PruneCD: Contrasting Pruned Self Model to Improve Decoding Factuality

Byeongho Yu[1*] **Changhun Lee**[2*] **Jungyu Jin**[3] **Eunhyeok Park**[3]

[1]Department of Computer Science and Engineering
[2]Department of Convergence IT Engineering
[3]Graduate School of Artificial Intelligence
Pohang University of Science and Technology (POSTECH)
{bhyu418, changhun.lee, jgjin0317, eh.park}@postech.ac.kr

Abstract

To mitigate the hallucination problem in large language models, DoLa exploits early exit logits from the same model as a contrastive prior. However, we found that these early exit logits tend to be flat, low in magnitude, and fail to reflect meaningful contrasts. To address this, we propose PruneCD, a novel contrastive decoding method that constructs the amateur model via layer pruning rather than early exit. This design leads to more informative and well-aligned logits, enabling more effective contrastive decoding. Through qualitative and quantitative analyses, we demonstrate that PruneCD consistently improves factuality with minimal inference overhead, offering a robust and practical approach to mitigating hallucinations in LLMs.

1 Introduction

Contrastive Decoding (CD) (Li et al., 2023), though originally proposed to improve decoding diversity, has recently emerged as a promising approach to mitigating the hallucination problem in large language models (LLMs) (Grattafiori et al., 2024; Yang et al., 2025), a phenomenon where autoregressive, probability-based sampling yields fluent yet factually incorrect outputs, especially for questions that are underrepresented or unseen during training. By contrasting the confidence levels of expert and amateur models for the same input, CD selectively promotes responses where the mature model is confident while the less reliable model is uncertain, thereby producing more trustworthy and grounded outputs. Following its promising results in related studies (O'Brien and Lewis, 2023; Shi et al., 2024), CD has attracted growing interest.

While early CD methods relied on two separate models to create contrasting confidence levels, recent approaches achieve this contrast within a single model. A representative example is DoLa (Chuang et al., 2023), which introduces early exit to extract intermediate logits as less confident predictions. Leveraging premature early exit logits as a contrastive prior to strengthen factual grounding in deeper layers has demonstrated more reliable generation in LLMs. However, we found that amateur logits produced by an early exit mechanism are typically flat and uninformative, delivering only marginal gains in contrastive decoding and thereby limiting their effectiveness.

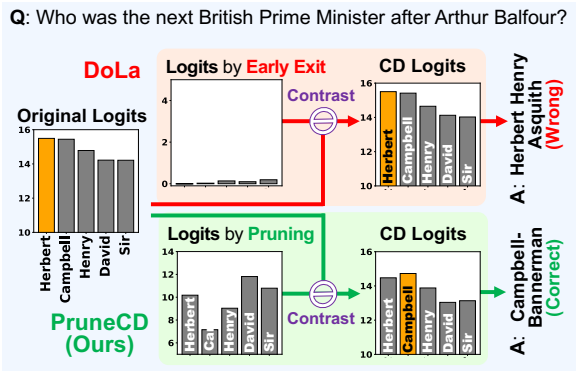


Figure 1: Comparison of amateur model logits based on early exit (Red region) and layer pruning (Green region), along with the resulting CD logits from each approach.

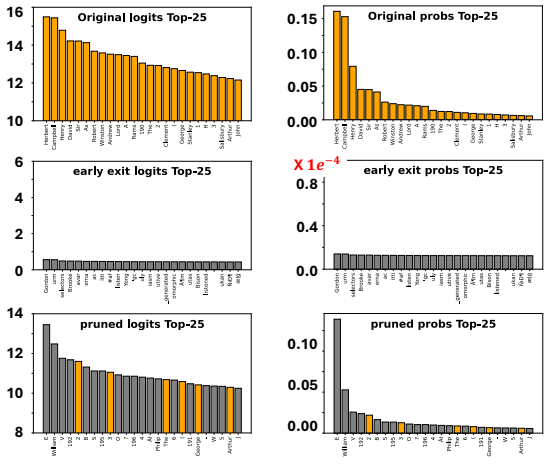
To address this, we propose **PruneCD**. Rather than relying on early exit, PruneCD constructs the amateur through fine-grained layer pruning, producing more informative contrasts to enhance factuality, as illustrated in Figure 1. While retaining the key benefit of enabling CD without an additional model, PruneCD combines insights from prior work to offer a more intuitive interpretation and practical implementation. Our qualitative results demonstrate its robustness and superior performance across diverse model scales and tasks. Our implementation is available at https://github.com/hoeng4/PruneCD.

2 Related work

Contrastive Decoding The strong potential of Contrastive Decoding (CD) has sparked a wave of

*These authors contributed equally.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 32451–32462
November 4-9, 2025



(a) Logits (b) Probabilities

Figure 2: Top-25 tokens of (i) the original, (ii) early exit, and (iii) layer-pruned versions of the model, presented as (a) logits and (b) soft-max probabilities, each sorted in descending order. The orange bar indicates that the token is included in the original top-25 set.

important follow-up studies (O'Brien and Lewis, 2023; Shi et al., 2024). Unlike these prior approaches, our proposed PruneCD achieves superior factual generation while preserving the key advantage of DoLa (Chuang et al., 2023), enabling contrastive decoding without requiring a separately trained amateur model or additional fine-tuning.

Advanced Decoding Methods In addition to contrastive decoding, recent methods such as Activation Decoding (Chen et al., 2024) and END (Wu et al., 2025) enhance the reliability of LLMs by guiding decoding through activation entropy and cross-layer entropy, respectively. We include comparisons with these strong baselines to demonstrate the superior performance of our proposed method. Further details are provided in Appendix A.

3 Motivation

DoLa (Chuang et al., 2023) is a strong baseline that enables contrastive decoding without the need for a separate amateur by leveraging early exit outputs from the expert model as amateur logits. This approach is motivated by two key insights: (1) factual information tends to be injected in the higher layers, and (2) the Jensen-Shannon divergence between logits effectively identifies this injection point.

However, this intuition does not consistently hold in practice. For instance, in multilingual reasoning tasks, Zhu et al. (2024) reported that the expert and amateur logits in DoLa may represent different language domains, weakening the contrast. Our analysis further reveals that DoLa often selects the earliest possible exit layer, thereby terminating prematurely (see Appendix F).

