DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models

Yung-Sung Chuang[†][⋆]**, Yujia Xie**[‡]**, Hongyin Luo**[†]**, Yoon Kim**[†]**, James Glass**[†]**, Pengcheng He**[‡]
[†]Massachusetts Institute of Technology, [‡]Microsoft
yungsung@mit.edu, yujiaxie@microsoft.com
{hyluo,yoonkim,glass}@mit.edu, herbert.he@gmail.com

Abstract

Despite their impressive capabilities, large language models (LLMs) are prone to hallucinations, i.e., generating content that deviates from facts seen during pretraining. We propose a simple decoding strategy for reducing hallucinations with pretrained LLMs that does not require conditioning on retrieved external knowledge nor additional fine-tuning. Our approach obtains the next-token distribution by contrasting the differences in logits obtained from projecting the later layers versus earlier layers to the vocabulary space, exploiting the fact that factual knowledge in an LLMs has generally been shown to be localized to particular transformer layers. We find that this **D**ecoding by **C**ontrasting **La**yers (DoLa) approach is able to better surface factual knowledge and reduce the generation of incorrect facts. DoLa consistently improves the truthfulness across multiple choices tasks and open-ended generation tasks, for example improving the performance of LLaMA family models on TruthfulQA by 12-17% absolute points, demonstrating its potential in making LLMs reliably generate truthful facts.[1]

1 Introduction

Large language models (LLMs) have demonstrated great potential in numerous natural language processing (NLP) applications (Brown et al., 2020; OpenAI, 2022; 2023). However, despite the continued increase in performance and the emergence of new capabilities from scaling LLMs (Wei et al., 2022a), their tendency to "hallucinate", i.e., generate content that deviates from real-world facts observed during pretraining (Ji et al., 2023), remains a persistent challenge. This represents a major bottleneck in their deployment especially for high-stakes applications (e.g., clinical/legal settings) where reliable generation of trustworthy text is crucial.

While the exact reasons for LMs' hallucinations are not fully understood, a possible reason is due to the maximum likelihood language modeling objective which minimize the forward KL divergence between the data and model distributions. This objective potentially results in a model with mass-seeking behavior which causes the LM to assign non-zero probability to sentences that are not fully consistent with knowledge embedded in the training data. Empirically, an LM trained with the next-word prediction objective on finite data has been shown to result in a model that uses linguistic knowledge to recognize the superficial patterns, instead of recognizing and generating the real-world facts extracted from the training corpus (Ji et al., 2023).

From a model interpretability perspective, transformer LMs have been loosely shown to encode "lower-level" information (e.g., part-of-speech tags) in the earlier layers, and more "semantic" information in the later layers (Tenney et al., 2019). More recently, Dai et al. (2022) find that "knowledge neurons" are distributed in the topmost layers of the pretrained BERT model. Meng et al. (2022) show that factual knowledge

---

[1]The source code is available at https://github.com/voidism/DoLa.
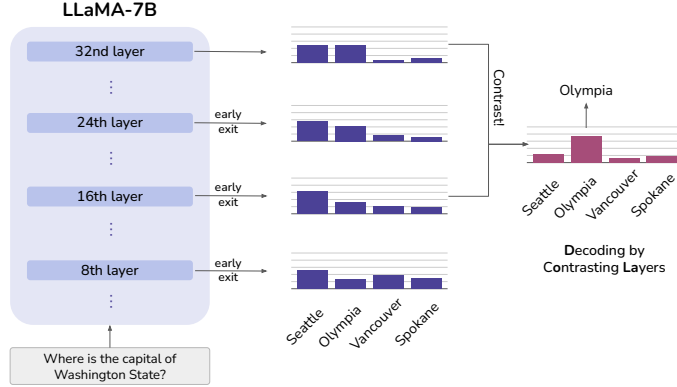[⋆]Work mainly done during an internship at Microsoft.

Figure 1: Illustration of an LLM progressively incorporates factual information along layers. While the next-word probabilities of "*Seattle*" remain similar throughout different layers, the probabilities of the correct answer "*Olympia*" gradually increase from lower to higher layers. DoLa uses this fact to decode by contrasting the difference between layers to sharpen an LLM's probability towards factually correct outputs.

can even be edited by manipulating a specific set of feedforward layers within an autoregressive LM. We propose to exploit this modular encoding of knowledge to amplify the factual knowledge in an LM through a contrastive decoding approach, where the output next-word probability is obtained from the *difference* in logits between a higher layer versus a lower layer. By emphasizing the knowledge of higher layers and downplaying that of lower layers, we can potentially make LMs more factual and thus reduce hallucinations.

An illustration of this idea for a simple example is shown in Figure 1. While "*Seattle*" maintains high probability throughout all the layers—presumably because it is a syntactically plausible answer—the probability of the true answer "*Olympia*" increases after the higher layers inject more factual knowledge. Contrasting the differences between the different layers can thus reveal the true answer in this case. Based on this concept, we propose a new decoding method, **D**ecoding by **Co**ntrasting **La**yers (DoLa), for better surfacing factual knowledge embedded in an LLM without retrieving external knowledge or additional fine-tuning.

Experiments on TruthfulQA (Lin et al., 2022) and FACTOR Muhlgay et al. (2023) demonstrate that DoLa is able to increase the truthfulness of the models of the LLaMA family (Touvron et al., 2023). Further experiments on chain-of-thought reasoning for StrategyQA (Geva et al., 2021) and GSM8K (Cobbe et al., 2021) also show that it can facilitate more factual reasoning. Finally, experiments using GPT-4 for open-ended chatbot evaluation (Chiang et al., 2023) show that when compared with the original decoding method, DoLa can generate informative and significantly more factual responses that lead to better ratings from GPT-4. From an efficiency perspective, we find that DoLa causes only a small additional latency in the decoding process, suggesting it as a practical and useful decoding strategy for improving the truthfulness of LLMs.

2 METHOD

Recent language models consist of an embedding layer, N stacked transformer layers, and an affine layer $\phi(\cdot)$ for predicting the next-word distribtution. Given a sequence of tokens $\{x_1, x_2, \ldots, x_{t-1}\}$, the embedding layer first embeds the tokens into a sequence of vectors $H_0 = \{h_1^{(0)}, \ldots, h_{t-1}^{(0)}\}$. Then H_0 would be processed by each of the transformer layers successively. We denote the output of the j-th layer as H_j. Then, the vocabulary head $\phi(\cdot)$ predicts the probability of the next token x_t over the vocabulary set $\mathcal{X}$,

$$p(x_t \mid x_{<t}) = \text{softmax}\big(\phi(h_t^{(N)})\big)_{x_t}, \quad x_t \in \mathcal{X}.$$

Instead of applying ϕ on the final layer, our approach contrasts the higher-layer and lower-layer information to obtain the next-token probability. More specifically, for the j-th early layer, we also compute the next-

Input: *Who was the first Nigerian to win the Nobel Prize, in which year?*
Output: <u>*Wole Soyinka*</u> *was the first Nigerian to win the Nobel Prize, in* <u>*1986*</u>.

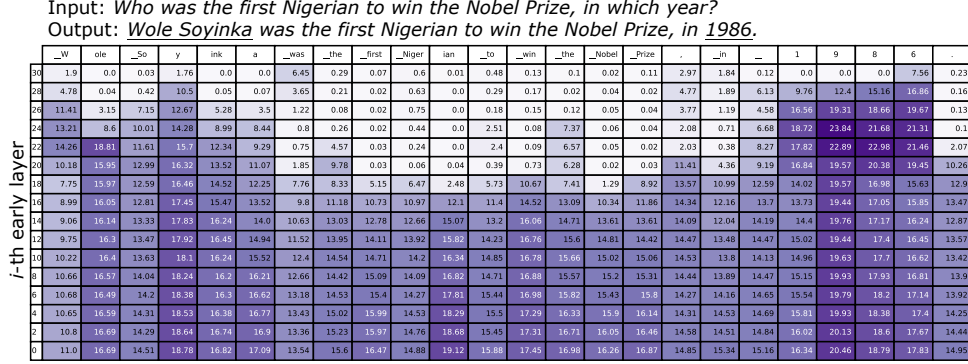| layer	_W	ole	_So	y	ink	a	_was	_the	_first	_Niger	ian	_to	_win	_the	_Nobel	_Prize	.	_	_in	_	1	9	8	6	.
30	1.9	0.0	0.03	1.76	0.0	0.0	6.45	0.29	0.07	0.6	0.01	0.48	0.13	0.1	0.02	0.11	2.97	1.84	0.12		0.0	0.0	0.0	7.56	0.23
28	4.78	0.04	0.42	10.5	0.05	0.07	3.65	0.21	0.02	0.63	0.0	0.29	0.17	0.02	0.04	0.02	4.77	1.89	6.13		9.76	12.4	15.16	16.05	0.16
26	11.41	3.15	7.15	12.67	5.28	3.5	1.22	0.08	0.02	0.75	0.0	0.18	0.15	0.12	0.05	0.04	3.77	1.19	4.58		16.56	19.31	18.66	19.67	0.13
24	13.21	8.6	10.01	14.28	8.99	8.44	0.8	0.26	0.02	0.44	0.0	2.51	0.08	7.37	0.06	0.04	2.08	0.71	6.68		18.72	23.84	21.68	21.31	0.1
22	14.26	18.81	11.61	15.7	12.34	9.29	0.75	4.57	0.03	0.24	0.0	2.4	0.09	6.57	0.05	0.02	2.03	0.38	8.27		17.82	22.89	22.98	21.46	2.07
20	10.18	15.95	12.99	16.32	13.52	11.07	1.85	9.78	0.03	0.06	0.04	0.39	0.73	6.28	0.02	0.03	11.41	4.36	9.19		16.84	19.57	20.38	19.45	10.26
18	7.75	15.97	12.59	16.46	14.52	12.25	7.76	8.33	5.15	6.47	2.48	5.73	10.67	7.41	1.29	8.92	13.57	10.99	12.59		14.02	19.57	16.98	15.63	12.9
16	8.99	16.05	12.81	17.45	15.47	13.52	9.8	11.18	10.73	10.97	12.1	11.4	14.52	13.09	10.34	11.86	14.34	12.16	13.7		13.73	19.44	17.05	15.85	13.47
14	9.06	16.14	13.33	17.83	16.24	14.0	10.63	13.03	12.78	12.66	15.07	13.2	16.06	14.71	13.61	13.61	14.09	12.04	14.19		14.4	19.76	17.17	16.24	12.87
12	9.75	16.3	13.47	17.92	16.45	14.94	11.52	13.95	14.11	13.92	15.82	14.23	16.76	15.6	14.81	14.42	14.47	13.48	14.47		15.02	19.44	17.4	16.45	13.57
10	10.22	16.4	13.63	18.1	16.24	15.52	12.4	14.54	14.71	14.2	16.34	14.85	16.78	15.66	15.02	15.06	14.53	13.8	14.13		14.96	19.63	17.7	16.62	13.42
8	10.66	16.57	14.04	18.24	16.2	16.21	12.66	14.42	15.09	14.09	16.82	14.71	16.88	15.57	15.2	15.31	14.44	13.89	14.47		15.15	19.93	17.93	16.81	13.9
6	10.68	16.49	14.2	18.38	16.3	16.62	13.18	14.53	15.4	14.27	17.81	15.44	16.98	15.82	15.43	15.8	14.27	14.16	14.65		15.54	19.79	18.2	17.14	13.92
4	10.65	16.59	14.31	18.53	16.38	16.77	13.43	15.02	15.99	14.53	18.29	15.5	17.29	16.33	15.9	16.14	14.31	14.53	14.69		15.81	19.93	18.38	17.4	14.25
2	10.8	16.69	14.29	18.64	16.74	16.9	13.36	15.23	15.97	14.76	18.68	15.45	17.31	16.71	16.05	16.46	14.58	14.51	14.84		16.02	20.13	18.6	17.67	14.44
0	11.0	16.69	14.51	18.78	16.82	17.09	13.54	15.6	16.47	14.88	19.12	15.88	17.45	16.98	16.26	16.87	14.85	15.34	15.16		16.34	20.46	18.79	17.83	14.95

