

DeLTa: A Decoding Strategy based on Logit Trajectory Prediction Improves Factuality and Reasoning Ability

Yunzhen He¹, Yusuke Takase¹, Yoichi Ishibashi^{2*}, Hidetoshi Shimodaira^{1,3}

¹Kyoto University, ²NEC Corporation, ³RIKEN AIP

he.yunzhen.25d@st.kyoto-u.ac.jp y.takase@sys.i.kyoto-u.ac.jp

yoichi-ishibashi@nec.com shimo@i.kyoto-u.ac.jp

Abstract

Large Language Models (LLMs) are increasingly being used in real-world applications. However, concerns about the reliability of the content they generate persist, as it frequently deviates from factual correctness or exhibits deficiencies in logical reasoning. This paper proposes a novel decoding strategy aimed at enhancing both factual accuracy and inferential reasoning without requiring any modifications to the architecture or pre-trained parameters of LLMs. Our approach adjusts next-token probabilities by analyzing the trajectory of logits from lower to higher layers in Transformers and applying linear regression. We find that this **Decoding by Logit Trajectory**-based approach (DeLTa) effectively reinforces factuality and reasoning while mitigating incorrect generation. Experiments on TruthfulQA demonstrate that DeLTa attains up to a 4.9% improvement over the baseline. Furthermore, it enhances performance by up to 8.1% on StrategyQA and 7.3% on GSM8K, both of which demand strong reasoning capabilities.¹

1 Introduction

Natural language processing has advanced significantly with the rise of large language models (LLMs) (OpenAI, 2024; Dubey et al., 2024). However, ensuring the factual accuracy of LLM-generated text remains challenging. A notable issue is hallucination, where models produce factually incorrect content, posing risks in fields like information retrieval, medicine, and law (Huang et al., 2024). Calculation errors in the logical reasoning further contribute to inaccuracies, stemming from incorrect token predictions during decoding. Mitigation strategies for these issues include the selection of dataset, modifications to loss functions (Ouyang et al., 2022), and the incorporation

*Work done while at Kyoto University.

¹Code is available at <https://github.com/githubyz/DeLTa>.

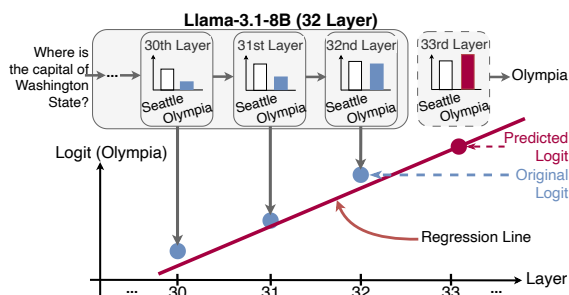


Figure 1: Overview of DeLTa. When input tokens are fed into the LLM, the logits from each layer (e.g., layers 30, 31, and 32) are computed and shown as bar graphs to illustrate changes between tokens (e.g., "Seattle" vs. "Olympia"). A linear regression (red line) approximates the logit trajectory (blue dots). Using this regression, we extrapolate the logits for a virtual 33rd layer (red dot) and improve prediction beyond the original outputs.

of external knowledge (Wan et al., 2024). However, implementing these methods requires refining models or acquiring additional data, which can incur substantial costs.

To overcome these limitations, we propose a decoding strategy, that boosts generation accuracy without extra training or data. Inspired by Chuang et al. (2024), who observed that correct token probabilities tend to rise in higher Transformer layers, we introduce **Decoding by Logit Trajectory**-based approach (DeLTa), which treats each layer's logits as a time-series and use linear regression to predict upper-layer logits (Figure 1).

Experiments demonstrate that DeLTa enhances factuality by up to 4.9% on TruthfulQA, 5.0% on TriviaQA, and 2.4% on Natural Questions, while also improving reasoning on StrategyQA and GSM8K by up to 8.1%. These findings confirm that DeLTa refines token prediction, leading to improved factuality and reasoning capabilities.

2 Related Work

Previous research on guiding LLMs to generate factually accurate text can be broadly categorized into training-based (Lin et al., 2024; Liang et al., 2024) and non-training-based approaches. DeLTa falls into the latter category. Among non-training-based methods, Chang et al. (2024) introduced Asymptotic Probability Decoding, which extrapolates output probabilities from LLMs of different sizes under Contrastive Decoding (CD) (Li et al., 2023). Another method, Sharma et al. (2024) showed that certain capabilities of Transformers are concentrated in the latter layers and achieved improvements in sentiment classification by applying linear extrapolation to a text classifier based on CD. These approaches estimate probabilities using linear regression or extrapolation, relying on only two data points (e.g., the outputs of two models or an intermediate layer and the final layer). In contrast, DeLTa predicts logits instead of probabilities and performs regression across the entire sequence of logits from intermediate layers to the final layer, then recalculates the probability values. Another non-training-based approach, DoLa (Chuang et al., 2024), uses the difference in log probabilities between a lower and higher layer of the model within CD to encourage factually based text generation.

3 Method

DeLTa aims to improve the probability of generating the correct token during decoding by focusing on changes in the logits across the Transformer’s layers. Specifically, when decoding the next token in an N -layer Transformer, we regard the logits produced by each layer as a time series (§ 3.1). Based on the observation that higher layers generally assign higher probabilities to the correct token (Chuang et al., 2024), we employ a simple linear regression model (§ 3.2) to predict the logits of the higher layers. By leveraging the upward trend from lower to higher layers, this approach enhances the final prediction performance.

3.1 Token Probability at Arbitrary Layers

Let x_t denote the token at position t , and let the preceding token sequence be $x_{<t} = \{x_1, \dots, x_{t-1}\}$. The application of the Logit Lens (nostalgebraist, 2020), enables the computation of the hypothetical probability distribution at any arbitrary layer denoted by the set of all possible values of the layer,

i.e., ℓ ($1 \leq \ell \leq N$).

$$P_\ell(x_t | x_{<t}) = \text{softmax}(\text{logit}^{(\ell)})_{x_t} \quad x_t \in \mathcal{X}.$$

Here, softmax represents the softmax function, and \mathcal{X} denotes the vocabulary set.

