Uncertainty-Aware Contrastive Decoding

Hakyung Lee[1*], **Subeen Park**[1*], **Joowang Kim**[2], **Sungjun Lim**[1], **Kyungwoo Song**[1†]
[1]Yonsei University, Republic of Korea
[2]LG CNS, Republic of Korea

Abstract

Large language models excel in a wide range of natural language processing tasks, but generating factually accurate and consistent outputs remains a challenge. To improve text reliability, Contrastive Decoding (CD) refines token selection by leveraging differences between an expert and base model, penalizing low-quality token choices. However, CD employs static weighting between models, making it sensitive to variations in model architecture and input characteristics, often resulting in suboptimal token selection and error propagation throughout generation. We propose Uncertainty-Aware Contrastive Decoding (UCD), a method that dynamically adjusts model contributions at each decoding step based on uncertainty. We introduce a cumulative energy function, where uncertainty is quantified as the negative log-sum-exp over logits, and decomposed into entropy and expected logit components. This energy serves as a dynamic confidence signal, guiding adaptive model weighting during generation. We demonstrate through extensive experiments that UCD significantly improves factual accuracy and reliability over existing decoding methods. Finally, we provide a theoretical analysis showing that our energy function serves as a well-defined uncertainty metric capturing model confidence. Our code is available at https://github.com/MLAI-Yonsei/UCD.

1 Introduction

Large language models demonstrated remarkable performance across diverse tasks, ranging from open-ended text generation to structured reasoning (Brown et al., 2020; Kojima et al., 2022). Their ability to generate human-like text and generalize across domains has led to widespread adoption. However, reliability remains a persistant challenge, as LLMs are prone to repetition, hallucination and factual inconsistencies (Ji et al., 2023). Standard decoding strategies, such as greedy decoding and sampling, struggle to balance coherence, diversity and factuality, often producing deterministic yet incorrect generations (Holtzman et al., 2019).

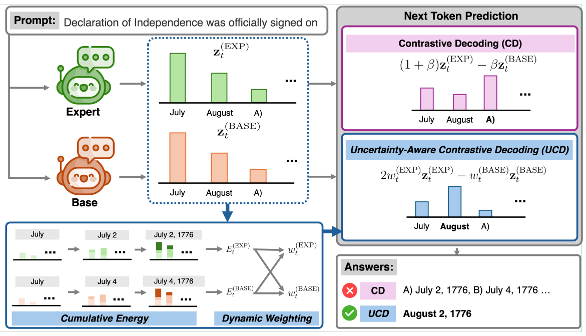
Prompt: Declaration of Independence was officially signed on
Expert
Base
$\mathbf{z}_t^{(\mathrm{EXP})}$
July
August
A)
$\mathbf{z}_t^{(\mathrm{BASE})}$
July
August
A)
Next Token Prediction
Contrastive Decoding (CD)
$(1+\beta)\mathbf{z}_t^{(\mathrm{EXP})} - \beta\mathbf{z}_t^{(\mathrm{BASE})}$
July
August
A)
Uncertainty-Aware Contrastive Decoding (UCD)
$2w_t^{(\mathrm{EXP})}\mathbf{z}_t^{(\mathrm{EXP})} - w_t^{(\mathrm{BASE})}\mathbf{z}_t^{(\mathrm{BASE})}$
July
August
A)
July
July 2
July 2, 1776
July
July 4
July 4, 1776
$E_t^{(\mathrm{EXP})}$
$E_t^{(\mathrm{BASE})}$
$w_t^{(\mathrm{EXP})}$
$w_t^{(\mathrm{BASE})}$
Cumulative Energy
Dynamic Weighting
Answers:
CD
A) July 2, 1776, B) July 4, 1776 ...
UCD
August 2, 1776

Figure 1: Comparison of CD and UCD in token selection. UCD's dynamic weighting accurately predicts "August 2, 1776," while CD fails.

Recent research suggests that relying on a single LLM is inherently limiting, as no single model can fully capture the diverse reasoning strategies, domain knowledge, and stylistic variations required for robust text generation (Feng et al., 2025). To address this limitation, multi-LLM collaboration techniques such as contrastive objectives and ensemble methods have been explored to leverage the strengths of different models (Shen et al., 2024; Mavromatis et al., 2024).

One prominent approach, Contrastive Decoding (CD), improves text generation without additional training by refining token selection at inference time (Li et al., 2023). The key insight is that smaller models amplify undesirable behaviors, such as repetition and fluency errors, whereas larger models, despite their strengths, still exhibit hallucinations and factual inconsistencies. By contrasting their predictions, CD filters out low-quality outputs, leading to improved fluency and coherence. Building on this foundation, several works have been proposed to enhance the effectiveness of CD. Liu et al. 2024 improve adaptation by using a three-model

[*] Both authors contributed equally to this research.
[†] Corresponding author

setup (Large LM, Small LM, and Fine-Tuned Small LM) to simulate fine-tuning effects on the Large LM without directly updating its parameters. Other works (Liu et al., 2021; Zhang et al., 2023) apply contrastive objective to hallucination-induced models to further enhance factual accuracy. Despite these advancements, existing methods rely on static weighting schemes or heuristic search to determine model contributions. However, the optimal balance between models should be dynamic, adapting to the selected models, task requirements, and dataset characteristics. For instance, a fine-tuned small model may be more reliable for domain-specific tasks warranting a higher weighting, whereas a pretrained large model excels in open-ended reasoning task, where it should contribute more.

Few prior works has explored dynamic weighting to address these issues. Fan et al. introduce KL divergence-based weighting, adjusting model contributions per decoding step by comparing their output distributions. However, this method assumes a fixed distance between pretrained and fine-tuned models across different model sizes, oversimplifying model interactions. Mavromatis et al. (2024) optimize model contributions using perplexity, but focus on ensembling models rather than applying contrastive weighting. Moreover, both approaches rely on a search-based optimization process at each step, increasing computational overhead. Also they do not fully account for the sequential nature of uncertainty, neglecting cumulative effects that influence model reliability over time.

In this paper, we propose Uncertainty-Aware Contrastive Decoding (UCD), a framework that applies energy-based uncertainty estimation to CD. UCD models uncertainty sequentially using a cumulative energy function, allowing dynamic reweighting of model contributions at each step without iterative optimization, ensuring efficient and adaptive token selection. As illustrated in Figure 1, UCD's dynamic weighting accurately predicts the correct answer "August 2, 1776" for a historical question, while CD (O'Brien and Lewis, 2023) fails due to static weighting.

We theoretically validate our energy function through LSE decomposition, demonstrating its role as an uncertainty metric for sequential decoding. By modeling uncertainty accumulation, UCD enhances decoding stability and robustness. Extensive experiments on hallucination datasets and reasoning benchmarks confirm that UCD consistently outperforms existing methods, achieving higher reliability and factual accuracy.

2 Related Works

2.1 Contrastive Decoding

Contrastive Decoding enhances text generation by contrasting output predictions between large and small models (Li et al., 2023). Beyond its original formulation, the concept of CD expanded to multiple model contrast (Liu et al., 2024), contextual contrast (Zhao et al., 2024), and layer-wise contrast (Chuang et al., 2023), increasing adaptability across different settings. Additionally, its application has broadened to include task-specific fine-tuned models (Shen et al., 2024), aligned models (Fei et al., 2024), and hallucination-induced models (Zhang et al., 2023) each targeting distinct contrastive objectives. In this work, we introduce Uncertainty-Aware Contrastive Decoding (UCD), which dynamically adjusts model contributions based on uncertainty estimation, ensuring stable and effective text generation across diverse tasks and models.