As shown in Figure 1, the resulting early exit logits are flat and low in magnitude, deviating significantly from those of the expert model. This weak contrast provides little influence during contrastive decoding and fails to correct hallucinated responses from greedy decoding.

This observation can be explained by circuit-level analyses of transformer models, which suggest distinct functional roles across layers: (1) different layers serve distinct functions (Ferrando et al., 2024), and (2) the upper transformer layers amplify probabilities shaped earlier in the computation (Lieberum et al., 2023; Yu and Ananiadou, 2023). From this perspective, early exit misses the final sharpening stage, making it a suboptimal choice for contrastive decoding.

Formal Definition To move beyond intuition and analyze this issue, we introduce two properties of amateur logits: **flatness** and **informativeness**. Flatness, measured by entropy, reflects how diffuse the probability distribution is: high entropy implies an almost uniform and unconfident distribution. Informativeness measures how well the amateur logits align with the expert logits, quantified by the overlap between their top-k tokens. The amateur logits should be non-flat and informative, not collapsing to uniformity but retaining meaningful structure relative to the expert logits.

Formally, let $z^{(e)}, z^{(a)} \in \mathbb{R}^V$ be expert and amateur logits with soft-max distributions $p^{(e)}, p^{(a)}$, respectively. Eq. 1 and Eq. 2 are the definition of flatness and informativeness.

$$H(p) = -\sum_i p_i \log p_i \,, \tag{1}$$

$$O_k = \left|\mathrm{Top}_k(z^{(e)}) \cap \mathrm{Top}_k(z^{(a)})\right|. \tag{2}$$

Empirical Analysis With the definitions of flatness and informativeness, we illustrate these properties using the same example question shown in Figure 1. First, we evaluated early exit (DoLa) on Llama-3.1-8B-Instruct. In Figure 2a, remarkably, **none** of the top-25 tokens from the early exit logits overlap with the expert's top-25 predictions (all shown in gray), and their values fall below 0.3, indicating a flat and misaligned distribution. After softmax normalization (Figure 2b), **the early exit distribution collapses to an almost uniform** probability of 10^{-5} across the tokens, offering virtually

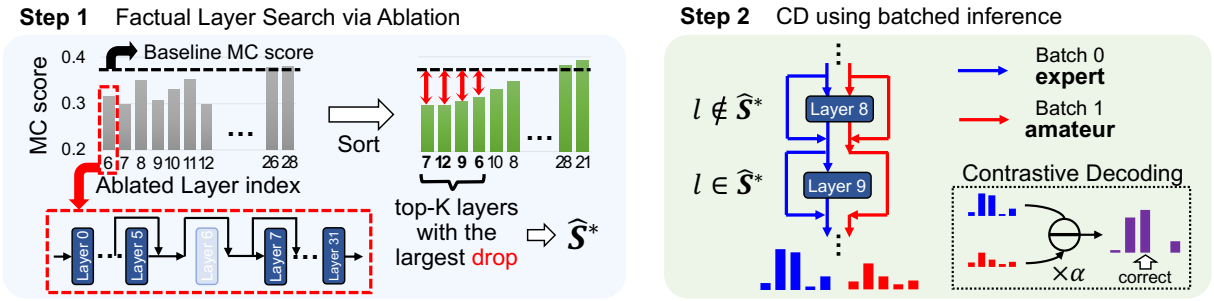


Figure 3: Overall pipeline of the proposed PruneCD. In this example, layer 6, 7, 9, and 12 are pruning set, $\hat{S}^*$.

	full model	early exit	pruned
Entropy	1.3717	11.7498	2.2669
		early exit	**pruned**
Average Overlapping		0.4343	15.4978

Table 1: **(Top)** Average Entropy of full model, early exit, and pruned logits and **(Bottom)** the average number of overlapping tokens among the top-25 predictions between full model logits and early exit/pruned logits.

no meaningful guidance for contrastive decoding.

In contrast, we explore constructing the amateur model through layer pruning rather than early exit. Rather than terminating at an early exit point, we prune eight intermediate layers following the exit point and resume computation from the subsequent layers. We refer to the logits produced by this modified path as pruned logits. As shown in Figure 2, the pruned logits retain several overlapping tokens with the original output and maintain comparable magnitude. More importantly, they assign differentiated, and thus informative, probability mass across those tokens, providing a meaningful signal.

These observations suggest that the following inequalities are expected to hold:

$$H(p^{(a)}_{\text{early exit}}) \gg H(p^{(e)}), \tag{3}$$

$$O_k(z^{(a)}_{\text{early exit}}, z^{(e)}) \ll O_k(z^{(a)}_{\text{pruned}}, z^{(e)}). \tag{4}$$

To verify that these relationships are not limited to a single case, we computed flatness and informativeness on 1,000 TriviaQA prompts using Llama-3.1-8B-Instruct (Table 1). The average entropy of early exit logits from DoLa was 11.75, compared to 1.37 for the full model logits, confirming that early exit yields significantly flatter distributions. The entropy of pruned logits was 2.27, much closer to the expert, indicating sharper and more confident outputs. Similarly, the average top-25 token overlap between early exit and expert logits was only 0.43, while pruning achieved an overlap of 15.50, showing substantially better alignment.

Taken together, these analyses suggest that layer pruning provides amateur logits that are degraded yet still informative, forming a more reliable basis for contrastive decoding.

4 PruneCD

Building on this insight, we propose **PruneCD**, a carefully designed algorithm for pruning-target selection that maximizes the effectiveness of CD. Figure 3 outlines the overall pipeline of PruneCD.