3.2 Decoding by Logit Trajectory

We employ a linear regression to model changes in logits across Transformer layers, thereby enabling the estimation of logits at virtual layers. Because the probability of the correct token generally increases in higher layers, we explicitly learn this upward trend to produce more reliable token probabilities. Concretely, we select an intermediate layer N_{mid} ($1 \leq N_{mid} \leq N - 1$) and use its logit vectors up to the final layer N to estimate logits. We then compute token probability $P_L(x_t | x_{<t})$ for a virtual layer L ($L \in \mathbb{R}$)

Linear Regression. We define the explanatory variable \mathbf{X}_{reg} as the Transformer layer indices and the response variable \mathbf{Y}_{reg} as the logit vectors:

$$\begin{aligned} \mathbf{X}_{reg} &= [N_{mid}, N_{mid} + 1, \dots, N], \\ \mathbf{Y}_{reg} &= [\text{logit}^{(N_{mid})}, \text{logit}^{(N_{mid}+1)}, \dots, \text{logit}^{(N)}] \end{aligned}$$

Based on the least squares method, the estimated logit at a virtual layer L is computed as follows:

$$\widehat{\text{logit}}^{(L)} = \hat{\beta}_0 + \hat{\beta}_1 L,$$

where $\hat{\beta}_0$ is the intercept and $\hat{\beta}_1$ is the regression coefficient. These parameters are determined by:

$$\hat{\beta}_0 = E(\mathbf{Y}_{reg}) - \hat{\beta}_1 E(\mathbf{X}_{reg}), \hat{\beta}_1 = \frac{C(\mathbf{X}_{reg}, \mathbf{Y}_{reg})}{V(\mathbf{X}_{reg})}$$

Here, E , V , and C represent the mean, variance, and covariance, respectively.

Token Probability Computation. The final token probability is computed from $\widehat{\text{logit}}^{(L)}$, filtered by the candidate token set \mathcal{V}_{head} :

$$\hat{P}_L(x_t | x_{<t}) = \text{softmax}(\widehat{\text{logit}}_{\mathcal{V}_{head}}^{(L)})_{x_t}.$$

Here, the candidate set \mathcal{V}_{head} is determined following Chuang et al. (2024) as:

$$\mathcal{V}_{head} = \{x_t \in \mathcal{X} : \hat{P}(x_t | x_{<t}) \geq \alpha \max_w P_N(w | x_{<t})\}. \quad (1)$$

Tokens that are not included in the candidate set are assigned a probability of 0.

Model	Factuality					CoT Reasoning	
	TruthfulQA (Open QA)			Closed QA		StrQA	GSM8K
	%True \uparrow	%Info \uparrow	%True*Info \uparrow	TriviaQA	NQ		
Qwen2.5-7B	68.9	92.4	64.1	39.1	11.5	76.9	78.7
+ filter	67.4	93.3	60.7	44.1	13.0	78.1	81.6
+ DoLa (early-layer)	71.2	91.6	62.9	41.9	12.8	73.8	76.3
+ DoLa (late-layer)	79.6	75.0	55.0	33.6	10.2	67.7	67.9
+ DeLTa	66.8	98.0	65.4	44.1	13.0	81.2	81.6
Mistral-7B-v0.1	56.3	95.3	53.8	51.3	16.0	65.3	31.0
+ filter	59.4	81.1	40.9	54.5	18.2	69.7	35.8
+ DoLa (early-layer)	50.5	91.7	42.9	53.2	17.2	69.3	33.4
+ DoLa (late-layer)	51.2	91.2	42.9	53.3	17.0	71.3	33.7
+ DeLTa	54.3	92.1	47.0	54.1	17.9	72.5	38.2
Llama-3.1-8B	50.8	90.1	44.0	50.0	14.0	64.0	42.8
+ filter	50.7	95.2	46.9	53.8	16.4	66.0	47.8
+ DoLa (early-layer)	48.9	99.0	48.2	53.2	15.6	66.4	46.1
+ DoLa (late-layer)	49.2	99.3	48.5	53.1	15.3	64.9	45.9
+ DeLTa	51.5	97.1	48.9	53.8	16.4	72.1	50.1

Table 1: Experimental results on (1) factuality tasks, including TruthfulQA, TriviaQA, and Natural Questions (NQ) and (2) reasoning tasks involving Chain-of-Thought (CoT), including StrategyQA (StrQA) and GSM8K. Bold values represent the highest scores. DeLTa achieves a strong performance on the %True*Info metric for TruthfulQA and shows substantial improvements across multiple benchmarks, including TriviaQA and GSM8K. Importantly, in GSM8K, which requires not only factual knowledge but also arithmetic reasoning, DeLTa outperforms the baseline by more than 7 points. These results indicate that DeLTa enhances both knowledge-intensive tasks and complex reasoning capabilities.

4 Experiments

4.1 Setup

Models and Baselines. We use Qwen2.5-7B (Qwen Team, 2024), Mistral-7B-v0.1 (Jiang et al., 2023), and Llama-3.1-8B (Dubey et al., 2024), comparing them with four baselines. The first baseline is the raw model output. The second baseline (filter) applies $\mathcal{V}_{\text{head}}$ (Equation (1)) to the raw model output. This baseline is specifically introduced to determine whether the performance improvement of our method primarily results from the filtering mechanism rather than from DeLTa. The third and fourth baselines, DoLa (early-layer) and DoLa (late-layer), are derived from DoLa (Chuang et al., 2024), a state-of-the-art decoding method that significantly enhances generation quality by leveraging the difference in log probabilities between an intermediate layer and the final layer. DoLa dynamically selects the intermediate layer from predefined layer buckets, which are primarily partitioned into two groups: early layers (lower half of the model) and late layers (upper half of the model). We denote these two configurations as DoLa (early-layer) and DoLa (late-layer), respec-

tively. Originally, DoLa determines the optimal bucket using a validation set. However, by comparing DeLTa with both DoLa (early-layer) and DoLa (late-layer), we assess whether DeLTa remains effective regardless of the specific intermediate layer bucket selection. This evaluation highlights the robustness and general applicability of DeLTa beyond DoLa’s predefined selection strategy. We exclude methods such as Chang et al. (2024) and Sharma et al. (2024) as baselines. The former requires fine-tuning for optimal results despite being a non-training method, making fair comparison with our training-free approach difficult. The latter is designed for classification, not generation, and is thus unsuitable for our evaluation.

