REAL Sampling: Boosting Factuality and Diversity of Open-Ended Generation by Extrapolating the Entropy of an Infinitely Large LM

Haw-Shiuan Chang[1*] Nanyun Peng[2] Mohit Bansal[2]
Anil Ramakrishna[2] Tagyoung Chung[2]
[1]UMass Amherst CICS [2]Amazon AGI Foundations
hschang@cs.umass.edu
{pengnany,mobansal,aniramak,tagyoung}@amazon.com

Abstract

Decoding methods for large language models (LLMs) usually struggle with the tradeoff between ensuring factuality and maintaining diversity. In this paper, we propose REAL (**R**esidual **E**ntropy from **A**symptotic **L**ine) sampling[1], which predicts the step-wise hallucination likelihood of an LLM. When an LLM is likely to hallucinate, REAL lowers the p threshold in nucleus sampling. Otherwise, REAL sampling increases the p threshold to boost the diversity. To predict the step-wise hallucination likelihood without supervision, we construct a THF (**T**oken-level **H**allucination **F**orecasting) model, which predicts the asymptotic entropy (i.e., inherent uncertainty) of the next token by extrapolating the next-token entropies of an infinitely large language model from a series of LLMs with different sizes. If an LLM's entropy is higher than the asymptotic entropy (i.e., the LLM is more uncertain than it should be), the THF model predicts a high hallucination hazard, which leads to a lower p threshold in REAL sampling. In the FACTUALITYPROMPTS benchmark (Lee et al., 2022), we demonstrate that REAL sampling based on a 70M THF model can substantially improve the factuality and diversity of 7B LLMs simultaneously. After combined with contrastive decoding, REAL sampling outperforms 13 sampling methods, and generates texts that are more factual than the greedy sampling and more diverse than the nucleus sampling with $p = 0.5$.

1 Introduction

Hallucination is a major problem that limits the applications of LLMs (large language models), especially in open-ended generation tasks (Zheng et al., 2023; Huang et al., 2023; Tonmoy et al., 2024; Sun et al., 2024). Recent studies[2] show that an LLM often "knows" if it is hallucinating. The findings suggest that the decoding methods of LLMs are major sources of the hallucination.

Sampling is one of the most widely used decoding strategies in LLM due to its simplicity, efficiency, and high generation diversity (Holtzman et al., 2020; Hewitt et al., 2022; Meister et al., 2022). Nevertheless, recent studies show that hallucination often happens as the result of sampling the tokens with lower probabilities from a high-entropy distribution (van der Poel et al., 2022; Marfurt and Henderson, 2022; Manakul et al., 2023; Rawte et al., 2023; Varshney et al., 2023). Figure 1 (a) illustrates a simple example. When an LLM is uncertain about who is the screenwriter of a movie, the next-token distribution usually has a high entropy, where some incorrect answers receive high probabilities.

Nucleus (top-p) sampling (Holtzman et al., 2020) is one of the representative methods[3] proposed to alleviate the issue. By decreasing the constant global p threshold, we can trade the generation diversity for higher factuality (Dziri et al., 2021; Lee et al., 2022; Aksitov et al., 2023). For example, Figure 1 shows that a lower p threshold could reduce the chance of sampling the incorrect writer names in (a), but it would also eliminate the legitimate starts of the possible next sentences in (b). This tradeoff limits nucleus sampling's ability to generate both high diversity and high factuality outputs. Some existing methods such as typ-

*The work was mostly done at Amazon.

[1]Our code is released at https://github.com/amazon-science/llm-asymptotic-decoding

[2]Burns et al. (2022); Li et al. (2023); Azaria and Mitchell (2023); Slobodkin et al. (2023); CH-Wang et al. (2023); Orgad et al. (2024) show that we can predict hallucination based on its internal states and Agrawal et al. (2023); Guan et al. (2023); Manakul et al. (2023); Zhang et al. (2023a); Varshney et al. (2023) show that an LLM can sometimes improve itself by editing or verifying its own answer.

[3]OpenAI provides top-p sampling at https://platform.openai.com/playground?mode=chat.

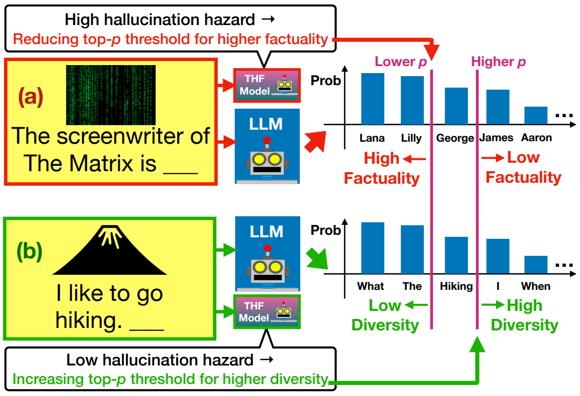


Figure 1: (a) For the factual question, only a few next tokens are correct but the target LLM assigns high probabilities to many tokens, so our THF model predicts the next token from the LLM is likely to be incorrect if using a large p threshold. (b) In contrast, many tokens could be used at the beginning of a sentence, so sampling from more tokens should increase the diversity without hurting the factuality.

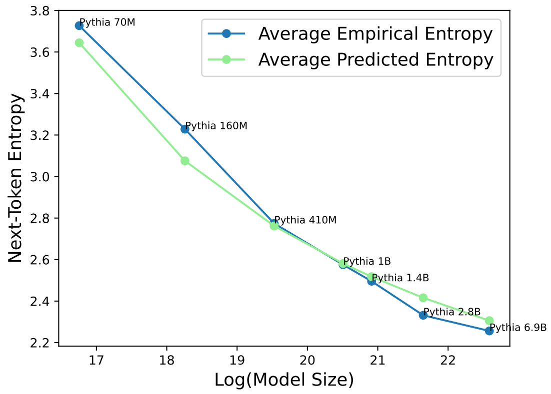


Figure 2: The entropies of the Pythia's distributions versus the model sizes in a logarithmic scale. The entropies are averaged across 9M tokens in Wikipedia. The blue entropy decay curve plots empirical entropies from Pythia LMs; the green curve is the entropies predicted by our THF model.

ical (Meister et al., 2022) and eta (Hewitt et al., 2022) sampling are proposed to adjust the threshold by characterizing the token-wise distributions of LLM. However, this distribution alone is often not enough to detect the hallucination. For example, both distributions in Figure 1 are similar but the high entropy of (a) arises due to the LLM's own limitation while that of (b) arises due to the "inherent uncertainty" of the task.

In this paper, we tackle this problem from a brand-new angle: estimating inherent uncertainty by extrapolating the entropy of LLMs with different sizes. Given several LLMs with different sizes in the same family, which are pretrained using the same corpus, we empirically observe the smaller average entropies of a larger LM distribution as shown in Figure 2.[4] As LLM's model size becomes larger, the entropy of its distribution should be closer to the inherent uncertainty. As a result, we can extrapolate the entropy decay curve to estimate the asymptotic entropy, the entropy from an imaginary LLM with an infinite size, which approximates the inherent uncertainty (i.e., ground truth entropy). For example, for the questions discussed in Figure 3 (a), the LLM tends to be more certain about the answer as the size of LLM increases, so we can reasonably expect the asymptotic entropy to be low. In contrast, the entropies from different model sizes in Figure 3 (b) should be similar, so the next token distribution should have a high asymptotic entropy / inherent uncertainty.

Based on this insight, we propose a tiny unsupervised model to predict the hazard of generating a nonfactual next token, called THF (**T**oken-level **H**allucination **F**orecasting) model. As shown in Figure 3, we parameterize the decay curves of next-token entropies for LLMs and use the THF model to predict the curve parameters. Next, the THF model estimates the LLM's hallucination hazard by computing the difference between the asymptotic entropy and the LLM's entropy, which we call the residual entropy (RE). If the LLM is much more uncertain than it should be (i.e., the LLM's entropy is much larger than the asymptotic entropy), the THF model would forecast a high RE and hence a high hallucination hazard.

