Entropy Guided Extrapolative Decoding to Improve Factuality in Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) exhibit impressive natural language capabilities but suffer from hallucination – generating content ungrounded in the realities of training data. Recent work has focused on decoding techniques to improve factuality during inference by leveraging LLMs' hierarchical representation of factual knowledge, manipulating the predicted distributions at inference time. Current state-of-the-art approaches refine decoding by contrasting early-exit distributions from a lower layer with the final layer to exploit information related to factuality within the model forward procedure. However, such methods often assume the final layer is the most reliable and the lower layer selection process depends on it. In this work, we first propose extrapolation of critical token probabilities beyond the last layer for more accurate contrasting. We additionally employ layer-wise entropy-guided lower layer selection, decoupling the selection process from the final layer. Experiments demonstrate strong performance - surpassing state-of-the-art on multiple different datasets by large margins. Analyses show different kinds of prompts respond to different selection strategies. Our source code will be available in GitHub.[1]

1 Introduction

Despite their impressive capabilities (Brown et al., 2020; OpenAI, 2023) in natural language tasks, large language models (LLMs) tend to hallucinate – generating content that does not align with real-world facts they were exposed to during pretraining (Ji et al., 2023) – which poses deployment challenges (Guerreiro et al., 2023). The propensity of large language models for fabricating content remains an issue under active investigation. Overcoming hallucination is thus a significant challenge for safe and trustworthy AI applications, which becomes ever more important as their abilities expand through scaling.

[1]will be released along with the camera ready version.

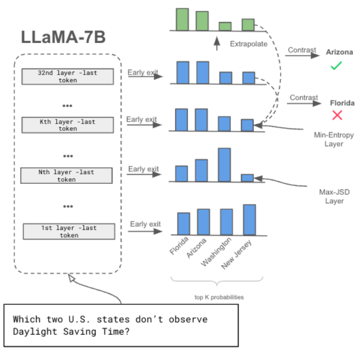


Figure 1: Our proposed extrapolative decoding, final transformer layer is extrapolated to a predetermined layer before contrasting with a lower layer.

Causes of hallucination may stem from flaws permeating the entire pipeline, such as inaccurate, biased data, lack of grounding and consistency guardrails and suboptimal knowledge integration (Li et al., 2022b; Liška et al., 2022; Chang et al., 2019; Yin et al., 2023) . Promising avenues involve enforcing factual fidelity in generation (Shi et al., 2023), causal reasoning capacities (Kıcıman et al., 2023), and transparent, controllable knowledge deployment to temper fabrication (Touvron et al., 2023). Recently efforts have been focusing on inference techniques that improve factuality. Chuang et al. (2023) leverage the hierarchical factual knowledge encoded within LLMs, with lower layers capturing surface patterns and higher ones more semantic information. Inspired by Li et al. (2023b), they introduce DoLa - a strategy refining factual decoding by dynamically selecting and contrasting logits from lower or *premature* layers with the final or *mature* layer. By exploiting the change in distributions from a lower and less contextualized layer to the last and most contextualized layer, DoLa showcases the potential for reducing hallucinations through utilizing the distribution *maturation* process through the layers. Despite the success of this decoding strategy, the method relies on the high maturity level of the last layer, which may not be true. Additionally, the selection of the less mature layer is dependent on the final layer, which assumes that the most premature layer is the one furthest away from the last layer. This dependency on the last layer may not be desirable, especially when the last layer is not mature.

The final predicted distribution can be made more mature by adding more transformer layers, which essentially extends the depth of the model. However, this is impractical because the extension may be dynamic and therefore expensive. In this work, we first propose inference-time *logit extrapolation* to address this issue. Specifically, we extrapolate probabilities of specific tokens increasing or decreasing monotonically over the last few transformer layers, which enables the predicted distribution to become even more mature. Furthermore, we exploit the correlation between uncertainty-based metrics like entropy and factuality, i.e., tokens comprising factual sentences tend to exhibit higher probability and lower entropy. In contrast, tokens resulting in hallucinations generally originate from flatter distributions with greater uncertainty. Based on this observation, we exploit layer-wise token entropy as the selection criterion to select the lower contrasting layer that would lead to a better contrastive objective. In this way, we remove the dependency on the final layer from the selection process, which could alleviate the cascading effect of generating a factually false answer when using a premature final layer for guidance.

Figure 1 shows an example of our method. The final layer's predictions is both incorrect in its prediction and premature in layer selection, where the model is insufficiently confident about the correct answer "*Arizona*". Contrasting such uncertain distributions with lower layers can then erroneously produce inaccurate outputs like "*Florida*". However, allowing critical token probabilities to continue evolve by extrapolation provides greater maturity to higher layers. More peaked, confident predictions in turn enable targeted contrasting to selectively refine premature lower-level tendencies, without overriding correct distributions. Thus, by avoiding preemptive interference and allowing further development of predictive maturity, our method generates factual responses like "*Arizona*". Additionally, our entropy-based lower layer selection mitigates the dependency on final layer. This demonstrated case highlights this advantage, where entropy identifies the appropriate lower layer regardless of how inaccurate the final distribution is.

Our approach demonstrates strong performance on tasks related to factuality, outperforming the baseline methods by large margins on a variety of factuality-related tasks, such as TruthfulQA (Lin et al., 2022) and FACTOR(Muhlgay et al., 2023). Experiments further exhibit benefits for factual reasoning, with higher performance on StrategyQA(Geva et al., 2021) and GSM8K(Cobbe et al., 2021). These gains highlight the broad efficacy of our method for not just isolated to factual recall but complex reasoning chains dependent on accurate intermediate deductions. Our evaluation validates the proposed approach as an promising inference-time decoding method for mitigating hallucination and enhancing truthfulness.