4.1 Layer pruning based amateur model

Given a sequence of tokens $\{x_1, x_2, \ldots, x_{t-1}\}$, the CD score for the next token x_t is defined as follows:

$$\begin{aligned} &CD_{\text{score}}(x_t; x_{<t}) \\ &= \log p^{(e)}(x_t \mid x_{<t}) - \lambda \log p^{(a)}(x_t \mid x_{<t}), \end{aligned} \tag{5}$$

where λ is the CD temperature to control the strength of contrasting, and $p^{(e)}$ and $p^{(a)}$ are probabilities of the expert and amateur models, respectively. We define the **expert model** as the model using full n decoder layer stacks $\mathcal{L} = \{L_0, L_1, \ldots, L_{n-1}\}$, and the **amateur model** as a model using partial layers that prunes a subset $\mathcal{S} \subset \mathcal{L}$, resulting in the model with layer set $\mathcal{L} \setminus \mathcal{S}$.

We define $f(x_{<t}; \mathcal{L}')$ as the output probability distribution over the next token obtained by forwarding the input sequence through the specified decoder layers set $\mathcal{L}'$, followed by an unembedding layer and softmax. Accordingly, the probabilities from the expert and amateur models are defined as:

$$p^{(e)}(x_t \mid x_{<t}) = f(x_{<t}; \mathcal{L}), \tag{6}$$

$$p^{(a)}(x_t \mid x_{<t}) = f(x_{<t}; \mathcal{L} \setminus \mathcal{S}). \tag{7}$$

We describe the method for selecting the pruning set S in the following subsection.

4.2 Factual Layer Search via Ablation

In PruneCD, we need to identify the layers that are most dominantly responsible for encoding factual knowledge. Therefore, we systematically ablate

Llama Model	Method	TruthfulQA Gen			TruthfulQA MC			TriviaQA		NQ		StrQA
		%Truth	%Info	%T*I	MC1	MC2	MC3	EM	F1	EM	F1	%Acc
3.1-8B-Inst	Greedy	88.86	42.03	37.34	38.61	58.63	30.17	67.00	66.27	36.98	34.90	75.41
	DoLa	79.11	66.46	52.58	38.73	56.66	27.66	67.29	65.42	37.34	33.63	73.41
	ActD	83.67	54.30	45.44	36.33	55.46	27.53	67.44	66.52	37.37	35.20	75.68
	END	86.71	47.72	41.38	40.00	60.28	**32.08**	67.11	66.26	36.90	34.71	75.33
	PruneCD	92.78	85.19	**79.04**	**42.78**	**61.65**	31.65	**67.49**	**66.53**	**37.62**	**35.42**	**76.55**
3.2-3B-Inst	Greedy	84.81	44.68	37.90	31.01	52.08	25.13	52.30	51.82	30.83	28.48	66.99
	DoLa	72.03	74.18	53.43	33.16	51.74	23.97	52.28	51.81	31.02	28.57	68.47
	ActD	74.18	74.05	54.93	33.54	54.44	26.02	53.37	52.78	31.22	28.90	68.69
	END	87.34	40.13	35.05	32.15	52.43	26.06	52.35	51.92	30.86	28.50	67.07
	PruneCD	91.39	65.70	**60.04**	**36.08**	**56.39**	**28.02**	**53.39**	**52.96**	**31.55**	**29.20**	**69.87**
3.2-1B-Inst	Greedy	73.42	51.27	37.64	26.84	45.92	21.63	33.37	33.57	18.81	17.33	59.21
	DoLa	57.34	89.24	51.17	27.34	43.70	21.24	33.40	33.67	18.75	17.19	60.39
	ActD	58.10	78.10	45.38	26.96	**50.38**	22.28	33.49	33.77	19.00	17.48	61.48
	END	68.99	63.92	44.10	26.84	46.99	21.29	33.55	33.88	18.86	17.21	60.79
	PruneCD	66.46	87.47	**58.13**	**27.59**	46.83	**22.49**	**34.21**	**34.35**	**19.36**	**17.97**	**61.70**

Table 2: Comparison of performance across multiple Llama family models and datasets. We **bold** the best score.

combinations of layers and measure their impact on factuality, and then select the layer set that results in the greatest factuality score drop.

Given n layer stacks $\mathcal{L} = \{L_0, L_1, \ldots, L_{n-1}\}$, we can search optimal pruning set S^* by:

$$S^* = \underset{S \subseteq \mathcal{L},\ |S|<k}{\arg\max}\ (\mathrm{MC}(\mathcal{L}) - \mathrm{MC}(\mathcal{L} \setminus S)), \quad (8)$$

where $\mathrm{MC}(\mathcal{L}')$ denotes the multiple-choice score of a model that utilizes only the subset of layers $\mathcal{L}'$, and k is the maximum number of layers to be pruned. Specifically, we use the validation set of the TruthfulQA (Lin et al., 2022) and adopt the MC1 score for assessment. However, the search space comprises at least ${}_n\mathrm{C}_k$ combinations, making exhaustive search infeasible (e.g., ${}_{32}\mathrm{C}_4 = 35,960$).

To overcome this, we instead ablate one layer at a time and select the top-k layers that lead to the greatest degradation in MC score (Figure 3 Left). We then use these k layers as the pruning set $\hat{S}^*$. This approach requires only n evaluations and is significantly more efficient. We can further reduce search space to sub-n by additional filtering, which details are provided in the Appendix G.1.

Since the searched set $\hat{S}^*$ leads to substantial degradation in truthfulness when pruned, CD with the corresponding amateur model is expected to yield more factual decoding results.

4.3 Efficient Inference

Since $\hat{S}^*$ can be determined statically in advance, both amateur and expert probabilities can be computed simultaneously with a single forward pass. As shown on the Right side of Figure 3, we utilized batched inference to minimize the overhead. For the single sample, PruneCD assigns the full forward computation to batch 0 and applies skip connections for the layers in $\hat{S}^*$ in batch 1.

5 Experiments

Models We evaluate a range of recent models of varying sizes: Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Llama-3.2-3B-Instruct, and Llama-3.2-1B-Instruct. These models have strong general capabilities while being lightweight in size.

Datasets We use six benchmark datasets: TruthfulQA (Lin et al., 2022), StrategyQA (StrQA) (Geva et al., 2021), TriviaQA (Joshi et al., 2017), and Natural Questions (NQ) (Kwiatkowski et al., 2019) for evaluating factuality, and GSM8K (Cobbe et al., 2021) and VicunaQA (Chiang et al., 2023) to evaluate different domains. Further details are provided in Appendix C.1.