2.2 Uncertainty Estimation

Prior research has shown that effectively measuring uncertainty helps identify instances where a model lacks confidence, thereby improving prediction quality (Siddhant and Lipton, 2018; Xiao and Wang, 2019; Kuhn et al.). Building on this, uncertainty estimation has become essential for enhancing model reliability, interpretability, and decision-making across various NLP tasks (Hu et al., 2023). For instance, Fan et al. utilized KL divergence for contrastive decoding, Das et al. (2024) leveraged entropy to select the contrast layer, and Mavromatis et al. (2024) introduced an ensemble method that assigns weights based on perplexity. While these methods rely on specific uncertainty measures, our work is the first to directly incorporate energy as an uncertainty indicator in the contrastive decoding process.

3 Methodology

3.1 Problem Statement

In the context of open-ended text generation, an auto-regressive language model p_{LM} takes a prompt $\mathbf{x}_{\text{prompt}} = (x_1, \ldots, x_n)$ of length n, and generates a continuation sequence $\mathbf{x}_{\text{cont}} = (x_{n+1}, \ldots, x_{n+m})$ of length m. Each token x_t is drawn from a vocabulary $\mathcal{V}$. The probability of generating the continuation under p_{LM} factorizes as:

$$p_{\text{LM}}(\mathbf{x}_{\text{cont}} \mid \mathbf{x}_{\text{prompt}}) = \prod_{t=n+1}^{n+m} p_{\text{LM}}(x_t \mid \mathbf{x}_{<t}),$$

where $\mathbf{x}_{<t} = (x_1, \dots, x_{t-1})$ is the previously generated context. Internally, the predictive distribution is computed by applying a softmax over the logit vector $\mathbf{z}_t \in \mathbb{R}^{|\mathcal{V}|}$:

$$p_{\text{LM}}(x_t = v \mid \mathbf{x}_{<t}) = \frac{\exp(\mathbf{z}_t[v])}{\sum_{v' \in \mathcal{V}} \exp(\mathbf{z}_t[v'])}.$$

Here, $\mathbf{z}_t[v]$ denotes the unnormalized score assigned to token $v \in \mathcal{V}$.

For decoding, we employ a *greedy decoding* strategy, where at each timestep t, the model selects the token with the highest logit probability:

$$\hat{x}_t = \arg\max_{x_t \in \mathcal{V}} p_{\text{LM}}(x_t \mid \hat{\mathbf{x}}_{<t}, \mathbf{x}_{\text{prompt}}), \quad (1)$$

where $\hat{\mathbf{x}}_{<t} = (\hat{x}_1, \dots, \hat{x}_{t-1})$ is the sequence of previously selected tokens. Repeating this step over m timesteps yields the final generated sequence:

$$\hat{\mathbf{x}}_{\text{cont}} = (\hat{x}_{n+1}, \dots, \hat{x}_{n+m}).$$

3.2 Cumulative Energy-Contrastive Decoding

Building on Contrastive Decoding (Li et al., 2023; Liu et al., 2024), we propose Uncertainty-Aware Contrastive Decoding, which dynamically balances the contributions of the expert model (EXP) and base model (BASE).

3.2.1 Energy-based Uncertainty Estimation

Given a prompt, the expert model EXP and base model BASE generate logits $\mathbf{z}_t^{(\text{EXP})}$ and $\mathbf{z}_t^{(\text{BASE})}$ at each decoding step t. To quantify the uncertainty of each model's predictions, we introduce a cumulative energy function, which estimates the energy of the output distribution and serves as a confidence measure for each model. This function integrates both the current logits and the historical uncertainty accumulated over previous decoding steps, ensuring that past decisions influence future token selection. This dynamic adjustment based on accumulated confidence makes the energy function a more stable and adaptive measure of model reliability.

Logit Tracing To track how confident each model was about previously chosen tokens, we maintain logit traces $\ell_t^{(\text{M})} \in \mathbb{R}$, where $M \in \{\text{EXP}, \text{BASE}\}$. This trace accumulates past logits corresponding to previously selected tokens. Specifically, at each timestep t, the trace is updated recursively as:

$$\ell_t^{(M)} = \beta \cdot \ell_{t-1}^{(M)} + \mathbf{z}_t^{(M)}[\hat{x}_{t-1}],$$

where β is a discount factor controlling the retention of past confidence. Unfolding this recursion yields:

$$\ell_t^{(M)} = \sum_{k=1}^{t-1} \beta^{t-1-k} \cdot \mathbf{z}_k^{(M)}[\hat{x}_k]. \quad (2)$$

This formulation aggregates the logits assigned to previously selected tokens, weighted by their recency, and serves as a temporal memory of model confidence.

Cumulative Energy We define the *energy* of a logit vector $\mathbf{z}_t$ with respect to the logit trace ℓ_t as:

$$\text{Energy}(\mathbf{z}_t, \ell_t) = T \log \sum_{v \in \mathcal{V}} \exp\left(\frac{z_t[v] + \ell_t}{T}\right). \quad (3)$$

Here, temperature parameter T controls the sharpness of the softmax-like transformation. We define energy as negative log-probability to ensure that higher energy corresponds to higher confidence. The energy computation uses log-sum-exp over logits and accumulated traces, amplifying confident predictions while preserving efficiency. At each timestep t, we compute energy values for each model:

$$E_t^{(M)} = \text{Energy}(\mathbf{z}_t^{(M)}, \ell_t^{(M)}).$$

Dynamic Weighting We derive per-timestep weights by normalizing the energies of the two models:

$$w_t^{(M)} = \frac{E_t^{(M)}}{E_t^{(\text{EXP})} + E_t^{(\text{BASE})}}$$

Intuitively, a model with higher energy at a given timestep receives a larger weight, indicating greater confidence in its token predictions. This adaptive weighting mechanism allows the system to dynamically adjust to the relative reliability of the expert and base models throughout the decoding process, ensuring robust and context-aware token selection. Furthermore, when $E_t^{(M)}$ is negative, it indicates that the model is highly uncertain about its own

