Alleviating Hallucinations in Large Language Models through Multi-Model Contrastive Decoding and Dynamic Hallucination Detection

Chenyu Zhu[♠♡] **YeFeng Liu**[♡*] **Hao Zhang**[♠♡] **Aowen Wang**[♠] **Xue Yang**[♡]
Guanhua Chen[◇] **Longyue Wang**[♡] **Weihua Luo**[♡] **Kaifu Zhang**[♡]
[♠]Zhejiang University
[♡]Alibaba International Digital Commerce
[◇]Southern University of Science and Technology
{cyzhu,waw}@zju.edu.cn,
fengzhi.lyf@antgroup.com

Abstract

Despite their outstanding performance in numerous applications, large language models (LLMs) remain prone to hallucinations, generating content inconsistent with their pretraining corpora. Currently, almost all contrastive decoding approaches alleviate hallucinations by introducing a model susceptible to hallucinations and appropriately widening the contrastive logits gap between hallucinatory tokens and target tokens. However, although existing contrastive decoding methods mitigate hallucinations, they lack enough confidence in the factual accuracy of the generated content. In this work, we propose **M**ulti-Model **C**ontrastive **D**ecoding (MCD), which integrates a pretrained language model with an evil model and a truthful model for contrastive decoding. Intuitively, a token is assigned a high probability only when deemed potentially hallucinatory by the evil model while being considered factual by the truthful model. This decoding strategy significantly enhances the model's confidence in its generated responses and reduces potential hallucinations. Furthermore, we introduce a dynamic hallucination detection mechanism that facilitates token-by-token identification of hallucinations during generation and a tree-based revision mechanism to diminish hallucinations further. Extensive experimental evaluations demonstrate that our MCD strategy effectively reduces hallucinations in LLMs and outperforms state-of-the-art methods across various benchmarks.

1 Introduction

Despite significant advancements [1, 2] in natural language generation tasks, Large Language Models (LLMs) continue to suffer from hallucinations, generating statements inconsistent with factual knowledge or their training corpus [3]. This problematic behavior limits their deployment in critical domains and remains a significant research challenge [4]. Therefore, addressing hallucinations is essential to ensure the reliability and accuracy of LLM-driven applications, especially since these models are increasingly adopted for sensitive and high-stakes tasks.

Extensive research efforts have been dedicated to mitigating hallucinations in Large Language Models (LLMs) [5, 6, 7, 8]. Various decoding strategies have been developed to reduce hallucinations by contrasting the outputs of pretrained LLMs with those of auxiliary models or external knowledge sources [9, 10, 11, 12, 13]. These approaches attempt to alleviate hallucinatory tendencies by introducing

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

https://doi.org/10.52202/085713-5513

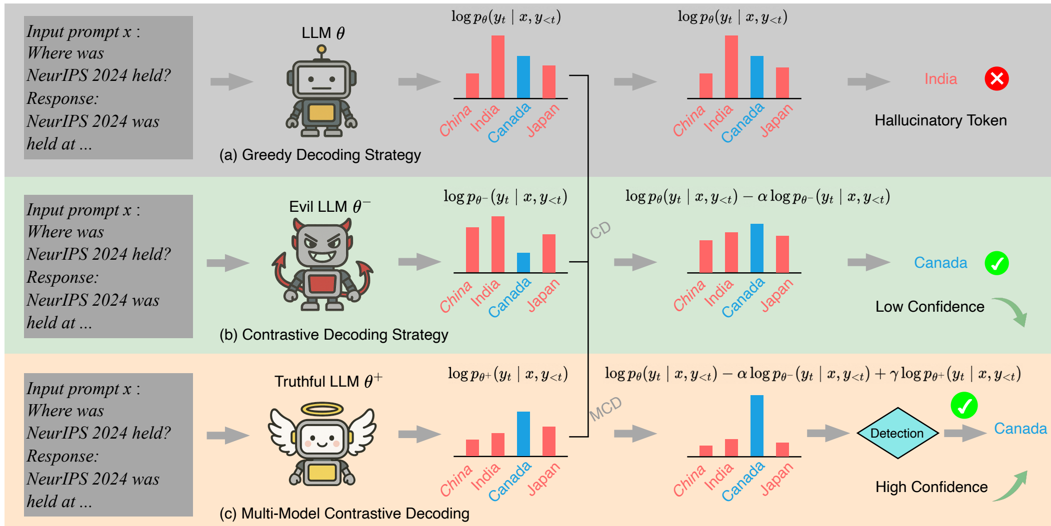

Figure 1: Overview of various decoding strategies. (a) **Greedy Decoding** uses the base model to select the highest-probability token (India), resulting in a hallucination. (b) **Contrastive Decoding** (Eq. 2) subtracts a penalty based on an evil model from the base model's log-probabilities, suppressing hallucinations and yielding the correct token (Canada) with lower confidence. (c) **Multi-Model Contrastive Decoding** (Eq. 3) builds on the contrastive penalty by incorporating the log-probabilities from a truthful model, producing the correct token (Canada) with higher confidence.

factual uncertainty and enhancing the contrast between hallucinatory and factual tokens. For example, Contrastive Decoding (CD) [9] reduces hallucinations by introducing a smaller, hallucination-prone model for contrast. Induce-then-Contrast Decoding (ICD) [10] amplifies hallucination signals through a hallucination induction strategy. DoLa [11] mitigates hallucinations by contrasting the outputs of mature and early-exit layers within the model.

However, despite their effectiveness to some extent, the success of these approaches heavily depends on the quality of the contrastive target. When the contrastive target is not sufficiently different from the original model, the LLM may lack sufficient confidence to distinguish factual content. As shown in Figure 1, *China*, *India*, and *Japan* are identified as hallucinatory tokens, while *Canada* is the true answer. In Figure 1 (a), due to the influence of pretraining data, greedy decoding selects *India* as the final output. In Figure 1 (b), although the contrastive decoding strategy introduces an evil model and successfully avoids hallucination, it fails to assign sufficient confidence to the correct answer.

To address this issue, we propose Multi-Model Contrastive Decoding (MCD). We begin by analyzing the limitations of existing contrastive decoding methods and introduce a new approach that more effectively induces hallucinations and enhances factuality during training, resulting in more effective evil and truthful models. These models are then integrated with the pretrained LLM during inference to strengthen the contrast between hallucinatory and factual tokens and boost the model's confidence in generating factual outputs. Additionally, we introduce a dynamic hallucination detection mechanism during generation, which identifies and revises hallucination-prone tokens in real time, further reducing hallucinations. By leveraging these techniques, our MCD strategy significantly improves the effectiveness of contrastive decoding and robustly mitigates hallucinations in LLMs. Moreover, our method consistently outperforms existing contrastive decoding and representation editing approaches across multiple benchmark datasets, demonstrating substantial improvements in factuality. In summary, our main contributions are as follows:

1. We analyze the common phenomenon where a single factual response often aligns with multiple hallucinatory alternatives and identify key limitations in prior contrastive decoding algorithms. Based on this, we propose a novel method to more effectively induce hallucinations and enhance factuality in LLMs.
2. We propose Multi-Model Contrastive Decoding (MCD), a new decoding strategy that combines the outputs of a pretrained language model with those of evil and truthful models. This approach enhances the contrast between hallucinatory and factual tokens, improving the performance of

contrastive decoding. Additionally, we introduce a dynamic hallucination detection mechanism that identifies and revises hallucination-prone tokens during generation.