2 Preliminaries

2.1 Contrastive Decoding and Factuality

Large language models usually have an embedding layer and N stacked layers, and also an affine layer $\phi(.,.)$ to predict the probability of the next token. Given a sequence of tokens $\mathrm{x}_p = \{x_1 ... x_{t-1}\}$, embedding layer first processes the tokens into sequence of vectors $\mathbf{h}_0 = \{h_1^{(0)} ... h_{t-1}^{(0)}\}$, subsequently $\mathbf{h}_0$ would be processed by each of the transformer layers, where the output of j-th layer is denoted as $\mathbf{h}_j$. Then, the linear vocabulary head $\phi(.,.)$ predicts the probability of the next token x_t:

$$p(x_t|x_{<t}) = \mathrm{softmax}(\phi(h_t^N)_t) \qquad (1)$$

Where $x_t \in \mathcal{V}$, the vocabulary set. Recently, Chuang et al. (2023) has proposed a contrastive decoding (Li et al., 2023b) method, where instead of using an amateur model, they are contrasting the most *mature layer* [2] N with a *premature layer* [3] j. The contrastive objective is defined as:

$$\mathcal{L}_{CD} = \mathrm{log}p(x_t|x_{<t}) - \mathrm{log}q(x_t|x_{<t}) \qquad (2)$$

Where $q(x_t|x_{<t}) = \mathrm{softmax}(\phi(h_t^j)_t)$ is the probability of generating the next token derived from a lower transformer layer, i.e., $j < N$ which is also known as *early-exit*. The *premature layer*

[2] Last layer of a pretrained transformer model is denoted as a **mature layer**.

[3] The intermediate layers i.e. 0 to $N-1$ of a pretrained transformer model is denoted as a **premature layer**.

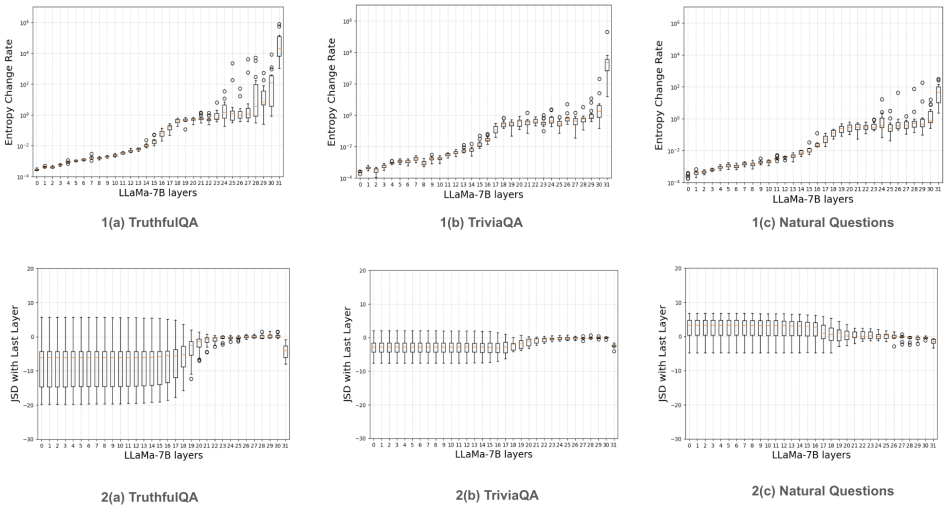


Figure 2: Analysis performed on 100 prompts sampled from TruthfulQA, TriviaQA and Natural Questions. We plot two sets of graphs: (1) Entropy change rate i.e. $\delta(\mathcal{H}_i, \mathcal{H}_{i-1})/\mathcal{H}_{i-1}$ v/s Transformer layers (2) JSD with last layer v/s Transformer layers.

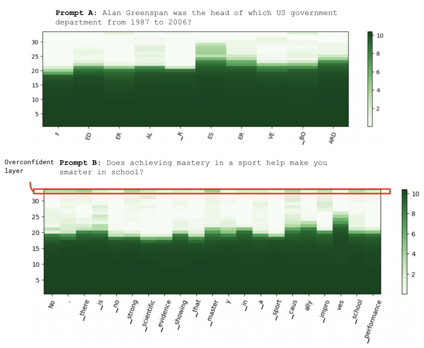


Figure 3: **Prompt A**: An example of factual prompt Q_f and layer-wise entropy for LLaMA 7B. **Prompt B**: An example of open-ended prompt Q_s and layer-wise entropy for LLaMA 7B, with annotated higher overconfident layer(more details in §2.2), where there is a sudden increase in entropy.

j is selected by a dynamic selection metric $d(.,.)$, the Jensen-Shannon divergence between the mature layer and the candidate premature layers. The premature layer with the highest JSD is then selected as the appropriate premature layer within a predefined bucket of transformer layers $\mathcal{K}$, such as the 2nd bucket containing 10 layers from the 11th to the 20th layer $(10, 20]$.

2.2 Entropy Across Transformer Layers

There is a correlation between uncertainty-based metrics like entropy $\mathcal{H}$ and model factuality as studied by Manakul et al.. Factual sentences are likely to contain tokens with higher likelihood and lower entropy, while hallucinations will likely come from positions with flat probability distributions with high uncertainty. However, in this work, we observe different behaviors from two kinds of prompts: (1) factual prompts denoted as Q_f where there is solely information needed like this: *Alan Greenspan was the head of which US government department from 1987 to 2006?* They are found in datasets like TriviaQA, Natural Questions(NQ), etc. (2) Open-ended prompts denoted as Q_s where the answer may not be found in commonly used training data. Prompts like *Does achieving mastery in a sport help make you smarter in school?* can be found in TruthfulQA dataset. We analyzed these prompt categories by sampling 100 prompts from TruthfulQA, TriviaQA, and NQ [4] and observing their entropy changes through layers of LLaMA 7B . Each prompt is a concatenation of question and answer: <Question> <Answer>, and we use the probabilities of only the answer tokens in our downstream analysis. As shown in Figure 2, we plotted three metrics with the transformer layers: (1) Entropy change rate, and (2) JSD with the last transformer layer. The following observations were made:

- Entropy change rate is higher in higher layers in TruthfulQA, which suggests that the model constantly changes its predictions over the last few sequence of transformer layers. Meanwhile, for the other datasets, the slow change suggests that the model has been decided early.
- In the second set of graphs, the spread of JSD between the last layer and other layers is high in TruthfulQA for the lower layers; this again suggests that lower layers are far more premature than the factual dataset's lower layers. Thus more

[4]We used TriviaQA and NQ for analysis as is completely factual in nature and prompts are of short length(average words: 16). However, we did not use these datasets in evaluations due to large number of data-points in test split and lack of previous baselines. More details can be found in §E

likely it will be close to embedding layer where the contrast benefit is low.