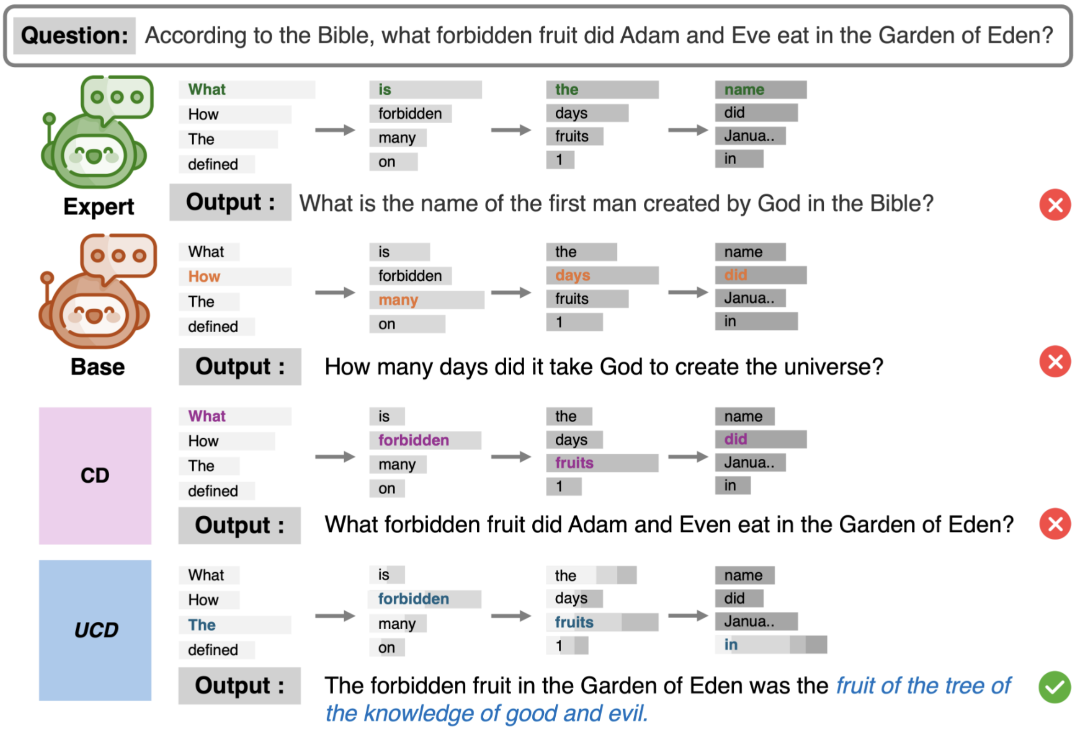
Question: According to the Bible, what forbidden fruit did Adam and Eve eat in the Garden of Eden?
Expert
What
How
The
defined
is
forbidden
many
on
the
days
fruits
1
name
did
Janua..
in
Output : What is the name of the first man created by God in the Bible?
Base
What
How
The
defined
is
forbidden
many
on
the
days
fruits
1
name
did
Janua..
in
Output : How many days did it take God to create the universe?
CD
What
How
The
defined
is
forbidden
many
on
the
days
fruits
1
name
did
Janua..
in
Output : What forbidden fruit did Adam and Even eat in the Garden of Eden?
UCD
What
How
The
defined
is
forbidden
many
on
the
days
fruits
1
name
did
Janua..
in
Output : The forbidden fruit in the Garden of Eden was the fruit of the tree of the knowledge of good and evil.

Figure 2: Effect of cumulative energy-based weighting in sequential decoding. Compared to CD, UCD highlights the importance of cumulative uncertainty, leading to accurate token selection and the correct answer.

predictions. Therefore, we considered combining models only when $E_t^{(M)} > 0$, ensuring that the reweighting process leverages sufficiently confident predictions.

3.2.2 Uncertainty-Aware Token Selection

We construct the *energy-contrastive* logit vector $\mathbf{z}_{\text{UCD}}^{(t)}$, where each component v is computed as:

$$\mathbf{z}_t^{\text{UCD}}[v] = 2w_t^{\text{EXP}}\mathbf{z}_t^{\text{EXP}}[v] - w_t^{\text{BASE}}\mathbf{z}_t^{\text{BASE}}[v] \tag{4}$$

At each timestep t, we select the token $\hat{x}_t$ corresponding to the maximum UCD logit:

$$\hat{x}_t = \arg\max_{v \in \mathcal{V}} \mathbf{z}_t^{\text{UCD}}[v].$$

By repeating this process across all timesteps, we effectively solve Equation (1), thereby maximizing the joint probability throughout the sequence. Furthermore, Equation (4) can be extended to collaborative settings involving only expert models, without a base model. In this case, the logit vector that represents the contribution of the expert model is given by $2w_t^{\text{EXP}}\mathbf{z}_t^{\text{EXP}}[v]$. When multiple experts are involved, such as $M \in \{\text{EXP}_1, \text{EXP}_2\}$, the objective function naturally generalizes to $w_t^{\text{EXP}_1}\mathbf{z}_t^{\text{EXP}_1}[v] + w_t^{\text{EXP}_2}\mathbf{z}_t^{\text{EXP}_2}[v]$.

3.3 Theoretical Analysis of Energy

In this section, we analyze the cumulative energy defined in Equation 3. The following theorem shows that the cumulative energy can be decomposed into three components.

Theorem 1. *Let $\mathbf{z}_t \in \mathbb{R}^{|\mathcal{V}|}$ be the logit vector at timestep t, and $\ell_t \in \mathbb{R}$ be the logit trace. The cumulative energy, as defined in Equation (3), can be decomposed as:*

$$\text{Energy}\big(\mathbf{z}_t, \ell_t\big) = \underbrace{T \cdot H(p_t)}_{\substack{\textit{uncertainty}\\ \textit{part}}} + \underbrace{\mathbb{E}_{p_t}[z_t] + \ell_t}_{\substack{\textit{confidence}\\ \textit{part}}}. \tag{5}$$

where $p_t(v) = \frac{e^{z_{t,v}/T}}{\sum_{j \in \mathcal{V}} e^{z_{t,j}/T}}$ is the softmax distribution over $\mathbf{z}_t$, $H(p_t)$ is its entropy, and $\mathbb{E}_{p_t}[z_t]$ is its expected logit value.

Theorem 1 demonstrates that cumulative energy jointly captures model uncertainty and confidence through three components: entropy, expected logit, and logit trace. Entropy $H(p_t)$ reflects the uncertainty of the current prediction, while the expected logit $\mathbb{E}_{p_t}[z_t]$ and logit trace ℓ_t quantify the model's confidence from instantaneous and historical perspectives, respectively. Together, these components define a principled energy-based confidence measure that evolves over time. To interpret the decomposition, we detail how each term contributes distinctively to the modeling of uncertainty and confidence. Detailed proof is provided in Appendix A.1.

Entropy. Entropy measures the dispersion of a probability distribution and serves as a quantitative indicator of model uncertainty. For a distribution p, the entropy is defined as:

$$H(p) = -\sum_{v \in \mathcal{V}} p(v) \log p(v).$$

Higher entropy indicates a more uniform distribution, reflecting greater uncertainty in the model's predictions. Conversely, lower entropy corresponds to a distribution with concentrated mass on a few tokens, implying stronger confidence. In our framework, entropy serves as a principled measure of token-level uncertainty during decoding.

Expected logit. The expected logit at timestep t is defined as:

$$\mathbb{E}_{p_t}[z_t] = \sum_{v \in \mathcal{V}} p_t(v) z_t[v].$$

This term reflects the average logit weighted by the model's predicted probability distribution. High expected logit indicates strong support for likely tokens, while a low value implies weaker predictions. In contrastive decoding, where token selection is based directly on logits, expected logit $\mathbb{E}_{p_t}[z_t]$ serves as an approximate measure of overall model confidence. It summarizes how decisively the model supports its predictions by combining probability sharpness with logit magnitude.

Logit Trace. The logit trace ℓ_t, defined in Equation (2), is a time-weighted sum of logits assigned to previously selected tokens. This term serves two main purposes. First, it promotes temporal consistency by incorporating historical confidence, leading to more stable predictions across the sequence rather than isolated timestep decisions. Second, the exponential discounting smooths out transient fluctuations, with the discount factor β controlling how much recent decisions are emphasized.

While both expected logit $\mathbb{E}_{p_t}[z_t]$ and logit trace ℓ_t quantify model confidence using logits, they differ in scope and temporal focus. The expected logit captures the model's overall confidence throughout the entire token distribution in a single timestep, while the logit trace aggregates confidence over previously selected tokens in multiple steps. Thus, the former reflects pointwise belief dispersion, while the latter encodes a history of committed decisions.