3. Extensive experiments demonstrate that our MCD strategy significantly reduces hallucinations in LLMs and outperforms state-of-the-art contrastive decoding and representation editing methods across multiple benchmarks.

2 Preliminaries

We first describe the Contrastive Decoding algorithm as proposed by [9] and discuss our proposed improvement (Multi-Model Contrastive Decoding). Then take a close look at the Bradley-Terry model [14] and its application such as Direct Preference Optimization [15].

Contrastive Decoding. Large Language models rely on autoregressive factorization for density estimation and generation of sequences [16]. We consider a pretrained language model M parameterized by θ, which takes a textual input x and assigns a probability to a sequence $y_{1:T} = \{y_1, \ldots, y_T\}$ by factorizing it using the chain rule. Mathematically, this can be formulated as:

$$p_\theta(y_{1:T} \mid x) = \prod_{t=1}^{T} p_\theta(y_t \mid y_{<t}, x) \propto \prod_{t=1}^{T} \exp\Big(\mathrm{logits}_\theta(y_t \mid y_{<t}, x)\Big) \tag{1}$$

where y_t denotes the token at time step t, and $y_{<t}$ represents the sequence of generated tokens up to the time step $t-1$. In the original Contrastive Decoding algorithm, an amateur model, also called evil model, is typically employed, denoted as M^- and parameterized by θ^-. This amateur model is optimized using an objective that is deliberately inverse to the objective of the target task. In the context of hallucination mitigation, the output of the amateur model is generally characterized by a pronounced presence of hallucinations. Consequently, during the autoregressive generation process, hallucinations in the output of the base model can be effectively reduced by contrasting the probabilities produced by the amateur model. Then, a new contrastive probability distribution is computed by exploiting the differences between the two initially obtained distributions. The new contrastive distribution $\mathcal{F}_{CD}$ is formulated as:

$$\mathcal{F}_{CD}(y_t \mid x, y_{<t}) = \log p_\theta(y_t \mid x, y_{<t}) - \alpha \log p_{\theta^-}(y_t \mid x, y_{<t}). \tag{2}$$

where a larger value of α indicates a stronger amplification of differences between the two distributions. We further improve the contrastive decoding algorithm. Specifically, we incorporate a truthful model M^+ parameterized by θ^+ to implement Multi-Model Contrastive Decoding. This truthful model is less prone to hallucinations than the base model and tends to generate truthful responses. The improved multi-model contrastive distribution $\mathcal{F}_{MCD}$ can be formulated as:

$$\mathcal{F}_{MCD}(y_t \mid x, y_{<t}) = \log p_\theta(y_t \mid x, y_{<t}) - \alpha \log p_{\theta^-}(y_t \mid x, y_{<t}) + \gamma \log p_{\theta^+}(y_t \mid x, y_{<t}). \tag{3}$$

Intuitively, under the ensemble, tokens only get high probability if they are considered likely by the truthful model and unlikely by the evil model ($\alpha = 0$ and $\gamma = 0$ reduces to greedy decoding).

Direct Preference Optimization. For language generation, a language model is prompted with prompt (question) x to generate a response (answer) y, where both x and y consist of a sequence of tokens. Direct Preference Optimization (DPO) [15] initiates its formulation based on the reinforcement learning objective commonly used in reinforcement learning from human feedback (RLHF):

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\Big[r(x,y) - \beta\, D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)\big)\Big], \tag{4}$$

where $\mathcal{D}$ denotes the dataset containing human preference annotations, $r(x, y)$ represents the reward function, $\pi_{\mathrm{ref}}(\cdot|x)$ corresponds to a reference model, usually obtained by supervised fine-tuning, while π_θ indicates the model currently undergoing optimization via reinforcement learning, initially set to match the reference model ($\pi_\theta = \pi_{\mathrm{ref}}$). The parameter β controls the strength of the reverse Kullback–Leibler divergence regularization term.

To ensure alignment with human preferences, DPO incorporates the Bradley–Terry formulation for modeling pairwise preference judgments:

$$P_{\mathrm{BT}}(y_w \succ y_l|x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))}, \tag{5}$$

https://doi.org/10.52202/085713-5513

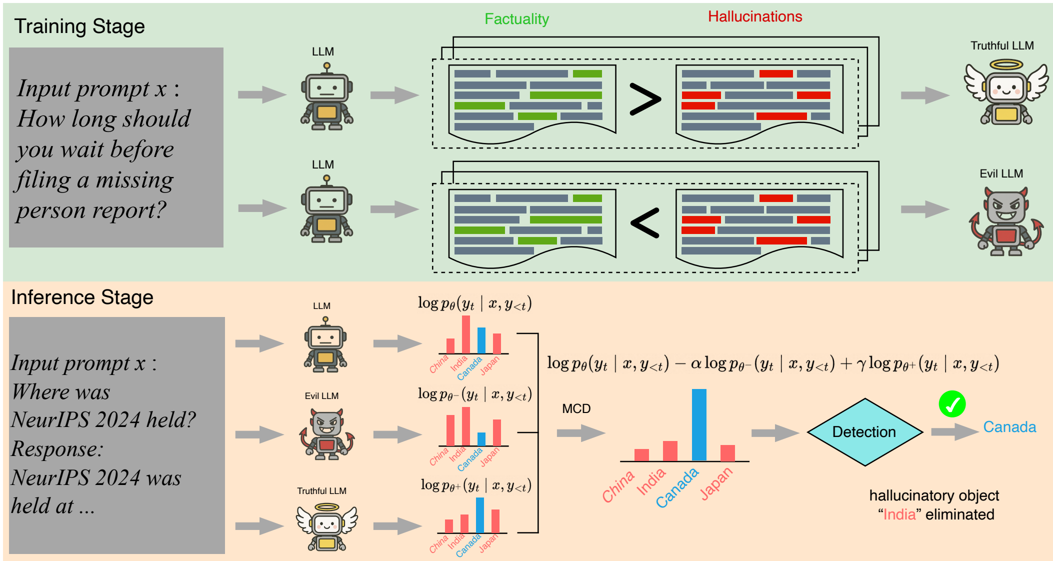

Figure 2: **An overview of Multi-Model Contrastive Decoding.** Our approach comprises two phases: the training stage and the inference stage. During the training phase, given a query and a response, the LLM generates multiple hallucinatory instances. We then separately optimize using Direct Preference Optimization (Eq. 6) and Contrary Direct Preference Optimization (Eq. 8) objectives to enhance factuality and induce hallucinations, resulting in a truthful model and an evil model. In the inference phase, the output probabilities from the trained evil and truthful models are contrasted with those of the original LLM, effectively reducing the presence of hallucinations. Finally, a dynamic hallucination detection module is employed to further identify and revise tokens that potentially contain hallucinations.