Tasks and Datasets. Following DoLa (Chuang et al., 2024), we evaluate open-ended generation tasks: TruthfulQA (Lin et al., 2022) (factual accuracy in open QA), StrategyQA (StrQA) (Geva et al., 2021), and GSM8K (Cobbe et al., 2021) (reasoning). To assess token-level accuracy across diverse tasks, we evaluate knowledge retrieval via closed QA tasks: TriviaQA (Joshi et al., 2017) and Natural Questions (NQ) (Kwiatkowski et al., 2019).

Evaluation metrics are in [Appendix A](#), and [Appendix B](#) details the prompt structure and generation hyperparameters. These settings follow DoLa for fair comparison. Additionally, [Appendix D](#) explains the selection of N_{mid} and virtual layer L for DeLTa.

4.2 Results

Factuality. [Table 1](#) summarizes the model performance across factuality benchmarks, demonstrating the effectiveness of DeLTa compared to strong baselines, including filtering and DoLa variants. On TruthfulQA, we focus on the $\%True*\%Info$ metric, which better reflects factual and informative responses than $\%True$ or $\%Info$ alone. This metric avoids rewarding trivial but technically correct answers. Under this measure, DeLTa improves Llama-3.1-8B from 44.0% to 48.9%, surpassing the best baseline (46.9%) by 2 points. In contrast, existing methods like filtering and DoLa (early-layer) show limited and inconsistent gains. For Closed QA tasks such as TriviaQA and NQ, generated answers typically consist of just few tokens, limiting opportunities for adjustment based on logit trajectories. Additionally, during validation, DeLTa often selects the middle layer $N_{mid} = N - 1$, causing logits from DeLTa and +Filter to coincide, thus leading to identical scores (e.g. Qwen2.5-7B achieves 44.1% on TriviaQA and 13.0% on NQ). Even in this constrained scenario, DeLTa consistently maintains or slightly exceeds baseline accuracy.

CoT Reasoning. DeLTa also substantially improves CoT reasoning accuracy, achieving up to a 7.3-point gain on GSM8K (e.g., Llama-3.1-8B: 42.8% to 50.1%), with similar improvements observed across other models. DoLa (early-layer) sometimes introduces minor improvements, while DoLa (late-layer) frequently fails to generalize, particularly on GSM8K. These results suggest that DeLTa enhances the accuracy of generated text, thereby leading to significant improvements in reasoning.

5 Analysis

In this section, we conduct a series of analyses to empirically validate the core hypotheses underpinning DeLTa. We first verify that deeper layers contribute more to task performance, then investigate the linearity of logit evolution across layers, and finally justify our choice of a linear regression model through an ablation study.

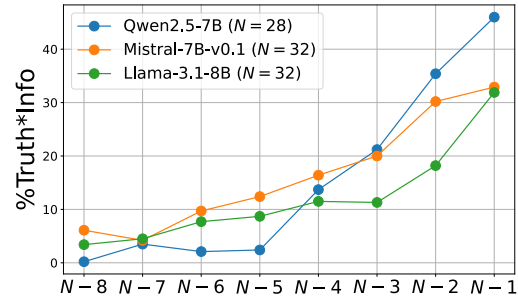


Figure 2: Direct decoding performance from intermediate layers on TruthfulQA ($\%Truth*\%Info\uparrow$). Performance consistently improves in deeper layers. The x-axis represents the layer depth from $N - 8$ (left) to $N - 1$ (right).

5.1 Information Saliency in Deeper Layers

DeLTa builds upon the hypothesis, inspired by [Chuang et al. \(2024\)](#), that task-relevant information for generating the correct token becomes more salient in the upper (deeper) layers of a Transformer model. To empirically validate this hypothesis in our setting, we performed direct decoding from the hidden states of the final eight intermediate layers (from layer $N - 8$ to $N - 1$) of three language models: Qwen2.5-7B, Mistral-7B-v0.1, and Llama-3.1-8B. We then evaluated performance on the TruthfulQA dataset using the $\%Truth*\%Info$ score.

[Figure 2](#) shows a clear trend across all models where performance improves as layers get deeper. This provides a strong empirical support for our foundational hypothesis.

5.2 Linearity of Logit Evolution Across Layers

Given the increasing importance of upper layers, we now investigate the nature of their internal dynamics. Specifically, we evaluate the extent to which the logits retain a linear structure across layers using the coefficient of determination (R^2).

Experimental Procedure. First, a text is input into the LLM, and the top 50 tokens with the highest logits in the final layer are extracted. Next, following the procedure described in [§3.2](#), the predicted and original logits of these tokens from layer N_{mid} to N are used to compute the R^2 for each token. This value is then averaged across all tokens and multiple input sentences from datasets in [§4.1](#).

Results on Mean Linearity. As shown in [Figure 3](#), all three LLMs exhibit a substantial increase in mean R^2 at higher layers, with Llama-3.1-8B

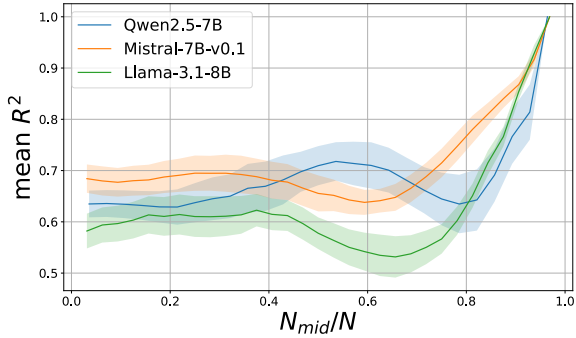


Figure 3: Mean coefficient of determination (mean R^2) and its standard deviation across input samples. The vertical axis represents the mean R^2 , and the horizontal axis represents the ratio of layer indices (N_{mid}/N).

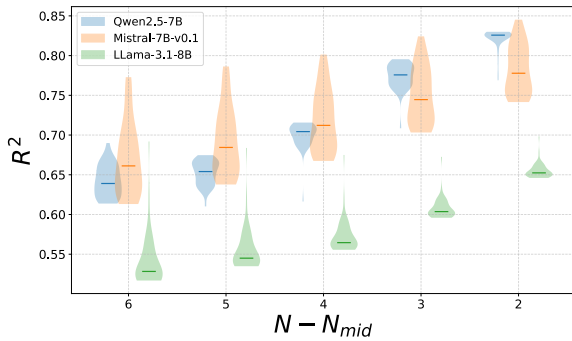


Figure 4: Violin plots of R^2 for Top-K tokens, grouped by model and layer difference ($N - N_{mid}$).

reaching approximately 0.9 near the final layer. These results indicate that a simple linear model can effectively capture logit relationships in higher layers. This finding is consistent with recent studies (Razzhigaev et al., 2024), which also demonstrate approximately linear behavior in later Transformer layers.