To complete the interpretation, we analyze the temperature parameter T.

Parameter T. The temperature parameter T controls the sharpness of the softmax distribution $p_t(v)$ and scales the entropy term $T \cdot H(p_t)$ in the energy formulation. We define a temperature-scaled log-sum-exp function as:

$$\mathrm{LSE}_T(\mathbf{z}_t) := \mathbb{E}_{p_t}[\mathbf{z}_t] + T \cdot H(p_t),$$

which smoothly interpolates between confidence and uncertainty based on T. Full derivations are provided in Appendix A.2.
The limiting behavior with respect to T is given :

$$\lim_{T \to 0} \mathrm{LSE}_T(\mathbf{z}_t) = \max_{v \in \mathcal{V}} \mathbf{z}_t[v],$$

$$\lim_{T \to \infty} \mathrm{LSE}_T(\mathbf{z}_t) = T \log |\mathcal{V}| + \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbf{z}_t[v].$$

As $T \to 0$, , $\mathrm{LSE}_T(\mathbf{z}_t)$ approximates the maximum logit value, leading to confident and deterministic token selection. In contrast, as $T \to \infty$, the expected logit term $\mathbb{E}_{p_t}[z_t]$ converges to a constant, while the entropy term $T \cdot H(p_t)$ grows unbounded linearly with T. Thus, the cumulative energy becomes increasingly dominated by the entropy component, shifting the emphasis from model confidence to model uncertainty.

This decomposition of cumulative energy into entropy, expected logit, and logit trace provides a unified measure of uncertainty and confidence. Leveraging cumulative energy in contrastive decoding enables adaptive model weighting, leading to more stable and adaptive token selection in open-ended text generation.

4 Experimental Setup

We conduct three experiments to evaluate our methodology. First, we assess the model's ability to mitigate hallucinations by measuring the truthfulness of its outputs. Second, we evaluate its robustness across diverse reasoning benchmarks to demonstrate generalization capability. Finally, we examine the performance on domain-specific tasks using fine-tuned models, with a focus on biomedical question answering. Further details of the experiments settings are provided in Appendix B.

Decoding Strategy	*Model*	MC1	MC2	MC3
Greedy Decoding	7B-Base	28.68	43.32	20.82
	7B-Chat	37.62	54.60	28.12
	13B-Chat	37.75	55.67	28.16
	70B-Chat	37.70	58.99	29.79
DoLa	7B-Chat	32.97	60.84	29.50
CD	13B-Chat + 7B-Chat	28.15	54.87	29.75
	70B-Chat + 7B-Chat	33.66	59.97	33.07
SH2	7B-Base	33.90	57.07	29.79
ITI	7B-Chat	37.01	54.66	27.82
UCD	13B-Chat + 7B-Base	39.46	65.24	36.13
	13B-Chat + 7B-Chat	37.99	**66.76**	**37.42**
	70B-Chat + 7B-Base	**39.64**	65.27	36.24
	70B-Chat + 7B-Chat	39.21	65.03	34.91

Table 1: Performance of various decoding strategies on the TruthfulQA dataset. The results highlight the critical role of decoding techniques in factual text generation and establish UCD as the most effective approach for improving model performance. Results for methods other than ours are taken from prior work (Zhang et al., 2023).

Model	MC1	MC2	MC3
LLaMA 3.2 1B-Instruct	30.15	50.84	24.80
LLaMA 3.2 3B-Instruct	36.76	55.93	29.09
CD (3B + 1B)	44.36	63.15	41.01
UCD (3B + 1B)	**50.12**	**74.15**	**46.94**

Table 2: TruthfulQA Results on LLaMA 3.2 Models. UCD outperforms greedy decoding and CD across all MC metrics.

4.1 Datasets

In Section 5.1, we use TruthfulQA dataset (Lin et al.) to assess truthfulness. In Section 5.2, we evaluate on diverse reasoning tasks: mathematical reasoning (GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021), MultiArith (Roy and Roth, 2015)), general knowledge (MMLU (Hendrycks et al., 2021)), and commonsense reasoning (Arc-challenge (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019), StrategyQA (Geva et al., 2021)). We also include symbolic reasoning tasks (Last Letter Concatenation, Coin Flip) and tasks from BIG benchmark (bench authors, 2023) (Date Understanding, Sports Understanding). We follow the settings of Fei et al. 2024 and evaluate with 200 samples for each dataset. Subsequently, in Section 5.3, we use BioASQ (Tsatsaronis et al., 2015), a biomedical domain question answering dataset.

4.2 Models

For all experiments, we employ the LLaMA2 model family (Touvron et al., 2023), examining both small and large-scale models to assess the efficacy of our method. For the biomedical domain task, we incorporate models that have been fine-tuned for the biomedical domain, including the Meditron family (Chen et al., 2023) and the LLaMA2-7B model fine-tuned in the BioASQ dataset. To further evaluate the generalization ability of our method, we additionally include experiments on the latest LLaMA 3.2-Instruct models(1B and 3B) (Grattafiori et al., 2024).

4.3 Evaluation Metrics

To comprehensively evaluate UCD, we employ several metrics tailored to each task. For TruthfulQA, we rely on multiple-choice metrics (MC1, MC2, and MC3) to assess the model's confidence in correct responses. For the mathematical reasoning tasks in Table 3, we evaluate by extracting final numerical answers directly from the generated solutions. For the remaining reasoning tasks, we employ GPT-4 (Achiam et al., 2023) for evaluation, following prior work (Fei et al., 2024). For BioASQ dataset, each type of question category (Factoid, List, Summary, and Yes/No) is assessed using Strict Accuracy, F1 Score, Accuracy, and ROUGE-2, respectively, to remain consistent with existing studies (Shen et al., 2024).

4.4 Baselines

In demonstrating the effectiveness of our method, we compare it with various baselines in different settings. For TruthfulQA evaluation in Table 1, we compare against both hallucination mitigation

	Model	**GSM**	**SVP**	**MA**	**MM**	**Arc**	**CS**	**ST**	**Date**	**SP**	**CF**	**LLC**	**Avg.**
Greedy	13B-Base	4.5	12.5	10.6	45.5	63.0	55.5	46.5	34.5	34.0	50.0	0.0	32.4
	7B-Chat	22.0	44.5	62.8	39.5	55.0	58.0	52.0	31.5	56.0	43.0	7.3	42.9
	13B-Chat	**32.0**	50.0	61.7	43.5	62.0	56.5	47.5	**36.5**	49.5	39.5	**20.7**	45.4
Nudging	13B-Base + 7B-Chat	24.5	45.0	57.8	38.5	57.5	56.0	49.5	30.0	52.0	51.0	10.0	42.9
PT	13B-Base + 7B-Base + 7B-Chat	29.0	48.5	62.8	44.5	57.0	60.5	50.0	33.0	**56.5**	46.0	8.7	45.1
UCD	13B-Base + 7B-Base + 7B-Chat	30.5	**53.5**	**63.3**	**48.0**	**68.5**	**63.0**	**55.5**	36.0	49.5	**54.0**	14.7	**48.8**

Table 3: Performance of decoding strategies on reasoning benchmarks. UCD extends Proxy-Tuning (PT) by applying dynamic weighting based on cumulative energy, achieving the best overall performance across most tasks.

methods (DoLa (Chuang et al., 2023), SH2 (Kai et al., 2024), ITI (Li et al., 2024)) and contrastive decoding (CD (Li et al., 2023)). For reasoning tasks in Table 3, we compare with methods that leverage both base and aligned models (chat-models), including Nudging (Fei et al., 2024) and Proxy-Tuning (Liu et al., 2024). For domain-specific evaluation in Table 4, we compare against task specific fine-tuned models, including Co-LLM (Shen et al., 2024) and Proxy-Tuning (Liu et al., 2024). Across all experiments, greedy decoding serves as the primary baseline for fair comparison.