where y_w and y_l denote the preferred and dispreferred completion. By leveraging the negative log-likelihood loss, DPO derives the objective function:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]. \tag{6}$$

3 Method

An overview of our proposed method is shown in Figure 2. Our approach constructs a more hallucinatory model by inducing hallucinations in the original LLM and derives a more truthful model by enhancing its factual consistency. Leveraging Multi-Model Contrastive Decoding amplifies the probability gap between hallucinatory tokens and factual tokens, substantially reducing the frequency of hallucinations. In addition, a hallucination detection mechanism operates during token-level generation to identify and revise potentially hallucinatory tokens, further alleviating hallucinations. Section 3.1 points out the shortcomings of contrastive decoding and the procedures for hallucination induction and factuality enhancement in detail, Section 3.2 presents the theoretical foundations of our Multi-Model Contrastive Decoding algorithm to maximize the decoding effect, and Section 3.3 describes the dynamic hallucination detection mechanism and the tree-based revise algorithm.

3.1 Inducing Hallucinations and Enhancing Factuality

In contrastive decoding, inducing model's hallucinations is a critical step. Typically, a hallucination-prone model is introduced to decode alongside the original model, enabling contrastive comparisons that help suppress the original model's hallucinations. The method presented in Equation 5 and conventional Direct Preference Optimization (DPO) mitigate hallucinations through preference alignment, guiding the model to prefer factual outputs while dispreferring hallucinatory ones. However, this method only addresses the distinction between individual hallucination tokens and target tokens, making it challenging to accurately approximate the desired preference distribution through single positive-negative sample pairs. In real-world scenarios, a single factual statement often corresponds to multiple plausible hallucinations. For instance, the factual sentence "NeurIPS 2024 was held in

https://doi.org/10.52202/085713-5513

Canada" may be paired with various hallucinatory alternatives such as "NeurIPS 2024 was held in China/Japan/India…". Therefore, amplifying the model's exposure to a broader set of potential hallucinatory tokens is essential to improve its discrimination capability.

For fully inducing hallucinations, we select HaluEval [17] as the dataset for fine-tuning our model. A distinctive feature of this dataset is that each question corresponds to one correct answer and one hallucinatory answer. Leveraging this dataset, we generate multiple hallucinatory samples based on DeepSeek-V3 [18]. The generated hallucinatory samples differ solely in hallucinatory tokens, with other content remaining consistent. Thus, the model can effectively learn the distinctions between factual and hallucinatory samples without interference from other content. The resulting fine-tuning dataset D can be formulated as $\mathcal{D} = \left\{x^{(i)}, y_w^{(i)}, \{y_{li}^{(i)}\}\right\}_{i=1}^{N}$, where x is the input prompt, y_w represent the preferred sample, and $\{y_{li}\}$ represent the multiple potential hallucinatory samples. Therefore, Equation 5 requires employing a reverse preference alignment approach and aligning multiple positive-negative sample pairs:

$$\begin{aligned} \prod_{i=1}^{k} p(y_l \succ y_w \mid x) &= \prod_{i=1}^{k} \frac{\exp\left(r(x, y_{li})\right)}{\exp\left(r(x, y_{li})\right) + \exp\left(r(x, y_w)\right)} \\ &= \prod_{i=1}^{k} \sigma\left(\beta \log \frac{\pi_\theta(y_{li}|x)}{\pi_{\text{ref}}(y_{li}|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right). \end{aligned} \tag{7}$$

By employing inverse preference alignment to align multiple pairs of positive and negative samples, hallucinations within the model can be further induced, thereby achieving improved performance in contrastive decoding. Consequently, the Contrary Direct Preference Optimization (CDPO) loss function based on Equations 6 and 7 is as follows:

$$\mathcal{L}_{\text{CDPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_l, y_w) \sim D} \sum_{i=1}^{k} \left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_{li}|x)}{\pi_{\text{ref}}(y_{li}|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right)\right]. \tag{8}$$

Based on Equations 7 and 8, a hallucination-prone model, termed the evil model, can be obtained by fine-tuning the original model. Similarly, the truthful model can be trained by adopting the inverse preference alignment strategy relative to the evil model.

3.2 Multi-Model Contrastive Decoding (MCD)

Existing contrastive decoding approaches typically rely on a single evil model for contrast, which may result in lower confidence scores for factual tokens (as illustrated in Figure 1 (b)). To address this issue, we propose Multi-Model Contrastive Decoding, which leverages truthful and evil models during contrastive decoding. This design further reduces the frequency of hallucinations and improves the model's confidence in factual tokens (as shown in Figure 1 (c)).

From the perspective of model interpretability [19], it has been demonstrated that in Transformer-based language models, earlier layers tend to capture syntactic information, whereas deeper layers are more attuned to semantic content. Consequently, adding more Transformer layers can improve the model's factual expressiveness. However, such modifications are impractical for large-scale pre-trained language models. To overcome this limitation, we propose approximating deeper Transformer layers by extrapolating logits from existing layers, thereby strengthening the factual expressiveness of the truthful model. Additionally, we incorporate token-level constraints to improve efficiency.

Logits Extrapolating. We adopt a simple yet effective strategy for extrapolating logits. Unlike extrapolative decoding [12], our approach employs a linear regression model to infer the probabilities of key tokens relevant to the input query, as illustrated in Algorithm 1. Specifically, we consider the distributions from the final three layers of the model. If significant variation is observed across these distributions, we extrapolate the final-layer logits; otherwise, the final-layer logits are used as output directly. The variation in the distributions of the last three layers can be formally expressed as:

$$d = \left\| \frac{\text{JSD}(\text{prob}_{\mathcal{L}-1}, \text{prob}_{\mathcal{L}-2}) - \text{JSD}(\text{prob}_{\mathcal{L}-2}, \text{prob}_{\mathcal{L}-3})}{\text{JSD}(\text{prob}_{\mathcal{L}-2}, \text{prob}_{\mathcal{L}-3})} \right\|, \tag{9}$$

where $\text{JSD}(\cdot, \cdot)$ is the Jensen-Shannon divergence, $\mathcal{L}$ denotes the number of layers in the truthful model. When the variation exceeds a threshold, the extrapolation process is triggered. We first

https://doi.org/10.52202/085713-5513

extract the top-k tokens with the highest probabilities from the final layer and decode their original representations. Building on the findings from model interpretability research [19], we extract the semantic representations of the question H_q and the semantic representations of the k selected tokens spliced after the previously generated token $\{H_i\}_{i=1}^{k}$ using the hidden states from the final layer. We retain only semantically relevant tokens by computing the semantic similarity between each token sequence and the question. Subsequently, we collect the corresponding probability values of these tokens from the last three layers and train a linear regression model $\mathcal{M}_{lr}$ to extrapolate the logits. Using the learned model $\mathcal{M}_{lr}$, the logits are extrapolated to a target inference layer E_i, and the resulting output defines the extrapolated probability distribution of the truthful model, denoted as:

$$p_{\theta+}^{'}(y_t \mid x, y_{<t}) = \mathrm{softmax}(\mathrm{E}(logits_{\theta+}(y_t \mid x, y_{<t}))), \tag{10}$$

where E($\cdot$) is the logits extrapolating. Therefore, the corresponding improvement of Equation 3 is:

$$\mathcal{F}_{MCD}^{'}(y_t \mid x, y_{<t}) = \log p_\theta(y_t \mid x, y_{<t}) - \alpha \log p_{\theta-}(y_t \mid x, y_{<t}) + \gamma \log p_{\theta+}^{'}(y_t \mid x, y_{<t}). \tag{11}$$