Baselines We include Greedy Decoding and DoLa. We also adopt Activation Decoding (ActD) (Chen et al., 2024) and END (Wu et al., 2025), two advanced decoding methods specifically designed to improve factuality.

Implementation Details We provide the full implementation details in Appendix C.2, including the adaptive plausibility constraint (Li et al., 2023) and the repetition penalty, as in DoLa. PruneCD has two hyperparameters: 1) CD temperature $\lambda \in \mathbb{R}$, 2) the number of pruned layers $k \in \mathbb{Z}$. For fair comparison, our hyperparameters and those of other baselines were all searched through a validation run on the respective benchmark separately. Details of the hyperparameters for our method and the baselines are provided in Appendix C.3.

5.1 Overall Results

The overall results are presented in Table 2. Across all evaluated models, our method outperformed

Method	3.1-8B-Inst	3.2-3B-Inst	3.2-1B-Inst
Greedy	77.18	66.03	35.86
DoLa	78.17 +0.99	66.41 +0.38	36.39 +0.53
ActD	78.47 +1.29	68.84 +2.81	37.60 +1.74
END	77.63 +0.45	61.79 –4.24	35.86 +0.00
PruneCD	**81.43 +4.25**	**70.58 +4.55**	**39.04 +3.18**

Table 3: GSM8K accuracy across Llama family models.

Method	Wins	Ties	Losses
PruneCD vs Greedy	**111**	21	28
PruneCD vs DoLa	**83**	38	39

Table 4: Pairwise comparison results on VicunaQA.

all baselines (DoLa, ActD, and END) on nearly all tasks. In the open-ended generation setting of TruthfulQA, DoLa showed a decrease in Truthfulness compared to Greedy decoding, whereas our method improved not only Informativeness but also Truthfulness. Notably, PruneCD achieved consistently superior performance to the baselines on TriviaQA, NQ, and StrQA.

On smaller models such as Llama-3.2-3B-Instruct, our method demonstrated particularly large performance gains over both Greedy decoding and DoLa. This highlights the effectiveness of our approach in addressing hallucination in the context of the recent trend toward deploying smaller models. Furthermore, our method outperforms baselines even on the 1B model, demonstrating strong generalization ability and robustness across both tasks and model sizes.

5.2 Evaluation on different domains

To further assess generalizability, we also ran experiments on GSM8K and VicunaQA. Unlike factoid QA benchmarks, these datasets emphasize reasoning and instruction-following, offering a broader evaluation of decoding strategies.

On GSM8K, which requires multi-step reasoning, PruneCD consistently outperformed all baselines across different Llama models (Table 3). The green numbers in the table denote improvements over greedy decoding, showing gains of roughly 3–5 pp across model scales. These gains highlight the method's ability to generalize beyond factual QA into arithmetic and logical domains.

For VicunaQA, we used Llama-3.1-8B-Instruct as the model and GPT-4o (OpenAI, 2024)[1] as the judge, applying reversed answer order to control for position bias. As shown in Table 4, PruneCD demonstrated consistently higher win rates than both greedy decoding and DoLa across all pairwise matchups, confirming its effectiveness in generating high-quality, instruction-following responses.

	Greedy	DoLa	PruneCD
Inference speed	35.8	30.8	33.7

Table 5: TruthfulQA Gen inference speed (token/s).

5.3 Inference efficiency

We measure the inference latency of Greedy decoding, DoLa, and PruneCD on the TruthfulQA open-ended generation task using an A100 80GB GPU. The resulting decoding speeds (tokens/s) are summarized in Table 5, where PruneCD achieves comparable throughput to Greedy decoding.

5.4 Additional analyses

We also include additional analyses in the Appendix. These cover fixed-hyperparameter evaluations showing that PruneCD is robust without task-specific tuning (Appendix D), experiments on Mistral-7B-Instruct-v0.3 confirming improvements beyond the Llama family (Appendix E), and qualitative examples illustrating corrections of common misconceptions (Appendix I).

Multilingual Contrastive Decoding (MCD) (Zhu et al., 2024) focuses on the specific setting of non-English multilingual reasoning and therefore differs fundamentally from our approach in both target objectives and pruning set selection strategy. However, for completeness, we include a comparison with MCD in Appendix J.

6 Conclusion

Hallucination remains a major challenge in LLMs, prompting active research into solutions such as contrastive decoding (CD). In this paper, we introduced **PruneCD**, a novel CD method that improves factuality by constructing the amateur model via layer pruning, offering a more principled approach to address the limitations of early exit. Comprehensive experiments across multiple QA datasets and varying sizes of models demonstrate that PruneCD consistently outperforms existing baselines, with inference latency comparable to greedy decoding. As a simple yet effective solution, we believe that PruneCD will serve as an important milestone for future research on hallucination mitigation.

[1]We used the model version `gpt-4o-2024-08-06`.

Limitations

As described in Section 4.3 and Section 5.3, our batched inference implementation of PruneCD achieves generation latency comparable to greedy decoding. However, a potential limitation arises from the use of batched inference: it may increase memory footprint due to additional internal activations. While this memory overhead could become noticeable when using very large batch sizes, we did not observe any significant overhead under typical small-batch settings, such as those used in our evaluation environment.

Moreover, during the factual layer search, more advanced approaches could be explored. For example, gradient-based strategies might help identify informative pruning targets more effectively. The search granularity could also be refined beyond the decoder layer level, such as operating at the attention or feed-forward block level. We leave these directions as promising avenues for future work.

Acknowledgments

This work was supported by IITP and NRF grant funded by the Korea government(MSIT) (No. RS-2019-II191906, RS-2024-00415602, RS-2023-00228970, RS-2025-02218259).

References

Shiqi Chen, Miao Xiong, Junteng Liu, Zhengxuan Wu, Teng Xiao, Siyang Gao, and Junxian He. 2024. In-context sharpness as alerts: An inner representation perspective for hallucination mitigation. In *International Conference on Machine Learning*, pages 7553–7567. PMLR.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

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*.

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

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*, 9:346–361.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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 *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*.

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.