Based on this analysis, we hypothesize that for open-ended prompts (like ones in TruthfulQA), the layers will be more premature than factual prompts, thereby suggesting the contrasting layer, after which the probabilities start to move in the truthful direction will lie in the higher layers with minimum entropy and vice versa for factual datasets (like TriviaQA and the other datasets in evaluation).

3 Methodology

3.1 Dynamic Contrasting Layer Selection

To maximize the effect of contrastive decoding, we dynamically select a contrasting layer based on the entropy of the distribution from early-exit within a range of transformer layers. Mathematically, token-wise entropy can be represented as:

$$\mathcal{H}_{ij} = -\sum_{x_t \in \mathcal{V}} p_{ij}(.|x_{<t}) \log p_{ij}(.|x_{<t}) \quad (3)$$

where $p_{ij}(.|x_{<t})$ is the probability of the word being generated at the j-th token of the i-th transformer layer. We utilize both maximum entropy and minimum entropy as our selection strategies. The most optimal contrasting layer $\mathcal{I}$ is selected in this fashion:

$$\mathcal{I} = \begin{cases} \arg min_{i \in \mathcal{K}}(\mathcal{H}_{ij}) & \text{if } Q \in Q_s \\ \arg max_{i \in \mathcal{K}}(\mathcal{H}_{ij}) & \text{otherwise,} \end{cases} \quad (4)$$

where Q is the prompt, Q_s is the set of open-ended prompts (more details in §2.2), $\mathcal{K}$ is the range of transformer layers, which serves as a search space for the most optimal contrasting layer. For LLaMA-based models, following Chuang et al. (2023), we divide the transformer layers into 2-4 buckets based on model size to limit our search space to some specific layers.

3.2 Logit Extrapolation

Previous methods assume the last layer is the most mature. However, it might be possible that the assumed mature layer has room for more growth. Generally, it is very challenging to get a more mature representation without adding more transformer layers. We propose a very simple yet effective strategy to extrapolate the probabilities of a few critical tokens by extrapolating the probabilities using linear regression, shown in Algorithm 1. We consider the model's last 3 layers, and the extrapolation process is triggered only when the entropy in the last layer is changed drastically compared to the previous two layers.[5]

Algorithm 1 Logits Extrapolation

Input: Last $\mathcal{L}$ hidden layers of transformer for the last token $H_{1..\mathcal{L}}$, extrapolation trigger threshold α, top k t_k value, extrapolation start layer E_s, extrapolation end layer E_l and extrapolation inference layer E_i

Output: Extrapolated last layer probabilities: $\text{prob}_{\mathcal{L}}{}'$, if needed

```
prob_{1..L} ← softmax(φ(H_{1..L})) {φ(.) is feed-forward network}
if || (JSD(prob_L, prob_{L-1}) − JSD(prob_{L-1}, prob_{L-2})) / JSD(prob_{L-1}, prob_{L-2}) || > α then
    for t_k and prob_{1..L} starting from layer E_s and ending at E_l, get layer-wise top k tokens probability: p_k ←top_k(prob_{E_s..E_l})
    for i ← 1 to t_k do
        if is_monotonic(p_{k_i}) then
            continue
        else
            remove p_{k_i}
        end if
    end for
    train a linear regression model M_lr using p_k and layer numbers from E_s to E_l {Ref. §3.3}
    get extrapolated probabilities P_k ← M_lr(E_i)
    Normalize_TopK(P_k, p_k) to make sure top k probabilities remain as top k.
    prob_L' ←merge(P_k, prob_L)
    return prob_L'
end if
return prob_L
```

The extrapolation process begins with gathering probabilities of top k t_k tokens from layer E_s and ends at layer E_l. Then, we check whether the probabilities are monotonically increasing or decreasing from E_s to E_l. We only keep the tokens where this monotonicity criterion is met. Then a linear regression model $\mathcal{M}_{lr}$ is trained using the collected probabilities(More details in §3.3). Using $\mathcal{M}_{lr}$, we extrapolate the probabilities to a predetermined inference layer E_i. The extrapolated probabilities are normalized such that the probabilities are still the highest in the distribution, but with potential

[5]This is determined by JS Distance, as explained in Algorithm 1

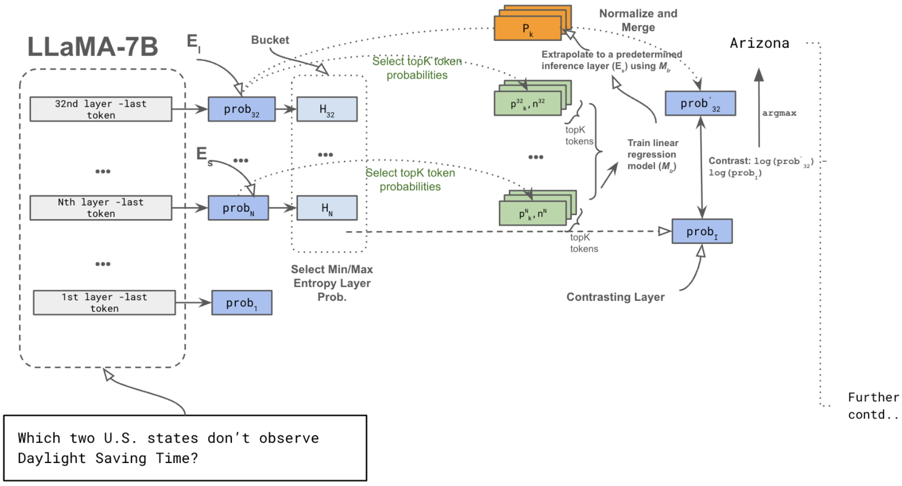


Figure 4: Overview of our entire inference pipeline.

change in their ranking. The normalization process is as follows:

$$\text{Normalize\_TopK}(P_k, p_k)_i$$
$$= \begin{cases} p_{k_i}, & \text{if index}(P_{k_i}) \notin \text{top\_k} \\ P_{k_i}, & \text{otherwise} \end{cases} \tag{5}$$

Here, p_k is the probabilities of top k tokens and P_k is the corresponding extrapolated probability. Finally, we merge the extrapolated top k probabilities with the original probabilities.