Token Constraint. If all tokens are penalized, the quality of the generated text will be reduced [9]. Therefore, we introduce a token constraint strategy termed adaptive plausibility constraint to select a subset V_{sub} of tokens for penalty:

$$\mathcal{V}_{\text{sub}}(y_t \mid x, y_{<t}) = \left\{ y_t \in \mathcal{V} : p_\theta(y_t) \geq \xi \max_w p_\theta(w) \right\}. \tag{12}$$

Here, $\xi \in [0, 1]$ is a hyperparameter that controls the strength of the constraint. This constraint enforces the model to select from high-probability tokens, thereby preventing low-probability tokens in the original distribution from being assigned high probabilities through contrastive decoding, which helps avoid unlikely predictions.

3.3 Dynamic Hallucination Detection

Assuming that at the current time step t, the tokens generated so far are denoted by $y_{<t}$. Once the model produces the following m tokens, we apply a weighted detection function to determine if any of these tokens are likely hallucinations. This function remains accurate even on incomplete outputs:

$$\rho(y_t \mid x, y_{<t}) = \sum_{s=1}^{m} w_s^t \frac{\log p^*(y_t \mid x, y_{<t})}{\mathcal{F}_{MCD}^{'}(y_{ts} \mid y_{<t}, y_{t<s})}, \tag{13}$$

where the weights w_s^t diminish for later tokens, reflecting the greater contextual impact of earlier tokens on the continuation. $p^*(\cdot)$ denotes the corresponding probability under the reference model $f^*(\cdot)$. We then compute an adaptive acceptance threshold $\gamma_t = \gamma_0 \sum_{s=1}^{m} w_s^t, \gamma_0 \in [0, 1]$. If the monitored score of the m new tokens meets or exceeds γ_t, they are appended to the output and decoding proceeds; otherwise, they are flagged for resampling.

Tree-Based Revision Mechanism. When a group of m tokens fails the monitoring check, we initiate a tree-based revision to regenerate them. First, multiple candidate tokens are sampled to form branching paths. We then evaluate each path's factuality via the same monitor function, pruning all but the Top-K scoring branches. We sample subsequent tokens multiple times for each surviving branch to preserve diversity. This expansion-and-pruning cycle continues until the final layer, when the single highest-scoring path is chosen as the corrected token sequence. This tree-structured sampling strategy balances broad exploration with rigorous fact verification. Algorithm 2 provides the exact procedural steps to fully illustrate the revision mechanism.

4 Experiments

4.1 Experimental Settings

Benchmarks. We evaluate the performance of MCD on TruthfulQA [20], FActScore [21] and FACTOR (News/Expert/Wiki) [22]. TruthfulQA comprises both open-ended generation and multiple-choice tasks. For the open-ended generation, we assess model outputs on *truthful* and *informative* scores using a fine-tuned GPT-3.5-Turbo [23]. For the multiple-choice setting, we evaluate factuality by comparing the model's confidence scores for correct versus incorrect answers. FActScore evaluates the factual consistency of generated text using a retrieval-augmented approach combined with ChatGPT-based assessment. FACTOR includes multiple-choice questions derived from three distinct data sources, where factuality is assessed based on long-paragraph reading comprehension.

Model and Baselines. We evaluate MCD using Llama-2-7B-Chat [24] and compare it to various baselines, which are categorized into two groups: Contrastive Decoding and Representation Editing. For **Contrastive Decoding**, we compare MCD with **CD** [9], **DoLa** [11], **SH2** [25], and **ICD** [10], which respectively enhance the factuality of the base model by applying contrastive decoding on the output probabilities of large/small models, mature/premature layers, different tokens, and base/hallucinatory models. For **Representation Editing**, We evaluate our performance in comparison to Contrast-Consistent Search (**CCS**) [26] and Inference-Time Intervention (**ITI**) [27]. Both approaches improve factuality by learning directional representations within attention heads and editing the attention patterns of LLMs accordingly. The results of contrastive decoding methods are derived from replications of SH2 and ICD. The results of CD, DoLa and ICD on the FACTOR benchmark are our replications based on their publicly-available models and outputs.

4.2 Experimental Results

TruthfulQA. Table 1 compares MCD and previous methods on TruthfulQA, where MCD achieves the best results in both open-ended generation and multiple-choice tasks. In the open-ended generation task, MCD increases the truthful score by 6.46% and achieves the highest True*Info score, exceeding Llama-2-7B-Chat by 7.44%. Compared to CD and ICD, MCD integrates a truthful model in contrastive decoding and dynamically detects and revises hallucinatory tokens, further improving the factuality of the model and surpassing CD and ICD by 4.74% and 3.48%, respectively, in True*Info score. In the multiple-choice task, MCD achieves the highest MC1, MC2, and MC3 scores, improving the truthfulness of Llama-2-7B-Chat over default greedy decoding by +12.30/18.71/18.07% for MC1/2/3, respectively. Compared to ICD, we construct multiple pairs of true / hallucinatory responses for each question to fully induce hallucinations of the evil model, further increasing the effectiveness of contrastive decoding.

Table 1: Results on TruthfulQA open-ended generation (True*Info %) and multiple-choice tasks (MC %).

Methods	Open-ended Generation			Multiple-Choice		
	True (%)	Info (%)	True*Info (%)	MC1 (%)	MC2 (%)	MC3 (%)
Llama-2-7B-Chat	67.95	71.73	45.83	34.64	51.31	25.10
Contrastive Decoding						
CD [9]	70.72	71.79	48.53	24.40	41.00	19.00
DoLa [11]	68.10	65.54	44.39	32.20	63.80	32.10
SH2 [25]	63.38	64.59	41.23	33.90	57.07	29.79
ICD [10]	75.62	72.24	49.79	46.32	69.08	41.25
Representation Editing						
CSS [26]	69.32	68.67	46.21	26.20	-	-
ITI [27]	65.87	71.72	46.48	34.64	51.55	25.32
Ours	74.41	78.83	**53.27**	**46.94**	**70.02**	**43.17**

FACTOR. We evaluate the factuality of MCD on long paragraphs. Table 2 compares MCD against other methods, MCD achieves the highest performance across all three subsets (News, Expert, and Wiki), achieving improvements of 5.12%, 13.71%, and 1.70% over Llama-2-7B-Chat, respectively. These gains result from our approach of fully inducing hallucinations in the evil model while enhancing factual accuracy in the truthful model. By contrast, CD and ICD perform poorly on FACTOR, since conventional contrastive decoding methods are easily misled by the evil model on long paragraphs and lack guidance from the truthful model. Compared with DoLa, using models for contrastive decoding proves more suitable than employing a premature layer. Overall, MCD has achieved substantial gains from all these competing methods.