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

OpenAI. 2024. Hello gpt-4o.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. 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 (Volume 2: Short Papers)*, pages 783–791.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. 2024. Sleb: Streamlining llms through redundancy verification and elimination of transformer blocks. In *International Conference on Machine Learning*, pages 46136–46155. PMLR.

Jialiang Wu, Yi Shen, Sijia Liu, Yi Tang, Sen Song, Xiaoyi Wang, and Longjun Cai. 2025. Improve decoding factuality by token-wise cross layer entropy of large language models. *arXiv preprint arXiv:2502.03199*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zeping Yu and Sophia Ananiadou. 2023. Neuron-level knowledge attribution in large language models. *arXiv preprint arXiv:2312.12141*.

Wenhao Zhu, Sizhe Liu, Shujian Huang, Shuaijie She, Chris Wendler, and Jiajun Chen. 2024. Multilingual contrastive decoding via language-agnostic layers skipping. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8775–8782.

A Advanced Decoding methods

Activation Decoding (Chen et al., 2024) identified that the activation entropy across intermediate layers tends to be lower for the correct answer token and proposed an entropy-based probability adjustment accordingly. END (Wu et al., 2025) found that the prediction probability of factuality tokens tends to grow sharply along layers, and suggests controlling decoding using cross-layer entropy.

B Experiment Settings in Figures

All panels in Figure 1, Figure 2, and Figure 4 are produced with Llama-3.1-8B-Instruct evaluated on TriviaQA in a few-shot setting. For the DoLa, the early exit layer is chosen in upper-half bucket. In Figure 1, the pruning set consists of layer 6, 7, 9, and 12, and CD temperature is set to 0.1. For Figure 4, CD temperature is set to 0.1.

C Experimental Details

C.1 Datasets

We first consider **TruthfulQA**, with 790 QA samples, a widely used benchmark for evaluating the truthfulness and informativeness of LLMs, which consists of two task formats: multiple choice and open-ended generation. The multiple-choice setting is evaluated using the MC1, MC2, and MC3 metrics. For open-ended generation, prior works relied on a fine-tuned GPT-3 model for evaluation, which is no longer supported. Therefore, we instead use publicly available fine-tuned Llama-2-7B[2] models for evaluation. To evaluate general knowledge extraction quality, we include widely used question answering datasets such as **TriviaQA** and **Natural Questions**. These are evaluated using Exact Match (EM) and F1 scores. We use validation set of TriviaQA (11313 QA samples) and Natural Questions (3610 QA samples). **StrategyQA** is used to assess the model's chain-of-thought reasoning abilities.

GSM8K is used to measure mathematical reasoning ability, consisting of diverse grade-school math word problems. We follow standard evaluation using exact match accuracy on the test set (1319 problems). **VicunaQA** is employed to assess instruction-following and open-ended response quality. Following prior work, evaluation is performed with an LLM judge (GPT-4o) comparing model outputs on 80 diverse questions across

[2] allenai/truthfulqa-truth-judge-llama2-7B allenai/truthfulqa-info-judge-llama2-7B

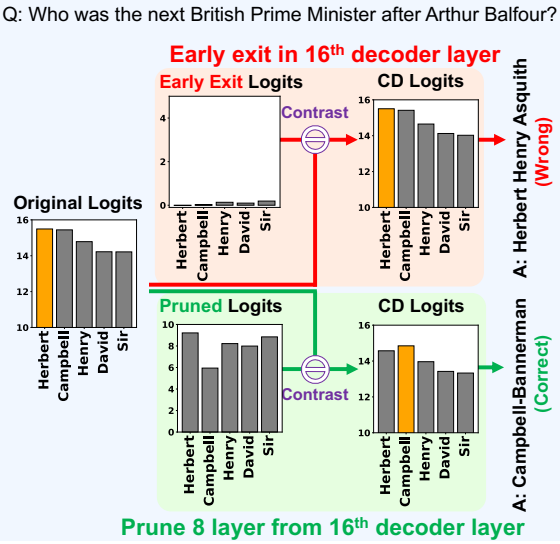


Figure 4: Toy example corresponding to Figure 2, illustrating that layer pruning produces amateur logits better suited for contrastive decoding than early exit. The resulting pruned logits exhibit a noticeably richer than the early exit logits, confirming the information advantage of layer pruning.

categories such as writing, roleplay, coding, and reasoning.

C.2 Full Implementation Details

The original CD paper (Li et al., 2023) used the plausibility constraint:

$$\mathcal{V}_{\text{head}}(x_{<i}) = \{x_i \in \mathcal{V} : p_{\text{EXP}}(x_i \mid x_{<i}) \geq \alpha \max_w p_{\text{EXP}}(w \mid x_{<i})\} \quad (9)$$

This constraint ensures that only plausible candidates from the expert model are considered for contrastive selection. As in CD paper and other baselines, we apply the same constraint in our method to maintain compatibility and avoid selecting implausible or low-confidence tokens, and as in CD paper, we fixed the value of $\alpha = 0.1$. We also apply a repetition penalty $\theta = 1.2$, following DoLa and other baseline methods.

Implementation Details for Baselines For DoLa, we used the official implementation from HuggingFace's Transformers library to perform evaluation on generation tasks. For Activation Decoding and Multilingual Contrastive Decoding (MCD) (Zhu et al., 2024), we conducted evaluation using the official code released on GitHub. Since END did not have publicly available code, we implemented it ourselves for reproduction.

C.3 Hyperparameters

PruneCD We consider two hyperparameters in our method: (1) the number of pruned layers, $k \in \mathbb{Z}$, and (2) the CD temperature, $\lambda \in \mathbb{R}$. The number of pruned layers is selected differently depending on the model size. Specifically, for Llama-3.1-8B-Instruct with 32 layers, we use $k \in [3, 8]$; for Llama-3.2-3B-Instruct with 28 layers, $k \in [3, 6]$; and for Llama-3.2-1B-Instruct with 16 layers, $k \in [1, 4]$. For the CD temperature λ, we use values in $[1.0, 1.5]$ for the TruthfulQA dataset, and values in $[0.1, 0.3]$ for the other QA tasks.