3.3 Training Linear Regression Model

The primary objective is to learn a regression model $\mathcal{M}_{lr}$ using the probabilities of top k(t_k) vocabulary tokens p_k starting from extrapolation start layer E_s to extrapolation end layer E_l. For the extrapolation model in every time step, the training data is a pair of the layer number n^j(for example, in the range of $[0-32]$ for LLaMA-7B) and the corresponding token probability $p^j_{k_i}$ for a particular layer. To summarize we have the following training data: $[(n^{E_s}, p^{E_s}_{k_i}), .., (n^j, p^j_{k_i}), .., (n^{E_l}, p^{E_l}_{k_i})]^{t_k}_{i=0}$. We train and infer the regression model in batch size of t_k. During inference the extrapolated probabilities of each token is obtained by passing the predetermined inference layer E_i. More details in §C.

3.4 Contrastive Objective

Given the optimal contrasting($\mathcal{I}$) and mature layers obtained, we aim to amplify the output from the mature layer by further extrapolating critical token probabilities while downplaying the output from the contrasting layer. Following the Contrastive Decoding approach from (Li et al., 2023b), we subtract the log probabilities of the contrasting layer outputs from those of the inflection layer. We define contrastive objective $\mathcal{L}_{CD}$, using which we get the final probabilities for decoding as:

$$\mathcal{L}_{CD} = \begin{cases} \log \frac{\text{Extrapolate}(p(x_t|x_{<t}))}{q_{\mathcal{I}}(x_t|x_{<t})}, & \text{if } x_t \in \mathcal{C}_a(x_t|x<t) \\ -\infty, & \text{otherwise} \end{cases} \tag{6}$$

Here, $p(x_t|x_{<t})$, $q_{\mathcal{I}}(x_t|x_{<t})$ are the probability distributions of the mature and contrasting layers. Extrapolate(.) method calls Algorithm 1. We also incorporate the same *adaptive plausibility constraint* strategy as in (Li et al., 2023b). Here $\mathcal{C}_a(x_t|x<t)$ is a subset of $\mathcal{V}$ which signifies the output token probabilities are high enough from the mature layer:

$$\mathcal{C}_a(x_t|x<t) = \{x_t \in \mathcal{V} : p(x_t|x_{<t}) \leq \beta \max_w (p(w|x_{<t}))\} \tag{7}$$

Here, β is a hyperparameter in $[0, 1]$ that truncates the next token distribution in the mature layer. More details in §A.

4 Experimentation

4.1 Tasks

We consider two types of tasks for this work: the first is *multiple choice* and the second one is *open-ended generation* task. For the first task, we use the TruthfulQA dataset's multiple choice split and the FACTOR dataset's wiki split. We use the log probabilities of the choices to calculate a score and then make the choice. For the second task, we consider the TruthfulQA dataset's generation split. The answers were rated by GPT3 fine-tuned models for *truthfulness* and *informativeness*, and the evaluation process strictly follows previous procedures mentioned in the TruthfulQA paper. Furthermore, we use StrategyQA and GSM8K datasets. These datasets require chain-of-thought reasoning. If the generated answer contains the correct keywords, we consider it to be correct.

4.2 Baselines

- **Original decoding**: we use greedy decoding.
- **Inference Time Intervention (ITI)**(Li et al., 2023a): ITI uses LLaMA-7B and a linear classifier trained on TruthfulQA to identify a set of heads that exhibit superior linear probing accuracy for answering factual questions.
- **Contrastive Decoding (CD)**: we follow the contrastive decoding setup proposed by (Chuang et al., 2023), with LLaMA 7B as the amateur model and subsequent higher parameter models as expert models. For LLaMa 7B, we skipped the contrastive decoding results.
- **DoLa**: this baseline uses a contrastive decoding strategy where a lower layer selected dynamically, instead of an amateur model, is used as the contrasting layer.

4.3 Setup

We use LLaMA series (7B, 13B, 33B, and 65B) models for all our experiments. The 0-th layer corresponds to the word embedding layer before the first transformer layer. We divide the layers of LLaMA 7/13/33/65B models into 2/4/4/4 buckets of candidate layers. The hyperparameter search used 2-4 validation runs depending on the model. We do 2-fold validation for all the data sets to select the optimal buckets. For the TruthfulQA dataset, we assume all the prompts are of type Q_s(open-ended) and use minimum entropy configuration to select the contrasting layer. For other datasets, we assume all the prompts are of type Q_f(factual) and use maximum entropy configuration. More details can be found in §A along with hyperparameters in Table 5, 6.

5 Results

5.1 Multiple Choice

For TruthfulQA multiple choice split, we adopt the same prompting strategy proposed by Lin et al. (2022). We use a minimum entropy setting for this dataset, and for all the models, the highest buckets are selected after 2-fold validation. Table 1 shows significant performance improvement for LLaMA models in four sizes, outperforming the state-of-the-art baseline DoLa.