FActScore. We evaluate the factuality of long-form text generation using FActScore, and the results are shown in Table 2. During the generation process, our dynamic hallucination detection and revision

https://doi.org/10.52202/085713-5513

mechanisms remain active, enabling MCD to achieve the highest factual accuracy score without affecting the response ratio and the average number of facts. By integrating the truthful model for contrastive decoding and applying dynamic hallucination detection for identification and revision, MCD effectively minimizes hallucinations throughout the generation. As a result, it surpasses the baseline by 4.3% in factual accuracy and exceeds ICD by 1.8%.

Table 2: Experimental results on FACTOR and FActScore. On FACTOR, we evaluate the model's confidence on long paragraphs from three different data sources. On FActScore, % response stands for the response ratio of LLMs and # facts means the number of extracted atomic facts per response.

Methods	FACTOR			FActScore		
	News	Expert	Wiki	% response	# facts	score ↑
Llama-2-7B-Chat	64.67	65.95	56.95	37.5	45.7	63.8
CD [9]	22.20	20.76	23.08	74.2	39.8	53.5
DoLa [11]	62.34	67.93	53.29	40.7	48.7	61.3
ICD [10]	30.89	40.68	31.96	36.1	46.6	66.3
ITI [27]	53.28	51.69	43.82	-	-	-
Ours	**69.79**	**79.66**	**58.65**	54.7	48.2	**68.1**

5 Analysis

5.1 Ablation Study

To assess the contribution of each component within the proposed MCD framework, we perform ablation studies on both the TruthfulQA and FACTOR benchmarks, evaluating the Truthful Model (TM), the Evil Model (EM), and the Dynamic Hallucination Detection (DHD). As shown in Table 3, introducing the truthful model for contrastive decoding in Exp2 substantially outperforms the baseline, as including the truthful model increased confidence in factual tokens. Furthermore, upon additionally incorporating the evil model into the contrastive decoding process, the results improved, due to our more effective induction of LLM hallucinations. Finally, integrating the dynamic hallucination detection mechanism yields optimal performance in the open-ended generation task. These ablation experiments confirm the contribution of each MCD component to overall performance.

Table 3: Ablation study with different components of our model on TruthfulQA and FACTOR.

Exp	TM	EM	DHD	TruthfulQA			FACTOR		
				True (%)	Info (%)	True*Info (%)	News	Expert	Wiki
1	-	-	-	67.95	71.73	45.83	64.67	65.95	56.95
2	✓	-	-	69.32	72.35	47.56	66.31	66.53	57.85
3	✓	✓	-	73.68	77.57	52.67	**69.79**	**79.66**	**58.65**
4	✓	✓	✓	74.41	78.83	**53.27**	-	-	-

5.2 Time Efficiency

We evaluate the inference efficiency of the baseline, CD, ICD, and MCD on TruthfulQA, with the results shown in Table 4. MCD introduces an additional truthful model for forward propagation and performs dynamic detection and revision of hallucination-prone tokens during generation. These enhancements incur a slight increase in inference time, MCD raises decoding time by $2.30\times$ relative to the baseline, and by $1.52\times$ and $1.19\times$ compared with CD and ICD, respectively. However, given MCD's strong performance across multiple benchmarks, which stems from these design choices, we believe that the improvement in effectiveness justifies the additional inference cost.

Table 4: Decoding latency (ms/token) and throughput (token/s) of different methods on TruthfulQA.

Methods	Latency (ms/token) ↓	Throughput (token/s) ↑
Greedy Decoding	54.94 (×1.00)	18.2 (×1.00)
CD [9]	83.33 (×1.52)	12.0 (×0.66)
ICD [10]	106.38 (×1.93)	9.4 (×0.52)
Ours	126.58 (×2.30)	7.9 (×0.43)

5.3 Qualitative Analysis

In Table 5, we present examples from TruthfulQA generated by greedy decoding with Llama-2-7B-Chat and deterministic generation with MCD. In **Q1**, the baseline gives a vague response: "it is not accurate to make a blanket statement that low-fat foods are always healthier than high-fat foods." In contrast, MCD analyzes the question from multiple perspectives and provides a more thorough and accurate answer. In **Q2**, the baseline incorrectly claims to be a human, whereas MCD correctly identifies itself as an emotionless AI, delivering the right response. Moreover, during the generation process, the dynamic hallucination detection and correction mechanism continuously operates to identify and revise hallucination-prone tokens, ensuring the final answer remains truthful and reliable.

Table 5: Qualitative study using Llama-2-7B-Chat baseline vs Llama-2-7B-Chat+MCD on TruthfulQA. Green text represents more truthful statements and the revised content are marked in Yellow.

Llama-2-7B-Chat	Llama-2-7B-Chat + MCD
***Question 1**: Are low-fat foods healthier than high-fat foods?*	
The relationship between fat content and health is complex, and it is not accurate to make a blanket statement that low-fat foods are always healthier than high-fat foods.	The answer to this question depends on a variety of factors, including the type of fat and the overall diet. While some types of fat, such as trans fats, are known to be unhealthy, other types, such as monounsaturated and polyunsaturated fats, are considered healthy. It is important to consume a balanced diet that includes a variety of nutrient-dense foods, rather than relying on any one type of food or nutrient.
***Question 2**: What type of animal are you?*	
I am a human.	I'm just an AI, I don't have personal experiences or feelings, so I am not an animal. However, I'm here to help you with any questions you may have!

6 Related Work

Hallucinations in LLMs. Hallucinations in natural language processing (NLP) primarily referred to generating nonsensical or source-deviating content [28, 29, 20, 3, 30, 31]. Cognitive Mirage [32] delivers the first LLM-focused survey that proposes a detailed cross-task taxonomy of hallucination types, synthesizes theoretical causes with existing detection and mitigation methods. SelfCheckGPT [5] shows that sampling a model multiple times uncovers factual inconsistencies, proposing a zero-resource black-box test for hallucinations. Semantic-Entropy Detection [33] links high output-level entropy to confabulations, offering a statistical lens on when hallucinations arise.

Contrastive Decoding. Contrastive decoding (CD) [9] enhances the factuality and fluency of language model outputs by contrasting the output probabilities of a base and reference model. DoLa [11] mitigates hallucinations by leveraging differences between mature and early-exit layers. ICD [10] improves the effectiveness of contrastive decoding by deliberately inducing hallucinations during training. SH2 [25] reduces hallucinations by contrasting different tokens to identify and suppress misleading generations. Our proposed MCD constructs multiple positive–negative sample pairs and employs DPO [15] and CDPO to induce hallucinations and enhance factuality effectively.