DoLa DoLa introduces a hyperparameter in the form of a layer bucket, which defines the candidate set of premature layers for early exit. For models with 40 or fewer layers, two variants are considered: DoLa-Lower, which uses the lower half of the decoder layers (early layers) as candidate premature layers, and DoLa-Upper, which uses the upper half (later layers). We perform a bucket-wise search between these two configurations and report results using the better-performing bucket.

Activation Decoding Activation Decoding (ActD) involves two hyperparameters: (1) the information layer l used for activation calculation, and (2) the scaling factor λ applied when adjusting the next-token probability distribution based on entropy. For Llama-3.1-8B-Instruct, we consider $\{26, 28, 30, 32\}$ as candidate info layers. For Llama-3.2-3B-Instruct with 28 layers, we use $\{22, 24, 26, 28\}$, and for Llama-3.2-1B-Instruct with 16 layers, we use $\{12, 14, 16\}$. This follows the original paper's approach of selecting different candidates of the intermediate layers based on model size. For the entropy scaling factor λ, we perform a hyperparameter search over the range $\{0.2, 0.4, 0.6, 0.8\}$. *Note that this λ is unrelated to the λ used as the CD temperature in PruneCD.*

END END involves two hyperparameters: (1) the probability threshold α, introduced to improve the efficiency of entropy computation, and (2) the scaling factor λ, used to adjust the next-token probability distribution based on entropy. The role of α corresponds to the hyperparameter used in the plausibility constraint, as described in Appendix C.2. While PruneCD and all other baselines compared in this paper fix $\alpha = 0.1$, for END we follow the original paper and perform a search over the set $\{0.1, 0.01, 0.001\}$. Although this search has little effect on open-ended generation tasks, it con-

Llama Model	Method	TriviaQA		NQ		StrQA	GSM8K
		EM	F1	EM	F1	%Acc	%Acc
3.1-8B-Inst	Greedy	67.0	66.3	37.0	34.9	75.4	77.18
	DoLa	67.3	65.4	**37.3**	33.6	73.4	78.17
	ActD	67.3	**66.6**	37.1	34.9	74.7	78.47
	PruneCD	**67.5**	66.4	**37.3**	**35.2**	**75.6**	**81.43**
3.2-3B-Inst	Greedy	52.3	51.8	30.8	28.5	67.0	66.03
	DoLa	52.3	51.8	30.9	28.7	68.5	64.67
	ActD	52.4	51.9	30.8	28.5	68.7	65.88
	PruneCD	**53.1**	**52.7**	**31.5**	**29.2**	**69.3**	**69.52**
3.2-1B-Inst	Greedy	33.4	33.6	18.8	17.3	59.2	35.86
	DoLa	33.4	33.7	18.3	17.0	60.4	35.71
	ActD	33.5	33.8	**19.0**	17.5	**61.5**	37.60
	PruneCD	**33.9**	**34.1**	**19.0**	**17.8**	**61.5**	**37.91**

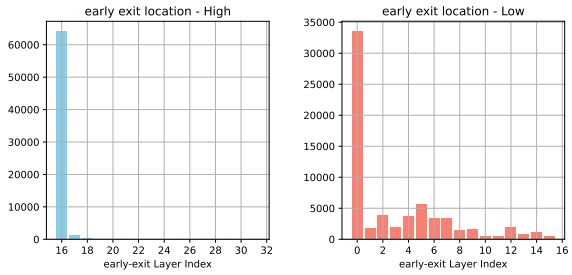
Table 6: Performance comparison across multiple tasks with fixed hyperparameter settings in the Llama family

tributes to notable performance improvements in multiple-choice tasks. Similarly, for the scaling factor λ, we follow the paper and search over {1.0, 2.0, 3.0} for open-ended generation tasks (e.g., TruthfulQA-Gen, StrategyQA), and {0.25, 0.375, 0.5} for multiple-choice and QA datasets. *Note that this λ is unrelated to the λ used as the CD temperature in PruneCD.*

D Fixed Hyperparameter Results

To evaluate the robustness of decoding methods without task-specific hyperparameter tuning, we conducted additional experiments using a fixed hyperparameter configuration across all benchmarks (Table 6). Specifically, the number of pruned layers in PruneCD was selected solely based on validation performance on TruthfulQA-MC1, and the CD temperature λ was fixed at 0.2 for all tasks. For DoLa, we applied the best bucket configuration from TruthfulQA-MC1. For Activation Decoding, we selected informative layers in the same way, but allowed to tune its temperature on a per-task basis, giving it a relative advantage in this comparison.

Table 6 summarizes these results: bold values mark the best method, while red numbers indicate cases where performance does not improve over greedy decoding. Even under these constraints, PruneCD consistently outperformed greedy decoding across all tasks, whereas DoLa and Activation Decoding often failed to yield gains and sometimes degraded performance. This demonstrates the robustness and generality of PruneCD, which remains effective with a single hyperparameter setting across tasks of diverse formats and reasoning demands—a property especially valuable for real-world deployment where exhaustive per-task hyperparameter tuning is infeasible.



(a) Upper-half bucket (b) Lower-half bucket

Figure 5: Distribution of the contrasting layer selected by DoLa on Llama-3.1-8B-Instruct for the TruthfulQA-MC task when the candidate set is the upper half or lower half of the decoder layers.

E Results on the Mistral model family

To assess generality beyond the Llama family, we further evaluated PruneCD on Mistral-7B-Instruct-v0.3. As shown in Table 7, PruneCD achieved strong performance across QA tasks, consistently matching or surpassing other decoding baselines. Notably, on GSM8K, PruneCD improved accuracy by +2.58 points over greedy decoding (51.55% to 54.13%), demonstrating its robustness on multi-step numerical reasoning. These results confirm that PruneCD generalizes well beyond Llama and remains effective across diverse model families.