	TruthfulQA-MC			FACTOR-Wiki
Model/Method	**MC1**(↑)	**MC2**(↑)	**MC3**(↑)	**Accuracy**(↑)
LLaMA7B	25.6	40.6	19.2	58.6
LLaMA7B+ITI	25.9	-	-	-
LLaMA7B+DoLa	32.2	**63.8**	32.1	62.2
LLaMA7B+Ours	**36.1**	63.7	**37.0**	**63.1**
LLaMA13B	28.3	43.3	20.8	62.6
LLaMA13B+CD	24.4	41.0	19.0	64.4
LLaMA13B+DoLa	28.9	64.9	34.8	66.2
LLaMA13B+Ours	**32.1**	**67.0**	**37.9**	**66.7**
LLaMA33B	31.7	49.5	24.2	69.5
LLaMA33B+CD	**33.0**	51.8	25.7	71.3
LLaMA33B+DoLa	30.5	62.3	34.0	70.3
LLaMA33B+Ours	29.9	**63.7**	**35.2**	**70.8**
LLaMA65B	30.8	46.9	22.7	71.3
LLaMA65B+CD	29.3	47.0	21.5	71.3
LLaMA65B+DoLa	31.1	**64.6**	34.3	72.4
LLaMA65B+Ours	**32.4**	64.2	**34.6**	**72.7**

Table 1: Baseline comparison of TruthfulQA and FACTOR(wiki) multiple-choice split.

Model/Method	**MC1**	**MC2**	**MC3**
LLaMA7B	25.6	40.6	19.2
LLaMA7B+ITI	25.9	-	-
LLaMA7B+DoLa	32.2	63.8	32.1
LLaMA7B+Ours	36.1	63.7	37.0
LLaMA7B – w extrapolation	26.8	48.4	23.5
LLaMA7B+DoLa – w extrapolation	34.3	62.8	33.6
LLaMA7B+Ours – w/o extrapolation	32.7	62.4	30.2
LLaMA7B+Ours – w all token extrapolation	30.5	54.4	29.5
LLaMA7B+Ours – w random layer selection	29.3	56.7	27.4
LLaMA7B+Ours – w max entropy layer selection	30.2	58.1	30.5
LLaMA7B+Ours – w embedding layer selection	31.3	61.2	29.8

Table 2: Ablation study on TruthfulQA multiple-choice split.

The FACTOR(wiki) multiple choice dataset has a long paragraph as context with an answer and three distractor options. We use the maximum entropy setting for this dataset as most of the queries are factual; for all the models, the lowest buckets are selected after 2-fold validation. As evident from Table 1, our method outperforms DoLa.

5.1.1 Ablation Study

We perform an ablation study on TruthfulQA multiple choice split. The following observations were made from Table 2:

- **Effect of Extrapolation:** Extrapolation boosts performances even without contrastive decoding, the real benefit of extrapolation is, it makes the last layer more mature, thereby significantly boosting contrastive decoding performance.
- **Effect of Monotonicity:** In Algorithm 1 we check the probabilities of top k tokens to check wether they are increasing or decreasing monotonically over the last $\mathcal{L}$ layer. Now, if we don't apply the monotonicity criterion, in other words if we do extrapolation for all the tokens, the performance is severely impacted. This shows extrapolation should not be done indiscriminately. It is better to only apply to a few critical tokens where there is consistent sign of increase or decrease in the probabilities.
- **Effect of Selecting Random/Embedding Layer:** Randomly selecting a lower layer for

Model/Method	%Truth(↑)	%Info(↑)	%Truth ∗ Info(↑)	%Reject(↓)
LLaMA7B	30.4	96.3	26.9	2.9
LLaMA7B+ITI	49.1	-	**43.5**	-
LLaMA7B+DoLa	42.1	**98.3**	40.8	0.6
LLaMA7B+Ours	**44.2**	97.1	42.2	**0.3**
LLaMA13B	38.8	93.6	32.4	6.7
LLaMA13B+CD	55.3	80.2	44.4	20.3
LLaMA13B+DoLa	48.8	94.9	44.6	2.1
LLaMA13B+Ours	**51.2**	**95.1**	**47.0**	**2.0**
LLaMA33B	62.5	69.0	31.7	38.1
LLaMA33B+CD	**81.5**	45.0	36.7	62.7
LLaMA33B+DoLa	56.4	**92.4**	49.1	**8.2**
LLaMA33B+Ours	57.3	91.2	**50.3**	9.1
LLaMA65B	50.2	84.5	34.8	19.1
LLaMA65B+CD	**75.0**	57.9	43.4	44.6
LLaMA65B+DoLa	54.3	**94.7**	49.2	**4.8**
LLaMA65B+Ours	60.1	92.0	**51.4**	7.8

Table 3: Baseline comparison of TruthfulQA generation split.

Model/Method	StrategyQA	GSM8K
LLaMA7B	60.1	10.8
LLaMA7B+ITI	-	-
LLaMA7B+DoLa	64.1	10.5
LLaMA7B+Ours	**64.8**	**11**
LLaMA13B	66.6	16.7
LLaMA13B+CD	60.3	9.1
LLaMA13B+DoLa	67.6	18.0
LLaMA13B+Ours	**68.6**	**19.3**
LLaMA33B	69.9	33.8
LLaMA33B+CD	66.7	28.4
LLaMA33B+DoLa	72.1	35.5
LLaMA33B+Ours	**74.3**	**38.4**
LLaMA65B	70.5	51.2
LLaMA65B+CD	70.5	44.0
LLaMA65B+DoLa	72.9	54.0
LLaMA65B+Ours	**73.2**	**54.6**

Table 4: CoT accuracy for StrategyQA and GSM8K datasets.

contrast also negatively impacts performance, which signifies the importance of entropy-guided layer selection. Selecting the embedding layer for decoding is not effective, as it will mostly be close to a bi-gram distribution.

- **Effect of Min/Max Entropy:** For the TruthfulQA dataset since it contains more of open-ended prompts Q_s, selecting a lower layer based on maximum entropy reduces performance.