Distributional Analysis. To provide a more granular view, we also visualize the full distribution of R^2 for the Top-50 tokens using violin plots. As depicted in Figure 4, the R^2 distributions for all models shift toward higher values and become narrower as N_{mid} approaches the final layer. This trend reveals not only an increase in the average linearity but also a reduction in its variance, indicating more stable linear behavior. Mistral-7B-v0.1 consistently shows the highest median and the tightest distribution, while Llama-3.1-8B exhibits broader distributions, suggesting greater variability. These results reveal the stability and model-dependence of logit linearity across layers.

Model	%Truth \uparrow	%Info \uparrow	%Truth*Info \uparrow
Qwen2.5-7B (+ DeLTa)	66.8	98.0	65.4
Qwen2.5-7B (+ DeLTa2)	64.7	94.3	45.2
Mistral-7B-v0.1 (+ DeLTa)	54.3	92.1	47.0
Mistral-7B-v0.1 (+ DeLTa2)	43.5	90.3	35.7
Llama-3.1-8B (+ DeLTa)	51.5	97.1	48.9
Llama-3.1-8B (+ DeLTa2)	39.4	98.4	38.1

Table 2: Performance comparison on TruthfulQA between linear regression (DeLTa) and quadratic regression (DeLTa2).

5.3 Ablation on Regression Model Choice

The observed linearity in upper layers motivates our choice of a linear regression model. To justify this design decision, we conducted an ablation study comparing DeLTa with a version using a more complex quadratic regression model, which we call DeLTa2.

The results in Table 2 show that the linear regression model (DeLTa) significantly outperforms the quadratic version (DeLTa2) across all models. This suggests that unnecessarily increasing the model’s expressiveness harms generalization performance. We conclude that a simple and robust linear regression, which aligns with the observed linear dynamics of the upper layers, is a more effective and efficient approach.

6 Conclusion

This study aimed to enhance the factual accuracy and reasoning of text generated by LLMs. The proposed method, DeLTa, operates without additional training or data. By leveraging token probability distributions across Transformer layers and employing linear regression, we developed a framework that is both computationally efficient and easily integrable. Empirical evaluations across multiple benchmarks demonstrate that DeLTa significantly improves factual accuracy and exhibits effectiveness in reasoning tasks.

7 Limitation

The proposed method (DeLTa) in this study has limitation, as outlined below:

- Due to computational resource constraints, we could not conduct experiments on large-scale language models. Whether our approach maintains its effectiveness in larger models needs to be investigated in future studies.

Future research should focus on overcoming this limitation to establish a more generalizable and highly accurate factuality correction method applicable to a broader range of language models.

Acknowledgments

This study was partially supported by JSPS KAKENHI 22H05106, 23H03355, JST CREST JP-MJCR21N3.

References

- Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. 2024. [Explaining and improving contrastive decoding by extrapolating the probabilities of a huge and hypothetical LM](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8503–8526, Miami, Florida, USA. Association for Computational Linguistics.
- Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. [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, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- DeepSeek-AI. 2024. [Deepseek-v3 technical report. Preprint](#), arXiv:2412.19437.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- 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.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. [A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions](#). *ACM Trans. Inf. Syst.* Just Accepted.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Mandar Joshi, Eunsol Choi, Daniel 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, Vancouver, Canada. Association for Computational Linguistics.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. [Contrastive decoding: Open-ended text generation as optimization](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.
- Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. 2024. Learning to trust your feelings: Leveraging self-awareness in llms for hallucination mitigation. *arXiv preprint arXiv:2401.15449*.
- 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.
- Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, yelong shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024. [Not all tokens are what you need for pretraining](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Yusuke Nakamura and Daisuke Kawahara. 2024. [Construction of japanese truthfulqa](#). In *Proceedings of the 30th Annual Meeting of the Association for Natural Language Processing*, pages 1709–1714. In Japanese.
- nostalgebraist. 2020. [Interpreting gpt: the logit lens](#).
- OpenAI. 2024. [Hello gpt-4o](#).
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Qwen Team. 2024. [Qwen2.5: A party of foundation models](#).

Anton Razzhigaev, Matvey Mikhalechuk, Elizaveta Goncharova, Nikolai Gerasimenko, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. 2024. [Your transformer is secretly linear](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5376–5384, Bangkok, Thailand. Association for Computational Linguistics.

Mayukh Sharma, Sean O’Brien, and Julian McAuley. 2024. [Linear layer extrapolation for fine-grained emotion classification](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20880–20888, Miami, Florida, USA. Association for Computational Linguistics.

Fanqi Wan, Xinting Huang, Leyang Cui, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. [Knowledge verification to nip hallucination in the bud](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 2616–2633. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Evaluation Metrics

The evaluation metrics used for the datasets in our experiments are as follows:

- **TruthfulQA:** Following [Lin et al. \(2022\)](#), We use a fine-tuned GPT-4 model to compute the scores of truthfulness (%Truth) and informativeness (%Info) for responses generated by LLMs. %Truth measures the degree to

Input Length	DoLa	DeLTa
128 tokens	0.0205	0.0013
256 tokens	0.0358	0.0014
512 tokens	0.0453	0.0015

Table 3: Comparison of inference time (seconds per sample) between DoLa and DeLTa across varying input lengths.

which a response is factually accurate, while %Info quantifies the amount of useful information contained in the response. Additionally, %Truth*Info is the product of %Truth and %Info, representing the degree to which a response maintains truthfulness while being informative. Higher values indicate better performance.

- **TriviaQA and NQ:** The accuracy is calculated based on the exact match between the responses generated by the LLM and the gold answers.
- **StrQA and GSM8K:** The accuracy is computed based on the exact match between the extracted final answer from the LLM-generated response and the gold answer.

B Generation Hyper-parameters

The hyper-parameters employed for generation were standardized across all experiments, with the temperature parameter fixed at 0.9 and the top- k and top- p sampling parameters set to 50 and 0.95, respectively. The repetition penalty was set to 1.0 for the raw model output and to 1.2 for other methods. Furthermore, the maximum token length was set to 50 for the TruthfulQA, TriviaQA, and Natural Questions datasets, and 256 for the StrategyQA and GSM8K datasets. About α in Equation (1), we set $\alpha = 0.1$. The above parameters are all derived from DoLa ([Chuang et al., 2024](#)).