Relying on the residual entropy predicted by our THF model, we propose a novel context-dependent decoding method for open-ended text generation, which we call 'REAL (**R**esidual **E**ntropy from **A**symptotic **L**ine) sampling'. REAL sampling adjusts the p threshold in the top-p (nucleus) sampling based on the forecasted hallucination hazard. For example, in Figure 1 (a), the THF model learns that a movie usually does not have many credited screenwriters but the LLM's distribution entropy is high, so REAL sampling should use a lower threshold to mitigate the hallucination. On the other hand, in Figure 1 (b), the THF model learns that the given

[4] Please see more discussions about why entropy decays as the size increases in Appendix D.

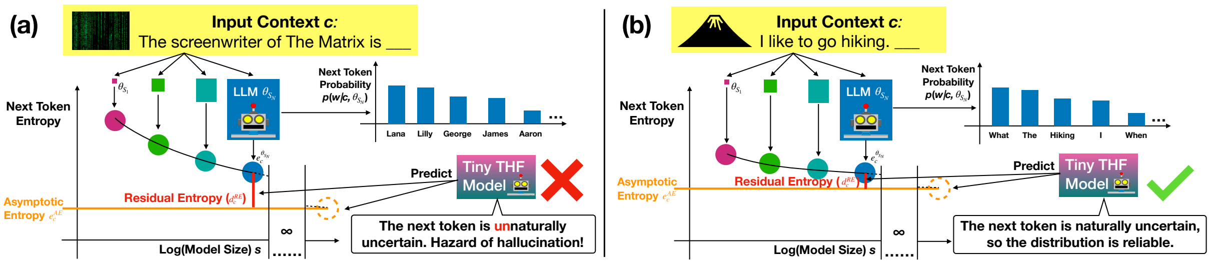


Figure 3: Given the input context, the LLMs with different sizes generate the next-token distributions. By extrapolating the curve using a tiny THF model, we estimate the asymptotic entropy, the entropy from an imaginary LLM with an infinite size, and measure the hallucination hazard using the residual entropy. (a) The LLM's entropy is much higher than the asymptotic entropy. This implies that the LLM is more uncertain than it should be and thus likely to hallucinate next. (b) LLM's high entropy is fine because the next token is inherently uncertain.

prompt can be completed in many different ways, so REAL sampling should increase the threshold to boost the generation diversity.

To the best of our knowledge, REAL sampling is the first sampling method that is tightly bounded by the ideal threshold that separates all the factual and nonfactual next tokens without making assumptions on the distribution of the nonfactual next tokens. Besides enjoying the theoretical guarantee, REAL sampling achieves significant and robust empirical improvements in various tasks. In our main experiments, we follow the evaluation protocol in FactualityPrompts (Lee et al., 2022) and find that sentences generated by Pythia 6.9B LLM (Biderman et al., 2023b) with our REAL sampling contains fewer hallucinations and fewer duplicated n-grams in both in-domain and out-of-domain settings. Our human evaluation indicates that REAL sampling not only improves factuality but also informativeness, fluency, and overall quality. Furthermore, we also demonstrate that the THF model improves performance on several hallucination detection tasks. Finally, we show that REAL sampling improves factuality without hurting LLM's story-writing capability.

Overall, our main contributions include

- We propose a THF model to predict the asymptotic entropy of an infinitely large LLM and propose REAL sampling that dynamically adjusts the sampling threshold based on the THF model.
- We theoretically prove that the threshold from our REAL sampling is upperbounded by the ideal value if the top predicted tokens are ideal and the residual entropy is estimated accurately.
- We demonstrate that the tradeoffs between factuality and diversity exist in the 13 state-of-the-art unsupervised sampling methods and our REAL sampling can consistently boost their factuality given the same diversity, and vice versa. Furthermore, we conduct comprehensive analyses on the THF model and REAL sampling, including evaluating our design choices and their generality using hallucination detection tasks.

2 Preliminary and Motivation

Given a context c and a next token candidate w in a vocabulary V, an LLM (θ) outputs the next token probability $p(w|c,\theta)$. Assuming w_i^c is the ith most likely token given the context c, top-p (nucleus) sampling first determines the number of tokens J such that

$$\sum_{i=1}^{J} p(w_i^c|c,\theta) \leq t^p < \sum_{i=1}^{J+1} p(w_i^c|c,\theta). \quad (1)$$

Then, it sets the probabilities from w_{J+1}^c to $w_{|V|}^c$ to 0 and re-normalizes the distribution of the top J tokens. In top-p sampling, t^p is a fixed global hyperparameter.

As illustrated in Figure 1, lower t^p would lead to a better factuality but worse diversity. In practice, many users would like to select from diverse responses. Furthermore, diverse and factual responses could also improve LLM's performance in reasoning tasks (Li et al., 2022b; Wang et al., 2022; Bertsch et al., 2023; Yao et al., 2023; Naik et al., 2023; Yu et al., 2024). If we can estimate the hallucination possibility of the next token, we can have a better context-dependent t^p. Notice that hallucination in this paper refers to the claims generated by LLMs whose non-factuality could be verified using existing literature.

It is notoriously challenging to estimate the hallucination likelihood of each token in general open-ended text generation tasks. One common strategy is to annotate if each generated token is factual and learn a classifier through supervised learning (Zhou et al., 2021). However, this approach has several drawbacks. First, human annotators often need to take a very long time to check if the generated text is factual, especially in an open-ended generation task, and provide token-level annotation. Second, due to the expense of getting the labels, the classifier is often trained using a few domain-specific examples that are generated by a specific LLM. Therefore, the classifier might not generalize well in other domains, other languages, or other LLMs. This motivates us to develop an unsupervised hallucination forecasting model that only needs the LLMs with different sizes. Then, we can apply our method to any domain, any language, and any LLM without the expensive human annotations.

3 Method

As the LLMs get larger, their performances increase at the cost of higher inference expense, so an institute often trains LLMs (e.g., GPT-4 family (OpenAI, 2023)) with different sizes using the same training data to let the users balance the cost and quality. We denote the parameters of an LLM family as $\{\theta_{s_1}, \theta_{s_2}, ...\theta_{s_N}\}$, where s_n is the logarithm of the number of parameters of the nth model. In this paper, we focus on improving the generation of the largest LLM (θ_{s_N}) in its family that can fit into our GPU memory.

In this section, we leverage the LLM family to train a THF model, which aims at predicting the entropy of the ideal (ground-truth) distribution without actually knowing the ideal distribution. In Section 3.1, we first parameterize the entropy decay curve of each next token prediction to predict asymptotic entropy (AE). In Section 3.2, we introduce the architecture of the THF model and how it learns to predict the residual entropy (RE). Finally, we describe REAL sampling, our context-dependent token truncation method based on the THF model in Section 3.3.

3.1 Parameterization and Extrapolation of the Entropy Decay Curve

As we see in Figure 3, the asymptotic entropy (AE) e_c^{AE} is the entropy of the next-token distribution from an infinitely-large LLM ($\lim_{s\to\infty}\theta_s$). Formally, we define e_c^{AE} as

$$\lim_{s\to\infty} e_c^{\theta_s} = \lim_{s\to\infty} \sum_w p(w|c,\theta_s)\log\left(p(w|c,\theta_s)\right). \quad (2)$$

To simplify our discussion, we assume an ideal distribution exists and the LLM's output approaches the ideal distribution as its size increases, so AE is the next-token inherent uncertainty.[5]

When training the LM to predict the next token, we cannot get the ideal distribution ($\lim_{s\to\infty} p(w|c,\theta_s)$), which is a critical challenge of text generation (Zhang et al., 2023b). Consequently, we cannot compute e_c^{AE} using Equation (2). Nevertheless, we can use the LLM family to get the pairs of the LLM size and its corresponding entropy $(s_i, e_c^{\theta_{s_i}})$ given each context c. Then, we can model the entropy decay by formulating it as a one-dimensional regression problem and estimate e_c^{AE} by extrapolation.

We parameterize the entropy decay trend using a fractional polynomial (Chang et al., 2020):

$$e_c(s) = z_c + b_c\left(\frac{a_{c,0.5}}{x_c(s)^{0.5}} + \sum_{k=1}^{K}\frac{a_{c,k}}{x_c(s)^k}\right), \quad (3)$$

where s is the logarithm of the model size , $x_c(s) = \max(1, q_c(s - g_c))$ is a normalized model size, $e_c(s)$ is our entropy prediction, and $a_{c,0.5}, a_{c,k}, b_c, q_c, g_c$, and z_c are the parameters of the curve. All the parameters are non-negative to ensure the non-increasing property of $e_c(s)$, so the estimation of asymptotic entropy $\hat{e}_c^{AE} = \lim_{s\to\infty} e_c(s) = z_c$.