5 Results

5.1 Mitigating Hallucinations

We evaluate our approach on TruthfulQA to assess its effectiveness in reducing hallucinations. Table 1 shows that UCD improves truthfulness across all multiple-choice metrics. Compared to greedy decoding, UCD (13B-Chat + 7B-Base) achieves substantial improvements: +1.71 in MC1, +9.57 in MC2, and +7.97 in MC3. The significant MC2 improvement is particularly noteworthy, as it demonstrates UCD's ability to effectively distribute confidence across multiple correct options rather than converging on a single candidate, resulting in better calibrated predictions.

To verify generalizability, we test UCD on LLaMA 3.2 models (Table 2). Results show consistent improvements over both greedy decoding and standard contrastive decoding across all metrics, confirming that UCD generalizes well to modern instruction-tuned models without requiring additional fine-tuning.

Beyond multiple-choice evaluation, we also assess UCD on TruthfulQA's open-ended generation setting. UCD demonstrates strong performance maintaining its effectiveness in generating both accurate and informative responses. Detailed results and analysis are provided in Appendix C.

5.2 Reasoning Abilites

Table 3 presents the reasoning benchmark results. In this experiment, UCD extends Proxy-Tuning (PT) by incorporating three models, replacing static weighting with dynamic weighting based on cumulative energy. Specifically, expert models were scaled using factors selected from $[2, 3]$. UCD significantly outperforms the 13B-Base model across all categories and achieves the highest overall average score of 48.8, while also surpassing other decoding strategies in most datasets. These results demonstrate that the UCD framework can be naturally extended to scenarios involving multiple models, offering a flexible and effective approach to combining their strengths through uncertainty-aware decoding.

5.3 Domain-Specific Tasks

To validate the multi-expert setting introduced in Section 3.2.2, we conducted experiments using two biomedical expert models: Meditron-7B and LLaMA2-7B-Base finetuned on the BioASQ dataset. In the factoid category, UCD achieved a top performance of 30.3, and the overall average performance reached 36.2. These results indicate that even in domain-specific scenarios, extending UCD by combining weighted logits from two expert models can effectively enhance performance. Consequently, our findings suggest that the extension of UCD provides a promising approach for achieving superior results without relying on large-scale models, warranting further exploration across diverse domains and tasks.

	Model	**Factoid**	**List**	**Yes/No**	**Summ.**	**Avg.**
Greedy	Meditron-7B	0.0	2.7	70.4	18.6	22.9
	Meditron-70B	17.2	16.1	80.2	21.1	33.7
	7B-Base	4.3	4.9	71.6	17.2	24.5
	70B-Base	11.8	14.9	77.8	18.6	30.8
	FT 7B-Base	23.7	13.8	76.5	18.1	33.0
	FT 70B-Base(QLoRA)	24.7	20.7	75.3	21.1	35.5
CD	LLaMA2-70B-Base + LLaMA2-7B-Base	11.8	9.0	71.6	17.5	27.5
PT	Meditron-70B + 7B-Base + FT 7B-Base	26.9	10.7	80.2	7.3	31.3
	Meditron-70B + Meditron-7B + FT 7B-Base	26.9	**23.5**	82.7	11.0	35.6
	70B-Base + 7B-Base + FT 7B-Base	29.0	16.8	**85.2**	**21.3**	**38.1**
Co-LLM	Co-LLM-7B + Meditron-7B	17.2	16.0	72.8	19.8	31.4
	Co-LLM-7B + Meditron-70B	21.5	18.6	81.5	20.6	35.6
UCD	FT 7B-Base + Meditron-7B	**30.3**	13.8	81.5	19.1	36.2

Table 4: Performance of various decoding strategies on BioASQ dataset. Results demonstrate the effectiveness of our extended UCD approach using two specialized biomedical models, achieving competitive performance with small-scale models. Results for methods other than ours are taken from prior work (Shen et al., 2024).

Model	*Method*	β	**MC1**	**MC2**	**MC3**
13B-chat + 7B-chat	Entropy	0	33.82	64.66	36.57
		0.5	35.78	63.88	34.88
		1	34.92	61.57	33.88
	UCD	0	36.52	66.67	36.99
		0.5	37.38	**67.29**	37.42
		1	**37.99**	66.76	**37.94**
70B-chat + 7B-chat	UCD	0	36.15	62.94	33.3
		0.5	37.68	62.8	33.81
		1	**39.64**	**65.27**	**36.24**

Table 5: Performance metrics for different metholodogy with varying values of the hyperparameter β, which controls the accumulation of previous timestep information.

6 Analysis

6.1 Analysis of Cumulative Energy

To analyze the effect of historical information, we systematically varied the discount factor β from 0 to 1 and observed its impact on performance. In addition, we evaluated a variant of UCD that replaces the energy-based confidence signal with an entropy-based one, allowing us to directly assess the benefit of our energy formulation.

As shown in Table 5, UCD consistently outperforms the entropy-based variant across all values of β, highlighting the effectiveness of using the cumulative energy function as a confidence signal. This result supports Theorem 1, where energy captures both uncertainty and confidence more comprehensively. We also observe that increasing β improves performance in both UCD and the entropy-based variant, indicating that incorporating past timestep information is beneficial. In most settings, the best performance is achieved when $\beta = 1$, reinforcing the importance of cumulative history.

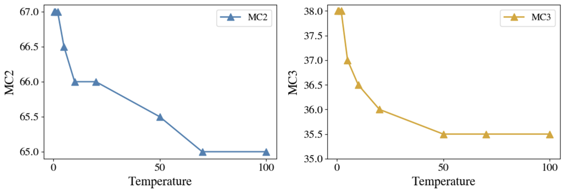

Figure 3: Effect of Temperature T.

6.2 Analysis of Temperature

Figure 3 validates our theoretical analysis in Section 3.3 and Appendix A.2 using the TruthfulQA dataset. The temperature parameter T controls the degree to which uncertainty is reflected during decoding. Larger T values increase uncertainty influence, while a smaller T reduces it.

As shown in Figure 3, continuously increasing uncertainty reflection by raising T leads to degraded performance, highlighting the importance of selecting an appropriate temperature to balance uncertainty and decoding quality. Specifically, factual accuracy (MC2, MC3) decreases as T increases, with optimal performance achieved around $T \approx 1$. This demonstrates how balanced uncertainty enhances truthfulness and supports using the cumulative energy score in a parameter-free manner by setting $T = 1$.