For each task, the LLM was provided with prompts and questions, and the generated responses were evaluated. The prompt structure and their selection followed [Chuang et al. \(2024\)](#). Specifically, we adopted the same few-shot examples as in [Chuang et al. \(2024\)](#) to ensure a fair comparison. However, due to computational resource constraints, we set the number of few-shot examples to 6.

C Computational Cost

To evaluate the computational cost of the algorithm itself, we measured the inference time on a synthetic model with 32 layers, hidden size of 4096, and vocabulary size of 32,000. For each input length—128, 256, and 512 tokens—we used 100 randomly generated samples and reported the average inference time in Table 3. The reason DeLTa demonstrates superior algorithmic efficiency lies in the fundamental differences between the underlying algorithms. DoLa requires sequential execution of multiple intermediate layers to dynamically determine the optimal layer using divergence-based criteria, including KL divergence and Jensen-Shannon divergence. This introduces significant computational overhead due to repeated forward passes and divergence evaluations.

In contrast, DeLTa employs a direct linear regression approach over precomputed hidden states, followed by normalization steps. Since it eliminates the need for iterative divergence computations and dynamic routing, DeLTa drastically reduces the overall computational cost.

D Configuration of DeLTa

DeLTa includes adjustable hyperparameters, N_{mid} and L . To select the optimal intermediate layer N_{mid} and the target virtual layer for estimation L for each model, we constructed validation and test datasets from each dataset. First, we determined the values of N_{mid} and L that maximize the accuracy of DeLTa for each model using the validation dataset. Then, using the selected N_{mid} and L , we conducted a comparison with the baseline on the test dataset.

For datasets without a validation set (TruthfulQA, StrQA), 10% of the test data was used as the validation data. On the other hand, for datasets with an existing validation set (TriviaQA, NQ, GSM8K), we extracted 10% of the existing validation dataset for use in our experiments.

In the experiments, we selected N_{mid} from $\{N-6, N-5, \dots, N-1\}$ and L from $\{N, N+0.5\}$. The selected values of N_{mid} and L are presented in Table 4.

After experiments, as shown in Table 4, the values of N_{mid} and L selected based on validation exhibit different tendencies depending on the dataset. Notably, in TruthfulQA, selecting an outer layer contributed to performance improvement, whereas in other datasets, optimization through smoothing

was found to be the most effective.

In conclusion, the range of selected N_{mid} and L values remains largely consistent across models, with no extreme differences observed between models. This suggests that DeLTa can be generally applied without dependence on specific datasets or models.

E Additional Experiments on Logit Linearity

E.1 Logit Linearity with Different Datasets

In this section, we examine the logit linearity across the intermediate layers of models for each dataset (TruthfulQA, TriviaQA, Natural Questions, StrategyQA, GSM8K), following the experimental procedure described in §5.2. The results are presented in Figure 5. The horizontal axis, N_{mid}/N , represents the starting point of the explanatory variables, while the vertical axis, mean R^2 , denotes the mean coefficient of determination.

When comparing the results across datasets, a general trend is observed: in the higher layers of the model (the last 4-5 layers), the mean R^2 values increase to around 0.8, indicating relatively high logit linearity. Notably, Mistral-v0.1-7B and Qwen2.5-7B consistently exhibit higher mean R^2 values than Llama-3.1-8B, suggesting that logit linearity is more pronounced in these models.

Conversely, in the lower layers, the mean R^2 values are relatively low, with significant variability across datasets and models. In particular, Llama-3.1-8B tends to have determination coefficients below 0.6 in the lower layers, suggesting lower linearity compared to other models.

Additionally, in the middle layers around $N_{mid} = 20$, a decline in mean R^2 is observed in some models. This phenomenon suggests that logits undergo nonlinear transformations in the intermediate layers. However, as the model approaches the final layers, mean R^2 increases again, indicating that logit representations become more linear.

Overall, consistent with the experimental results described in §5.2, these findings suggest that while logit linearity across layers is dataset-dependent, it generally stabilizes and improves as the model approaches the final layers.

E.2 Distributional Analysis of Logit Linearity with Different Datasets

Following §5.2, we analyze the distribution of logit linearity for the Top-K ($= 50$) tokens across inter-

Dataset	Qwen2.5-7B	Mistral-7B-v0.1	Llama-3.1-8B
TruthfulQA	26, 28.5	31, 32.5	30, 32
TriviaQA	27, 28	28, 32	31, 32
Natural Questions	27, 28	27, 32	31, 32
Strategy QA	25, 28	29, 32	28, 32
GSM8K	27, 28	26, 32	28, 32

Table 4: Results of the selected M and L . The left and right numbers in each cell represent M and L , respectively.