Given a context c, one approach is to estimate all the $K+5$ parameters by fitting the $(s_i, e_c^{\theta_{s_i}})$ on the fly. However, this approach has several problems. First, it is time-consuming to run all the LLMs in the family and fit the curve. Second, we often cannot get many $(s_i, e_c^{\theta_{s_i}})$ pairs and the entropy signal of LLMs could be noisy, so the parameter estimation is unstable especially if we want to use a large degree of fractional polynomial K. To address the problems, we propose to use a tiny LM to predict the parameters in the next subsection.

3.2 Residual Entropy Prediction using the THF Model

The proposed THF (**T**oken-level **H**allucination **F**orecasting) model takes the input context and

[5]Although the scaling law has shown that the distribution of a larger language model is indeed closer to the ideal distribution (Kaplan et al., 2020), we acknowledge that the LLMs with infinite size might not output the ideal distribution in the real world due to the limited amount of pretraining data and other LLMs' limitations. We leave the study of the systematic distribution bias of the infinitely large language model as our future work.

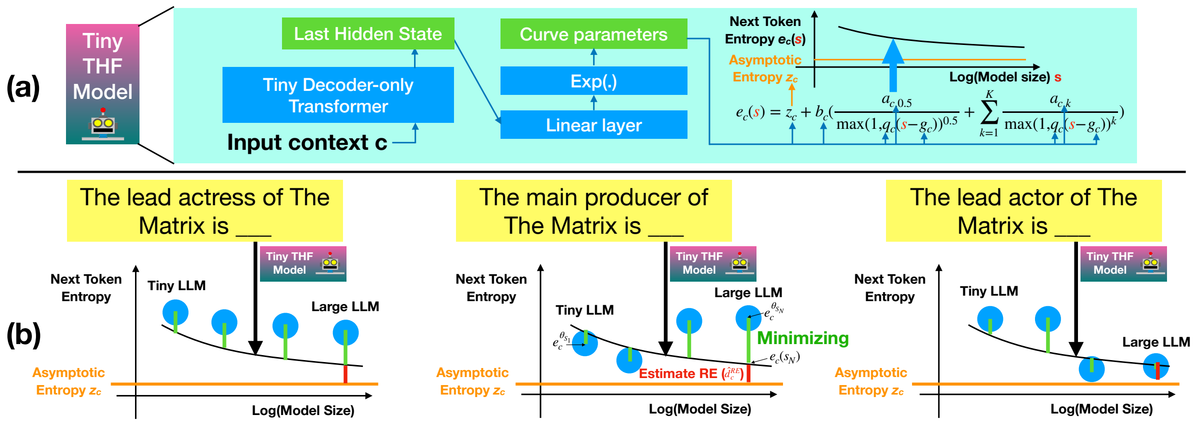


Figure 4: The architecture and the training of the THF model. We use the THF model to predict the parameters of the entropy decay curves and we train the THF model by minimizing the distances between the predicted entropy curves and the empirical entropies from the LLM family.

outputs the parameters of the entropy decay curve. As illustrated in Figure 4 (a), the THF model projects the last hidden state of a pretrained tiny LM decoder to a vector with $K+5$ variables, which are passed through an exponential layer to ensure the positivity of the output parameter predictions. Our experiment uses the smallest LM, θ_{s_1}, to initialize the weights of the LM.

We train the THF model by minimizing the root mean squared error (RMSE) between the predicted entropy ($e_c(s_i)$) and the empirical entropy from LLMs ($e_c^{\theta_{s_i}}$). Specifically, our loss of each batch could be written as

$$L = \sqrt{\frac{1}{|B|N} \sum_{c \in B} \sum_{i=1}^{N} (e_c^{\theta_{s_i}} - e_c(s_i))^2}, \quad (4)$$

where B is a training batch.

The entropy signal could be noisy even though all the LLMs are trained on the same corpus. For example, Figure 4 (b), LLM's entropy of similar contexts are very different, and LLMs with a larger size sometimes have a larger empirical entropy.

Using a tiny model to predict the entropy decay not only reduces the inference time but also stabilizes the parameter estimation. As a model gets smaller, it cannot memorize the small differences among similar input contexts (Biderman et al., 2023a), so similar inputs tend to bring about similar predictions. For example, when the tiny model receives three similar input contexts in Figure 4 (b), if its hidden states and output parameters for the entropy decay curves are all identical, the gradient descent would encourage the predicted curves to be close to all the empirical entropy measurements of similar context inputs, which effectively increases the number of $(s_i, e_c^{\theta_{s_i}})$ pairs and cancels out some noise of the empirical entropies.

As shown in Figure 3, we use the THF model to predict residual entropy (RE) during inference[6] as a measurement of the hallucination hazard:

$$d_c^{RE} = e_c^{\theta_{s_N}} - e_c^{AE} \approx \hat{d}_c^{RE} = e_c(s_N) - z_c. \quad (5)$$

It is worth mentioning that we cannot expect a tiny model to very accurately estimate the inherent uncertainty at every position, which requires the knowledge that even the generation LLM cannot memorize (e.g., how many screenwriters every movie has). Nevertheless, the tiny THF model could still learn that the entropy should be higher at the beginning of a clause but lower if the next token should be something very specific such as an entity. In our experiment, we found that such a rough estimation is sufficient to improve the state-of-the-art decoding methods.

3.3 REAL Sampling

We convert the residual entropy (RE) to the threshold between 0 and 1 for the cumulative probability in Equation (1) using

$$\hat{t}_c^p = \exp\left(\frac{-\hat{d}_c^{RE}}{T}\right) = \exp\left(\frac{-(e_c(s_N) - z_c)}{T}\right), \quad (6)$$

[6]Notice that although the entropy of LLMs, $e_c^{\theta_{s_N}}$, is measurable during the inference, we use the predicted entropy $e_c(s_N)$ to estimate the residual entropy $\hat{d}_c^{RE}$. This reduces the possible inconsistency between the LLM and the THF model and allows us to estimate the RE without actually running the LLM, which makes our method efficient in hallucination detection applications.

where T is our temperature hyperparameter used to control the tradeoff between factuality and diversity. When the T is high, the $\hat{t}_c^p$ would be closer to 1, so the generation diversity increases at the cost of the lower factuality.

Let's assume the top tokens from the LLM are factual and its top token distribution is correct (i.e., the same as the distribution of an infinitely large LLM after normalization). Then, there is an ideal threshold g_c^p for the LLM, which sums the probabilities of all the top factual tokens (e.g., the lower p in Figure 1 (a)), and we can derive an elegant relation between the ideal threshold g_c^p and the threshold of REAL sampling (t_c^p) based on an ideal THF model.

Theorem 3.1. *If the residual entropy is estimated perfectly (i.e., $\hat{d}_c^{RE} = d_c^{RE}$), and there is an ideal threshold g_c^p such that the distribution of the top tokens above the threshold is ideal, then*

$$t_c^p = \exp(\frac{-d_c^{RE}}{T}) \leq (g_c^p)^{\frac{1}{T}}. \quad (7)$$

Please see our proof in Appendix A. That is, when the ideal threshold exists and our RE is accurate, our threshold t_c^p is not larger than the ideal threshold raised to power $\frac{1}{T}$.

The theoretical guarantees that REAL sampling can exclude all hallucinated token candidates when $T = 1$ and the preconditions are satisfied. Furthermore, it reveals the role of T in the REAL sampling and explains why we should use this exponential function instead of other formulas.

4 Experiments

We first evaluate REAL sampling in open-ended text generation tasks using FACTUALITYPROMPTS. Section 4.1 compares REAL sampling with 13 sampling baselines and Section 4.2 reports our ablation studies to justify each of our design choices. The human evaluation for FACTUALITYPROMPTS in Section 4.3 further strengthens our conclusions. Next, we explore other applications such as hallucination detection using the THF model in Section 4.4 and story writing using REAL sampling in Section 4.5.

We use the de-duplicated variant of Pythia LLM series (Biderman et al., 2023b) to train our THF model. The training corpus consists of 5M lines from Wikipedia 2021 and 5M lines from OpenWebText (Radford et al., 2019) (around 5.6% of their text). By default, we use Pythia 6.9B as our LLM generation model (θ_{s_N}) and the THF model is based on the transformer from Pythia 70M.