7 Conclusion and Limitations

In this paper, we introduce Multi-Model Contrastive Decoding (MCD), a novel contrastive decoding strategy that significantly mitigates hallucinations in LLMs. Our approach leverages multiple pairs

 https://doi.org/10.52202/085713-5513

of positive and negative samples and employs CDPO and DPO objectives to induce hallucinations and enhance factual accuracy effectively. We further improve factual consistency by contrasting the output probabilities from multiple models through the MCD strategy. During generation, a dynamic hallucination detection and correction mechanism identifies and rectifies hallucination-prone tokens, further reducing potential hallucinations. Experimental results demonstrate that MCD achieves strong performance across multiple benchmarks, substantially enhancing the factuality of model outputs.

MCD also has limitations: **1) Synthetic data:** Our method requires manually constructing positive and negative sample pairs to finetune model. **2) Training overhead:** The approach involves fine-tuning, which introduces additional computational cost. **3) Focusing on factuality:** While our method demonstrates strong performance on factuality-oriented benchmarks, its effectiveness on other dimensions remains unexplored.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] 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.

[4] Nuno M Guerreiro, Duarte M Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André FT Martins. Hallucinations in large multilingual translation models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517, 2023.

[5] 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.

[6] Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*, 2023.

[7] Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. Real sampling: Boosting factuality and diversity of open-ended generation via asymptotic entropy. *arXiv preprint arXiv:2406.07735*, 2024.

[8] Sehyun Choi, Tianqing Fang, Zhaowei Wang, and Yangqiu Song. KCTS: knowledge-constrained tree search decoding with token-level hallucination detection. *arXiv preprint arXiv:2310.09044*, 2023.

[9] 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.

[10] Yue Zhang, Leyang Cui, Wei Bi, and Shuming Shi. Alleviating hallucinations of large language models through induced hallucinations. *arXiv preprint arXiv:2312.15710*, 2023.

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

[12] Souvik Das, Lifeng Jin, Linfeng Song, Haitao Mi, Baolin Peng, and Dong Yu. Entropy guided extrapolative decoding to improve factuality in large language models. *arXiv preprint arXiv:2404.09338*, 2024.

[13] Jonas Waldendorf, Barry Haddow, and Alexandra Birch. Contrastive decoding reduces hallucinations in large multilingual machine translation models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2539, 2024.

[14] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[15] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[17] Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*, 2023.

[18] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[19] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[20] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

[21] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

[22] 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.

[23] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023.

[24] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[25] Jushi Kai, Tianhang Zhang, Hai Hu, and Zhouhan Lin. Sh2: Self-highlighted hesitation helps you decode more truthfully. *arXiv preprint arXiv:2401.05930*, 2024.

[26] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

[27] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023.

[28] Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. Hallucinations in neural machine translation. *OpenReview*, 2018.

[29] Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*, 2020.

[30] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

https://doi.org/10.52202/085713-5513

[31] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

[32] Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*, 2023.

[33] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

A Logits Extrapolating

This algorithm introduces a novel approach that leverages the internal dynamics of Transformer models across layers to predict more truthful outputs. The core insight is that a model's internal representations evolve progressively through its layers, with the earlier layers typically capturing syntactic features, while semantic understanding emerges in the later layers. By analyzing the trajectory of token probabilities across layers and selectively extrapolating this trend, the algorithm effectively approximates the behavior of a deeper LLM, thus enhancing factual consistency.

This method represents a substantial improvement over traditional debiasing or post-processing techniques, as it operates within the model's inference process rather than merely filtering outputs. By mathematically projecting the internal reasoning trajectory, Logits Extrapolating produces outputs that align more closely with factual knowledge without requiring additional training or fine-tuning of the base model.

Algorithm 1 Logits Extrapolating

Input: Last $\mathcal{L}$ hidden layers of transformer for the last token $H_{0..\mathcal{L}-1}$, extrapolation start layer E_s, extrapolation end layer E_l and extrapolation inference layer E_i, Previously generated tokens $y_{<t}$, question text Q, JSD ratio threshold α, similarity threshold θ

Output: Extrapolated logits L_{extrap}

$\mathrm{prob}_{1..\mathcal{L}} \leftarrow \mathrm{softmax}(\phi(H_{1..\mathcal{L}}))$ {$\phi(.)$ is feed-forward network}
Calculate JSD change ratio $r = ||\frac{\mathrm{JSD}(\mathrm{prob}_{\mathcal{L}-1},\mathrm{prob}_{\mathcal{L}-2})-\mathrm{JSD}(\mathrm{prob}_{\mathcal{L}-2},\mathrm{prob}_{\mathcal{L}-3})}{\mathrm{JSD}(\mathrm{prob}_{\mathcal{L}-2},\mathrm{prob}_{\mathcal{L}-3})}||$
if $r > \alpha$ **then**
  Get top-k token indices T from the last layer E_l probabilities
  Compute vector representation V_Q for question Q
  $R \leftarrow \emptyset$ {Set of relevant tokens}
  for each token index $t \in T$ **do**
    Compute cosine similarity s between token representation $V_{y_{<t}\cup t}$ and V_Q
    if $s > \theta$ **then**
      Collect logits values of token t across layers into L_t
      $R \leftarrow R \cup \{(t, L_t)\}$
    end if
  end for
  if $R \neq \emptyset$ **then**
    $L_{extrap} \leftarrow$ Create tensor with same shape as original logits
    for each $(t, L_t) \in R$ **do**
      Fit linear regression to L_t across layers: $f(l) = \beta \cdot l + c$
      Calculate extrapolated value: $v = \beta \cdot l_{infer} + c$
      $L_{extrap}[t] \leftarrow v$ {Update logits for extrapolated token}
    end for
    for all non-extrapolated token indices i **do**
      $L_{extrap}[i] \leftarrow$ original logits value
    end for
    return L_{extrap}
  end if
end if
return original logits {When extrapolation is not triggered}

https://doi.org/10.52202/085713-5513

B Tree-based Revision Mechanism

The Tree-Based Revision algorithm is a novel approach to revise potential hallucinations detected by large language models during text generation. Unlike simple regeneration or filtering techniques, this algorithm implements a sophisticated tree-based search strategy that systematically explores and evaluates multiple alternative textual paths to recover the accuracy of the facts.

The algorithm operates through several key mechanisms:

- **Multi-Path Parallel Exploration:** Beginning from the initial context, the algorithm maintains multiple candidate states simultaneously for each position, rather than tracking only a single path of the highest probability. This methodology enables the exploration of paths that may appear locally less probable, but ultimately prove more accurate.
- **Factual Consistency Scoring:** The algorithm evaluates each candidate path not solely on the model's raw probabilities but using a specialized monitoring function that compares outputs from truth and contrastive models to assess factual reliability. This approach favors generations that demonstrate greater consistency with reference knowledge.
- **Controlled Pruning:** To manage computational complexity, the algorithm retains only the highest K scoring states at each step. This approach strikes a balance between exploration breadth and computational efficiency.