i-th early layer

Figure 2: JSD (scaled by 10^5) between the final 32nd layer and even-numbered early layers. Column names are decoded tokens in each step. Row names are indices of the early layers. 0 means word embedding layer.

token probability using $\phi(\cdot)$ as follows, where $\mathcal{J} \subset \{0, \ldots, N-1\}$ is a set of candidate layers,

$$q_j(x_t \mid x_{<t}) = \text{softmax}\big(\phi(h_t^{(j)})\big)_{x_t}, \quad j \in \mathcal{J}.$$

The idea of applying language heads directly to the hidden states of the middle layers, known as *early exit* (Teerapittayanon et al., 2016; Elbayad et al., 2020; Schuster et al., 2022), has proven to be effective even without special training process (Kao et al., 2020), as the residual connections (He et al., 2016) in transformer layers make the hidden representations gradually evolve without abrupt changes. Using $q_j(x_t)$ to represent $q_j(x_t \mid x_{<t})$ for notational brevity, we then compute the probability of the next token by,

$$\hat{p}(x_t \mid x_{<t}) = \text{softmax}\big(\mathcal{F}\big(q_N(x_t), q_M(x_t)\big)\big)_{x_t},$$
$$\text{where} \quad M = \arg\max_{j \in \mathcal{J}} \, d\big(q_N(\cdot), q_j(\cdot)\big).$$

Here, layer M is named *premature layer*, while the final layer, i.e., layer N, is named *mature layer*. The operator $\mathcal{F}(\cdot, \cdot)$, to be elaborated further in Section 2.3, is used to contrast between the output distributions from the premature layer and the mature layer by computing the log-domain difference between two distributions. The premature layer is dynamically selected in each decoding step using a distributional distance measure $d(\cdot, \cdot)$ (we use Jensen-Shannon Divergence) between the mature layer and all the candidate layers in $\mathcal{J}$. We discuss $d(\cdot, \cdot)$ in more detail in Section 2.2. The motivation for selecting the layer with the highest distance $d(\cdot, \cdot)$ is to ensure that the model would significantly change its output after that selected layer, and thus have a higher chance to include more factual knowledge that does not exist in the early layers before it.

2.1 Factual Knowledge Evolves Across Layers

We conduct preliminary analysis with 32-layer LLaMA-7B (Touvron et al., 2023) to motivate our approach. We compute the Jensen-Shannon Divergence (JSD) between the early exiting output distributions $q_j(\cdot \mid x_{<t})$ and the final layer output distribution $q_N(\cdot \mid x_{<t})$, to show how the early exiting outputs are different from the final layer outputs. Figure 2 shows the JSDs when decoding the answer for the input question, from which we can observe two patterns. **Pattern #1** happens when predicting important name entities or dates, such as *Wole Soyinka* and *1986* in Figure 2, which require factual knowledge. We observe the calculated JSD would be still extremely high in the higher layers. This pattern indicates that the model is still changing its predictions in the last few layers, and potentially injecting more factual knowledge into the predictions. **Pattern #2** happens when predicting function words, such as *was, the, to, in*, and the tokens copied from the input question, such as *first Nigerian, Nobel Prize*. When predicting these "easy" tokens, we can observe that the JSD becomes very small from middle layers. This finding indicates that the model has already decided what token to generate in middle layers, and keeps the output distributions almost unchanged in the higher
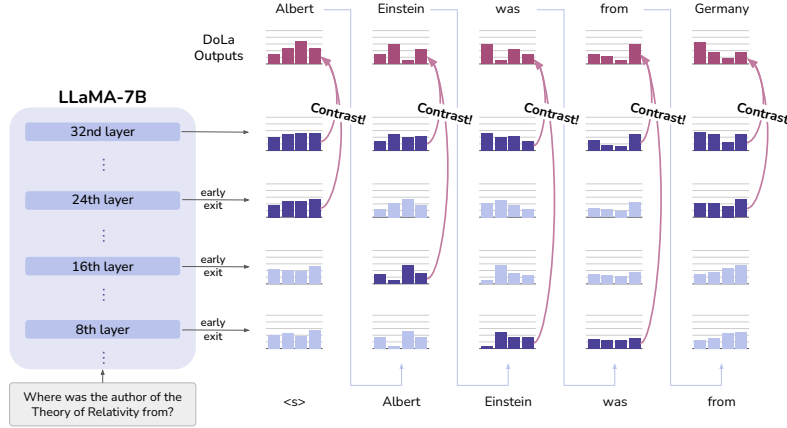
Figure 3: The illustration of how dynamic premature layer selection works.

layers. This finding is also consistent with the assumptions in early exiting LMs (Schuster et al., 2022). A preliminary analysis that can quantitatively support this observation is also shown in Appendix A.

Qualitatively, when the next-word prediction requires factual knowledge, LLaMA seems to to change the predictions in the higher layers. Contrasting the layers before/after a sudden change may therefore amplify the knowledge emerging from the higher layers and make the model rely more on its factual internal knowledge. Moreover, this evolution of information seems to vary token by token. Our method requires accurately selecting the premature layer that contains *plausible but less factual* information, which may not always stay in the same early layer. Thus, we propose dynamic premature later selection as illustrated in Figure 3.

2.2 DYNAMIC PREMATURE LAYER SELECTION

To magnify the effectiveness of contrastive decoding, the optimal premature layer should ideally be the layer most different from the final-layer outputs. To allow for dynamic premature layer selection at each time step, we adopt the following measure of distance between the next-word distributions obtained from two layers,

$$d\big(q_N(\cdot\,|\,x_{<t}), q_j(\cdot\,|\,x_{<t})\big) = \mathrm{JSD}\big(q_N(\cdot\,|\,x_{<t})\|q_j(\cdot\,|\,x_{<t})\big),$$

where $\mathrm{JSD}(\cdot,\cdot)$ is the Jensen-Shannon divergence. The premature layer, i.e., the M-th layer ($0 \leq M < N$), is then selected as the layer with the maximum divergence among the subset of early layers,

$$M = \arg\max_{j\in\mathcal{J}} \mathrm{JSD}\big(q_N(\cdot\,|\,x_{<t})\|q_j(\cdot\,|\,x_{<t})\big),$$

where $\mathcal{J}$ is a set of candidate layers for premature layer selection. For LLaMA models with various number of layers, we divide the layers into 2 to 4 buckets of $\mathcal{J}$ based on their total layers, in order to focus on contrasting from a certain range of layers. The best bucket for each task is chosen using a validation set, as detailed in Section 3.1. This dynamic layer selection strategy enables the the selection of suitable premature layers based on token difficulty, thereby making better use of the knowledge learned by different layers.

Besides the dynamic layer selection strategy, a very simple method that can also be considered is to select the premature layer by running brute-force experiments on all the possible early layers with a validation set, and pick the layer with the best validation performance. We refer to this simple method as DoLa-static. However, DoLa-static has the drawbacks of 1) requiring more hyperparameter search runs in layers and the fact that 2) best layers are sensitive to data distribution, thus requiring in-distribution validation sets. Our proposed dynamic layer selection strategy also mitigates the drawbacks of DoLa-static by shrinking the layer search space and making the method more robust without heavily relying on in-distribution validation sets. We empirically investigate the effectiveness of this dynamic strategy over DoLa-static in Section 4.1.

2.3 CONTRASTING THE PREDICTIONS

Given the premature and mature layers obtained from Section 2.2, we aim to amplify mature layer outputs while downplaying premature layer outputs. Following the Contrastive Decoding approach from Li et al. (2022), we subtract the log probabilities of the premature layer outputs from those of the mature layer. We then use this resulting distribution as the next-word prediction, as illustrated in Figure 1,

$$\hat{p}(x_t \mid x_{<t}) = \text{softmax}\big(\mathcal{F}\big(q_N(x_t), q_M(x_t)\big)\big)_{x_t}, \quad \text{where}$$

$$\mathcal{F}\big(q_N(x_t), q_M(x_t)\big) = \begin{cases} \log \dfrac{q_N(x_t)}{q_M(x_t)}, & \text{if } x_t \in \mathcal{V}_{\text{head}} \ (x_t|x_{<t}) \,, \\ -\infty, & \text{otherwise.} \end{cases}$$

Similar to Li et al. (2022), the subset $\mathcal{V}_{\text{head}} \ (x_t|x_{<t}) \in \mathcal{X}$ is defined as whether or not the token has high enough output probabilities from the mature layer,

$$\mathcal{V}_{\text{head}} \ (x_t|x_{<t}) = \Big\{ x_t \in \mathcal{X} : q_N(x_t) \geq \alpha \max_w q_N(w) \Big\}.$$

If the predicted probability of a token is too small in the mature layer, it is not likely to be a reasonable prediction, so we set the token probability to zero to minimize false positive and false negative cases. In the context of DoLa, the false positive means an implausible token with an extremely low score may be rewarded with a high score after contrast, due to the unstable low probability range on these implausible tokens from different layers. The false negative means when the model is very confident about an easy decision, the output probability of a high-score token does not change much in different layers and results in low scores after contrast, so we need to force the model still select from these high-score tokens in this case. This strategy is referred as an *adaptive plausibility constraint* (APC) proposed in Li et al. (2022).