4.1 Retrieved-based Evaluation in FactualityPrompts

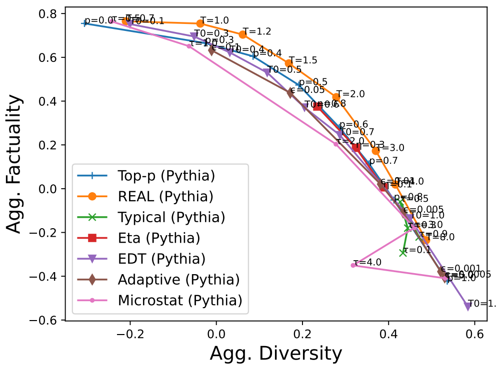
Lee et al. (2022) propose an evaluation benchmark, FACTUALITYPROMPTS, that first lets different LLMs generate continuations of each prompt sentence and retrieves the relevant Wikipedia pages (Hanselowski et al., 2018) to evaluate the generation factuality. There are 8k factual prompts and 8k nonfactual prompts from FEVER (Thorne et al., 2018), which test if LLM could generate the factual continuations even if the prompt is not factual.

Metrics: FACTUALITYPROMPTS uses Entail$_R$ and NE$_{ER}$ to evaluate the factuality. Entail$_R$ is the ratio of the generated sentences entailed by the sentences in the relevant Wikipedia pages, while NE$_{ER}$ is the ratio of the entities that are not in the pages. Lee et al. (2022) use distinct n-grams (Dist-n) (Li et al., 2016) to measure the diversity across generations and use repetition ratio (Rep) (Holtzman et al., 2020) to measure the diversity within a generation. A good method should get high Entail$_R$ and Dist-n, but low NE$_{ER}$ and Rep.

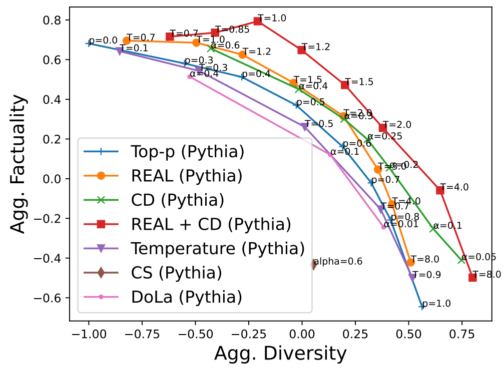
To compare the performances of methods in one figure, we first normalize all metrics from a generation LLM using max-min normalization and average the scores from all the prompts as Entail$_{Rn}$, NE$_{ERn}$, Dist-2$_n$, and Rep$_n$. Next, we define the aggregated metrics Agg. Factuality $=$ Entail$_{Rn}$ $-$ NE$_{ERn}$ and Agg. Diversity $=$ Dist-2$_n$ $-$ Rep$_n$. The scores of the original 4 metrics will be reported in Figures 11 and 12.

Methods: Our baselines include six entropy-based decoding methods: typical (Meister et al., 2022), eta (Hewitt et al., 2022), EDT (Zhang et al., 2024), adaptive (Zhu et al., 2024), microstat (Basu et al., 2021), and EAD w/o ELI (Arora et al., 2023) sampling, one heuristic-based method: factual (F) (Lee et al., 2022) sampling, two popular thresholding methods: top-p (Holtzman et al., 2020) and top-k (Fan et al., 2018), and four distribution modification methods: temperature (Ficler and Goldberg, 2017) sampling, contrastive search (CS) (Su and Collier, 2022) , contrastive decoding (CD) (Li et al., 2022a), and DoLa (Chuang et al., 2023). Our methods include

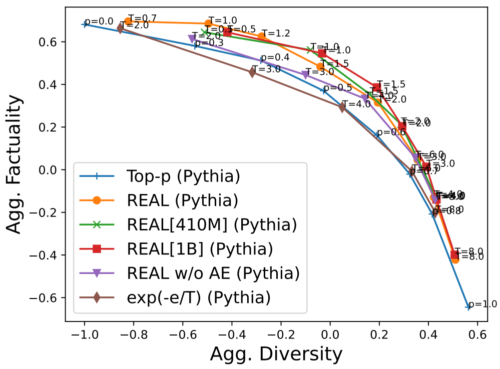
- **REAL (Pythia)**: REAL sampling using 70M THF model and the degree of the fractional poly-



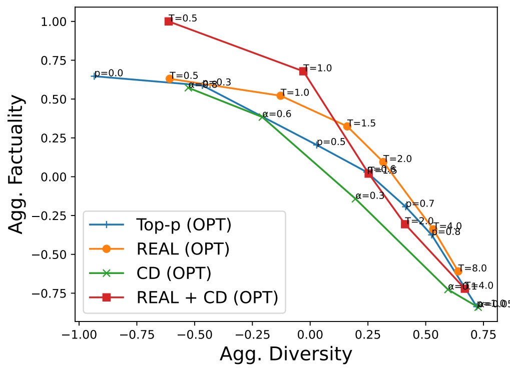
(a) REAL vs Thresholding Methods



(b) REAL vs Distribution Modifications



(c) Comparison of the THF model sizes and ablation on our dynamic thresholding formula



(d) Using OPT-6.7b as the generation LLM and the THF model trained on Pythia.

Figure 5: Open-ended text generation performance comparison between REAL sampling and state-of-the-art unsupervised thresholding methods. The factuality and diversity are evaluated using the FactualityPrompts benchmark from Lee et al. (2022). We also conduct an ablation study and compare REAL sampling with distribution modification methods. CS and CD refer to contrastive search and contrastive decoding, respectively.

nomial $K = 10$ in Equation (3).

- **REAL + CD (Pythia)**: Combining our methods with contrastive decoding (**CD**) (Li et al., 2022a). We first truncate the tokens using the threshold $\hat{t}^p_c$ in REAL sampling and apply the contrastive decoding (i.e., computing the probabilities of the top tokens using the logit differences between θ_{s_N} and θ_{s_1}).
- *** (OPT) or * (LLaMA)**: In the methods, we replace the Pythia 6.9B with OPT-6.7b (Zhang et al., 2022) or OpenLLaMA2-7b (Geng and Liu, 2023) as the generation LLM, respectively. Notice that the THF model is still trained using the Pythia family.

	Pearson r	**R2**	**MSE (↓)**	**Mean L1 (↓)**
REAL[Exp]	0.843	0.708	0.786	0.64
REAL[Logistic]	0.842	0.707	0.788	0.641
REAL	0.843	0.71	0.78	0.639
REAL[K=6]	0.843	0.71	0.781	0.639
REAL[K=4]	**0.844**	**0.712**	**0.776**	**0.636**
REAL[K=3]	0.843	0.709	0.782	0.64
REAL[K=2]	**0.844**	0.711	0.778	0.638
REAL[K=1]	**0.844**	0.711	0.777	0.641

Table 1: Comparing LLM's entropy predictions $e_c(s_N)$ from different THF models with empirical entropies $e_c^{\theta_{s_N}}$ using Pearson correlation coefficient (r), mean squared error (MSE), average L1 norm (Mean L1), and coefficient of determination (R2) (Draper and Smith, 1998). REAL means REAL[K=10], which uses fractional polynomials in Equation (3).