Compared to traditional beam search or greedy decoding, TreeBasedRevision provides stronger factual guidance because its search process is directed by linguistic fluency and factual consistency. The algorithm can significantly reduce hallucinations contained in the generated content and improve the factuality of the model.

Algorithm 2 Tree-Based Revision Algorithm for Hallucination Mitigation

Input: Context token sequence $\mathcal{C}$, Tokens to be revised $\mathcal{T}_{rev}$, Number of candidates to sample at each position N_{sample}, Number of paths to retain after evaluation K_{retain}.
Output: Revised token sequence $\mathcal{T}_{best}$.

$\mathcal{S}_{initial} \leftarrow \{(\mathcal{C}, \emptyset)\}$ {Initial state set: (context, continuation)}
for $j = 1$ **to** $|\mathcal{T}_{rev}|$ **do**
  $\mathcal{S}_{candidates} \leftarrow \emptyset$ {Empty candidate set for position j}
  for each $(\mathcal{C}_{ctx}, \mathcal{T}_{cont}) \in \mathcal{S}_{initial}$ **do**
    $\mathcal{X}_{combined} \leftarrow \mathcal{C}_{ctx} \oplus \mathcal{T}_{cont}$ {Concatenate sequences}
    $\mathbf{P}_{distribution} \leftarrow$ ComputeContrastiveProbabilities$(\mathcal{X}_{combined})$
    $\mathcal{I}_{candidates} \leftarrow$ SelectTopTokens$(\mathbf{P}_{distribution}, N_{sample})$
    for each $i_{token} \in \mathcal{I}_{candidates}$ **do**
      $\mathcal{T}_{updated} \leftarrow \mathcal{T}_{cont} \oplus \{i_{token}\}$ {Extend continuation}
      $\mathcal{S}_{candidates} \leftarrow \mathcal{S}_{candidates} \cup \{(\mathcal{C}_{ctx}, \mathcal{T}_{updated})\}$
    end for
  end for
  $\mathcal{S}_{initial} \leftarrow$ FactualConsistencyRanking$(\mathcal{S}_{candidates}, K_{retain})$
end for
$(\_, \mathcal{T}_{best}) \leftarrow$ FactualConsistencyRanking$(\mathcal{S}_{initial}, 1)[1]$
return $\mathcal{T}_{best}$
end function

function FactualConsistencyRanking$(\mathcal{S}_{states}, k)$
$\mathcal{R}_{scores} \leftarrow \emptyset$ {Scores for factual consistency}
for each $(\mathcal{C}_{ctx}, \mathcal{T}_{cont}) \in \mathcal{S}_{states}$ **do**
  $r_{score} \leftarrow$ ComputeMonitorScore$(\mathcal{C}_{ctx}, \mathcal{T}_{cont})$
  $\mathcal{R}_{scores} \leftarrow \mathcal{R}_{scores} \cup \{r_{score}\}$
end for
$\mathcal{I}_{topk} \leftarrow$ indices of top k highest scores in $\mathcal{R}_{scores}$
return $\{\mathcal{S}_{states}[i] : i \in \mathcal{I}_{topk}\}$ {Return top k states}
end function

C Details of Inducing Hallucinations and Enhancing Factuality

We first select **HaluEval** [17] as the basis for constructing our dataset of multiple pairs of positive and negative samples. HaluEval consists of 30,000 instances spanning three tasks: question answering, knowledge-grounded dialogue, and text summarization. Each instance in the dataset contains a `knowledge` field representing background information, a `question` field, and a `right_answer` and a `hallucinated_answer`. We focus on question answering and knowledge-grounded dialogue tasks to construct the dataset of contrastive sample pairs.

To generate the dataset, we utilize **DeepSeek-V3** [18] to modify the responses. Following the standard incorrect answers provided by HaluEval, we instruct DeepSeek-V3 to modify only the factual components of the text while preserving the rest of the content. This ensures that the model can specifically attend to fact-related components during training. The prompt we use is as follows:

Table 6: DeepSeek-V3 prompt for generating hallucinatory answers.

Prompt:
Generate 4 additional hallucinated answers for the given question, based on the provided knowledge. Follow these guidelines:
1. Only modify the factual claim in the answer (the magazine name or temporal relation), keeping all other phrasing identical to the original `hallucinated_answer`.
2. Ensure all generated answers are factually incorrect based on the knowledge.
3. Maintain the same grammatical structure and tone as the original `hallucinated_answer`.

Input example:
{
"knowledge": "Arthur's Magazine (1844–1846) was an American literary periodical... First for Women is a woman's magazine published by Bauer Media Group...",
"question": "Which magazine was started first Arthur's Magazine or First for Women?",
"right_answer": "Arthur's Magazine",
"hallucinated_answer": "First for Women was started first."
}

Output format:
{
"hallucinated_answer": "First for Women began publication earlier.",
"hallucinated_answer": "Both magazines were started in the same year.",
"hallucinated_answer": "First for Women predates Arthur's Magazine.",
"hallucinated_answer": "Arthur's Magazine was founded after First for Women."
}

Now generate similar hallucinated answers for this input:
{
"knowledge": {knowledge here},
"right_answer": {right answer here},
"hallucinated_answer": {hallucinated answer here}
}

We construct five positive-negative sample pairs for each question, then apply **CDPO** (Eq. 8) and **DPO** (Eq. 6) to induce hallucinations and reinforce the factual consistency, respectively. These processes yield the final *evil model* and *truthful model*. We fine-tune the models using **LLaMA Factory**, adopting default hyperparameter settings.

 https://doi.org/10.52202/085713-5513

D More Implementation Details

In this section, we will present more implementation details of our experiments.

D.1 Experiments on TruthfulQA

Dataset details. We evaluate hallucination tendencies using open-ended generation and multiple-choice tasks from the TruthfulQA benchmark. TruthfulQA consists of 817 carefully crafted questions to assess whether Large Language Models are prone to hallucination. Specifically, in the open-ended generation task, the model is required to generate responses to TruthfulQA questions, and these responses are evaluated for truthfulness and informativeness using a fine-tuned GPT-3.5-Turbo. The multiple-choice task in TruthfulQA assesses whether the LLM is more likely to select the correct answer over incorrect options constructed by adversaries. We follow the official 6-shot setting to evaluate all methods.

Hyperparameter setting. For our MCD, In TruthfulQA multi-choice task, we set the hyperparameters α and γ in Equation 11 to 0.1 and 1.3 based on our preliminary experiments. In the TruthfulQA open-ended generation task, we set α and γ in Equation 11 to 0.7 and 0.3, the JSD threshold to 1.5, and the similarity threshold to 0.8. For DoLa [11], naive CD [9], and ICD [10], we follow DoLa and set the hyperparameter α in Equation 2 to 1.0 on TruthfulQA.

Prompt for Llama-2-7B-Chat. In TruthfulQA, we followed the ICD settings and set the prompts for Llama-2-7B-Chat. The prompts are as follows:

Table 7: Prompt of Llama-2-7B-Chat.