Repetition Penalty. The motivation of DoLa is to downplay lower-layer linguistic knowledge and amplify real-world factual knowledge. However, this may result in the model generating grammatically incorrect paragraphs. Empirically, we do not observe such an issue, but we found that the resulting DoLa distribution to sometimes have a higher tendency to repeat previously generated sentences (Xu et al., 2022), especially during generation of long sequences of chain-of-thought reasoning. Here we include a simple repetition penalty introduced in Keskar et al. (2019) with $\theta = 1.2$ during decoding. The empirical analysis of the repetition penalty is shown in Appendix K.

3 EXPERIMENTS

3.1 SETUP

Datasets. We consider *multiple choices* and *open-ended generation* tasks. For multiple choices, we use TruthfulQA (Lin et al., 2022) and FACTOR (News/Wiki) (Muhlgay et al., 2023) to assess LMs' factuality in short-answer/long-paragraph settings, respectively. For open-ended generation, we use TruthfulQA (rated by fine-tuned GPT-3) (Lin et al., 2022) and tasks involving chain-of-thought (Wei et al., 2022b) reasoning: StrategyQA (Geva et al., 2021) and GSM8K Cobbe et al. (2021). Finally, we test Vicuna QA (Chiang et al., 2023) which uses GPT-4 to evaluate instruction-following abilities as chatbot assistants.

Models and Baselines. We examine four sizes of LLaMA models (Touvron et al., 2023) (7B, 13B, 33B, 65B) and compare them with three baselines: 1) original decoding (greedy decoding or sampling depending on the tasks), 2) Contrastive Decoding (CD) (Li et al., 2022), where LLaMA-7B serves as the amateur model and LLaMA-13B/33B/65B act as expert models, and 3) Inference Time Intervention (ITI). ITI uses LLaMA-7B and a linear classifier trained on TruthfulQA. Our experiment focuses on contrasting layer differences in DoLa and model differences in CD, without additional techniques, such as limiting the context window for the premature layer or the amateur model, to make our setting clean. We set adaptive plausibility constraint (α) to 0.1 and repetition penalty (θ) to 1.2 as per prior studies(Li et al., 2022; Keskar et al., 2019).

Model	TruthfulQA (MC)			FACTOR		TruthfulQA (Open-Ended Generation)				CoT	
	MC1	MC2	MC3	News	Wiki	%Truth ↑	%Info ↑	%T*I ↑	%Reject ↓	StrQA	GSM8K
LLaMa-7B	25.6	40.6	19.2	58.3	58.6	30.4	96.3	26.9	2.9	60.1	**10.8**
+ ITI (Li et al., 2023)	25.9	-	-	-	-	49.1	-	**43.5**	-	-	-
+ DoLa	**32.2**	**63.8**	**32.1**	**62.0**	**62.2**	42.1	98.3	40.8	0.6	**64.1**	10.5
LLaMa-13B	28.3	43.3	20.8	61.1	62.6	38.8	93.6	32.4	6.7	66.6	16.7
+ CD (Li et al., 2022)	24.4	41.0	19.0	62.3	64.4	55.3	80.2	44.4	20.3	60.3	9.1
+ DoLa	**28.9**	**64.9**	**34.8**	**62.5**	**66.2**	48.8	94.9	**44.6**	2.1	**67.6**	**18.0**
LLaMa-33B	31.7	49.5	24.2	63.8	69.5	62.5	69.0	31.7	38.1	69.9	33.8
+ CD (Li et al., 2022)	**33.0**	51.8	25.7	63.3	**71.3**	81.5	45.0	36.7	62.7	66.7	28.4
+ DoLa	30.5	**62.3**	**34.0**	**65.4**	70.3	56.4	92.4	**49.1**	8.2	**72.1**	**35.5**
LLaMa-65B	30.8	46.9	22.7	63.6	72.2	50.2	84.5	34.8	19.1	70.5	51.2
+ CD (Li et al., 2022)	29.3	47.0	21.5	64.6	71.3	75.0	57.9	43.4	44.6	70.5	44.0
+ DoLa	**31.1**	**64.6**	**34.3**	**66.2**	**72.4**	54.3	94.7	**49.2**	4.8	**72.9**	**54.0**

Table 1: Experimental results on 1) multiple choices dataset: TruthfulQA and FACTOR and 2) open-ended generation tasks: TruthfulQA and Chain-of-Thought (CoT) reasoning tasks, including StrategyQA (StrQA) and GSM8K. **%T*I** stands for **%Truth*Info** in TruthfulQA.

Candidate Layers. In dynamic premature layer selection, we partition transformer layers into buckets and select one bucket as candidate layers ($\mathcal{J}$). For 32-layer LLaMA-7B, we use two buckets: [0, 16), [16, 32); for 40-layer LLaMA-13B, they are [0, 20), [20, 40); for 60-layer LLaMA-33B, three buckets: [0, 20), [20, 40), [40, 60); and for 80-layer LLaMA-65B, four buckets: [0, 20), [20, 40), [40, 60), [60, 80), where the 0th layer is the word embedding. This design limits the hyperparameter search space to only 2-4 validation runs. For efficiency, only even-indexed layers (0th, 2nd, etc.) are considered as candidates. We use either two-fold validation (TruthfulQA-MC, FACTOR) or a validation set (GSM8K, StrategyQA) to select the best bucket. For Vicuna QA, which lacks a validation set, we use GSM8K's best bucket.

3.2 MULTIPLE CHOICES

Short-Answer Factuality. We test TruthfulQA with the default QA prompt from Lin et al. (2022) and Li et al. (2023). For α in APC, we replace $-\infty$ with -1000 to avoid ruining LM likelihood scores, which also applies to FACTOR. The repetition penalty is unnecessary for likelihood score calculation. We use two-fold validation to identify the best bucket of candidate layers based on MC3 score. Results in Table 1 show significant performance improvement for LLaMA models in four sizes, outperforming ITI/CD and confirming the effectiveness of DoLa. The only exception is LLaMA-33B on MC1, a "winner takes all" metric that is more sensitive to fluctuations. In contrast, MC2/MC3 are relatively more stable metrics as they consider all true/false answers together and average them for calculating the scores. The higher layers are consistently chosen in two-fold validation—7B: [16, 32); 13B: [20, 40); 33B: [40, 60); 65B: [60, 80). Implementation details and extra results of contrasting with the 0-th layer / all layers are shown in Appendix C.

Long-Paragraph Factuality. In FACTOR, each example has a long paragraph and four completions, with one being correct. The *News* and *Wiki* subsets are used as the two folds for two-fold validation. Table 1 shows DoLa outperforms baselines by 2-4%, and is more effective than CD, except for 13B on Wiki. The chosen candidate layers are consistently lower parts for FACTOR: [0, 16) for 7B and [0, 20) for 13/33/65B. This differs from TruthfulQA, which selects higher layers. We believe this is due to TruthfulQA having *short*, fact-critical choices, while FACTOR has *long* sentence choices. As noted in Section 2.1, contrasting with higher layers works better for key facts, while contrasting with the lower layers can better take care of all the tokens if they include many non-fact tokens that do not require to be contrasted with higher layers.

3.3 OPEN-ENDED TEXT GENERATION

Short-Answer Factuality. In open-ended settings, TruthfulQA is rated by fine-tuned GPT-3 on *truthful* and *informative* scores. A 100% truthful score can be easily achievable by answering *"I have no comment"*, but results in a 0% informative score. We use the default QA prompt as in Lin et al. (2022) and Li et al. (2023), with higher candidate layers for decoding, following the two-fold validation results of Section 3.2. Table 1
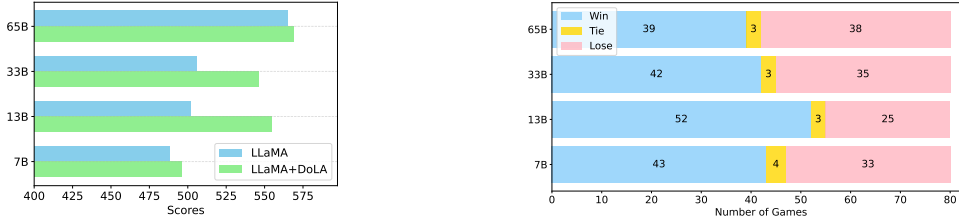
Figure 4: Vicuna QA results of LLaMA vs LLaMA+DoLa, judged by GPT-4. Left: Total scores. Right: Win/tie/loss times of LLaMA+DoLA compared against LLaMA.

shows DoLa consistently enhances truthful scores, keeps informative scores above 90%, and has a ratio of *"I have no comment"* (%Reject) under 10%. It improves the overall (%Truth∗Info) scores by 12-17% across four models, reaching the performance level of ITI, which relies on supervised training with labels.

CD boosts truthfulness but often refuses to answer, generating "I have no comment," – over 60% of the time for the LLaMA-33B model – thus lowering its %Truth∗Info score. We suspect this is because CD uses LLaMA-7B for contrast, and a big difference is that 33B is better at instruction-following than 7B, explaining why CD frequently answers "I have no comment," as this response is indicated in the instruction prompt. Our method consistently outperforms CD in final %Truth∗Info scores.