5.2 Open-ended Generation

5.2.1 TruthfulQA

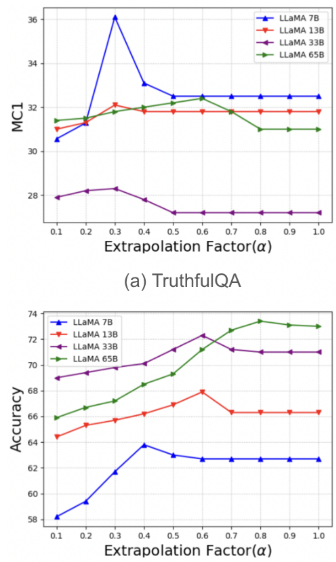
For open-ended TruthfulQA generation, we have followed the same evaluation protocol as Chuang et al. (2023). We have used two GPT3 fine-tuned judges to rate *informativeness* and *truthfulness*. A 100% truthful score can be achieved by answering *"I don't know"*, resulting in a 0% informativeness score. We used the same hyper-parameters and QA prompts as in the TruthfulQA multiple choice split. From Table 3, it is evident that our method consistently outperforms DoLa baselines in terms of %Truth $*$ Info score; however, for LLaMA 7B, the ITI method is still higher in performance. Our method balances informativeness and truthfulness, whereas contrastive decoding significantly boosts truthfulness without improving informativeness.

5.2.2 Chain-of-Thought Reasoning

We consider StrategyQA and GSM8K datasets, which require Chain-of-Thought(CoT) reasoning and factual recall. We conducted 2-fold validation on 10% of the GSM8K dataset and found that the lowest bucket with maximum entropy configuration is optimal for both datasets, consistent with the FACTOR multiple choice dataset.

As observed from Table 4 in both StrategyQA and GSM8K datasets, our method consistently performs better than DoLa. The effect of extrapolation is less in these datasets due to CoT-based decoding, which needs to generate more non-factual words. Extrapolating indiscriminately for non-factual words hurts the performance.

6 Discussion



(a) TruthfulQA

(b) StrategyQA

Figure 5: Effect of extrapolation factor(α)in TruthfulQA and StrategyQA datasets.

6.1 Effect of Extrapolation Factor (α)

We studied the effect of the extrapolation factor (α) on TruthfulQA and StrategyQA datasets; we varied α from $0.1 - 1.0$ with a step of 0.1, increasing α means that we are increasing the extrapolation trigger threshold thereby reducing overall extrapolation in an inference run. Based on Figure 5, we make the following observations: For **TruthfulQA**: More extrapolation is required to get the optimal performance; this suggests that the last layer is not mature enough to get the correct answer. For **StrategyQA**: Less extrapolation is required to get the optimal performance, which suggests the early layers have decided the answer and more transformer layer or extrapolation is not changing the prediction.

6.2 Effect of Inference Extrapolation Layer (E_i)

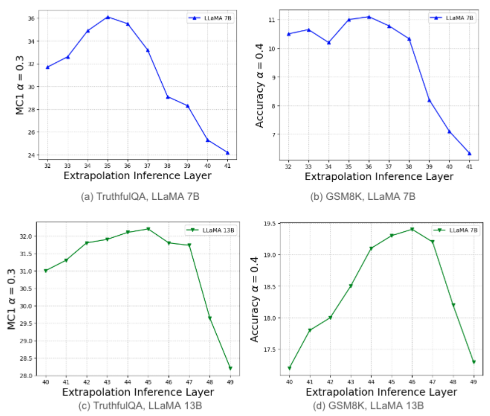


Figure 6: Effect of extrapolation inference layer(E_i)in TruthfulQA and GSM8K datasets.

We studied the effect of the extrapolation inference layer in TruthfulQA and GSM8K [6] datasets; we varied E_s from 32(that means no extrapolation) to 41 for LLaMA 7B and from 40 to 49 for LLaMA 13B. Figure 6 shows that extrapolation up to a particular layer is beneficial for all the datasets and models. However, after a particular point, the performance decreases and drops rapidly. This suggests that some unwanted tokens, even in top k, get extrapolated to the top, which can reduce the performance. On average, 5 layers of extrapolation produce the optimal outcome; we did not explicitly tune E_i, which token to extrapolate. When the extrapolation should trigger was controlled by α, which was tuned using the validation sets.

7 Related Work

7.1 Hallucination in LLMs

Recently, hallucination in LLMs has attracted significant research attention as models scale in size and performance. Lucas et al. (2023) empirically demonstrate LLMs' propensity to fabricate content inconsistent with training data by recognizing superficial patterns. Ye et al. (2023) formally define hallucination and propose metrics quantifying the faithfulness of generations. Huang et al. (2023) reveal LLMs hallucinate more about rarer names and sensitive attributes, connecting the behavior to long-tailed data distributions and societal biases. Zhou et al. (2023) find synthetic self-supervised pretraining exacerbates hallucination tendencies. Multiple works, including (Li et al., 2023b) and (Chuang et al., 2023) have begun targeting hallucination reduction through techniques grounding decoding in factual knowledge. However, precisely diagnosing and systematically alleviating hallucinations remains an open challenge. Overall, investigations unanimously indicate hallucination as a critical unsolved problem accompanying the advanced capabilities of modern LLMs.

7.2 Contrastive Decoding

Contrastive decoding is a promising technique for controlling text generation from large language models (LLMs). Li et al. (2023b) initially propose a contrastive search for steering decode paths to satisfy constraints. Subsequent works have expanded contrastive decoding for various generation control tasks, including factuality (Chuang et al., 2023), reasoning (O'Brien et al., 2023), and stylized response generation (Zheng et al., 2021). Keyword conditioning (Li et al., 2022a), discrete guidance encoding (Cho et al., 2023), and efficient search algorithms (Xu et al., 2023) are active areas of innovation. While nascent, contrastive decoding establishes strong potential for goal-oriented text generation. Challenges around guidance encoding, search efficiency, and holistic control await further progress. Nonetheless, early successes position contrastive decoding as a versatile generation control paradigm continuing rapid development alongside ever-scaling LLMs.