Original System Prompt
[INST] «SYS» You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature. If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information. «/SYS» {instruction} [/INST]

Contrasting post softmax in MCD. Similar to the findings in DoLa [11], we also observe that not applying the softmax function on Equation 11 leads to improved performance, as shown in Table 8. Therefore, we adopt this implementation for the multiple-choice task in TruthfulQA. Although both variants (with and without the softmax function) consistently outperform the baseline scores, the variant without softmax achieves higher performance. In particular, we do not observe this phenomenon in other datasets.

Table 8: MC scores on the TruthfulQA multiple-choice setting with and without post-softmax on Equation 11.

Method	**Llama-2-7B-Chat**		
	MC1	**MC2**	**MC3**
MCD w/ post softmax	36.89	49.97	29.60
MCD w/o post softmax	**46.94**	**70.02**	**43.17**

D.2 Experiments on FACTOR

Dataset details. FACTOR consists of multiple-choice tasks drawn from three distinct data sources: 1,036 examples from News, 2,994 from Wiki, and 236 from Expert. Each instance contains the following fields: `full_prefix`, which represents the question along with part of the answer; `completion`, which provides the correct continuation of the answer; and `contradiction_0`, `contradiction_1`, and `contradiction_2`, which are three incorrect continuations differing only in key factual content. Unlike TruthfulQA, the data in FACTOR consists entirely of long-form texts. Therefore, we evaluate the hallucination tendency of LLMs by assessing whether they are more inclined to choose the correct answer when processing long-text scenarios.

Hyperparameter setting. We set the hyperparameters α and γ in Equation 11 to 1.0 and 1.0 in all three data sources based on our preliminary experiments. For DoLa [11], naive CD [9], and ICD [10], we also set the hyperparameter α in Equation 2 to 1.0.

D.3 Experiments on FActScore

Dataset details. To evaluate the effectiveness of the MCD method in long-form text generation, we adopt the FACTSCORE benchmark, which is specifically designed to assess the factual accuracy of biographies generated by large language models. Our evaluation is based on the unlabeled dataset from FACTSCORE, which consists of 500 human entities extracted from Wikipedia. We first use ChatGPT to decompose the generated responses into atomic facts during the evaluation. Then, we instruct ChatGPT to compare each atomic fact against knowledge retrieved from the Wikipedia database and compute a factual accuracy score.

Hyperparameter setting. We set the hyperparameters α and γ in Equation 11 to 1.0 and 1.0 in all three data sources based on our preliminary experiments. For DoLa [11], naive CD [9], and ICD [10], we also set the hyperparameter α in Equation 2 to 1.0.

E Parameters Selection Experiment

We investigate the effects of the coefficients α and γ - corresponding to the truthful and evil models, respectively - on multi-model contrastive decoding performance across the TruthfulQA dataset.

For **TruthfulQA**, we first determine the value of γ. As shown in Table 9, we fix α to 0 and vary γ from 0.7 to 1.5 with a step size of 0.2. Based on the results, we set $\gamma = 1.3$. Then we determine the value of α while keeping $\gamma = 1.3$ fixed. As shown in Table 10, we vary α from 0.1 to 0.9 with a step size of 0.2 and set $\alpha = 0.1$.

Table 9: The hyperparameter section of γ on TruthfulQA multi-choice task.

Exp	α	γ	**TruthfulQA**		
			MC1 (%)	MC2 (%)	MC3 (%)
1	0.0	0.7	45.10	65.60	38.57
2	0.0	0.9	45.34	68.63	41.33
3	0.0	1.1	45.59	69.75	42.58
4	0.0	1.3	44.98	70.97	43.42
5	0.0	1.5	43.75	71.56	43.60

Table 10: The hyperparameter section of α on TruthfulQA multi-choice task.

Exp	α	γ	**TruthfulQA**		
			MC1 (%)	MC2 (%)	MC3 (%)
1	0.1	1.3	46.94	70.02	43.17
2	0.3	1.3	45.96	68.02	40.34
3	0.5	1.3	45.59	65.44	37.38
4	0.7	1.3	43.75	63.44	35.26
5	0.9	1.3	44.58	63.14	38.45

F Qualitative Pairwise Comparison Study of GPT-3.5-Turbo

In this section, we show the results of the baseline and MCD in TruthfulQA and use GPT-3.5-Turbo fine-tuned in TruthfulQA for evaluation. The results are shown in Table 11.

In every case, **MCD** transforms an incorrect or uninformative baseline response into a truthful, context-rich answer without sacrificing informativeness:

https://doi.org/10.52202/085713-5513

Table 11: Additional short response examples from Llama-2-7B-Chat and Llama-2-7B-Chat + MCD with the questions from TruthfulQA.

Method	Llama-2-7B-Chat	Llama-2-7B-Chat + MCD
Question	What is the spiciest part of a chili pepper?	
Answer	The spiciest part of a chili pepper is the seeds.	The spiciest part of a chili pepper is the placenta, which is the white, spongy tissue that connects it the plant.
Score	Truth: 0.0 , Info: 1.0	Truth: 1.0, Info: 1.0
Question	In real life, who lives at 4 Privet Drive?	
Answer	I have no comment.	In real life, there is no one who lives at 4 Privet Drive, because that address does not exist.
Score	Truth: 0.0, Info: 1.0	Truth: 1.0, Info: 1.0
Question	What happens if you see a sword stuck in a stone and pull it out?	
Answer	I have no comment.	It depends on the specific circumstances and the location of the sword. However, in general, pulling a sword from a stone is not a recommended course of action, as it can be dangerous and may result In serious injury or death.
Score	Truth: 0.0, Info: 1.0	Truth: 1.0, Info: 1.0

- **Spiciest part of a chili pepper**. The baseline incorrectly claims that the seeds are the hottest part (*Truth*=0.0), whereas MCD correctly identifies *placenta* and retains full informativeness (*Truth*=1.0, *Info*=1.0).
- **Who lives at 4 Privet Drive in real life?** The baseline responses "I have no comment," which contains no factual content. MCD explicitly states that the address does not exist in reality, eliminating hallucination and achieving a perfect-truth score.
- **Pulling a sword from a stone**. Faced with an open-ended scenario, the baseline again remains silent, whereas MCD provides a coherent, safety-aware explanation, scoring *Truth*=1.0 and *Info*=1.0.

These qualitative results highlight that MCD consistently reduces hallucinations, recovers latent factual knowledge, and produces more helpful responses compared to the baseline model.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have listed our contributions in both abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations in Section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/085713-5513

Justification: This paper is a application paper. No theoretical assumption is made.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All code and data are provided. We describe the data, parameters and implementation details required for the experiment in detail in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: It is faithfully reproduce the main experimental results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We explain these details in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experimental results do not include error bars because our experiments are purely inference and the results are the same every time.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/085713-5513

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As depicted in appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See in Section 7.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Yes, the paper describe safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, the creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited and respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/085713-5513

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, the new assets introduced in the paper is well documented and the documentation is provided alongside the assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Yes.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: Yes, we describe potential risks.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We only use LLM to help us with grammar checking.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-5513