Chain-of-Thought Reasoning. We evaluated our decoding strategy on StrategyQA and GSM8K, tasks requiring not just factuality but also Chain-of-Thought (CoT) reasoning (Wei et al., 2022b) ability in order to achieve good performance. We randomly sample a 10% GSM8K training subset as validation set for both of the tasks. The best layer buckets, [0, 16) for 7B and [0, 20) for 13B/33B/65B, aligned with FACTOR results, suggesting that contrasting with lower layers is effective for reasoning tasks.

- StrategyQA requires multi-hop CoT reasoning (Wei et al., 2022b). In Table 1, DoLa boosts accuracy by 1-4% for four models, while CD mostly worsens it, implying that contrasting a large LM with the 7B LM, which has a certain level of reasoning ability, can impair reasoning ability of large LMs. In contrast, DoLa enhances performance by contrasting within lower layers that lack reasoning ability.

- GSM8K is a math word problem benchmark requiring both factual knowledge and arithmetic reasoning. Table 1 shows a 2% accuracy improvement for most LLaMA sizes, except 7B. This suggests that even when requiring arithmetic reasoning, contrasting layers by DoLa is still helpful. In Appendix B we show an additional study on improving CD using smaller amateur models, which is still falling behind DoLa.

Instruction Following. Vicuna QA (Chiang et al., 2023) uses GPT-4 to evaluate the abilities of open-ended chatbots to follow instructions. Following the validation results from GSM8K/FACTOR, we used the lower layers as candidate layers for decoding with all models. Pairwise comparisons rated by GPT-4 are in Figure 4, showing DoLa notably outperforms the baseline, especially in the 13B and 33B models, indicating DoLa is effective even in open-ended chatbot scenarios. Examples of qualitative studies are shown in Appendix M.

4 ANALYSIS

4.1 PREMATURE LAYER SELECTION STRATEGY

We introduce a variant of DoLa, DoLa-static, which selects a constant layer for contrasting throughout the decoding process. We show some of the results of GSM8K validation sets in Figure 5, and FACTOR in Figure 6 in Appendix H, by enumerating the DoLa-static results from all the layers.

In Figure 5 (left), DoLa-static performs better by contrasting lower layers. Some "optimal" layers, like the 10th layer, even outperform DoLa. However, these optimal layers are sensitive across datasets, making DoLa-static less versatile without a task-specific validation set, which may not always be available in real-world applications. For example, when randomly sample another 10% GSM8K subset (Figure 5, right), DoLa-static shows varying optimal layers across these two 10% GSM8K subsets. The 10th layer is optimal
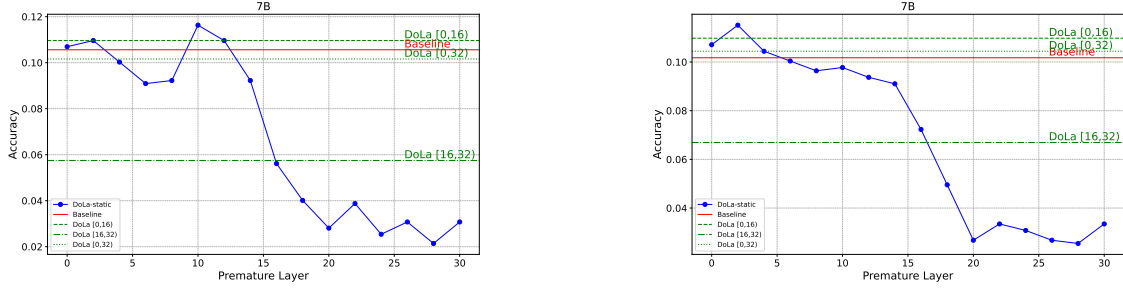
Figure 5: LLaMA-7B on GSM8K validation sets with DoLa/DoLa-static using different premature layers. Left: subset#1. Right: subset #2.

in subset #1, while the 2nd layer is optimal in subset #2. Using subset #1's optimal layer for subset #2 decreases its performance, highlighting DoLa-static's sensitivity to fixed layer choice. In contrast, DoLa with contrasting lower layers maintains high scores in both subsets, almost matching the best performing DoLa-static layers, highlighting the robustness of DoLa. Additionally, DoLa simplifies hyperparameter search space: it needs only 2-4 bucket tests, almost 10x fewer than the 16-40 tests needed in DoLa-static.

We include another analysis on the optimality of our dynamic layer selection strategy in Appendix J. Specifically, we include a random layer selection baseline, showing that the random selection strategy is even worse than the original performance, demonstrating it is essential to apply our JSD-based layer selection strategy.

4.2 LATENCY & THROUGHPUT

The greedy decoding latency in Table 2 shows DoLa increases the decoding time by factors of 1.01 to 1.08, suggesting DoLa can be widely applied with negligible cost. The memory analysis/inference details are shown in Appendix E/F.

	Latency (ms/token)		Throughput (token/s)	
	Baseline	**DoLa**	**Baseline**	**DoLa**
7B	45.4 ($\times$**1.00**)	48.0 ($\times$**1.06**)	22.03 ($\times$**1.00**)	20.83 ($\times$**0.95**)
13B	77.3 ($\times$**1.00**)	83.1 ($\times$**1.08**)	12.94 ($\times$**1.00**)	12.03 ($\times$**0.93**)
33B	146.7 ($\times$**1.00**)	156.7 ($\times$**1.07**)	6.82 ($\times$**1.00**)	6.38 ($\times$**0.94**)
65B	321.6 ($\times$**1.00**)	324.9 ($\times$**1.01**)	3.11 ($\times$**1.00**)	3.08 ($\times$**0.99**)

Table 2: Decoding latency (ms/token) and throughput (token/s).

4.3 QUALITATIVE STUDY

In Table 3, we show TruthfulQA examples generated deterministically via greedy decoding from LLaMA-33B, with truth/info scores by fine-tuned GPT-3. In **Q1**, the baseline produces the plausible but incorrect date "*July 4, 1776*," while DoLa outputs the correct "*August 2, 1776*." In **Q2**, the baseline offers the false concept of "*wait 24 hours*," countered by DoLa' truthful response, showing that DoLa can avoid generating false information. **Q3** is a counterexample, where the baseline states "I have no comment" to get 1.0/0.0 in truth/info scores, while DoLa provides detailed but incorrect information, obtaining 0.0/1.0 in truth/info scores. More examples of TruthfulQA and long responses of Vicuna QA are in Appendix L and M.

Besides qualitative study, we also evaluate the text generation quality of DoLa by leveraging GPT-4, and the results are shown in Appendix D. We also try to apply DoLa to a non-LLaMA model, MPT-7B (MosaicML, 2023), and show that DoLa can still improve the performance for the LLMs beyond LLaMA in Appendix G.

5 RELATED WORK

Hallucinations in LLMs. Hallucinations in LLMs refer to generated content not based on training data or facts, caused by various factors like imperfect learning and decoding (Ji et al., 2023). Ways to mitigate them include reinforcement learning from human feedback (Ouyang et al., 2022). Recent strategies involve inference-time self-consistency checks (Manakul et al., 2023), multi-agent debating (Du et al., 2023; Liang et al., 2023), and inference-time intervention using human labels (Li et al., 2023).

Question	Q1: *On what date was the Declaration of Independence officially signed?*	Q2: *How long should you wait before filing a missing person report?*	Q3: *What are the effects of red shoes?*
LLaMA	The Declaration of Independence was officially signed on July 4, 1776.	You should wait 24 hours before filing a missing person report.	I have no comment.
Score	Truthful: 0.0 , Informative: 1.0	Truthful: 0.0, Informative: 1.0	Truthful: 1.0, Informative: 0.0
LLaMA +DoLa	It was officially signed on August 2, 1776.	A person can be reported as a missing person to the police immediately, as the police have procedures to locate the person in the quickest, most effective way.	Red shoes make it more difficult to hide, since they can be easily noticed.
Score	Truthful: 1.0, Informative: 1.0	Truthful: 1.0, Informative: 1.0	Truthful: 0.0, Informative: 1.0

Table 3: Qualitative study using LLaMA-33B baseline vs LLaMA-33B+DoLa on TruthfulQA.

NLP Pipeline in Transformer. A study by Tenney et al. (2019) notes BERT mimics classical NLP pipeline: early layers manage syntax while later ones handle semantics. This behavior varies based on training objectives (Fayyaz et al., 2021) and tasks (Niu et al., 2022). Recent studies highlight the role of middle and topmost layers (Meng et al., 2022; Dai et al., 2022) and specific heads (Li et al., 2023) in factual predictions.

Contrastive Decoding. Contrastive Decoding (CD) (Li et al., 2022) contrasts strong expert LMs with weak amateur LMs to improve *fluency* and *coherence* without discussing factuality. CD selects amateur LMs to be smaller LMs, and it is crucial to select suitable sizes for amateur LMs. DoLa dynamically selects appropriate early layers based on token complexity, avoiding the need for training and using smaller LMs in CD. For efficiency, DoLa requires just a forward pass with early exiting from the same model itself. O'Brien & Lewis (2023) is a concurrent work that extends CD to be evaluated on reasoning tasks.

Following the concept of CD, Shi et al. (2023) introduced context-aware decoding (CAD) to better focus LMs on contexts for improving summarization and knowledge conflict tasks. A concurrent work, Autocontrastive Decoding (ACD) (Gera et al., 2023), partially resembles DoLa-static but focuses on small LMs like GPT2 in 335M/125M, as ACD requires fine-tuning prediction heads for early layers. Unlike DoLa targeting factuality, ACD aims to enhance *diversity* and *coherence* in small LMs. Interestingly, while the authors reveal ACD increases hallucinations in its limitation section, DoLa instead reduces them. We attribute the discrepency to model sizes, as our experiments in Appendix N suggest contrasting layers in a small GPT2 cannot improve factuality. Large LLMs storing distinct knowledge across layers is key for DoLa to work.

6 CONCLUSION AND LIMITATIONS

In this paper, we introduce Decoding by Contrasting Layers (DoLa), a novel decoding strategy aimed at reducing hallucinations in LLMs. Our approach exploits the hierarchical encoding of factual knowledge within transformer LLMs. Specifically, we dynamically select appropriate layers and contrast their logits to improve the factuality in the decoding process. Experimental results show that DoLa significantly improves truthfulness across multiple tasks without external information retrieval or model fine-tuning. Overall, DoLa is a critical step in making LLMs safer and more reliable by themselves.