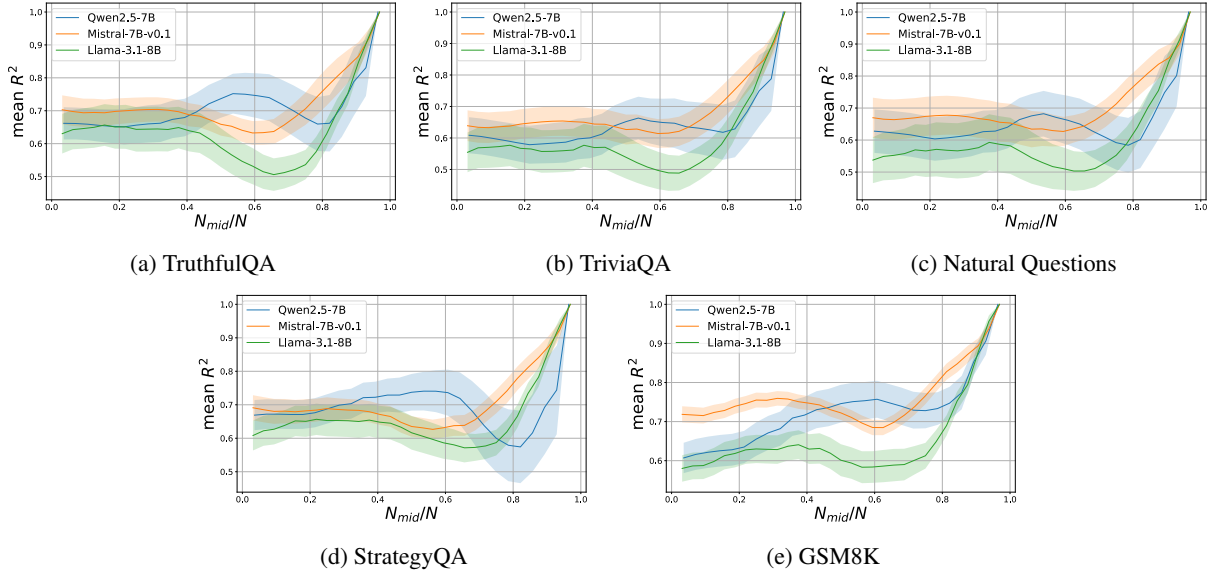


Figure 5: Logit linearity of different models (Qwen2.5-7B, Mistral-v0.1-7B, Llama-3.1-8B) on various datasets (TruthfulQA, TriviaQA, Natural Questions, StrategyQA, GSM8K), as calculated in § 5.2. The horizontal axis represents the layer ratio, while the vertical axis shows the mean R^2 , which denotes the average coefficient of determination.

mediate layers, separately for each dataset (TruthfulQA, TriviaQA, Natural Questions, StrategyQA, GSM8K), following the experimental procedure described in § 5.2. The results are presented in Figure 6.

Comparing the distributions across datasets in Figure 6, several consistent trends and dataset-specific differences can be observed. In the upper layers (small $N - N_{mid}$), the R^2 distributions for the Top-K tokens shift towards higher values and become significantly narrower, indicating that the linearity between predicted and original logits becomes both stronger and more stable as the model approaches the final layers.

In contrast, Llama-3.1-8B shows markedly lower median R^2 values and broader distributions across all datasets and layers, suggesting that its logit linearity is both weaker and less stable, especially for high-probability tokens. This model-dependent difference is especially notable in more challenging

datasets such as Natural Questions and GSM8K, where the separation between models becomes even more apparent in the upper layers.

Across all models and datasets, the lower layers (large $N - N_{mid}$) display lower R^2 values and broader distributions, indicating that the predictive power of the linear model is limited in the earlier stages of computation. In several datasets, such as TriviaQA and StrategyQA, a gradual and monotonic improvement in R^2 is observed as the model moves toward the output layer, while for others, such as TruthfulQA, some non-monotonicity and broadening of distributions in the intermediate layers can be seen, reflecting possible nonlinear transformations at these stages.

Overall, these results demonstrate that the distributional characteristics of logit linearity for Top-K tokens are jointly influenced by both model architecture and dataset properties. Nevertheless, the general tendency across all settings is that logit

linearity is strengthened and stabilized in the upper layers, consistent with findings in § 5.2 and previous sections.

E.3 Qualitative Evaluation of DeLTa

In this section, we qualitatively evaluate DeLTa using yes/no question datasets. Specifically, we input particular questions into Qwen2.5-7B and constrain their outputs to either **yes** or **no**, allowing for a detailed analysis of changes in logit scores.

First, in E.3.1, we investigate the impact of different settings of M and L on logit scores, and clarify under what conditions incorrect answers are corrected. Next, in E.3.1, we compare the logit scores with and without the application of DeLTa, verifying the effectiveness of the correction. In particular, we focus on cases where the correct answer is **no** but the model originally outputs the incorrect answer **yes**, and analyze how the logit score of **no** changes after applying DeLTa.

Through this analysis, we qualitatively evaluate how the appropriate selection of M and L influences the correction of incorrect answers, thereby demonstrating the effectiveness of DeLTa.

E.3.1 Logit Changes for Different M and L

As an example using Qwen2.5-7B, we consider the input question: What is the population of the country?
 \nA: The population is about 320 million.
 \nTrue:. In this case, the correct answer is **no**; however, the model’s original output is **yes**, resulting in an incorrect answer. The results of applying DeLTa under different values of M and L are shown in Figure 7.

Figure 7a and Figure 7b display the changes in the logits for "yes" and "no" over $M \in \{0, \dots, 27\}$ for $L = 28$ and $L = 29$, respectively. In the case of $L = 28$ (Figure 7a), the logit for the incorrect answer **yes** remains higher than that for the correct answer **no** across different M .

On the other hand, for $L = 29$ (Figure 7b), the logit for **no** begins to exceed that of **yes** around $M = 23$, indicating that DeLTa has successfully corrected the model’s error. These results demonstrate that selecting an appropriate L is crucial for effective correction of incorrect answers.

E.3.2 Detailed Analysis of the Effect of Regression-Based Correction

The effect of regression-based correction using Qwen2.5-7B is shown in Figure 8. Figure 8a ($M = 24$) and Figure 8b ($M = 27$) indicate the

Model	Accuracy	Gain
Qwen2.5-7B-Instruct	61.9	–
+filter	63.7	(+1.8)
+DoLa (early-layer)	63.2	(+1.3)
+DoLa (late-layer)	62.7	(+0.8)
+DeLTa	68.6	(+6.7)
Mistral-7B-Instruct-v0.1	28.2	–
+filter	34.7	(+6.5)
+DoLa (early-layer)	27.9	(–0.3)
+DoLa (late-layer)	28.0	(–0.2)
+DeLTa	35.9	(+7.7)
Llama-3.1-8B-Instruct	60.7	–
+filter	62.7	(+2.0)
+DoLa (early-layer)	56.2	(–4.5)
+DoLa (late-layer)	56.5	(–4.2)
+DeLTa	64.1	(+3.4)

Table 5: Factual accuracy on JTruthfulQA (Japanese). Evaluation is based on DeepSeek-V3.

original logit scores with dashed lines and the corrected logit scores after applying DeLTa with solid lines.

For $M = 24$ (Figure 8a), the logit for **yes** significantly exceeds that for **no** in the original scores, but after regression-based correction, the score for **no** increases and the gap between **yes** and **no** narrows. However, this correction is not complete, and **yes** still remains dominant.

Conversely, for $M = 27$ (Figure 8b), the corrected logit for **no** surpasses that for **yes**, leading to the correct answer. These results indicate that as M increases, the correction effect becomes more pronounced. Thus, by appropriately setting M , DeLTa can suppress incorrect answers and induce correct ones.