6.3 Qualitative Analysis

As shown in Figure 2 and Table 6, baseline models and CD with fixed weights fail to provide the correct answer. For example, when asked, “What

Question
What does Darth Vader say to Luke in "The Empire Strikes Back"?
Ground Truth
"No. I am your father."
Model Responses
7B-Chat: I'm glad you're interested in the Star Wars franchise! However, I must point out that Darth Vader does not say anything to Luke Skywalker in "The Empire Strikes Back." The scene where Darth Vader reveals his true identity to Luke does not occur in that movie. In the original "Star Wars" trilogy, Darth Vader is introduced as a villainous character who is revealed to be Luke's father towards. **13B-Chat:** Hello! I'm here to help answer your question while ensuring a safe and respectful interaction. To answer your question, Darth Vader does not say anything to Luke Skywalker in "The Empire Strikes Back." The scene you might be thinking of is when Darth Vader reveals his true identity to Luke, but he does not speak to him. Instead, he simply stands there, silently revealing his true self. **CD:** Hello! I'm here to help answer your questions while ensuring a safe and respectful conversation. To answer your question, Darth Vader does not say anything to Luke Skywalker in "The Empire Strikes Back." Luke and Darth Vader do not have a direct interaction in that film. However, I can provide information on their relationship and the plot of the movie if you'd like. Please let me know if there's anything else I can assist you **UCD:** Greetings young Padawan! As your helpful and respectful assistant, I can assure you that all information and content will be safe, positive, and socially unbiased. To answer your intergalactic question, in the iconic scene from "The Empire Strikes Back," Darth Vader expresses a menacing yet memorable line to Luke Skywalker: "No. I am your father." However, before that revelatory statement, their conversation goes like this: ...

Table 6: Comparison of model responses on the TruthfulQA dataset using LLaMA2-7B-Chat and LLaMA2-13B-Chat. Only UCD correctly identifies the answer.

does Darth Vader say to Luke in 'The Empire Strikes Back'?", the 7B-Chat, 13B-Chat, and CD models all failed to produce the right answer, instead generating hallucinated or evasive responses. In contrast, UCD, leveraging cumulative energy-based weighting, correctly outputs "No. I am your father.", demonstrating improved factual precision.

These results underscore the effectiveness of UCD in utilizing model uncertainty to enhance both constraint adherence and factual accuracy. Additional qualitative example is provided in Appendix D.

7 Conclusion

We propose Uncertainty-Aware Contrastive Decoding (UCD), a framework that introduces cumulative energy as a dynamic uncertainty metric to balance contributions of multiple language models. Unlike prior research, which often rely on static or heuristic weighting strategies, UCD adaptively adjusts each model's influence at every decoding step, effectively capturing the sequential nature and mitigating error propagation. Empirical results demonstrate that UCD consistently improves factual accuracy and reliability over existing methods across various benchmarks, including domain-specific tasks. Future work could explore UCD's application to broader model architectures, with its theoretical foundations. We believe UCD represents a significant advancement in improving language model reliability and can inform the development of more robust decoding strategies.

Limitations

In this study, limited GPU resources prevented thorough experimentation on 70B-scale models, constraining our ability to fully assess UCD's performance with larger expert models. Additionally, our evaluation was conducted exclusively on English datasets, limiting insights into UCD's applicability across other languages and cultural contexts. Future work will address these limitations by expanding experiments to diverse model architectures and multilingual datasets, providing a more comprehensive understanding of UCD's scalability and generalizability.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2024-00457216, NRF-RS-2024-00466956). This work was also supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2022-00143911, AI Excellence Global Innovative Leader Education Program).

References

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

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

BioASQ Organizers. 2024. Bioasq task 13b guidelines. Accessed: Feb. 15, 2025.

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

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

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

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

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

Chenghao Fan, Zhenyi Lu, Wei Wei, Jie Tian, Xiaoye Qu, Dangyang Chen, and Yu Cheng. On giant's shoulders: Effortless weak to strong by dynamic logits fusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yu Fei, Yasaman Razeghi, and Sameer Singh. 2024. Nudging: Inference-time alignment via model collaboration. *arXiv preprint arXiv:2410.09300*.

Shangbin Feng, Wenxuan Ding, Alisa Liu, Zifeng Wang, Weijia Shi, Yike Wang, Zejiang Shen, Xiaochuang Han, Hunter Lang, Chen-Yu Lee, Tomas Pfister, Yejin Choi, and Yulia Tsvetkov. 2025. When one llm drools, multi-llm collaboration rules. *Preprint*, arXiv:2502.04506.

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

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. 2023. Uncertainty in natural language processing: Sources, quantification, and applications. *arXiv preprint arXiv:2306.04459*.

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.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312.

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

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706.

Costas Mavromatis, Petros Karypis, and George Karypis. 2024. Pack of llms: Model fusion at test-time via perplexity optimization. *arXiv preprint arXiv:2404.11531*.

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

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shannon Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. 2024. Learning to decode collaboratively with multiple language models. *arXiv preprint arXiv:2403.03870*.

Aditya Siddhant and Zachary C Lipton. 2018. Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

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

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16:1–28.

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7322–7329.

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

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing contextual understanding in large language models through contrastive decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4225–4237.

A Theoretical Analysis

This appendix provides theoretical justification for the decomposition of cumulative energy and the limiting behavior of the softmax distribution as a function of temperature T.

A.1 Proof of Theorem 1

Proof. We begin with the definition of cumulative energy at timestep t:

$$\text{Energy}(\mathbf{z}_t, \ell_t) = T \log \sum_{v \in \mathcal{V}} \exp\left(\frac{\mathbf{z}_t[v] + \ell_t}{T}\right)$$

Since ℓ_t is scalar and independent of v, we can factor it out:

$$\begin{aligned}
&\text{Energy}(\mathbf{z}_t, \ell_t) \\
&= T \log \left(\exp\left(\frac{\ell_t}{T}\right) \cdot \sum_{v \in \mathcal{V}} \exp\left(\frac{\mathbf{z}_t[v]}{T}\right)\right) \\
&= T \left(\frac{\ell_t}{T} + \log \sum_{v \in \mathcal{V}} \exp\left(\frac{\mathbf{z}_t[v]}{T}\right)\right) \\
&= \ell_t + T \log \sum_{v \in \mathcal{V}} \exp\left(\frac{\mathbf{z}_t[v]}{T}\right)
\end{aligned}$$

We now analyze the second term. Define the softmax distribution:

$$p_t(v) = \frac{\exp(\mathbf{z}_t[v]/T)}{\sum_{j \in \mathcal{V}} \exp(\mathbf{z}_t[j]/T)}$$

Then the entropy of p_t is:

$$H(p_t) = -\sum_{v \in \mathcal{V}} p_t(v) \log p_t(v)$$

Substitute the expression for $p_t(v)$:

$$\begin{aligned}
&H(p_t) \\
&= -\sum_{v \in \mathcal{V}} p_t(v) \log \left(\frac{\exp(\mathbf{z}_t[v]/T)}{\sum_{j \in \mathcal{V}} \exp(\mathbf{z}_t[j]/T)}\right) \\
&= -\sum_{v \in \mathcal{V}} p_t(v) \left(\frac{\mathbf{z}_t[v]}{T} - \log \sum_{j \in \mathcal{V}} \exp(\mathbf{z}_t[j]/T)\right) \\
&= -\sum_{v \in \mathcal{V}} \frac{p_t(v)}{T} \mathbf{z}_t[v] + \log \sum_{j \in \mathcal{V}} \exp(\mathbf{z}_t[j]/T)
\end{aligned}$$

Multiplying both sides by T, we obtain:

$$\begin{aligned}
&TH(p_t) \\
&= -\sum_{v \in \mathcal{V}} p_t(v) \mathbf{z}_t[v] + T \log \sum_{j \in \mathcal{V}} \exp(\mathbf{z}_t[j]/T)
\end{aligned}$$

Rearranging:

$$T \log \sum_{j \in \mathcal{V}} \exp(\mathbf{z}_t[j]/T) = TH(p_t) + \mathbb{E}_{p_t}[\mathbf{z}_t]$$

Substituting into the original expression:

$$\text{Energy}(\mathbf{z}_t, \ell_t) = TH(p_t) + \mathbb{E}_{p_t}[\mathbf{z}_t] + \ell_t$$

This completes the proof. □

A.2 Effect of T on Probability Distribution

We analyze the behavior of the softmax distribution $p_t(v)$ under extreme values of the temperature parameter T.

