li et al., 2023b), DoLa (Chuang et al., 2024), and CD (Li et al., 2023c). ITI shifts the model activations by using attention heads. DoLa contrasts the output distributions from different layers of the LLM. CD contrasts the output distributions from LLMs of different parameter sizes [2].

4.2 Results and Discussion

Does IMAE improve truthfulness? Table 1 shows the experimental results. IMAE significantly improved the truthfulness of Llama2-7B-Chat on TruthfulQA. IMAE and ICD improved all truthfulness scores, but the other methods did not improved MC1 score. Moreover, IMAE outperformed the existing methods in MC1, except for the conventional anti-expert method. Since MC1 evaluates the correctness of the most plausible response, we consider that it simulates the greedy setting. In comparison, MC2/3 consider all correct answers. IMAE increased the probability of some tokens and decreased the probability of others. As a result, it increased the probability of the most plausible answer but decreased the probability of some correct answers.

We also found that DoLa and CD, which are widely used for mitigating hallucination, are effective for base models (Chuang et al., 2024; Li et al., 2023c) but not effective for MC1 in chat models. DoLa and CD contrast the distributions and adjust token probabilities relatively modestly, which is effective for MC2/3 but ineffective for MC1 because it does not substantially boost the probability of the top candidate. We consider that our method and DoLa/CD possess characteristics that make them particularly effective for MC1 and MC2/3, respectively.

[2]We used Llama2-13B-Chat for contrasting.

	MC1	MC2	MC3
baseline	36.96	54.62	27.95
ICD (upper bound)	46.32	69.08	41.25
ITI	37.01	54.66	27.82
DoLa	32.97	60.84	29.50
CD	28.15	54.87	29.75
IMAE (ours)	40.02	57.12	28.96

Table 1: Experimental results on TruthfulQA.

	memory	latency
baseline	13.2 (**1.0x**)	2.09 (**1.0x**)
ICD	28.6 (**2.2x**)	4.05 (**1.9x**)
ITI	16.2 (**1.2x**)	2.09 (**1.0x**)
DoLa	15.1 (**1.2x**)	2.21 (**1.1x**)
CD	41.0 (**3.1x**)	6.42 (**3.1x**)
IMAE (ours)	18.4 (**1.4x**)	2.60 (**1.2x**)

Table 2: Computational costs. We measured the GPU memory usage (GB) and latency (ms/token).

Does IMAE reduce computational costs? Table 2 shows the computational costs on TruthfulQA. We evaluate the average GPU memory usage and latency per token. ICD required high additional computational costs. IMAE improved the GPU memory usage from 2.2x to 1.4x and the latency from 1.9x to 1.2x. Moreover, it alleviated the increase in additional computational costs to a level comparable to that of ITI and DoLa.

Is IMAE also effective for other model sizes? We conducted additional experiments using Llama3.2 (3B) and Llama3.1 (8B) (Meta, 2024). Table 3 shows the experimental results. IMAE significantly improved the truthfulness of both LLMs on TruthfulQA. These results indicate that IMAE is also effective for smaller LLMs.

5 Conclusions

We proposed a novel and efficient hallucination mitigation method. An anti-expert is effective and only requires hallucination data, which is easier to prepare than paired data. However, their approach comes with significant computational costs because they runs two LLMs simultaneously. To overcome this limitation, our method intervenes in the internal representations in order for the LLM to mitigate hallucination by itself.

Experimental results showed the effectiveness

	MC1	MC2	MC3
Llama3.2-3B-Instruct	35.25	54.75	27.44
+ ICD	40.13	73.67	44.67
+ IMAE (ours)	38.19	59.36	32.10
Llama3.1-8B-Instruct	40.88	59.90	31.35
+ ICD	44.80	79.03	50.99
+ IMAE (ours)	43.08	63.62	36.01

Table 3: Experimental results of Llama3 family.

and efficiency of the proposed method: the in-model anti-expert outperformed existing methods except for the anti-expert method on truthfulness and reduced the GPU memory usage and latency of the conventional anti-expert methods. These results suggest that our method provides an effective and scalable solution to hallucination mitigation.