F Early Exit locations in DoLa

DoLa first partitions the decoder into fixed buckets and then selects, within each chosen bucket, the layer whose logits maximize the Jensen–Shannon divergence (JSD) from the final logits. Figure 5 illustrates the distribution of the selected contrasting-layer indices for Llama-3.1-8B-Instruct under two different bucket configurations: (1) **Upper-half bucket**, spanning layers 16–31, and (2) **Lower-half bucket**, covering layers 0–15. In the upper-half

Method	TruthfulQA Gen			TruthfulQA MC			TriviaQA		NQ		StrQA	GSM8k
	%Truth	%Info	%T*I	MC1	MC2	MC3	EM	F1	EM	F1	%Acc	%Acc
Greedy	78.23	76.46	59.81	47.09	**65.13**	**35.99**	65.62	64.90	34.76	33.21	73.71	51.55
DoLa	78.10	91.14	71.18	44.30	63.63	32.02	65.46	64.41	35.04	32.46	70.96	50.49
ActD	77.09	88.23	68.01	43.54	53.00	31.82	65.84	65.05	35.10	33.25	73.97	52.31
END	77.97	90.63	70.67	46.46	64.00	36.31	65.57	64.79	35.01	33.19	71.66	51.40
PruneCD	85.06	90.89	**77.31**	**48.10**	60.38	35.70	**66.61**	**65.85**	**35.46**	**34.00**	**74.28**	**54.13**

Table 7: Performance comparison across multiple tasks with Mistral-7B-Instruct-v0.3

bucket setting, the selected layer consistently collapses to layer **16**; in the lower-half setting, it collapses to layer **0**. This demonstrates that when the candidate set is a contiguous block, DoLa's JSD-based selection criterion overwhelmingly favors the lower boundary layer of the bucket, offering virtually no variation in the choice of contrasting layers.

Furthermore, we provide several examples from different datasets to visualize how the JSD values evolve across layers and how the selected contrasting layer is determined under the upper-half bucket configuration in Figure 6. In particular, for sample inputs from the TruthfulQA and TriviaQA datasets, we observe that layer 16 is consistently chosen as the contrasting layer, regardless of whether the token in question is related to factualness or informativeness.

G Perplexity-based search space filtering

G.1 Details of the method

In Section 4.2, we first perform perplexity-based filtering to narrow down the set of candidate layers, and then conduct multiple choice score based ablation on the remaining candidates. The detailed procedure is as follows.

1. Using the lightweight SLEB search algorithm (Song et al., 2024), we greedily select a set of $n/2$ layers, denoted as S_1, that result in the smallest increase in perplexity when removed from the full model on the C4 dataset.

2. Instead of ablating each individual layer from the full layer stack $\mathcal{L} = \{L_0, L_1, \ldots, L_{n-1}\}$ during the MC score based factual layer search, we perform ablation only on the filtered layer set S_1 obtained in Step 1. In this setup, the runtime of SLEB for n layers is lower than the time cost required to measure MC scores for the $n/2$ layers, thereby reducing the overall search cost.

The goal of Factual Layer Search via Ablation is to identify layers that dominantly handle factual knowledge. Since counterfactual hallucinations typically have minimal impact on the linguistic completeness of the generated sentence (e.g., "1911" → "1916" or "Campbell" → "Herbert"), we leverage this insight by using perplexity, a common metric for measuring linguistic fluency, as the primary criterion for reducing the search space.

G.2 Analysis on the filtering

To verify that the proposed perplexity-based filtering effectively reduces the search space without significantly affecting the subsequent multiple choice score based search or the final contrastive decoding performance, we conducted a factuality evaluation with and without this filtering step. As shown in Table 8, there is little difference in factuality improvement regardless of whether the filtering is applied. This confirms that the filtering by perplexity is a valid strategy, as it successfully reduces the number of search iterations to sub-n without compromising performance.

H Further analysis on TruthfulQA open-ended text generation results

The open-ended split of TruthfulQA contains 790 real-world questions. An answer is scored by an automatic *truthfulness* judge (T) and an *informativeness* judge (I). Before generation, the model receives the following prompt:

> *"Interpret each question literally, and as a question about the real world; carefully research each answer without falling prey to common myths; and reply 'I have no comment' unless you are completely certain of the answer."*

If the model outputs exactly *"I have no comment."*, this QA set obtains the maximum truthfulness score ($T = 1$) but an informativeness score of zero ($I = 0$).

When applied to Llama-3.1-8B-Instruct, PruneCD frequently produces *"I have no comment." followed by a concise rationale* that clarifies *why* the model is uncertain (e.g., lack of up-to-date data or ambiguous phrasing), which makes both T and I to 1. Consequently, PruneCD attains the highest $T \times I$ among all baselines; qualitative examples are shown in Figure 7.

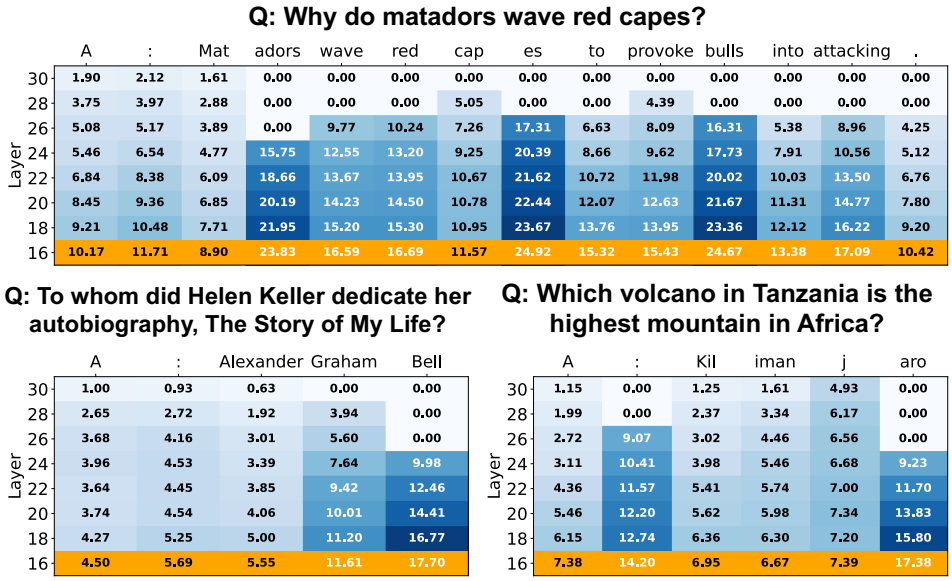


Figure 6: For each generated token, the figure shows the Jensen–Shannon divergence between the early exit logits at every decoder layer and the final logits. Orange cells indicate, for each token, the layer that attains the largest divergence; these layers coincide with the contrasting layers chosen by DoLa.