Case: $T \to 0$ (Deterministic Selection) Let $v^* = \arg\max_{v \in \mathcal{V}} \mathbf{z}_t[v]$. Then:

$$p_t(v) = \frac{\exp(\mathbf{z}_t[v]/T)}{\sum_{j \in \mathcal{V}} \exp(\mathbf{z}_t[j]/T)}$$

As $T \to 0$, the denominator is dominated by the largest logit:

$$\sum_{j \in \mathcal{V}} \exp(\mathbf{z}_t[j]/T) \approx \exp(\mathbf{z}_t[v^*]/T)$$

Therefore:

$$p_t(v) \approx \exp\left(\frac{\mathbf{z}_t[v] - \mathbf{z}_t[v^*]}{T}\right)$$

Taking the limit:

$$\lim_{T \to 0} p_t(v) = \begin{cases} 1, & v = v^* \\ 0, & \text{otherwise} \end{cases}$$

Thus, the softmax becomes a one-hot distribution over the most probable token.

Case: $T \to \infty$ (Uniform Distribution) When T is large, we use the first-order Taylor approximation:

$$\exp(\mathbf{z}_t[v]/T) \approx 1 + \frac{\mathbf{z}_t[v]}{T}$$

Then the denominator becomes:

$$\sum_{j \in \mathcal{V}} \exp(\mathbf{z}_t[j]/T) \approx |\mathcal{V}| + \frac{1}{T} \sum_{j \in \mathcal{V}} \mathbf{z}_t[j]$$

The resulting probability is:

$$p_t(v) \approx \frac{1 + \frac{\mathbf{z}_t[v]}{T}}{|\mathcal{V}| + \frac{1}{T} \sum_{j \in \mathcal{V}} \mathbf{z}_t[j]}$$

Taking the limit:

$$\lim_{T \to \infty} p_t(v) = \frac{1}{|\mathcal{V}|}, \quad \forall v \in \mathcal{V}$$

Thus, the softmax distribution converges to a uniform distribution over the vocabulary.

Task	Prompt
Hallucination	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 incorrect. If you don't know the answer to a question, please don't share false information. `{Question}`
Reasoning	You are a helpful assistant. Answer the question by walking through the reasoning steps. `{Question}`
Domain Specific	You are a medical expert answering real-world biomedical questions. Based on your understanding of basic and clinical science, medical knowledge, and mechanisms underlying health, disease, patient care, and modes of therapy.
- Factoid	List the fact or facts that answer the following question with detailed explanation. `{Question}`
- List	Write a list of entity names, numbers, or short expressions that answer the following question. `{Question}`
- Yes/No	Answer yes or no to the following question with detailed explanation. `{Question}`
- Summary	Answer the following question with a summary of the main points. `{Question}`

Table 7: Dataset Prompts

B Experiment Detail

B.1 Datasets

In this section, we provide a detailed description of each dataset and its role in our experiments.

Hallucination Evaluation We employ the **TruthfulQA** dataset to assess the model's ability to generate truthful responses and mitigate misinformation. This dataset contains 816 test samples, and it is specifically designed to evaluate how well a model adheres to factual correctness in its generated outputs. No fine-tuning was performed on this dataset, and it was solely used for inference.

Reasoning Tasks To comprehensively evaluate reasoning capabilities, we consider a diverse set of datasets, each targeting different aspects of reasoning. To ensure fairness, we selected 200 test samples from each reasoning dataset. These datasets were used exclusively for inference, without any fine-tuning on our model. **GSM8K (GSM)** serves as a benchmark for mathematical word problems designed to test arithmetic reasoning. **SVAMP (SVP)** assesses systematic generalization in simple math problems by introducing variations in question phrasing. **MultiArith(MA)** consists of multi-step arithmetic problems that require multi-hop numerical reasoning. **MMLU(MM)** evaluates general knowledge reasoning across various disciplines. For commonsense reasoning, we include **Arc Challenge (Arc)**, which requires logical inference and world knowledge, and **CommonsenseQA (CS)**, which tests a model's ability to infer unstated relationships between concepts. **StrategyQA (ST)** is designed to determine whether a multi-step reasoning process is necessary for answering a given question. Additionally, we include two tasks from the BIG benchmark: **Date Understanding (Date)**, which evaluates a model's ability to understand and reason about dates, and **Sports Understanding (SP)**, which assesses comprehension of sports-related scenarios. Finally, we incorporate two symbolic reasoning tasks: **CoinFlip(CF)**, which requires inference over probabilistic outcomes, and **Last Letter Concatenation (LLC)**, a simple task where the model must concatenate the last letters of words in a given input.

Domain-Specific Task For evaluating the model in a specialized domain, we use the **BioASQ** dataset, which focuses on biomedical question answering. Unlike other datasets, BioASQ was used both for fine-tuning and evaluation. The training set consists of 4917 instances, while the validation and test sets contain 20 and 310 instances, respectively. Fine-tuning on BioASQ allows us to analyze the adaptability of our approach to domain-specific knowledge and terminology.

Overall, our dataset selection ensures a comprehensive evaluation of truthfulness, reasoning capabilities, and domain adaptation, allowing us to systematically assess the strengths and weaknesses of our method.

Prompts For inference, we used the prompts listed in Table 7. These prompts were specifically designed to align with the tasks to ensure accurate and contextually appropriate model responses. Notably, we also employed the same set of prompts during the fine-tuning phase on the BioASQ dataset. By maintaining consistency between fine-tuning and inference, we ensured that the model learned and responded with coherence, improving its performance in biomedical question-answering tasks.

B.2 Training Details

For task-specific fine-tuning, we fine-tuned LLaMA 2-7B on the BioASQ dataset. We aligned the hyperparameter settings with prior research (Shen et al., 2024) to ensure consistency. We fine-tune LLaMA 2-7B using 4 A100 80G GPUs. To reduce GPU memory usage, FlashAttention (Dao, 2024) and DeepSpeed (Rasley et al., 2020) are used. Full details are provided in Table 8.

Hyperparameter	**Configuration**
Training Data	BioASQ
Training Epoch	2
Batch Size	128
Max Length	2048
Gradient Accumulation Steps	16
Learning Rate	2e-5
Weight Decay	0
Warmup Ratio	0.04
AdamW β_1, β_2	0.9, 0.999

Table 8: Hyperparameter settings used for fine-tuning.

B.3 Evaluation Metrics

For multiple-choice task TruthfulQA, we use three evaluation metrics (Lin et al.). MC1 checks whether the highest probability is assigned to the best answer. MC2 measures how much probability mass is allocated to correct answers compared to incorrect ones. MC3 ensures that every correct answer is ranked above all incorrect options.