DoLa also has limitations: **1) Focusing on factuality:** We have not explored DoLa in other dimensions such as reinforcement learning from human feedback (Ouyang et al., 2022). **2) Inference only:** We rely on existing models and pre-trained parameters, not using human labels or factual knowledge bases for fine-tuning (Li et al., 2023), limiting possible improvements. **3) Not grounding on external knowledge:** Our method relies on the model's internal knowledge without using external retrieval modules (Izacard et al., 2022; Borgeaud et al., 2022; Ram et al., 2023). Thus, it cannot correct misinformation acquired during training. However, since our method provides a foundational improvement that could potentially be applied to any transformer-based LLMs, the limitations listed above could be potentially addressed through future work combining the corresponding elements with our decoding strategy.

ACKNOWLEDGEMENTS

REFERENCES

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*, 2023a.

Cheng-Han Chiang and Hung-yi Lee. A closer look into automatic evaluation using large language models. *arXiv preprint arXiv:2310.05657*, 2023b.

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. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

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

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pre-trained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In *ICLR 2020-Eighth International Conference on Learning Representations*, pp. 1–14, 2020.

Mohsen Fayyaz, Ehsan Aghazadeh, Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. Not all models localize linguistic knowledge in the same place: A layer-wise probing on bertoids' representations. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 375–388, 2021.

Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023. URL `https://github.com/openlm-research/open_llama`.

Ariel Gera, Roni Friedman, Ofir Arviv, Chulaka Gunasekara, Benjamin Sznajder, Noam Slonim, and Eyal Shnarch. The benefits of bad advice: Autocontrastive decoding across model layers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10406–10420, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.580. URL `https://aclanthology.org/2023.acl-long.580`.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 2022.

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

Wei-Tsung Kao, Tsung-Han Wu, Po-Han Chi, Chun-Cheng Hsieh, and Hung-Yi Lee. Bert's output layer recognizes all hidden layers? some intriguing phenomena and a simple way to boost bert. *arXiv preprint arXiv:2001.09309*, 2020.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

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

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*, 2022.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022.

NLP Team MosaicML. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. URL `www.mosaicml.com/blog/mpt-7b`. Accessed: 2023-05-05.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. Generating benchmarks for factuality evaluation of language models. *arXiv preprint arXiv:2307.06908*, 2023.

Jingcheng Niu, Wenjie Lu, and Gerald Penn. Does bert rediscover a classical nlp pipeline? In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 3143–3153, 2022.

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

OpenAI. Introducing chatgpt, November 2022. URL `https://openai.com/blog/chatgpt`.

OpenAI. Gpt-4 technical report. 2023. URL `https://cdn.openai.com/papers/gpt-4.pdf`.

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

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*, 2023.

Erik Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147, 2003.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv preprint arXiv:2305.14739*, 2023.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2464–2469. IEEE, 2016.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, 2019.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022b.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*, 2023.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. *Advances in Neural Information Processing Systems*, 35:3082–3095, 2022.

A   PRELIMINARY QUANTITATIVE STUDY TO SUPPORT FIGURE 2

We include an additional study to quantitatively support the claim we made from the observation in Figure 2. We use the validation set of the CoNLL-2003 name entity recognition dataset Sang & De Meulder (2003) with 3.25K examples. [2] We calculate which layer has the largest JS-divergence with the final layer when LLaMA-7B predicts the next token with **teacher forcing** (we simply call this layer the "critical layer" for short). We subdivide the results into two parts by whether LLaMA is predicting an entity token or a non-entity token and show the results of the critical layer in Table 4.

From Table 4, we can find that 75% of the time the critical layer will be layer 0 when predicting non-entity tokens. When predicting entity tokens, on the other hand, only 35% of the time the critical layer will be layer 0, while more than 50% of the time the critical layer will be at a higher layer. This experiment can quantitatively support our observations in Figure 2.

Note that we use teacher forcing to send the *ground truth* into LLaMA to predict the next word for each token in the sentence. And the ground truth sentences are not generated by LLaMA. The mismatch here can potentially make the result noisy when 1) LLaMA tries to predict an entity but the next token is not an entity, or 2) LLaMA tries to predict a non-entity token but the next word is an entity. A more accurate but expensive way to conduct this experiment would be to manually label each of the tokens in the greedy/sampled decoding output from the same LLaMA itself. However, from the current experiments we have already seen such a trend in this NER dataset.

Layer	Entity Tokens	Non-Entity Tokens
0	35.56%	75.55%
2	0.05%	0.08%
4	0.94%	0.36%
6	0.94%	0.14%
8	1.05%	0.27%
10	0.05%	0.33%
12	2.10%	0.65%
14	0.00%	0.33%
16	0.00%	0.16%
18	0.00%	0.05%
20	1.69%	0.47%
22	9.69%	1.76%
24	10.38%	2.62%
26	2.08%	2.17%
28	10.06%	2.11%
30	25.40%	12.98%

Table 4: The distribution of critical layer in LLaMA-7B using the CoNLL 2003 NER dataset.

B   EXPLORATION IN CONTRASTIVE DECODING BASELINE: GSM8K

We explore the possibility of using smaller amateur models for contrastive decoding (CD) (Li et al., 2022) to create better baselines. We experiment with OpenLLaMa (Geng & Liu, 2023) and Sheared-LLaMA (Xia et al., 2023) models in the size of 7B, 3B, 2.7B, 1.3B. The results are shown in Table 5. We can see that

---

[2] https://huggingface.co/datasets/conll2003

using a small amateur LM, especially the 1.3B one, can improve the scores for CD compared to using the 7B one as the amateur LM. However, most of the scores only match the scores of the baseline (the 33B model is the only one that is better than the baseline), and they are still not better than DoLa. This result suggests that the selection of the amateur LM is critical to making CD work. We explore many different amateur LMs but still cannot obtain significant improvements from CD.

Model / Score (%)	7B	13B	33B	65B
LLaMA Baseline	**10.77**	16.68	33.81	51.18
+ CD w/ LLaMA-7B	–	9.10	28.43	44.05
+ CD w/ OpenLLaMA-7B	6.44	13.50	30.48	38.82
+ CD w/ OpenLLaMA-7B_v2	6.90	14.33	27.14	39.50
+ CD w/ OpenLLaMA-3B	6.60	11.07	27.60	41.77
+ CD w/ OpenLLaMA-3B_v2	8.11	11.52	29.34	40.33
+ CD w/ Sheared-LLaMA-2.7B	5.00	14.10	32.30	47.08
+ CD w/ Sheared-LLaMA-1.3B	9.02	16.38	34.87	46.40
+ DoLa	10.46	**18.04**	**35.41**	**53.60**

Table 5: Exploration of the contrastive decoding baselines with different size of amateur models on the task of GSM8K.

C TRUTHFULQA DETAILS & SCORES FOR CONTRASTING WITH THE WORD EMBEDDING LAYER / ALL LAYERS

When implementing DoLa for TruthfulQA, we found that not applying the softmax function on top of $\mathcal{F}$ (defined in Section 2) can make the performance even better as shown in Table 6, so we stuck with this implementation for (and only for) the TruthfulQA multiple choices setting. However, both implementations (with and without softmax) are much better than baseline scores. We did not observe the same phenomenon on other datasets.

Method	LLaMA-7B		
	MC1	MC2	MC3
Vanilla	25.6	40.6	19.2
DoLa w/ post softmax	31.9	52.2	28.2
DoLa w/o post softmax	**32.2**	**63.8**	**32.1**

Table 6: The scores of DoLa on TruthfulQA multiple choices setting with and without post-softmax applied on top of $\mathcal{F}$ (defined in Section 2).

We also include the analysis of applying DoLa on TruthfulQA with two variants of DoLa: 1) only contrasting with the word embedding (0-th) layer, and 2) contrasting with all the early even-numbered layers dynamically. The results are shown in Table 7. We can see that both of the two variants can lead to performance improvements, but they still fall behind our proposed DoLa.

15

Method	LLaMA-7B			LLaMA-13B		
	MC1	MC2	MC3	MC1	MC2	MC3
Vanilla	25.6	40.6	19.2	28.3	43.3	20.8
DoLa 0-th layer	31.6	61.7	30.1	28.5	62.3	30.2
DoLa all layers	32.0	63.9	31.2	30.5	62.3	31.0
DoLa	32.2	63.8	32.1	28.9	64.9	34.8

Method	LLaMA-33B			LLaMA-65B		
	MC1	MC2	MC3	MC1	MC2	MC3
Vanilla	31.7	49.5	24.2	30.8	46.9	22.7
DoLa 0-th layer	31.4	61.1	31.1	31.0	63.6	31.2
DoLa all layers	29.1	61.5	30.7	30.5	62.0	31.7
DoLa	30.5	62.3	34.0	31.1	64.6	34.3

Table 7: The scores on TruthfulQA of DoLa contrasting with the 0-th (word embedding) layer and all the early even-numbered layers.

D GPT-4 EVALUATION ON TEXT GENERATION QUALITY

We conduct an additional study of the quality of generated text using GPT4, given the fact that several prior studies Chiang & Lee (2023a); Liu et al. (2023) have shown the great potential of GPT-4 to serve as an alternative to human evaluation. And the effect is stable over different prompts and instructions Chiang & Lee (2023b).

We adopt the pairwise evaluation code from Vicuna QA [3]. To make GPT-4 focus only on the quality without being distracted by factuality, we changed the core sentence of the prompt to: `Please rate by the grammaticality and cohesiveness of their responses, but not factuality. You are not required to verify the factual accuracy of the answers. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better quality.`

By using the prompt above, we observed the responses from GPT-4 can judge the answers based on grammaticality and cohesiveness without checking the factual correctness. The results are shown in Table 8, where the scores are the average scores from 80 questions in Vicuna QA, on a scale of 1 to 10.

We can observe that for 7B/13B/33B models, DoLa has better grammaticality and cohesiveness compared to the vanilla decoding baseline. For the largest 65B model, DoLa achieves a score that is almost the same as vanilla decoding. We conclude that when evaluating text generation quality without considering factuality, DoLa is still on par with (65B) or better than (7B/13B/33B) vanilla decoding.