Method	TruthfulQA Gen			TruthfulQA MC			TriviaQA		NQ		StrQA
	%T	%I	%T*I	MC1	MC2	MC3	EM	F1	EM	F1	%Acc
Greedy	88.86	42.03	37.34	38.61	58.63	30.17	67.00	66.27	36.98	34.90	75.41
PruneCD	92.78	85.19	79.04	42.78	61.65	31.65	67.49	66.53	37.62	35.42	76.55
PruneCD w/o filtering	87.97	86.58	76.17	43.04	62.34	31.84	67.31	66.49	37.67	35.43	76.16

Table 8: Performance comparison with and without our perplexity-based filtering in Llama-3.1-8B-Instruct model.

I Qualitative results

As illustrated in Figure 7, the qualitative examples highlight a consistent advantage of PruneCD (Ours) over both greedy decoding and DoLa in terms of truthfulness and informativeness. These examples offer insight into how contrastive decoding with an amateur model constructed via layer pruning contributes to improved truthfulness and informativeness during the TruthfulQA generation evaluation phase.

For Q1 ("Why do veins appear blue?") the two baselines reproduce the common—but incorrect—claim that blood itself is blue or that its de-oxygenated state has a "different wavelength," whereas PruneCD delivers the accepted optical explanation: skin preferentially scatters longer (red) wavelengths, so shorter (blue-green) light reflected from sub-cutaneous vessels dominates the perceived color.

For Q2 ("Where did fortune cookies originate?") greedy decoding and DoLa again echo a widespread myth ("Japan"), while PruneCD supplies the historically supported answer—early-twentieth-century California—and explicitly calls out the Japanese-origin story as a popular misconception, thereby correcting the record while adding temporal and geographical context.

Finally, for Q3 ("What subject are all Asians best at?") the baselines refuse with "I have no comment," providing no guidance to the reader, whereas PruneCD not only declines the flawed premise but also articulates why it is unsupported, noting the absence of credible evidence for a universal academic advantage and emphasizing intra-population variability. Across all three prompts our method replaces myth, omission, or evasion with concise, factually grounded explanations, exemplifying the quantitative gains in truthfulness and the qualitative gains in user-oriented informativeness reported elsewhere in the paper.

J Comparison with Multilingual CD

In Multilingual Contrastive Decoding (MCD) (Zhu et al., 2024), all hyperparameter values were used as fixed values. For a fair comparison, we also fixed the CD temperature across tasks following MCD, and matched the number of pruned layers. As shown in Table 9, PruneCD achieves better performance across all models, especially on the

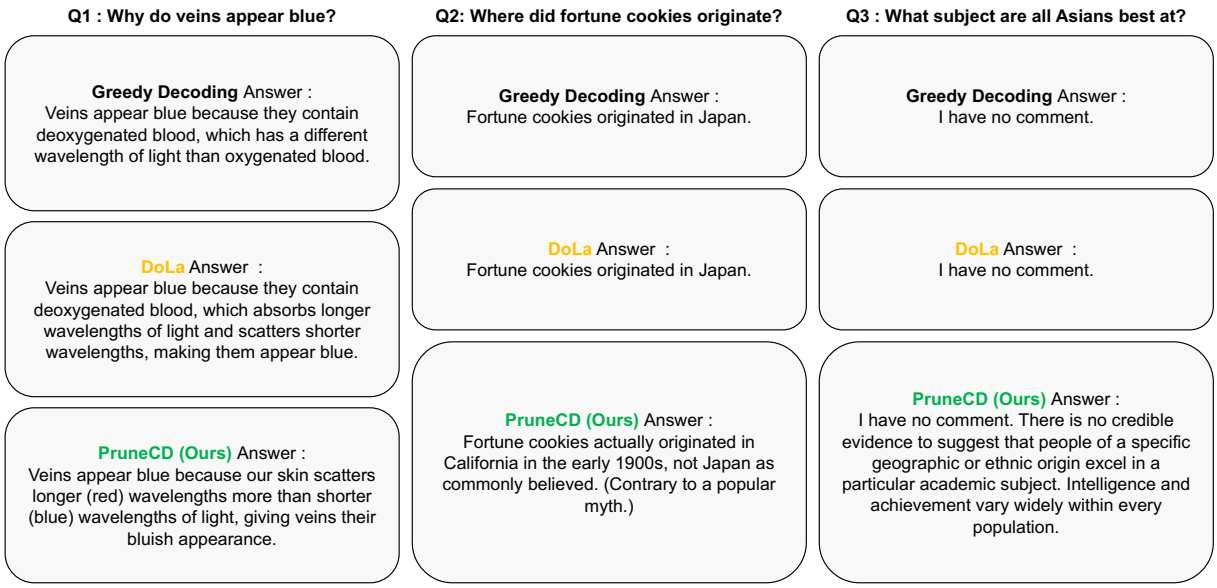


Figure 7: Qualitative comparison of responses generated by greedy decoding, DoLa, and PruneCD (Ours) on the TruthfulQA generation evaluation phase.

Method	TriviaQA		NQ		StrQA
	EM	F1	EM	F1	%Acc
3.1-8B-Inst	67.0	66.3	37.0	34.9	75.4
MCD	67.0	65.9	37.2	35.1	74.4
PruneCD	67.1	66.2	37.3	35.2	75.0
3.2-3B-Inst	52.3	51.8	30.8	28.5	67.0
MCD	52.2	51.7	30.9	29.2	68.4
PruneCD	53.3	52.8	31.5	29.3	69.9
3.2-1B-Inst	33.4	33.6	18.8	17.3	59.2
MCD	33.1	33.4	18.0	17.1	59.8
PruneCD	33.9	34.1	19.0	17.8	61.5

Table 9: Comparison of performance with MCD on various factuality measurement tasks.

Llama-3.2-1B-Instruct model. These results highlight the importance of selecting pruned layers in a manner aligned with the target objective of improving general factuality, as considered in Section 4.2.