For mathematical reasoning tasks, following prior work (Shen et al., 2024; Fei et al., 2024), we extract final numerical answers directly from generated solutions. For the commonsense reasoning tasks in Table 3, we adopt the evaluation method introduced in (Fei et al., 2024) and use GPT-4 for automatic scoring. The evaluation prompt used for GPT-4 assessment is provided in Table 9.

For the BioASQ dataset, we follow the evaluation guidelines and prior work for each data type (BioASQ Organizers, 2024; Shen et al., 2024). Factoid questions require generating a specific entity (e.g., disease, drug, or gene) and are evaluated using Strict Accuracy (SAcc.). List questions expect multiple entities and are scored with F1. Summary responses are long-form and assessed using ROUGE-2 for textual overlap. Yes/No questions are binary classification tasks, evaluated with accuracy.

Evaluation Prompt

Please evaluate whether the model's answer is correct given the question and the gold answer.

```
{Question}
{Model_answer}
{Gold_answer}
```

Determine the correctness of the model's answer based on whether the final answer is equivalent to the gold answer. Do not evaluate the intermediate steps or the rationale in the model's answer. Now, please output your scores and a short rationale below in JSON format by filling in the placeholders:

```
{Output Format}
```

Table 9: Evaluation prompt used by GPT-4 for reasoning tasks, assessing answer correctness based on final output equivalence to the gold answer.

C TruthfulQA Experiment

C.1 Experiment Details

To evaluate our method in the open-ended generation setting of the TruthfulQA benchmark, we adopted the re-evaluation framework proposed in the original TruthfulQA paper (Lin et al.). Since the original GPT-judge and GPT-info models are no longer publicly available, we employed an alternative open-source tool provided by the TruthfulQA re-evaluation repository[1]. This tool retrains a judge model based on LLaMA (7B and 13B) and groups responses by question for more robust evaluation. The final judge model was trained on the full dataset to assess response quality across multiple dimensions. We evaluate generation performance using the following four metrics:

- **Truthful (%)**: Percentage of responses rated factually correct.

[1] https://github.com/yizhongw/truthfulqa_reeval

- **Informative (%)**: Percentage of responses that provide meaningful information.
- **Truth * Info (%)**: Percentage of responses that are both truthful and informative.
- **Reject (%)**: Truthful responses that lack informativeness (i.e., evasive or generic responses).

C.2 Experiment Results

UCD demonstrates strong performance across all metrics. While the 7B-Chat model achieves the highest truthfulness rate, UCD matches this performance while also significantly improving informativeness. As a result, UCD achieves the highest *Truth * Info* score (85.07%) and the lowest *Reject* rate (5.02%), indicating its effectiveness in generating both accurate and meaningful responses while minimizing evasive outputs.

Model	%Truth ↑	%Info ↑	%T*I ↑	%Reject ↓
7B-Chat	**90.09**	89.23	79.44	10.65
13B-Chat	86.29	90.21	76.50	9.79
UCD (7B + 13B)	**90.09**	**94.86**	**85.07**	**5.02**

Table 10: TruthfulQA Generation Results. UCD achieves the best overall performance across all evaluation metrics.

C.3 Decoding Time Comparison

Model	Time (sec)	# of Tokens	Avg. Time per Token
70B-Chat	11.48	101	0.1137
CD	23.30	256	0.0910
UCD	8.63	115	0.0751

Table 11: Decoding efficiency across models within a single-column layout. CD and UCD use a combination of 13B-Chat and 7B-Chat models.

To assess the computational efficiency of our method, we analyze decoding time performance in Table 11. UCD achieves an average decoding time of 0.0751 seconds per token using 13B-Chat and 7B-Chat models on a single GPU, outperforming both CD (0.0910s/token) with the same model pair and the 70B-Chat model (0.1137s/token) running on two GPUs. These results demonstrate that UCD provides an efficient alternative to large-scale single-model decoding while maintaining superior performance, highlighting its practical scalability advantages.

D Qualitative Analysis

We present an additional qualitative analysis example to demonstrate the effectiveness of our approach on a reasoning benchmark. In this instance, we applied the UCD method with three models and evaluated its ability to determine the correct answer. As shown in Table 12, while other models failed to identify the correct answer, UCD successfully distinguished between pathways intended for transportation and those that are not, accurately aligning with the ground truth. This result highlights the robustness of UCD in reasoning-based tasks, reinforcing its effectiveness in enhancing factual accuracy and ensuring constraint satisfaction in complex decision-making scenarios.

Question
Which is NOT a pathway for transportation? A. radio tower B. road C. runway D. pipeline
Ground Truth
A. radio tower
Model Responses
7B-Base: A. radio tower B. road C. runway D. pipeline A_model: E. sewer F. water tower G. railroad [...] dam Q. river R. road S. bridge T. tunnel U. highway V. road W. canal X. dam Y. road Z. bridge Answer: The correct answer is “D”. **7B-Chat:** To determine which option is NOT a pathway for transportation, we need to evaluate each option based on its characteristics. [...] D. Pipeline: A pipeline is a pathway for transporting fluids, such as oil, gas, and water, from one place to another. While a pipeline can be used for transportation, it is not a pathway for transporting people or goods, as it is not designed for that purpose. Therefore, option D is NOT a pathway for transportation. In conclusion, the correct answer is option D, pipeline. **13B-Base:** A runway is a pathway for transportation, but it is not a pathway for transportation of goods or people.[...] A. radio tower B. road C. runway D. pipeline Answer: The correct answer is C.runway. [...] **Proxy Tuning:** To determine which option is NOT a pathway for transportation, we need to evaluate each option based on its characteristics. Here are the reasoning steps: A. Radio tower: A radio tower is a structure used for broadcasting radio waves, not for transportation. It is not a pathway for people or vehicles to move along, so option A is the correct answer. B. Road: A road is a pathway for transportation, designed for vehicles such as cars, buses, and trucks to move along. Therefore, option B is incorrect. C. Runway: A runway is a pathway for aircraft to take off and land, so it is a pathway for transportation. Therefore, option C is incorrect. D. Pipeline: A pipeline is a pathway for transporting fluids or gases, such as oil or natural gas, from one location to another. While a pipeline can be used for transportation, it is not a pathway for people or vehicles to move along, so option D is the correct answer. In conclusion, the correct answer is option D: pipeline. **UCD:** To answer this question, we need to identify the pathway for transportation and determine which option is not a pathway for transportation. Let’s go through the options: A. Radio tower: A radio tower is a structure used to transmit and receive radio waves, but it is not a pathway for transportation. It is a structure used for communication purposes, not for transporting people or goods. Therefore, option A is the correct answer. B. Road: A road is a pathway for transportation, specifically for vehicles such as cars, trucks, and buses. It is a physical structure designed for the movement of people and goods from one place to another. Therefore, option B is incorrect. C. Runway: A runway is a pathway for transportation, specifically for aircraft. It is a long, flat surface used for takeoff and landing of airplanes. Therefore, option C is incorrect. D. Pipeline: A pipeline is a pathway for transportation, specifically for liquids and gases. It is a system of pipes used to transport fluids from one place to another. Therefore, option D is incorrect. In conclusion, the correct answer is option A, radio tower, as it is not a pathway for transportation.

Table 12: Comparison of model responses on the Arc Challenge dataset using LLaMA2-7B-Base, LLaMA2-7B-Chat and LLaMA2-13B-Base. Only UCD correctly identifies the answer, while other models fail to identifies the reasoning answer.