F Cross-Lingual Evaluation on Japanese

To examine whether the proposed decoding strategy generalizes beyond English, we evaluate its performance on a Japanese benchmark, **JTruthfulQA** (Nakamura and Kawahara, 2024), a Japanese counterpart of the TruthfulQA dataset. The factual accuracy of generated responses is automatically assessed using DeepSeek-V3 (DeepSeek-AI, 2024), which assigns a score between 0 (incorrect) and 1 (correct) to each prediction, with the final accuracy computed as the average across all instances.

DeLTa achieves consistent performance gains across all three models. On Qwen2.5-7B-Instruct, it improves accuracy to 68.6, outperforming both the base model (61.9) and the filter baseline (63.7). On Mistral-7B-Instruct-v0.1, it yields the highest

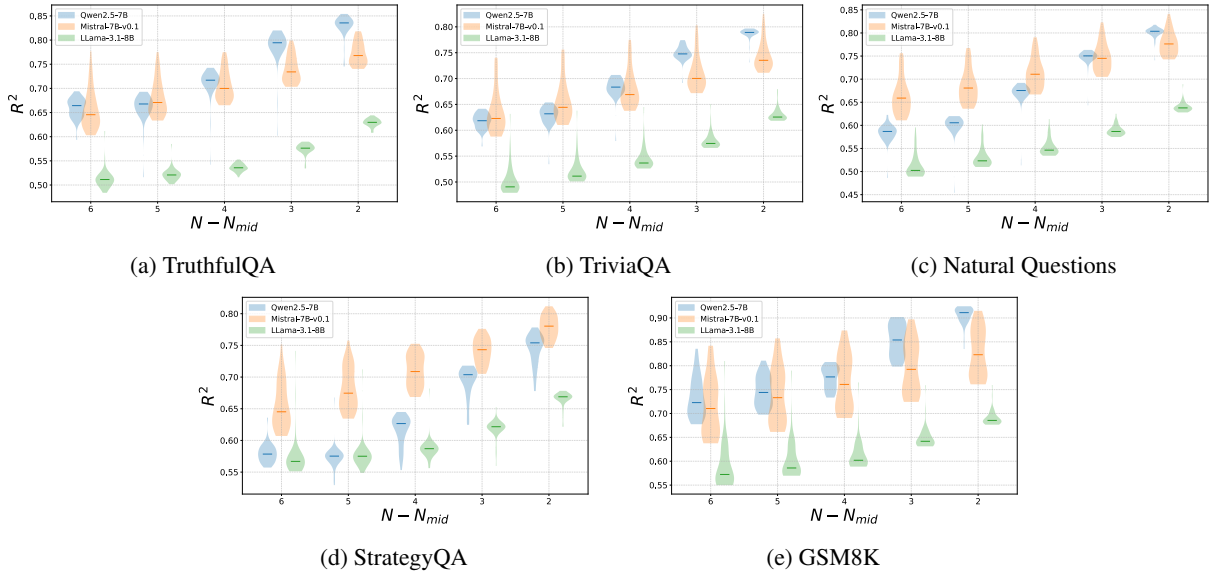


Figure 6: Distribution of logit linearity (R^2) for different models (Qwen2.5-7B, Mistral-v0.1-7B, Llama-3.1-8B) across various datasets (TruthfulQA, TriviaQA, Natural Questions, StrategyQA, GSM8K), as calculated in §E.2. The horizontal axis denotes the difference between the final layer and the intermediate layer ($N - N_{mid}$), while the vertical axis shows the coefficient of determination (R^2) for Top-50 tokens. Each violin plot visualizes the distribution of R^2 values across input samples, allowing for a comparison of both the central tendency and variability of logit linearity among models, layers, and datasets.

improvement of 7.7 points over the base model. Similarly, on Llama-3.1-8B-Instruct, it attains 64.1, surpassing all other variants. These results demonstrate that the benefits of DeLTa extend to Japanese, highlighting its cross-lingual effectiveness in enhancing factuality.

G Implementation library

We used Pytorch (Paszke et al., 2019) and huggingface transformers (Wolf et al., 2020) for all experiments.

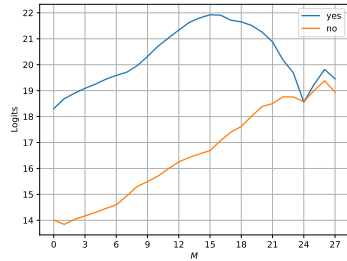
H Computational Resources

In all experiments in this study, the GPUs used were NVIDIA RTX 6000 Ada Generation or NVIDIA RTX A6000. All experiments were performed using torch.float16, and the VRAM used was approximately 15GB to 40GB.

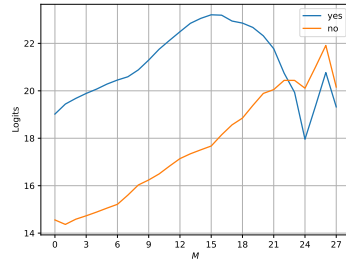
I Generated Examples

In this section, we present response examples from Qwen2.5-7B and DeLTa for questions in the GSM8K dataset. The example sentences included in the few-shot prompt are omitted, and only the pure model responses are compared. These examples illustrate cases where DeLTa functioned effectively. Among the responses to each question

shown in Table 6, the output of DeLTa is listed in the right column.

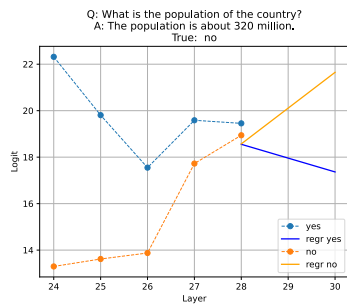


(a) $L = 28$

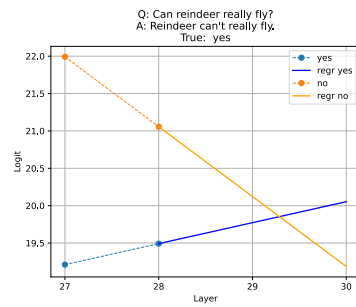


(b) $L = 29$

Figure 7: Results of applying DeLTa to the Qwen2.5-7B model. The plots show the trajectories of the logit scores for tokens **yes** and **no** as M varies under different values of L ($L = 28, 29$). The proposed method tends to reduce the gap between the logit scores of **yes** and **no** for certain M .