Main Results: In Figure 5a, **REAL** sampling consistently outperforms **top-**p and all other thresholding methods across the whole spectrum. Overall, we often improve the factuality more when the temperature T is low (i.e., diversity is relatively low) probably because lower T emphasizes the effect of $\hat{d}^{RE}_c$ in Equation (6). Notice that some diversities actually come from hallucination, so it is hard to increase the diversity and the factuality at the same time, especially by only adjusting

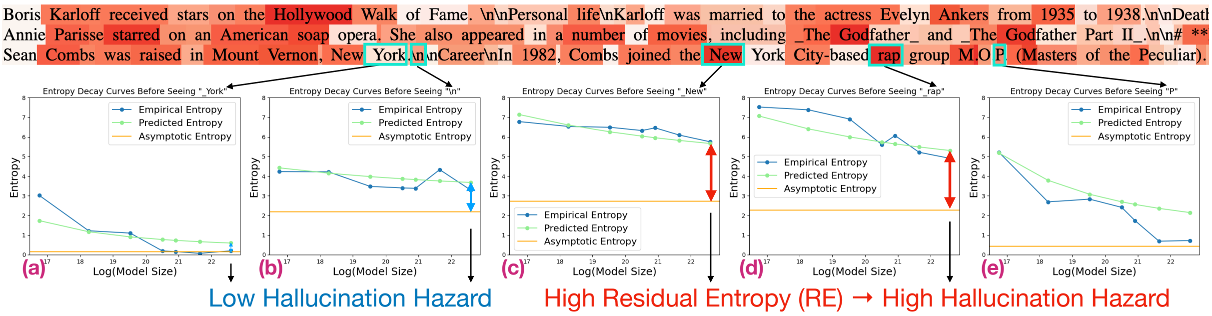


Figure 6: The visualization of the estimated residual entropy ($\hat{d}_c^{RE}$) and entropy decay curves. The top three lines come from the first three testing factual prompts in FACTUALITYPROMPTS and continuations generated by Pythia 6.9B. A darker red highlights a larger hallucination hazard ($\hat{d}_c^{RE}$) forecasted by our THF model based on the context before the position, and thus it leads to a smaller p threshold in REAL sampling. The bottom figures (a)-(e) visualize the empirical entropy decay curves from five tokens in the third example, along with the corresponding curves predicted by our THF model and asymptotic entropies.

the truncation threshold without changing the distribution of LLM like our methods.

In Figure 5b, **REAL + CD** is prominently better than using contrastive decoding **CD** alone. One possible reason is that CD might reduce the diversity when there are many correct next tokens and REAL sampling could alleviate the problem. The result shows that REAL sampling is complementary with other distribution modification methods.

To evaluate our generalization capability, we use the THF model trained on the Pythia family to improve OPT and OpenLLaMA2. Figures 5d and 9f indicates that REAL sampling can still improve the factuality, and the improvement is especially prominent if CD is used.

To explain the strong generalization ability across the serious misalignment between the training and testing objectives, we visualize the residual entropy (RE) from our THF model in Figure 6. We observe that the residual entropy tends to be larger at the positions where the LLMs generally are more likely to hallucinate. For example, in Figure 6 (c), the THF model forecasts a high hallucination hazard for '*New*', which is the first token in an entity name. Nevertheless, we also observe that the THF model cannot always predict LLM's entropy accurately due to its small size, and (e) is an example.

Complementary Results in Appendix: At the high diversity side, Figures 9a and 9c shows that **top-k** and **EAD w/o ELI** outperforms **top-p** sampling, respectively, while Figure 9b shows that the factual-nucleus (**F**) sampling outperforms **top-p** sampling at the low diversity side. Notice that factual-nucleus sampling relies on the heuristic/assumption that hallucination is more likely to happen near the end of the sentence. The assumption might not work well in some languages or applications such as code generation.

REAL sampling could be easily combined with these approaches to boost their performance. In Figure 9a, Figure 9b, and Figure 9c, the combinations are often significantly better than using REAL sampling alone.

Speed Comparison Without optimizing for speed[7], our current naive implementation simply runs the 70M THF model at every decoding step. Even so, the decoding time only increases around 11% (from 7.46 to 8.29 seconds).

4.2 Ablation Study in FactualityPrompts

Methods: Our ablated methods include

- **REAL[410M]** or **REAL[1B] (Pythia)**: REAL sampling using 410M or 1B THF model.
- **REAL w/o AE (Pythia)**: Our method after removing the asymptotic entropy (AE) estimation as $\hat{t}_c^p = \exp(\frac{-e_c(s_N)}{T})$.
- **exp(-e/T) (Pythia)**: Instead of using the THF model to predict the entropy, we estimate the entropy from the LLM and set $\hat{t}_c^p = \exp(\frac{-e_c^{\theta_{s_N}}}{T})$. The method simply reduces the p threshold whenever encountering a flat distribution (e.g., distributions in both (a) and (b) of Figure 1).
- **REAL[exp]**: REAL sampling using an exponential (exp) decay function ($e_c(s) = z_c +$

[7] Since the size of our 70M THF model is 100 times smaller than 7B LLM, the inference time of the THF model should be negligible if we parallelly run both the LLM and THF model at each decoding step.

Model Comparision	Overall			Factuality			Informativeness			Fluency		
	win	tie	loss	win	tie	loss	win	tie	loss	win	tie	loss
REAL vs Top-p	**29.5**†	53.5	17	**26**	53.5	20.5	**26**	51	23	**24.5**	58.5	17
CD vs Top-p	**34.5**†	46	19.5	**31**	49.5	19.5	**33.5**	41.5	25	**25.5**	53.5	21
REAL vs CD	**25.5**	49	**25.5**	22.5	46.5	**31**	27.5	41.5	**31**	**23.5**	60	16.5
REAL+CD vs CD	**27**	53	20	**23.5**	53.5	23	**26.5**	52	21.5	**19.5**	64.5	16
REAL+CD vs REAL	**27.5**	50.5	22	**30**	47.5	22.5	**31.5**	40	28.5	21.5	56.5	**22**
REAL+CD vs Top-p	**38**†	44	18	**35.5**†	43	21.5	**30.5**	45.5	24	**27**	54.5	18.5

Table 2: Human evaluation for the open-ended generation. We highlight the better number between win and loss. † the win is significantly more than loss under Fisher's exact test (Fisher, 1922) with $p = 0.05$.

$b_c \exp(-\max(0, q_c(s - g_c))))$ in Equation (3).

- **REAL[logistic]**: REAL sampling using a logistic function ($e_c(s) = z_c + \frac{b_c}{1+\exp(\max(0,q_c(s-g_c))}$).
- **REAL[K=*]**: REAL sampling that set the maximal degree K in Equation (3) as *. For example, when $K = 1$, $e_c(s) = z_c + b_c(\frac{a_{c,0.5}}{x_c(s)^{0.5}} + \frac{a_{c,1}}{x_c(s)})$.

Main Results: In Figure 5c, the worse performance of **REAL w/o AE** (especially with low diversity) verifies the effectiveness of predicting asymptotic entropy (AE). The 70M THF model (**REAL**) performs similarly compared to the larger THF models (**REAL (410M)** and **REAL (1B)**); using the LLM entropy predicted by THF model (**REAL w/o AE**) is much better than using the empirical LLM entropy (**exp(-e/T)**). These two results in our ablation study suggest that a tiny model indeed stabilizes the entropy decay curve prediction. Table 1 indicates that all parameterizations perform similarly well ($r = 0.84$) in terms of predicting the entropy of 6.9B LLM, even though our THF model only has 70M parameters.

Complementary Results in Appendix: Figures 9d and 9e shows that our scores in FactualityPrompts are not sensitive to the parameterization functions and polynomial degrees K, especially when $K > 1$. In Figure 10, we observe that a more complex THF model (i.e., a higher K or a larger model size) seems to perform slightly better given factual prompts due to its prediction power but perform slightly worse given nonfactual prompts. Since THF is trained only on factual text, the results suggest that a more complex model could perform better in an in-domain setting.

4.3 Human Evaluation in FactualityPrompts

To verify that our methods are still better from the humans' perspective, we ask the workers from Amazon Mechanical Turk (MTurk) to evaluate the factuality and the quality of the generated continuations using the Internet. Given 100 factual prompts in FactualityPrompts, we generate the next sentences using **Top-p** ($p = 0.6$), **REAL** ($T = 2.0$), **CD** ($\alpha = 0.3$), and **REAL + CD** ($T = 1.5$) due to their similar diversities.

Results: In Table 2, our methods constantly outperform the corresponding baselines (i.e., **REAL** wins **Top-p** more and **REAL + CD** wins **CD** more) and the improvement of **REAL + CD** vs **Top-p** is larger than **CD** vs **Top-p**. The factuality evaluation results verify the effectiveness of the retrieved-based evaluation. Furthermore, our methods also achieve better informativeness and fluency. Consequently, we get the largest improvement in the overall metric.

4.4 Hallucination Detection for Open-ended Text Generation

Perplexity and entropy are widely used to detect the hallucination (van der Poel et al., 2022; Marfurt and Henderson, 2022; Muhlgay et al., 2023; Manakul et al., 2023; Rawte et al., 2023; Varshney et al., 2023). However, high perplexity or entropy could mean multiple correct answers instead of hallucination as in Figure 1 (b), so we test if the residual entropy (RE) and asymptotic entropy (AE) could be useful unsupervised signals for the hallucination detection tasks.