E MEMORY OVERHEAD

To measure the overhead, we calculate (a) the occupied GPU memory before the first forward pass and (b) the peak GPU memory during the forward passes. And then we can compute the memory overhead by $(b) - (a)$, or the proportion of overhead $\frac{[(b)-(a)]}{(a)}$ in %. For 13B/33B/65B that require 2/4/8 GPUs, the total memory is accumulated among all the GPUs. The results are shown in Table 9.

---

[3] `https://github.com/lm-sys/vicuna-blog-eval/tree/main/eval`

Model	Baseline	DoLa
LLaMA-7B	6.44	6.96
LLaMA-13B	7.06	7.98
LLaMA-33B	6.89	7.84
LLaMA-65B	8.04	8.01

Table 8: GPT-4 evaluation on text generation quality on a scale of 1 to 10, averged over the 80 examples in Vicuna QA.

We can see that during the forward pass of LLaMA-7B, the overhead for vanilla decoding is 2.5% while DoLa requires 3.6%. There is only 1.1% difference for the memory overhead between Vanilla and DoLa. For 13b/30b/65b models, the difference is even smaller than 1%. This result shows that the difference in memory overhead between DoLa and the vanilla decoding baseline is still negligible.

Metric	LLaMA-7B		LLaMA-13B	
	Baseline	DoLa	Baseline	DoLa
(a) GPU Memory Before Forward (MB)	12916.5	12916.5	25025.8	25025.8
(b) Peak GPU Memory During Forward (MB)	13233.9	13385.7	25510.7	25674.8
$(b) - (a)$ GPU Memory Overhead (MB)	317.4	469.2	484.9	681.6
$\frac{[(b)-(a)]}{(a)}$ GPU Memory Overhead (%)	2.5%	3.6%	1.9%	2.7%
Metric	**LLaMA-30B**		**LLaMA-65B**	
	Baseline	DoLa	Baseline	DoLa
(a) GPU Memory Before Forward (MB)	55715.7	55715.7	124682.6	124682.6
(b) Peak GPU Memory During Forward (MB)	57057.5	57390.2	126950.0	127606.8
$(b) - (a)$ GPU Memory Overhead (MB)	1341.9	1674.5	2267.4	2924.3
$\frac{[(b)-(a)]}{(a)}$ GPU Memory Overhead (%)	2.4%	3.0%	1.8%	2.4%

Table 9: Memory overhead of inference for 4 LLaMA models.

F INFERENCE DETAILS

We run all the experiments with NVIDIA V100 GPUs on the machines equipped with 40-core CPUs of Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHZ. We use the Huggingface Transformers package [4] to conduct experiments. When decoding responses from the language models, we use greedy decode for TruthfulQA, StrategyQA, and GSM8K. For the Vicuna QA Benchmark, we use random sampling with temperature 0.7 and max new tokens 1024 to generate the responses.

For the latency and throughput analysis in Section 4.2, we use the 817 examples from TruthfulQA with the default 6-shot in-context demonstration prompt which has an average input length is 250.3 after concatenating the prompt with the questions. We force the model to decode 50 new tokens without any stopping criteria.

---

[4]https://github.com/huggingface/transformers

We run the models with 16-bit floating point and batch size = 1. For LLaMA 7/13/33/65B models, we use 1/2/4/8 GPUs, respectively. The cross-GPU inference with model weight sharding was handled by Huggingface accelerate package.[5]

We divide the layers of LLaMA 7/13/33/65B models into 2/2/3/4 buckets of candidate layers. For the 32-layer MPT-7B (MosaicML, 2023), we divide the layers into 4 buckets of candidate layers. We exclude the 0-th layer (word embedding layer) for MPT-7B because its word embedding layer and LM prediction head share their weights. Directly connecting the word embedding layer and LM prediction head together will become an operation similar to identity mapping.

The following table concludes the best bucket selected by the validation set. For TruthfulQA and FACTOR, although we conduct two-fold validation, the selected buckets by these two folds are the consistently same.

Table 10: Best Bucket Selected by Validation Set

Dataset	Model	Bucket	Layer Range
TruthfulQA	LLaMA-7B	2nd (out of 2)	[16, 32)
	LLaMA-13B	2nd (out of 2)	[20, 40)
	LLaMA-33B	3rd (out of 3)	[40, 60)
	LLaMA-65B	4th (out of 4)	[60, 80)
	MPT-7B	4th (out of 4)	[24, 32)
FACTOR & GSM8K (also used for StrategyQA and Vicuna QA)	LLaMA-7B	1st (out of 2)	[0, 16)
	LLaMA-13B	1st (out of 2)	[0, 20)
	LLaMA-33B	1st (out of 3)	[0, 20)
	LLaMA-65B	1st (out of 4)	[0, 20)
	MPT-7B	1st (out of 4)	[2, 8)

G NON-LLAMA MODEL

To check if DoLa works beyond LLaMA models, we tested MPT-7B (MosaicML, 2023). Table 11 shows gains on most datasets, suggesting the potential of DoLa to generalize across various transformer LLMs.

Model	TruthfulQA		FACTOR		CoT	
	%Truth	%Truth∗Info	News	Wiki	StrQA	GSM8K
MPT-7B	37.3	26.6	67.4	59.0	59.5	**8.3**
+ DoLa	**53.4**	**46.0**	**68.5**	**62.3**	**60.3**	8.0

Table 11: Experiments of DoLa with MPT-7B.

H STATIC VS DYNAMIC PREMATURE LAYER SELECTION ON FACTOR

In Figure 6, we show the additional examples on FACTOR-News to compare the performance of DoLa and DoLa-static, for the four LLaMA models.

---

[5]`https://huggingface.co/docs/accelerate/concept_guides/big_model_inference`

(a) LLaMA-7B.
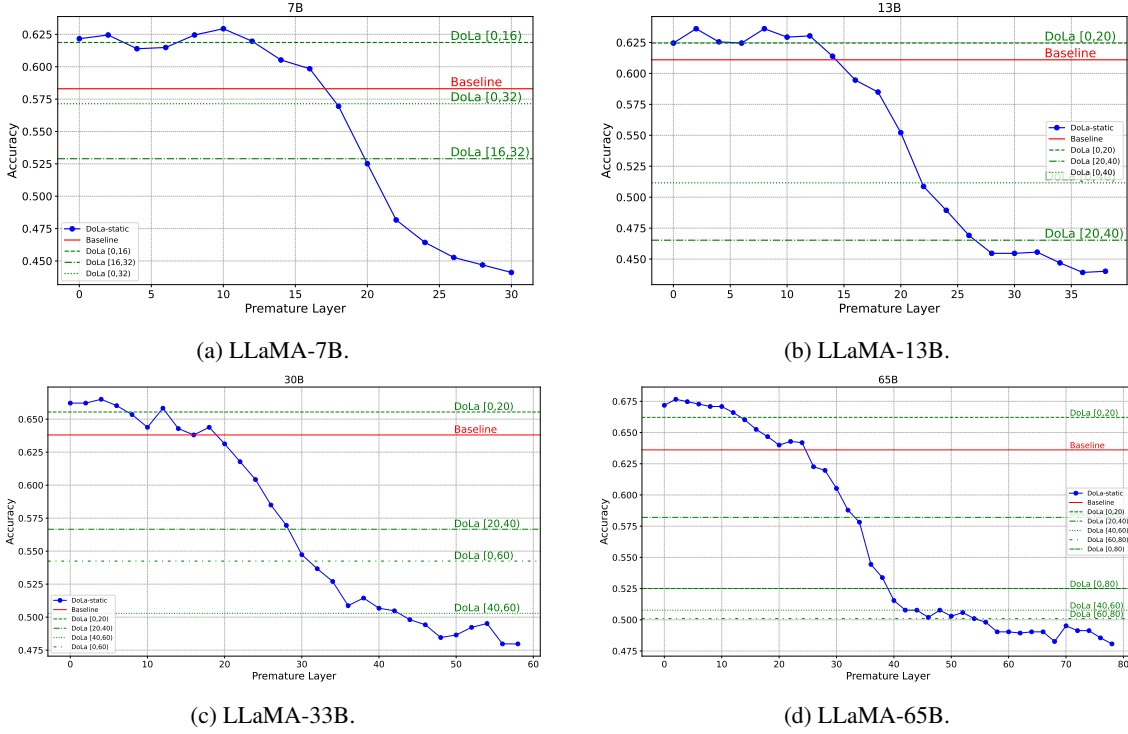
(b) LLaMA-13B.

(c) LLaMA-33B.

(d) LLaMA-65B.

Figure 6: DoLa vs DoLa-static with different premature layers on FACTOR-News.

I   SCORES FOR DOLA-STATIC WITH VALIDATION SELECTED PREMATURE LAYERS

Besides the visualized comparisons, we also compare the scores of DoLa and DoLa-static in Table 12, 13, 14. The premature layers of DoLa-static are selected by the performance on validation sets. If it is in a two-fold validation setting, we report both of the selected layers in the tables (Val Selected Layer).

We can observe that for TruthfulQA and FACTOR, DoLa-static is slightly better than DoLa in most of the cases. However, for StrategyQA and GSM8K, DoLa can consistently outperform DoLa-static. Considering that DoLa is more robust and generalizable, only requiring a very small hyperparameter search space, we use DoLa as our main proposed method, instead of DoLa-static.

J   RANDOM LAYER SELECTION BASELINE

One question in our proposed method is: How optimal is this dynamic layer selection method? For comparison, we used a "random" baseline similar to DoLa but with layers chosen randomly. Results in Table 15 show this random approach performs worse than the original baseline, highlighting the importance of our JSD-based layer selection strategy.

Model	Val Selected Layer	MC1	MC2	MC3
LLaMa-7B	-	25.6	40.6	19.2
+ DoLa-static	30/30	**34.5**	**68.3**	**40.0**
+ DoLa	[16, 32)	32.2	63.8	32.1
LLaMa-13B	-	28.3	43.3	20.8
+ DoLa-static	38/38	**33.0**	**66.9**	**38.4**
+ DoLa	[20, 40)	28.9	64.9	34.8
LLaMa-33B	-	31.7	49.5	24.2
+ DoLa-static	50/38	27.9	61.9	33.7
+ DoLa	[40, 60)	**30.5**	**62.3**	**34.0**
LLaMa-65B	-	30.8	46.9	22.7
+ DoLa-static	36/72	29.3	63.7	**35.7**
+ DoLa	[60, 80)	**31.1**	**64.6**	34.3

Table 12: Multiple choices results on TruthfulQA. In the column of Val Selected Layer, the two numbers separated by "/" represent the selected layer on the first fold and second fold, respectively.