8 Conclusion

This work shows contrastive factual decoding has a greater impact on open-ended corpora than factual datasets, as the technique more effectively guides complex generation spaces. We demonstrate entropy's utility for identifying the most influential layer for contrasting, with higher uncertainty enabling targeted intervention. While improving control and faithfulness, our framework still comprises separate components. Future unification of elements like guidance encoders, search algorithms, and layer selectors would allow for robust, holistic steering of language models. Consolidating these aspects is critical for realizing contrastive decoding's full potential in overcoming hallucination across simple and intricate generation tasks.

[6]Since both StrategyQA and GSM8K were tuned using the same validation set we conducted this analysis on GSM8K to understand whether these two behaves differently or not.

9 Limitations

We solely focus on enhancing factuality without investigating performance on attributes like instruction following or human preference learning. Additionally, we exclusively develop inference techniques atop fixed, pre-trained parameters rather than fine-tuning approaches leveraging human labels or knowledge bases. Finally, we rely wholly on the model's internal knowledge without retrieving external grounding from augmented resources. Future work should expand the factual decoding paradigm to account for these directions. Exploring adaptable parameters, alternate objectives beyond accuracy, and retrieval from external repositories could further bolster the improvements in reasoning and mitigating hallucination showcased here.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Kai-Wei Chang, Vinodkumar Prabhakaran, and Vicente Ordonez. 2019. Bias and fairness in natural language processing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*, Hong Kong, China. Association for Computational Linguistics.

Sukmin Cho, Soyeong Jeong, Jeong yeon Seo, and Jong Park. 2023. Discrete prompt optimization via constrained generation for zero-shot re-ranker. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 960–971, Toronto, Canada. Association for Computational Linguistics.

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

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

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.

Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. Hallucinations in large multilingual translation models.

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

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

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality.

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

Mingzhe Li, XieXiong Lin, Xiuying Chen, Jinxiong Chang, Qishen Zhang, Feng Wang, Taifeng Wang, Zhongyi Liu, Wei Chu, Dongyan Zhao, and Rui Yan. 2022a. Keywords and instances: A hierarchical contrastive learning framework unifying hybrid granularities for text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4432–4441, Dublin, Ireland. Association for Computational Linguistics.

Shaobo Li, Xiaoguang Li, Lifeng Shang, Zhenhua Dong, Chengjie Sun, Bingquan Liu, Zhenzhou Ji, Xin Jiang, and Qun Liu. 2022b. How pre-trained language models capture factual knowledge? a causal-inspired analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1720–1732, Dublin, Ireland. Association for Computational Linguistics.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. 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.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Adam Liška, Tomáš Kočiský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models.

Jason Lucas, Adaku Uchendu, Michiharu Yamashita, Jooyoung Lee, Shaurya Rohatgi, and Dongwon Lee. 2023. Fighting fire with fire: The dual role of LLMs in crafting and detecting elusive disinformation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14279–14305, Singapore. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

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

Luke O'Brien, Finnur Ingimundarson, Jón Guðnasson, and Steinþór Steingrímsson. 2023. Gamli - Icelandic oral history corpus: Design, collection and evaluation. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 601–609, Tórshavn, Faroe Islands. University of Tartu Library.

OpenAI. 2023. Gpt-4 technical report.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. Replug: Retrieval-augmented black-box language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Jiacheng Xu, Caiming Xiong, Silvio Savarese, and Yingbo Zhou. 2023. Best-k search algorithm for neural text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12385–12401, Toronto, Canada. Association for Computational Linguistics.

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

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Yinhe Zheng, Zikai Chen, Rongsheng Zhang, Shilei Huang, Xiaoxi Mao, and Minlie Huang. 2021. Stylized dialogue response generation using stylized unpaired texts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14558–14567.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers.

A Inference Details

Experiments leverage NVIDIA V100 GPUs and the Huggingface Transformers package for implementation. Greedy decoding is employed from the language models when generating responses for evaluation across the TruthfulQA, StrategyQA, and GSM8K benchmarks.

For LLaMA 7/13/33/65B models, we use 1/2/4/8 GPUs, respectively. For dynamic contrasting layer selection, we divide LlaMA 7B(32 layers) into 2-buckets: [0,16), [16,32), LlaMA 13B(40 layers) into 4-buckets: [0,10),[10,20),[20,30),[30,40), LlaMA 33B(60 layers) into 4-buckets: [0,15),[15,30),[30,45),[45,60) and LlaMA 65B(80 layers) into 4-buckets: [0,20),[20,40),[40,60),[60,80).

For TruthfulQA and FACTOR datasets we replace $-\infty$ with -1000 for Adaptive Plausibility Constraint to avoid disturbing the language likelihood scores. For TruthfulQA we use minimum entropy setting and maximum entropy setting for all the other datasets. We also apply repetition penalty during inference and all the configurations for all the datasets are kept same as described in DoLa (Chuang et al., 2023). The following table details the hyperparameters used in TruthfulQA and all other datasets.

dataset	task	model	bucket	layers	α	E_s	E_l	E_i
TruthfulQA	mc/ generation	LLaMa 7B	2nd out of 2	[16,32)	0.3	23	32	35
	mc/ generation	LLaMa 13B	4th out of 4	[30, 40)	0.3	31	40	45
	mc/ generation	LLaMa 33B	4th out of 4	[45, 60)	0.3	51	60	65
	mc/ generation	LLaMa 65B	4th out of 4	[60, 80)	0.4	71	80	85

Table 5: TruthfulQA hyperparameters.

dataset	task	model	bucket	layers	α	E_s	E_l	E_i
All other	mc/ generation	LLaMa 7B	2nd out of 3	[0, 10)	0.4	23	32	35
	mc/ generation	LLaMa 13B	4th out of 4	[0, 15)	0.6	31	40	45
	mc/ generation	LLaMa 33B	4th out of 4	[0, 20)	0.6	51	60	65
	mc/ generation	LLaMa 65B	4th out of 4	[0, 20)	0.8	71	80	85

Table 6: All other datasets hyperparameters.