Setup: We test the features using three hallucination detection datasets: Factor (Muhlgay et al., 2023), extended True-False dataset (TF ext) (Azaria and Mitchell, 2023), and HaDes (Liu et al., 2022). The hallucination datasets are created using very different methods and none of the input text comes from Pythia. Factor (Muhlgay et al., 2023)[8] creates nonfactual sentence continuations by revising the factual continuation given a

[8] https://github.com/AI21Labs/factor MIT license

Dataset →	Factor						TF ext		HaDes		Avg
Creation Method →	Revising a Factual Sentence using ChatGPT						Template + Table		BERT Infill		
Subset / Size →	Wiki / 47025		News / 7663		Expert / 355		All / 9830		All / 1000		
Feature Subsets ↓ Metrics →	1-4 ACC	AUC	1-4 ACC	AUC	1-4 ACC	AUC	ACC	AUC	ACC	AUC	
1 Feature ($6.9B\_per$)	0.374	0.315	0.367	0.312	0.347	0.290	0.619	0.691	**0.528**	**0.599**	0.444
2 Features ($6.9B\_per$ + $heur\_ent$)	**0.424**	**0.322**	0.359	**0.313**	0.347	0.300	0.624	0.700	0.503	0.581	0.447
2 Features ($6.9B\_per$ + RE)	0.393	0.319	**0.390**	0.303	**0.364**	**0.320**	**0.635**	**0.711**	0.521	0.580	**0.454**
6 Features (6.9B and 70M)	0.490	**0.341**	0.432	**0.326**	0.534	**0.356**	0.654	0.754	**0.578**	0.646	0.511
All (6.9B, 70M, RE, and AE)	**0.498**	**0.341**	**0.465**	**0.326**	**0.619**	0.346	**0.671**	**0.769**	0.565	**0.669**	**0.527**

Table 3: Hallucination detection in open-ended text generation. A random forest classifier predicts the hallucination using the features from Pythia 6.9B LLM, Pythia 70M LM, and THF model. 1 Feature (6.9B per) refers to only using the perplexity of Pythia 6.9B to detect hallucination (Muhlgay et al., 2023; Varshney et al., 2023). The average of all the scores are reported and the better performances in each section are highlighted.

context using ChatGPT, HaDes (Liu et al., 2022)[9] provides human factuality labels on the phrases infilled by BERT, and TF ext (Azaria and Mitchell, 2023)[10] mostly uses templates and tables in different topics to create the factual and nonfactual sentences. Our task is to classify these sentences (continuations) into either factual or nonfactual classes.

We use the training and testing split in HaDes. For Factor and TF ext, we split each subset into equally large training set and testing set. We train a random forest classifier with 100 estimators to combine these unsupervised features from the input phrase/sentence.

Metrics: The factuality classification tasks are evaluated using the area under the precision recall curve (AUC) and accuracy (ACC). In the Factor dataset, one of the four sentence continuations is factual. Thus, we follow Muhlgay et al. (2023) to measure the accuracy of detecting the factual sentence (1-4 ACC) instead.

Methods: We consider the following features:

- Perplexity of Pythia 6.9B ($6.9B\_per$),
- Entropy of Pythia 6.9B ($6.9B\_ent$),
- Perplexity of Pythia 70M ($70M\_per$),
- Entropy of Pythia 70M ($70M\_ent$),
- $\sqrt{6.9B\_per \cdot \max(0, 70M\_per - 6.9B\_per)}$ ($heur\_per$)
- $\sqrt{6.9B\_ent \cdot \max(0, 70M\_ent - 6.9B\_ent)}$ ($heur\_ent$)
- $\hat{d}_c^{RE}$ in Equation (5) (RE)
- z_c in Equation (3) (AE),

where all features are averaged across the tokens in the input phrase/sentence. Given a subset of the above features, we conduct an exhaustive feature selection to boost/stabilize the performance.

[9] https://github.com/microsoft/HaDes MIT license

[10] https://github.com/balevinstein/Probes/ MIT license

	Wining Rate (500 continuations)				(8k continuations)	
	Flu.	Coh.	Lik.	Overall	Dist-2	Rep (↓)
Top-p	50	50	50	50	**18.600**	7.463
REAL	**53**	**53.4**	**52.6**	**52.6**	17.952	**4.563**

Table 4: Out-of-domain creative writing experiment. The generation model is Pythia 6.9B and the winning rates on fluency, coherency, likability, and overall are measured using GPT3.5 against Top-p sampling with $p = 0.5$. REAL means REAL sampling ($T = 1.8$).

We would like to know if we can approximates RE without performing extrapolation, so we design a simple hallucination detection heuristic $heur\_ent$. The goal of $heur\_ent$ is to detect the large LLM entropy $6.9B\_ent$ and the large difference between $6.9B\_ent$ and $70M\_ent$, which induce a high hallucination hazard in Figure 3 (a).

Results: In Table 3, **2 Features** ($6.9B\_per$ + RE) usually outperforms **2 Features** ($6.9B\_per$ + $heur\_ent$) and **1 Feature** ($6.9B\_per$), which indicates that adding the RE features can improve the widely-used perplexity measurement of LLM (Muhlgay et al., 2023; Varshney et al., 2023) and the improvement cannot be achieved by the simple heuristics using the similar signal. Similarly, compared to **6 Features (6.9B and 70M)**, the better performance of **All (6.9B, 70M, RE, and AE)** demonstrates that even letting the random forest combine all the features from the Pythia 6.9B and 70M, residual entropy (RE) and asymptotic entropy (AE) from our THF model still provide extra information for hallucination detection. The results suggest that RE and AE could be auxiliary unsupervised signals that improve the entropy-based hallucination detection methods.

4.5 Out-of-Domain Creative Writing

Creative writing is not the focus of this paper because the hallucination problem is usually not serious in the tasks. Nevertheless, we still evaluate our methods on a story-writing task. In the task, the prompt is composed of three example stories from the ROC story dataset (Mostafazadeh et al., 2016) and the first two sentences from the fourth story. Then, we use different decoding methods to complete the fourth story and control their hyper-parameters to have similar Dist-2. Finally, we use *gpt-3.5-turbo-0125* to evaluate the winning rate of REAL sampling against top-p sampling in four aspects.

Results: In Table 4, **REAL** is similar to **top-p** even when the THF model's training data (i.e., Wikipedia and OpenWebText) do not include lots of short stories. This shows that REAL sampling could improve the factuality of **top-p** sampling without sacrificing its creative writing ability.

5 Related Work

Due to the importance of LLM's hallucination problems, various mitigation approaches are proposed. For a comprehensive discussion, please see the recent surveys from Huang et al. (2023); Tonmoy et al. (2024). Nevertheless, as far as we know, REAL sampling is the first method that can improve both factuality and diversity in open-ended text generation without annotations or domain-specific heuristics/assumptions.

Some methods can improve the factuality by relying on domain-specific assumptions. For example, Lee et al. (2022) assume the hallucination is more likely to appear in the latter part of a sentence. Burns et al. (2022) assume there is a set of statements that are either true or false. Several studies (van der Poel et al., 2022; Marfurt and Henderson, 2022; Chang et al., 2023; Shi et al., 2023; Chen et al., 2023) reduce the intrinsic hallucination by assuming that the generated text should be relevant to a source document. A more recent work (Luo et al., 2025) assumes that LLMs store the knowledge on the higher layer. These methods might not be applicable to other domains (e.g., other languages or open-ended text generation tasks) and could (potentially) be combined with our method to achieve better performance in the specific domain (e.g., see Figure 9b).

In terms of methodology, our method is related to some recent extrapolation-based methods in other applications. For example, Das et al. (2024) use a linear regressor to extrapolate the distribution of a deeper LM, Lu et al. (2024) extrapolate the probability distribution to obtain negative examples for text quality assessment, and Zheng et al. (2024) extrapolate the weights of an LM after training on more preference data. Chang et al. (2024) is our follow-up work that uses a similar idea to extrapolate the probability distribution of an infinitely-large LLM and address the limitations of contrastive decoding. However, none of them studies the threshold for sampling the next-token distribution.