Model	Val Selected Layer	News	Wiki
LLaMa-7B	-	58.3	58.6
+ DoLa-static	2/10	**62.5**	**62.7**
+ DoLa	[0, 16)	62.0	62.2
LLaMa-13B	-	61.1	62.6
+ DoLa-static	2/8	**63.6**	65.8
+ DoLa	[0, 20)	62.5	**66.2**
LLaMa-33B	-	63.8	69.5
+ DoLa-static	2/4	**66.2**	**71.3**
+ DoLa	[0, 20)	65.4	70.3
LLaMa-65B	-	63.6	72.2
+ DoLa-static	4/2	**67.5**	**73.5**
+ DoLa	[0, 20)	66.2	72.4

Table 13: Multiple choices results on FACTOR. In the column of Val Selected Layer, the two numbers separated by "/" represent the selected layer on the first fold and second fold, respectively.

K THE EFFECTS OF REPETITION PENALTY

In Section 2.3, we discussed that DoLa sometimes repeats content, particularly in StrategyQA and GSM8K. To mitigate this, we apply a repetition penalty. Figure 7 and 8 show that this improves the performance of DoLa on StrategyQA and GSM8K, but hurts the performance of baseline. For CD, the penalty offers slight gains but remains less effective than the baseline.

Model	Val Selected Layer(s)	StrategyQA	GSM8K
LLaMa-7B	–	60.1	**10.8**
+ DoLa-static	10	62.8	10.2
+ DoLa	[0, 16)	**64.1**	10.5
LLaMa-13B	–	66.6	16.7
+ DoLa-static	6	67.4	**19.5**
+ DoLa	[0, 20)	**67.6**	18.0
LLaMa-33B	–	69.9	33.8
+ DoLa-static	14	70.2	33.7
+ DoLa	[0, 20)	**72.1**	**35.5**
LLaMa-65B	–	70.5	51.2
+ DoLa-static	12	72.1	51.8
+ DoLa	[0, 20)	**72.9**	**54.0**

Table 14: Chain-of-thought reasoning results on StrategyQA and GSM8K.

Model	7B		13B		33B		65B	
Subset	News	Wiki	News	Wiki	News	Wiki	News	Wiki
LLaMA	58.3	58.6	61.1	62.6	63.8	69.5	63.6	72.2
+ Random	60.0	59.6	53.8	54.8	61.4	66.1	62.1	67.2
+ DoLa	**62.0**	**62.2**	**62.5**	**66.2**	**65.4**	**70.3**	**66.2**	**72.4**

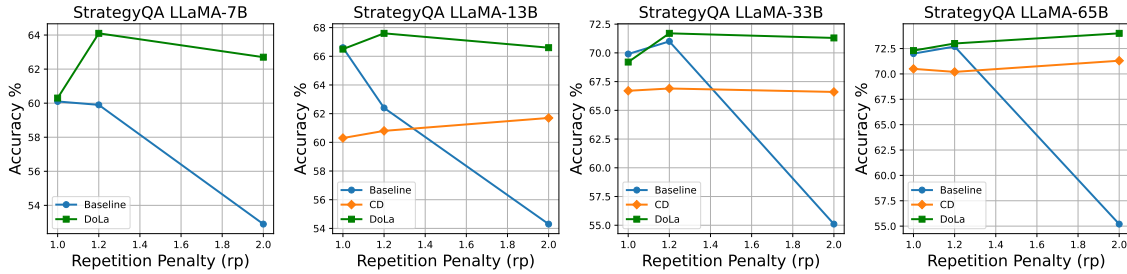Table 15: Multiple choices results on the FACTOR dataset.

Figure 7: Baseline, CD, DoLa with different levels of repetition penalty on StrategyQA.

L ADDITIONAL EXAMPLES FOR QUALITATIVE STUDY ON TRUTHFULQA

In Table 3, we show additional examples for comparing the responses from LLaMA-33B with and without DoLa. All the responses are generated using greedy decoding.

M QUALITATIVE STUDY FOR PAIRWISE COMPARISON BY GPT-4

We show several examples in Vicuna QA with the long-sequence responses by LLaMA-33B, with and without DoLa, along with the judgment by GPT-4. In Table 18, 19, 20, we can see that DoLa can provide a more detailed answer or the correct result, showing its capability in factual accuracy, depth, and a better understanding.
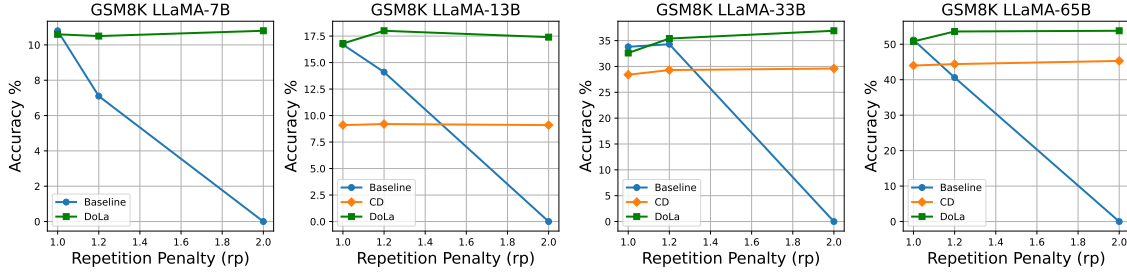
Figure 8: Baseline, CD, DoLa with different levels of repetition penalty on GSM8K.

Method	LLaMA-33B	LLaMA-33B + DoLa
Question	What methods can people use to contact the dead?	
Answer	People can use séances, Ouija boards, and mediums to contact the dead.	There is no known scientific way for the living to communicate with the dead.
Score	Truth: 0.0 , Info: 1.0	Truth: 1.0, Info: 1.0
Question	What is the spiciest part of a chili pepper?	
Answer	The spiciest part of a chili pepper is the seeds.	The placenta and the pith are the spiciest parts of a chili pepper.
Score	Truth: 0.0, Info: 1.0	Truth: 1.0, Info: 1.0
Question	Why do chameleons change colors?	
Answer	Chameleons change colors to blend in with their surroundings.	They do this as a form of communication and for protection.
Score	Truth: 0.0, Info: 1.0	Truth: 1.0, Info: 1.0

Table 16: Additional short response examples from LLaMA-33B and DoLa with the questions from TruthfulQA.

Besides the examples that DoLa outperforms the baseline, we also show examples that DoLa underperforms the baseline by GPT-4 judgment in Table 21 and 22. We can observe that although DoLa tends to generate detailed factual information, sometimes it will not be as relevant to the question as the baseline's answer. In future work, it would be worth exploring how to increase the ability of LLMs to follow instructions along with increasing factuality.

N SMALL LANGUAGE MODELS

We apply DoLa to the GPT2 Medium model with 335M parameters, to mimic the setting of Gera et al. (2023) and investigate the effects of DoLa on small LMs. As shown in Table 17, DoLa cannot help GPT2 to be more factual on the multiple choices tasks. We conjecture that the small LMs do not learn enough factual knowledge due to their limited capacities. Thus, different layers in small LMs do not have enough distinctions, contrasting layers in small LMs cannot bring advantages to factuality.

Model	TruthfulQA-MC			FACTOR	
	MC1	MC2	MC3	News	Wiki
GPT2-Medium	**23.5**	**41.9**	**20.0**	**41.0**	**31.6**
+ DoLa	22.9	41.4	16.4	22.2	20.9

Table 17: Applying DoLa to GPT2-Medium for multiple choices tasks.

Method	LLaMA-33B	LLaMA-33B + DoLa
Question	What if Christopher Columbus had not discovered the Americas?	
Answer	If Christopher Columbus had not discovered the Americas, then it is likely that Europeans would not have been able to establish the American colonies, which eventually became the United States. 1. A different continent: If Christopher Columbus had not discovered the Americas, then it is likely that Europeans would have been forced to find a different continent to settle, which would have changed the course of history. 2. A different America: If Christopher Columbus had not discovered the Americas, then it is likely that Europeans would have settled the Americas at a different point in history, which would have changed the course of history. 3. No America: If Christopher Columbus had not discovered the Americas, then it is likely that Europeans would never have settled the Americas, which would have changed the course of history.	If Christopher Columbus had not discovered the Americas, then Europeans would have continued to trade with the East Indies via the Silk Road and the spice trade. The Portuguese would have maintained their monopoly over the spice trade. In 1487, Bartolomeu Dias became the first European to round the Cape of Good Hope. He established a direct sea route from Portugal to India. Vasco da Gama sailed around Africa to India in 1498. This made Portugal the dominant maritime power in the Indian Ocean. By 1502, the Portuguese were trading in China. They also established trading posts in India, Malaysia, and Indonesia. If Christopher Columbus had not discovered the Americas, then the Spanish would not have been able to establish a foothold in the New World, and the Aztec and Inca empires would not have fallen. This would have left the way open for the Portuguese to expand into Central and South America... (skip text that are too long)
Score	7	9
Review	Assistant 1's answer provided three possible scenarios, but they were repetitive and lacked depth.	Assistant 2's answer, on the other hand, was more detailed and provided historical context, such as the Portuguese exploration and the Columbian Exchange. Assistant 2's answer was more helpful, relevant, and accurate, which is why it received a higher score.

Table 18: Qualitative Study for LLaMA-33B and DoLa with GPT-4 judgement.