(a) $M = 24$



(b) $M = 27$

Figure 8: Examples of DeLTa applied to Qwen2.5-7B. The dashed lines show the original logit scores, while the solid lines show the logit scores after regression-based correction by DeLTa (e.g. label “regr yes” stands for regression line for logit “yes”).

Method	Qwen2.5-7B	Qwen2.5-7B + DeLTa
Question	Cecilia just bought a new puppy. According to her veterinarian, she has to feed the puppy 1 cup of dog food every day for the first 180 days. Then she has to feed the puppy 2 cups of dog food every day for the rest of its life. If one bag of dog food contains 110 cups, how many bags of dog food will Cecilia use in the first year?	
Answer	The puppy needs to be fed 1 cup of dog food per day for 180 days, so the total amount of food needed for the first 180 days is 1 cup/day x 180 days = 180 cups of dog food. If one bag of dog food contains 110 cups, then Cecilia will need 180 cups / 110 cups/bag = 1.64 bags of dog food. Since she can't buy a fraction of a bag, she will need to buy 2 bags of dog food for the first year. The answer is 2.	In the first 180 days, the puppy will consume $180 * 1 = 180$ cups of dog food. In the first year, there are 365 days. So after the first 180 days, there are $365 - 180 = 185$ days remaining. In those remaining days, the puppy will consume $185 * 2 = 370$ cups of dog food. In total, the puppy will consume $180 + 370 = 550$ cups of dog food in the first year. Since one bag contains 110 cups, Cecilia will need $550 / 110 = 5$ bags of dog food in the first year. The answer is 5.
Question	Sadie slept 8 hours on Monday. For the next two days, she slept 2 hours less, each, because she had to complete some assignments. If the rest of the week she slept 1 hour more than those two days, how many hours did she sleep in total throughout the week?	
Answer	Sadie slept 8 hours on Monday. On the next two days, she slept 2 hours less each day, so $8 - 2 = 6$ hours each day. For those two days, she slept $6 * 2 = 12$ hours in total. For the rest of the week, she slept 1 hour more than those two days, so $6 + 1 = 7$ hours each day. There are 5 days in the week, so for the rest of the week, she slept $7 * 5 = 35$ hours. In total, she slept 8 (Monday) + 12 (next two days) + 35 (rest of the week) = 55 hours throughout the week. The answer is 55.	Sadie slept 8 hours on Monday. On Tuesday and Wednesday, she slept 2 hours less than on Monday, so she slept $8 - 2 = 6$ hours on each of those days. In total for those three days, she slept $8 + 6 + 6 = 20$ hours. For the rest of the week, she slept 1 hour more than on those two days, so she slept $6 + 1 = 7$ hours on each of those days. She has four more days in the week, so she slept $7 * 4 = 28$ hours on those days. In total, she slept $20 + 28 = 48$ hours throughout the week. The answer is 48.
Question	There are 700 bees in a hive. There are twice as many worker bees as baby bees, and there are twice as many babies as queens. How many worker bees are there?	
Answer	The total ratio is 2 (workers) + 1 (baby) + 1 (queen) = 4 parts. Each part has $700 / 4 = 175$ bees. Therefore, there are $2 * 175 = 350$ worker bees. The answer is 350.	Let's represent the number of worker bees as W, the number of baby bees as B, and the number of queens as Q. We know that there are 700 bees in total, so $W + B + Q = 700$. We also know that there are twice as many worker bees as baby bees, so $W = 2B$, and there are twice as many babies as queens, so $B = 2Q$. We can substitute these equations into the first equation to get $2B + B + Q = 700$. Combining like terms, we get $3B + Q = 700$. Now we can substitute $B = 2Q$ into this equation to get $3(2Q) + Q = 700$. Simplifying, we get $6Q + Q = 700$. Combining like terms, we get $7Q = 700$. Dividing both sides by 7, we get $Q = 100$. Now we can substitute $Q = 100$ back into $B = 2Q$ to get $B = 2(100) = 200$. Finally, we can substitute $B = 200$ back into $W = 2B$ to get $W = 2(200) = 400$. The answer is 400.
Question	Billy is peeling and cutting potatoes in preparation for making potato salad for his big family reunion barbecue. It's a big event, so he has 60 potatoes to get through. The potatoes are roughly the same size, so it takes about the same amount of time to peel and cut each one. It takes Billy about a minute and a half to peel a potato, but only about 5 seconds to cut it up. How long will it take Billy to finish prepping the potatoes?	
Answer	It takes Billy about 1.5 minutes to peel a potato, which is $1.5 * 60 = 90$ seconds. It takes him about 5 seconds to cut each potato. So it takes him about $90 + 5 * 60 = 90 + 300 = 390$ seconds. To convert this into minutes, we divide by 60, so it takes Billy about $390 / 60 = 6.5$ minutes to finish prepping the potatoes. The answer is 6.5.	It takes 1.5 minutes to peel each potato, so for 60 potatoes, it will take $1.5 * 60 = 90$ minutes to peel them all. It takes 5 seconds to cut each potato, so for 60 potatoes, it will take $5 * 60 = 300$ seconds to cut them all. To convert seconds to minutes, we divide by 60, so it will take $300 / 60 = 5$ minutes to cut the potatoes. In total, it will take 90 minutes + 5 minutes = 95 minutes to prep the potatoes. The answer is 95.
Question	Juan and his brother Carlos are selling lemonade. For each gallon they make it costs \$3 for lemons and \$2 for sugar. They sell each glass for \$0.50 and get 20 glasses per gallon. If they made \$25 in profit, how much did they spend on lemons?	
Answer	They spent \$5 per gallon because $3 + 2 = 5$. They sold 8 gallons because $25 / 3.125 = 8$. They spent \$40 on lemons because $8 * 5 = 40$. The answer is 40	They spent \$3 per gallon on lemons and \$2 per gallon on sugar, so a total of $\$3 + \$2 = \$5$ per gallon. They sell 20 glasses per gallon at \$0.50 each, so they make $20 * \$0.50 = \10 per gallon. They made \$25 in profit, so they must have made $\$25 + \$5 = \$30$ in total. Since they make \$10 per gallon, they must have made $\$30 / \$10 = 3$ gallons. So they spent $3 * \$5 = \15 on lemons. The answer is 15.

Table 6: Examples of responses to the GSM8K dataset by Qwen2.5-7B and DeLTa. The baseline exhibits numerical calculation errors and reasoning mistakes, whereas DeLTa yields more accurate computation results.