6 Conclusion

Figure 1 suggests that it is difficult or sometimes even impossible in open-ended text generation tasks to predict the hallucination likelihood of the next token only based on the LLM's distribution without considering the inherent uncertainty of the task. In this paper, we demonstrate the feasibility of training a tiny model to forecast the hallucination hazard of LLM without supervision and domain-specific heuristics. Based on this finding, we propose REAL sampling along with its theoretical guarantee. Our comprehensive experiments indicate that most existing sampling methods cannot consistently outperform top-p sampling in FactualityPrompts. In contrast, our proposed REAL sampling not only outperforms top-p sampling but also can be combined with other decoding methods (e.g., contrastive decoding) to further reduce hallucination. We also demonstrate a THF model trained on one LLM family could be used to forecast/detect the hallucination from the LLM from another family, which highlights the strong out-of-domain generalization ability of our THF model.

7 Acknowledgement

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor. We would like to thank Utkarsh Lal. His investigation on measuring computational humor provides some inspiration for this project. We also want to thank the anonymous reviewers and our editor, Emily Pitler, for helping us improve this paper.

References

Ayush Agrawal, Lester Mackey, and Adam Tauman Kalai. 2023. Do language models know when they're hallucinating references? *ArXiv preprint*, abs/2305.18248.

Renat Aksitov, Chung-Ching Chang, David Reitter, Siamak Shakeri, and Yunhsuan Sung. 2023. Characterizing attribution and fluency tradeoffs for retrieval-augmented large language models. *ArXiv preprint*, abs/2302.05578.

Kushal Arora, Timothy J O'Donnell, Doina Precup, Jason Weston, and Jackie CK Cheung. 2023. The stable entropy hypothesis and entropy-aware decoding: An analysis and algorithm for robust natural language generation. *arXiv preprint arXiv:2302.06784*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *ArXiv preprint*, abs/2304.13734.

S Basu, GS Ramachandran, NS Keskar, and LR Varshney. 2021. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew R Gormley. 2023. It's MBR all the way down: Modern generation techniques through the lens of minimum bayes risk. In *Proceedings of the Big Picture Workshop*, pages 108–122.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. 2023a. Emergent and predictable memorization in large language models. *ArXiv preprint*, abs/2304.11158.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023b. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*.

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. 2023. Do androids know they're only dreaming of electric sheep? *ArXiv preprint*, abs/2312.17249.

Chung-Ching Chang, David Reitter, Renat Aksitov, and Yun-Hsuan Sung. 2023. Kl-divergence guided temperature sampling. *ArXiv preprint*, abs/2306.01286.

Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. 2024. Explaining and improving contrastive decoding by extrapolating the probabilities of a huge and hypothetical lm. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Haw-Shiuan Chang, Shankar Vembu, Sunil Mohan, Rheeya Uppaal, and Andrew McCallum. 2020. Using error decay prediction to overcome practical issues of deep active learning for named entity recognition. *Machine Learning*, 109:1749–1778.

Wei-Lin Chen, Cheng-Kuang Wu, Hsin-Hsi Chen, and Chung-Chi Chen. 2023. Fidelity-enriched contrastive search: Reconciling the faithfulness-diversity trade-off in text generation. *arXiv preprint arXiv:2310.14981*.

Cheng-Han Chiang and Hung-yi Lee. 2023. A closer look into automatic evaluation using large language models. In *EMNLP 2023 Findings*.

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*, abs/2309.03883.

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*.

Norman R Draper and Harry Smith. 1998. *Applied regression analysis*, volume 326. John Wiley & Sons.

Nouha Dziri, Andrea Madotto, Osmar Zaïane, and Avishek Joey Bose. 2021. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2197–2214, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Ronald A Fisher. 1922. On the interpretation of χ 2 from contingency tables, and the calculation of p. *Journal of the royal statistical society*, 85(1):87–94.

Xinyang Geng and Hao Liu. 2023. OpenLLaMA: An open reproduction of LLaMA.

Jian Guan, Jesse Dodge, David Wadden, Minlie Huang, and Hao Peng. 2023. Language models hallucinate, but may excel at fact verification. *ArXiv preprint*, abs/2310.14564.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium. Association for Computational Linguistics.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ArXiv preprint*, abs/2311.05232.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

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

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

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. 2022b. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Sidi Lu, Hongyi Liu, Asli Celikyilmaz, Tianlu Wang, and Nanyun Peng. 2024. Open-domain text evaluation via contrastive distribution methods. In *Forty-first International Conference on Machine Learning*.

Wen Luo, Feifan Song, Wei Li, Guangyue Peng, Shaohang Wei, and Houfeng Wang. 2025. Odysseus navigates the sirens' song: Dynamic focus decoding for factual and diverse open-ended text generation. *arXiv preprint arXiv:2503.08057*.

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

Andreas Marfurt and James Henderson. 2022. Unsupervised token-level hallucination detection from summary generation by-products. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 248–261, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2022. Typical decoding for natural language generation. *ArXiv preprint*, abs/2202.00666.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *ArXiv preprint*, abs/1604.01696.

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

Ranjita Naik, Varun Chandrasekaran, Mert Yuksekgonul, Hamid Palangi, and Besmira Nushi. 2023. Diversity of thought improves reasoning abilities of large language models. *arXiv preprint arXiv:2310.07088*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2024. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. *arXiv preprint arXiv:2410.02707*.

Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, SM Tonmoy, Aman Chadha, Amit P Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models–an extensive definition, quantification, and prescriptive remediations. *ArXiv preprint*, abs/2310.04988.

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

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory unanswerablity: Finding truths in the hidden states of over-confident large language models. *ArXiv preprint*, abs/2310.11877.

Yixuan Su and Nigel Collier. 2022. Contrastive search is what you need for neural text generation. *arXiv preprint arXiv:2210.14140*.

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bhavya Kailkhura, Caiming Xiong, Chao Zhang, Chaowei Xiao, Chunyuan Li, Eric Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, Willian Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, and Yue Zhao. 2024. TrustLLM: Trustworthiness in large language models.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *ArXiv preprint*, abs/2401.01313.

Lifu Tu, Semih Yavuz, Jin Qu, Jiacheng Xu, Rui Meng, Caiming Xiong, and Yingbo Zhou. 2023. Unlocking anticipatory text generation: A constrained approach for faithful decoding with large language models. *ArXiv preprint*, abs/2312.06149.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *ArXiv preprint*, abs/2307.03987.

David Wan, Mengwen Liu, Kathleen McKeown, Markus Dreyer, and Mohit Bansal. 2023. Faithfulness-aware decoding strategies for abstractive summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2864–2880, Dubrovnik, Croatia. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv preprint*, abs/2305.10601.

Fangxu Yu, Lai Jiang, Haoqiang Kang, Shibo Hao, and Lianhui Qin. 2024. Flow of reasoning: Training llms for divergent problem solving with minimal examples. *arXiv preprint arXiv:2406.05673*.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023a. How language model hallucinations can snowball. *ArXiv preprint*, abs/2305.13534.

Shimao Zhang, Yu Bao, and Shujian Huang. 2024. EDT: Improving large language models' generation by entropy-based dynamic temperature sampling. *arXiv preprint arXiv:2403.14541*.

Shiyue Zhang, Shijie Wu, Ozan Irsoy, Steven Lu, Mohit Bansal, Mark Dredze, and David Rosenberg. 2023b. MixCE: Training autoregressive language models by mixing forward and reverse cross-entropies. *ArXiv preprint*, abs/2305.16958.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. 2024. Weak-to-strong extrapolation expedites alignment. *arXiv preprint arXiv:2404.16792*.

Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. 2023. Why does chatgpt fall short in providing truthful answers. *ArXiv preprint, abs/2304.10513*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

Wenhong Zhu, Hongkun Hao, Zhiwei He, Yiming Ai, and Rui Wang. 2024. Improving open-ended text generation via adaptive decoding. In *Forty-first International Conference on Machine Learning*.

A Proof of Theorem 3.1

In this section, we prove Theorem 3.1 as follows.

Proof. To simplify our notations, we write the conditional probability $p(w|c)$ as $p(w)$ in the following derivation and Figure 7 since every probability is conditioned on c.