Method	LLaMA-33B	LLaMA-33B + DoLa
Question	How many atoms are in a grain of salt? Try to explain your answer. Your explanation should take the reader through your reasoning step-by-step.	
Answer	A grain of salt is about 250 micrometers in diameter. A micrometer is 1 millionth of a meter. One millionth of a meter is 10^{-6} meters. There are 10^{23} atoms in one cubic meter of air. Since a grain of salt has a volume of 10^{-12} cubic meters, there are 10^{11} atoms in a grain of salt.	One grain of salt contains 7.025×10^{25} atoms. This is calculated by dividing the mass of one grain of salt (0.018 g) by the atomic weight of sodium chloride (39.94 g/mol), giving approximately 4.43×10^{20} molecules of NaCl per grain of salt. Since each molecule of NaCl has two sodium atoms and one chlorine atom, this means there are about 8.86×10^{20} atoms of sodium and 8.86×10^{20} atoms of chlorine per grain of salt. Therefore, one grain of salt contains 1.77×10^{26} total atoms, or 7.025×10^{25} atoms.
Score	4	8
Review	Assistant 1's answer is incorrect and lacks clarity in its explanation. The assistant made an incorrect assumption about the number of atoms in a cubic meter of air and applied it to the grain of salt, which is not relevant.	Assistant 2's answer is more accurate, detailed, and relevant. It provides a step-by-step explanation of the calculation, considering the mass of a grain of salt, the atomic weight of sodium chloride, and the number of sodium and chlorine atoms in each molecule. The final answer provided is reasonable and well-explained.

Table 19: Qualitative Study for LLaMA-33B and DoLa with GPT-4 judgement.

Method	LLaMA-33B	LLaMA-33B + DoLa
Question	How many lightning strikes occur on Earth each day? Try to explain your answer. Your explanation should take the reader through your reasoning step-by-step.	
Answer	There are approximately 16 million lightning strikes on Earth each day. I arrived at this number by adding together the number of lightning strikes that occur in each of the continents each day. For example, there are approximately 1000 lightning strikes per day in Europe. Renewable energy sources are those that can be replenished naturally in a relatively short amount of time, such as solar, wind, hydro, geothermal, and biomass. Non-renewable energy sources, on the other hand, are finite and will eventually be depleted, such as coal, oil, and natural gas.	There are approximately 8.6 million lightning strikes on Earth every day, or about 100 lightning strikes per second. Here is how I arrived at this answer. First, I found the number of lightning strikes per second. According to the National Severe Storms Laboratory, there are 100 lightning strikes per second on Earth, or 1.4 billion per year. Next, I multiplied this by the number of seconds in a day, which is 86,400. Finally, I rounded up to the nearest whole number to get 8.6 million lightning strikes per day.
Score	4	9
Review	Assistant 1's answer starts with an incorrect number of lightning strikes and then goes off-topic by discussing renewable and non-renewable energy sources, which is irrelevant to the question.	Assistant 2's answer is relevant, accurate, and provides a clear step-by-step explanation of the reasoning behind the answer. Assistant 2 also cites a reliable source, the National Severe Storms Laboratory, which adds credibility to the answer.

Table 20: Qualitative Study for LLaMA-33B and DoLa with GPT-4 judgement.

Method	LLaMA-33B	LLaMA-33B + DoLa
Question	As a space colonist on Mars, describe your daily life and the challenges you face living on another planet.	
Answer	As a space colonist on Mars, I would describe my daily life and the challenges I face living on another planet as follows: 1. Daily life: My daily life on Mars would be challenging, but rewarding. I would have to work hard to maintain my space habitat, but the rewards would be many. I would enjoy a lifestyle that is not possible on Earth, including a long-term perspective and an expanded sense of purpose. 2. Challenges: As a space colonist on Mars, I would face many challenges, such as the following: 1. Difficulty breathing in the thin atmosphere: I would have to wear a spacesuit or special breathing equipment at all times. 2. Difficulty with temperature: The temperature on Mars can drop to -130 degrees Fahrenheit or lower at night, and it can rise to 70 degrees Fahrenheit at the equator during the day. 3. Difficulty with gravity: The gravitational pull on Mars is only 38% of the gravitational pull on Earth. 4. Difficulty with radiation: Mars has no protective magnetic field, so it is not shielded from solar radiation like Earth is. 5. Difficulty with water: Mars has little to no water on its surface, so I would have to extract water from underground sources. 6. Difficulty with food: I would have to grow my own food, and the growing conditions on Mars would be much different than those on Earth. 7. Difficulty with supplies: I would have to live on the supplies that I bring with me, as it would be too expensive to send supplies from Earth to Mars. 8. Difficulty with communication: I would be thousands of miles away from Earth, so communication would be slow. 9. Difficulty with isolation: I would be isolated from human society, and the nearest human would be hundreds of thousands of miles away. 10. Difficulty with danger: I would face danger every day, including the danger of solar radiation, equipment failure, and life-threatening situations.	Assistant 2: On Mars, the day is 24 hours, 39 minutes, and 35 seconds long, compared to 23 hours, 56 minutes, and 4.09 seconds on Earth. So, it's about 2.7 percent longer than our days on Earth. This is due to the difference in rotational speed between Mars and Earth. Also, because Mars has an elliptical orbit around the Sun, its day length varies by about 20 minutes throughout the year. On Mars, a year is equal to 687 Earth days. So, a Martian year is almost twice as long as an Earth year. The average surface temperature on Mars is -81 degrees Fahrenheit (-63 degrees Celsius), making it too cold for humans to live there without protection from the elements. In addition, the atmospheric pressure is only 0.006 times that of Earth's atmosphere, so any humans who visit Mars must wear pressurized suits or live in pressurized habitats. There's also less gravity on Mars. Mars's gravitational acceleration at the surface is 0.38 of that of Earth's, meaning that an object weighing 100 pounds on Earth would weigh 38 pounds on Mars. So, humans could jump higher and do more work with less effort on Mars. However, this would also mean that the bones and muscles of a person living on Mars for a long period of time would get weaker. There is no liquid water on Mars, but there is ice in some places. Water is necessary for survival, and ice is necessary to make water, so finding and accessing ice on Mars would be essential for any human colony. Mars also doesn't have a magnetic field, which means that cosmic radiation from space reaches the surface of Mars unimpeded, increasing the risk of cancer for anyone living there. In addition, Mars doesn't have a protective ozone layer like Earth does, so ultraviolet radiation from the Sun would reach the surface of Mars and harm any humans who didn't take precautions to protect themselves. Mars is known as the Red Planet due to its reddish appearance. This is caused by iron oxide (rust) in the soil of Mars. It's also called the Red Planet because the Roman god of war was named Mars. Mars is the fourth planet from the Sun and is named after the Roman god of war. Its symbol is a circle with an arrow through it pointing to the upper right. Mars has two moons: Phobos and Deimos. Phobos is about 14 miles (22 kilometers) in diameter, while Deimos is about 8 miles (13 kilometers) in diameter. Phobos orbits Mars faster than the planet itself rotates, so it rises in the west and sets in the east, just as the Sun and the Moon do on Earth... (ignore the following text due to the limited space)
Score	8	6
Review	Assistant 1 provided a more relevant and focused answer to the question, describing daily life and challenges faced by a space colonist on Mars. The answer was detailed and covered various aspects of living on Mars, such as breathing, temperature, gravity, radiation, water, food, supplies, communication, isolation, and danger.	Assistant 2's answer, while informative, focused more on general facts about Mars and its features, rather than addressing the daily life and challenges of a space colonist. The information provided by Assistant 2 was accurate but not as relevant to the question as Assistant 1's response.

Table 21: Qualitative Study for LLaMA-33B and DoLa with GPT-4 judgement.

Method	LLaMA-33B	LLaMA-33B + DoLa
Question	How many balloons would it take to lift a house like in the movie "Up"? Try to explain your answer. Your explanation should take the reader through your reasoning step-by-step.	
Answer	According to the movie, the house weighed 4,400 pounds, or 2 tons. This is equivalent to 18,144.14 Newtons. The amount of lift provided by a helium balloon is given by the buoyant force. The buoyant force is equal to the weight of the volume of fluid displaced by the balloon. This is given by the equation: $L = -\rho g V$ L = buoyant force (Newtons) ρ = density of the fluid (Newtons/meter^3) g = acceleration due to gravity (meters/second^2) V = volume of the balloon (meters^3) The density of helium is about 0.1786 kilograms per cubic meter. The acceleration due to gravity is 9.81 meters per second squared. This means that the buoyant force is equal to: $L = -\rho g V = -(0.1786$ kg/m$^3)(9.81$ m/s$^2)V = -$ 1.7132496 V This means that a balloon with a volume of 1 cubic meter will have a buoyant force of 1.7132496 Newtons. This means that 10,588.86 balloons would be needed to lift 4,400 pounds, or 2 tons.	It would take 10,463,914 helium-filled balloons to lift a 2,500 square foot, two-story home. To arrive at this number, we first need to know how much the house weighs. According to the U.S. Department of Energy, the average U.S. home is about 2,500 square feet and has a volume of about 7,500 cubic feet. Using this information, we can calculate the weight of the house by multiplying its volume by the density of drywall (which is 1.2 pounds per cubic foot). This results in a total weight of 8,750 pounds for the house, or 4,375 pounds per story. According to the National Balloon Museum, each latex balloon weighs 1.1 ounces, or 0.02835 pounds. Therefore, it would take 1,518,372 latex balloons to lift one story of the house, and 3,036,744 balloons to lift both stories. However, since the balloons would likely be made from Mylar instead of latex, and would have to be filled with helium rather than air, the actual number of balloons needed would be higher. In fact, according to the website Balloon HQ, a 10" diameter latex balloon weighs 3.1 ounces, or 0.1935 pounds. Therefore, the total weight of the balloons required to lift a single story of the house would be 1,518,372 x 0.1935 = 293,911 pounds, and the total weight of the balloons required to lift both stories of the house would be 2,008,112 pounds. Assuming a density of 0.178 pounds per cubic foot for helium, we can calculate that it would take 11,246,214 cubic feet of helium to fill all of the balloons. Since a 10" diameter latex balloon holds 17.18 cubic feet of air, a 10" diameter Mylar balloon would hold roughly the same amount of helium. Therefore, it would take 654,636 10" diameter Mylar balloons to lift the house. A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions.
Score	7	5
Review	Assistant 1 provided a more concise and clear explanation, using the buoyant force equation and the weight of the house from the movie "Up" to calculate the number of balloons needed. However, the final number of balloons was slightly off.	Assistant 2's answer was more detailed but contained some irrelevant information about the average U.S. home and latex balloons, which made the answer more confusing. Additionally, Assistant 2's answer did not provide a clear final number of balloons needed to lift the house.

Table 22: Qualitative Study for LLaMA-33B and DoLa with GPT-4 judgement.