Limitations

While IMAE showed promising results in mitigating hallucinations with lower computational costs compared with existing anti-expert approaches, this research has several limitations.

Dependency on parametric knowledge of LLMs. One of the main causes of hallucination is knowledge recall failure (Zheng et al., 2023). The proposed method helps to mitigate hallucinations caused by knowledge recall failures. Mallen et al. (2023) suggested lack of knowledge as another cause of hallucination in LLMs. While our method can not mitigate hallucinations caused by lack of knowledge, methods referring to external knowledge, such as retrieval-augmented generation (RAG) (Lewis et al., 2021), are effective in mitigating these sorts of hallucination (Gao et al., 2023; Ram et al., 2023). Recently, various RAG-based methods have proposed (Yu et al., 2023; Asai et al., 2024; Cuconasu et al., 2024). This study is orthogonal to these lines of research, but it is likely that the proposed method is compatible with and may yield additional performance gains when used together with RAG-based methods.

Broader applicability of anti-expert methods. While this study focused on truthfulness, it is important to note that anti-expert methods were originally proposed in the field of toxicity mitigation (Liu et al., 2021). Therefore, anti-expert methods have the potential to be effective in various fields. We hope this study inspires further cross-cutting research on anti-expert methods.

Limited Scope of TruthfulQA Evaluations. In this work, we have only evaluated our method in the multiple-choice setting of TruthfulQA. While multiple-choice evaluation is the most widely adopted metric, it provides only a limited view of model factuality. As future work, we plan to extend our evaluation to the open-ended setting of TruthfulQA to further assess the generality and robustness of our method in more realistic generation scenarios.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *ICLR*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *IJCNLP-AACL*, pages 675–718.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *arXiv:2309.03883*.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *arXiv:2401.14887*.

David Dale, Elena Voita, Loic Barrault, and Marta R. Costa-jussà. 2023. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity Even better. In *ACL*, pages 36–50.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *NAACL-HLT*, pages 5271–5285.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *ACL*, pages 16477–16508.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *ICLR*.

Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S. Yu, and Zhijiang Guo. 2023. Do large language models know about facts? In *arXiv:2310.05177*.

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

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Factuality enhanced language models for open-ended text generation.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich KÃijttler, Mike Lewis, Wen tau Yih, Tim RocktÃd'schel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *arXiv:2005.11401*.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *EMNLP*, pages 6449–6464.

Kenneth Li, Oam Patel, Fernanda ViÃl'gas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. In *NeurIPS*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023c. Contrastive decoding: Open-ended text generation as optimization. In *ACL*, pages 12286–12312.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *ACL*, pages 3214–3252.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *ACL-IJCNLP*, pages 6691–6706.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *ACL*, pages 9802–9822.

Meta. 2023. Llama 2: Open foundation and fine-tuned chat models. In *arXiv:2307.09288*.

Meta. 2024. The llama 3 herd of models. In *arXiv:2407.21783*.

Nvidia. 2024. Nemotron-4 340b technical report. In *arXiv:2406.11704*.

OpenAI. 2024. Gpt-4 technical report. In *arXiv:2303.08774*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2023. Med-HALT: Medical domain hallucination test for large language models. In *CoNLL*, pages 314–334.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *TACL*, 11:1316–1331.

Tohida Rehman, Ronit Mandal, Abhishek Agarwal, and Debarshi Kumar Sanyal. 2023. Hallucination reduction in long input text summarization. In *arXiv:2309.16781*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael SchÃd'rli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *PMLR*.

David Wan, Shiyue Zhang, and Mohit Bansal. 2023. HistAlign: Improving context dependency in language generation by aligning with history. In *EMNLP*, pages 2941–2960.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2024. Alignment for honesty. In *NeurIPS*.

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023. Improving language models via plug-and-play retrieval feedback. In *arXiv:2305.14002*.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R. Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023a. R-tuning: Teaching large language models to refuse unknown questions. In *arXiv:2311.09677*.

Shaolei Zhang, Tian Yu, and Yang Feng. 2024. TruthX: Alleviating hallucinations by editing large language models in truthful space. In *ACL*, pages 8908–8949.