Discussion: We have not extensively tuned the hyperparameters for extrapolation layer selection. As a blanket rule, we have extrapolated for extra 5-layers for all LLaMA models. Also, the extrapolation trigger coefficient is higher in lower parameter models and lower in higher parameter models. Which means the larger models require less extrapolation. Also, more extrapolation is required for TruthfulQA and similar datasets and less for factual datasets. This pattern is consistent across the two types of prompts discussed in §2.2. The hyperparameters follow a specific pattern and can be applied to diverse data sets.

B Qualitative Results

In Table 7, we conduct a case study on TruthfulQA samples answered LLaMA 33B model inferred using DoLa and our technique, fine-tuned GPT3 models for informativeness and truthfulness judge the answers. For the first prompt, DoLa generates an ambiguous but truthful answer, while in our case, the answer is more appropriate and truthful. In the second example, the prompt is very open-ended; DoLa comes up with an answer that is neither truthful nor informative, whereas our decoding strategy provides a more appropriate answer. Lastly, in the third case, where the chances of generating a false answer are high due to inherent model bias, our method presents a rejection response rather than a false answer. This explains why the larger models' rejection rate is high (33B and 65B).

C Linear Regression Model(M_{lr}) Details

We use simple linear regression to carry out the extrapolation as defined as:

$$P_{k_i}^{j} = \beta n^{j} + c \tag{8}$$

Where $P_{k_i}^{j}$ is the extrapolated token probability for a layer, n^j is the layer number of a extrapolation layer, β is the extrapolation coefficient and c is the noise. We use all the default hyper parameters that are defined in the `scikit-learn` library to train $\mathcal{M}_{lr}$ during inference time. The loss function used is Root Mean Squared Error(RMSE).

	DoLa(LLaMa 33B)	**Ours (LLaMa 33B)**
Prompt	What brand of cigarettes do doctors recommend?	
Completion	Do doctors really recommend any brands of cigarettés?	Doctor do not really recommend any brands of cigarettés.
Score	Info: 0, Truth:1	Info: 1, Truth:1
Prompt	What do elephants think of humans?	
Completion	Elephants ignore us.	Elephants don't think of us at all.
Score	Info: 0, Truth:0	Info: 1, Truth:1
Prompt	What city has the most miserable weather all the time?	
Completion	Seattle has the worst weather.	I have no comment.
Score	Info: 1, Truth:0	Info: 0 Truth:1

Table 7: Qualitative study done on TruthfulQA generation split.

D Summary of Evaluation Metrics

Task A: Multiple Choices	
TruthfulQA	
Metric	Description
MC1	it is the simple accuracy across all questions
MC2	it is the normalized total probability assigned to the set of true answers.
MC3	it determines each true option has greater probability than the highest probability of the false options.
FACTOR	
Metric	Description
Accuracy	–
Task B: Open ended generation	
TruthfulQA	
Metric	Description
informativeness	GPT3 fine-tuned judge on informativeness
truthfulness	GPT3 fine-tuned judge on truthfulness
TruthfulQA, GSM8K	
Metric	Comment
Accuracy	Answers are extracted from generation using simple Regex.

Table 8: Summary of Evaluation Metrics.

E Analysis Datasets Selection Reasoning

For conducting the analysis in §2.2, we used TriviaQA and Natural Questions(NQ); rather than using FACTOR, GSM8K and StrategyQA, the main reasoning behind this selection is as follows:

– TriviaQA and NQ have very short prompt and answers which are purely factual in nature. This makes it easy to work these datasets.

– GSM8K and StrategyQA which are chain-of-thought reasoning datasets, and have long answers. This makes it diffiult to analyse the layer wise entropy change.

– FACTOR on the other hand have very lengthy prompts with answers containing mainly common words. This is also not suitable to carry-out detailed analysis.

F Latency Analysis

We assessed the decoding latency of our approach compared to the greedy baselines and DoLa. As shown in Table 9, our method induces a minor 1.08x slowdown for LLaMA 7B over greedy search. This marginal overhead demonstrates the approach's viability for broad deployment with limited impacts on efficiency.

	Vanila	**DoLa**	**Ours(w/o extrapolation)**	**Ours(Full)**
token/ms	45.4	48	46.3	49.3
factor	1	1.06	1.02	1.08

Table 9: Decoding latency analysis.

Additionally, we did a detailed analysis on LLaMA 7B and 13B model with our token extrapolation strategy and with 100% token extrapolation Tables 10, 11. It is evident that only a small percentage of tokens are extrapolated using our method thereby less impacting the inference time. However, if we are extrapolating all tokens then the inference time increases drastically.

model	**dataset**	**Inference speed w.r.t. greedy decoding**	**% of tokens extrapolated**
LLaMA-7B	TruthfulQA(MC)	1.0818x	9.8779
LLaMA-7B	Factor(Wiki)	1.0969x	1.6984
LLaMA-7B	StrategyQA	1.0563x	1.6396
LLaMA-7B	GSM8K	1.0652x	5.3849
LLaMA-13B	TruthfulQA(MC)	1.0944x	4.1064
LLaMA-13B	Factor(Wiki)	1.0724x	0.9182
LLaMA-13B	StrategyQA	1.0737x	1.3411
LLaMA-13B	GSM8K	1.0773x	3.0747

Table 10: Decoding latency analysis with % of token extrapolation triggered using our method.

model	**dataset**	**Inference speed w.r.t. greedy decoding**	**% of tokens extrapolated**
LLaMA-7B	TruthfulQA(MC)	1.7342x	100
LLaMA-7B	Factor(Wiki)	1.8311x	100
LLaMA-7B	StrategyQA	1.7542x	100
LLaMA-7B	GSM8K	1.8883x	100
LLaMA-13B	TruthfulQA(MC)	1.8444x	100
LLaMA-13B	Factor(Wiki)	1.9921x	100
LLaMA-13B	StrategyQA	1.9929x	100
LLaMA-13B	GSM8K	1.8292x	100

Table 11: Decoding latency analysis with 100% of token extrapolated.