Yue Zhang, Leyang Cui, V. W., and Shuming Shi. 2025. Alleviating hallucinations of large language models through induced hallucinations. In *NAACL Findings*, pages 8218–8232.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023b. Siren's song in the ai ocean: A survey on hallucination in large language models. In *arXiv:2309.01219*.

Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. 2023. Why does chatgpt fall short in providing truthful answers? In *ICBINB@NeurIPS*.

Configuration	Value
Epochs	5
Batch size	24
Learning rate	5×10^{-5}
Intermedeate size of $\mathrm{MLP}_{\mathrm{anti}}$	4096
Penalty strength β in $p_{\mathrm{expert}}(\cdot)$	2.0
Filtering strength γ in T_{fact}	1.0
Penalty strength δ in ν_{valid}	1.0

Table 4: Hyperparameters of IMAE.

	MC1	MC2	MC3
baseline	36.96	54.62	27.95
IMAE	40.02	57.12	28.96
w/o gate layer	39.05	55.96	29.07
w/o L_{anti}	37.09	54.15	27.68
w/o L_{expert}	36.35	54.08	26.64
w/o L_{gate}	37.33	54.28	27.24
w/o token filtering	38.80	55.46	28.43

Table 5: Ablation study of IMAE on TruthfulQA. All methods in this table are based on Llama2-7B-Chat.

A More Implementation Details.

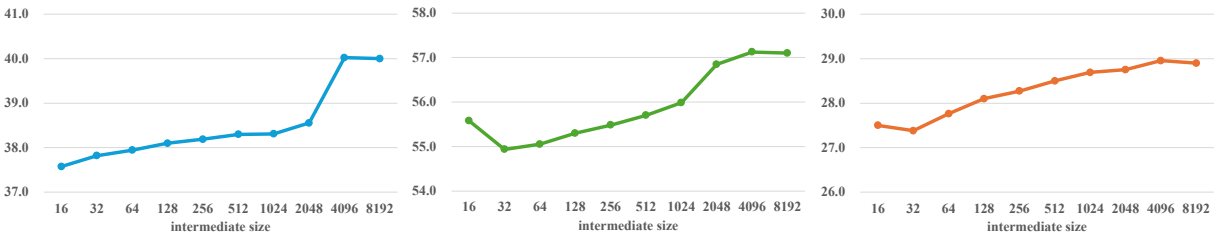
Dataset details. As training data, we used 10k hallucinated QA pairs from the HaluEval dataset (Li et al., 2023a). Li et al. (2023a) created hallucinated answers by prompting ChatGPT to generate non-factual answers. As evaluation data, we used the popular TruthfulQA (Lin et al., 2022). TruthfulQA consists of 817 QA pairs on which LLMs tend to generate hallucinate answers and covers 38 domains, such as medical, legal, and political.

Finetuning details. We ran experiments with 8 NVIDIA RTX 6000 (24GB) GPUs. We trained the models with the AdamW optimizer (Loshchilov and Hutter, 2019). Table 4 shows the hyperparameters of IMAE.

B Ablation Study

Which parts of the proposed method contribute to improving the truthfulness score? Table 5 shows an ablation study on TruthfulQA using Llama2-7B-Chat. Removing loss reveals their contributions: excluding the anti-expert mode loss (L_{anti}), expert mode loss (L_{expert}), or gate loss (L_{gate}) led to significant decreases in truthfulness scores, highlighting their importance. The gate layer and token filtering both provided additional gains.

Does the MLP layer size of the anti-expert unit affect the truthfulness score? Figure 3 shows the relation between the intermediate size of $\mathrm{MLP}_{\mathrm{anti}}$ and MC1/2/3 scores. MC1/2/3 scores tended to improved up to 4096 for the intermediate size of $\mathrm{MLP}_{\mathrm{anti}}$, but did not improve from 4096. This indicated that the truthfulness score improves as the intermediate size of $\mathrm{MLP}_{\mathrm{anti}}$ increases up to a certain value.



(a) MC1 (b) MC2 (c) MC3

Figure 3: Comparison of different intermediate sizes of $\mathrm{MLP}_{\mathrm{anti}}$ on TruthfulQA. The base LLM is Llama2-7B-Chat.