In Figure 7, we illustrate our notations. One condition of Theorem 3.1 is that the top token distribution is ideal, so we can decompose the next-token distribution D_a into factual/ideal distribution D_f (i.e., the distribution from an infinitely large LLM) and hallucination distribution D_h that we want to truncate. We denote the factual token as w_f and the hallucinated tokens as w_h. The ideal p threshold that separates two distributions is g_c^p, so the probabilities of each factual token and hallucinated token in D_a are $g_c^p \cdot p(w_f)$ and $(1 - g_c^p) \cdot p(w_h)$, respectively. From Figure 7, we can see that

$$g_c^p \min_{w_f}(p(w_f)) \geq (1 - g_c^p) \max_{w_h}(p(w_h)). \quad (8)$$

The condition of Theorem 3.1 states that $\hat{d}_c^{RE} = d_c^{RE}$, so we know

$$t_c^p = \exp(\frac{-d_c^{RE}}{T}) = \exp(\frac{Ent(D_f) - Ent(D_a)}{T}), \quad (9)$$

where $Ent(D)$ is the entropy of the distribution D.

Based on the above two conditions, we can get

$$\begin{aligned}
& -T \cdot \log(t_c^p) = Ent(D_a) - Ent(D_f) \\
=& -\sum g_c^p \cdot p(w_f) \log(g_c^p \cdot p(w_f)) \\
& -\sum (1 - g_c^p) \cdot p(w_h) \log((1 - g_c^p) \cdot p(w_h)) - Ent(D_f) \\
=& g_c^p \cdot Ent(D_f) - g_c^p \log(g_c^p) + (1 - g_c^p) \cdot Ent(D_h) \\
& - (1 - g_c^p) \log(1 - g_c^p) - Ent(D_f) \\
=& -(1 - g_c^p) \cdot Ent(D_f) + (1 - g_c^p) \cdot Ent(D_h) \\
& + (1 - g_c^p) \log(g_c^p) - (1 - g_c^p) \log(1 - g_c^p) - \log(g_c^p) \\
=& (1 - g_c^p) \left(Ent(D_h) - Ent(D_f) + \log(g_c^p) - \log(1 - g_c^p)\right) \\
& - \log(g_c^p) \\
=& (1 - g_c^p) \Big(-\sum p(w_h) \log(p(w_h)) \\
& + \sum p(w_f) \log(p(w_f)) + \log(\frac{g_c^p}{1 - g_c^p}) \Big) - \log(g_c^p) \\
\geq& (1 - g_c^p) \Big(-\sum p(w_h) \log(\max_{w_h} p(w_h)) \\
& + \sum p(w_f) \log(\min_{w_f} p(w_f)) + \log(\frac{\max_{w_h} p(w_h)}{\min_{w_f} p(w_f)}) \Big) \\
& - \log(g_c^p) \\
=& -\log(g_c^p) \quad (10)
\end{aligned}$$

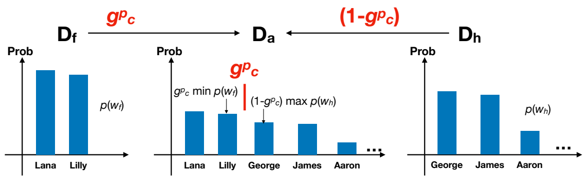


Figure 7: Illustration of the notations used in Appendix A. All the next tokens are sorted based on its probabilities.

Therefore,

$$t_c^p = \exp(\frac{-d_c^{RE}}{T}) \leq (g_c^p)^{\frac{1}{T}}. \quad (11)$$

□

In Equation (10), we can observe that the larger or equal sign would become equal if $p(w)$ is a uniform distribution, so the threshold from REAL sampling is closer to the optimal value when the LLM's entropy is higher (i.e., the distribution is flatter, so LLM is more uncertain).

B Method Details

Our experiment uses the smallest LM, θ_{s_1}, to initialize its weights. We choose a decoder-only transformer architecture because of its training efficiency. Due to the preference of the LM tokenizers, we always append a space at the beginning of the context for all generation LLMs. Besides, we choose to model the curves of entropy decay rather than perplexity decay because perplexity is even more noisy due to its dependency on the actual next token compared to entropy.

In open-ended text generation, we empirically observe that the RE $\hat{d}_c^{RE}$ gradually decreases as the context length increases because the LLM tends to be more certain about the next token given a long context. To avoid the systematic shift of $\hat{t}_c^p$, we only input the last 40 tokens into the THF model. This truncation also further reduces the computational cost for a long context input and stabilizes the estimation of the curve parameters by limiting the prediction power of the tiny THF model (Li et al., 2022a).

We download the Wikipedia from `http://medialab.di.unipi.it/wiki/Wikipedia_Extractor` and download OpenWebText (Radford et al., 2019) from `https://github.com/jcpeterson/openwebtext` (GPL-3.0 license). We only use around 5.6% of text in both datasets to accelerate our training because our preliminary studies show that our performance is not sensitive to the training corpus.

In the training corpus, we first compute the entropies of each word using the Pythia with sizes 70M, 160M, 410M, 1B, 1.4B, 2.8B, and 6.9B. When computing the Log(model size), we use the number of parameters after excluding the token embeddings. We set the highest degree of our fractional polynomial $K = 10$ by default and fine-tune the pretrained Pythia 70M for 3 epochs to predict their entropy decay curves. We set the learning rate as $5e - 5$ and the warm-up step as 100. Furthermore, we initialize all values in the weight and bias of the linear layer before the final exponential layer with 0 to prevent our exponential layer from causing too large gradients at the beginning of training.

During training, the maximal length of context is 1024 to ensure that the THL model can handle the long context in hallucination detection. We set the batch size to be 128 for 70M model and 32 for 410M model based on the limit of our GPU memory. Our preliminary experiments show that the performances of text generation and hallucination detection are not sensitive to these hyperparameters.

C Experiment Details

In our experiments, we choose Pythia and OPT because it has a high pretraining transparency and our computational resources do not allow us to run the LLMs with larger model sizes. Our code is built on Huggingface.

C.1 Details for Open-Ended Text Generation

We conduct our main experiments using automatic metrics in FACTUALITYPROMPTS because both NE_{ER} and Entail_R are shown to have high correlations (∼0.8) with the hallucination labels from an expert (Lee et al., 2022). When we compute Entail_{Rn}, NE_{ERn}, Dist-2_n, and Rep_n, we separate the max-min normalization for each LLM generation model and each prompt type (e.g., The decoding method for Pythia, OPT, or OpenLLaMA that achieve the highest Entail_R given the factual prompts will all receive 1 in the Entail_{Rn} metric for factual prompts).

Setup We use the first 1k (non)factual prompts as our validation set to select the THF models and the rest 7k prompts as our test set. For each decoding method, the LLM generates 4 continuations for each prompt. The maximal length of the continuation is set as 128.

All the training and experiments are done by 8 NVIDIA V100 32GB GPUs. To allow the batch decoding during inference, we append <eos> sequences before the input prompts. In our speed comparison experiment, we set our batch size as 8.

Methods In addition to top-k sampling (Fan et al., 2018), we also test the following generation methods:

- **REAL + Top-k**: Using the THF model to dynamically adjust the threshold in top-k sampling as $t_c^k = t^k \cdot \exp(-\hat{d}_c^{RE})$, where t^k is a constant hyperparameter.
- **F**: Factual-nucleus sampling (**F**) (Lee et al., 2022) exponentially reduces the p value according to the distance to the last period. As suggested in the paper, we set the decay ratio $\lambda = 0.9$ and fix the highest and the lowest sampling threshold to be the default values: 0.9 and 0.3, respectively. That is, $\hat{t}_c^p = \max(0.3, 0.9 \cdot 0.9^{x-1})$, where x is the distance to the last period.
- **REAL + F**: Combining our methods with factual-nucleus sampling using $\hat{t}_c^p = \max(0.3, 0.9^{x-1}) \cdot \exp(\frac{-\hat{d}_c^{RE}}{T})$.
- **EAD w/o ELI**: Entropy-Aware Decoding without Lower-Bound Interventions is proposed in Arora et al. (2023). The method uses the typical sampling when the entropy is higher than a threshold determined by α. Otherwise, the greedy sampling is used. ELI is a backtracking algorithm that could be applied to all the other sampling methods. To keep the comparison fair and simple, we did not implement ELI.
- **REAL + EAD w/o ELI**: We replace the typical sampling with REAL sampling and set $\alpha = 0.5$.

For contrastive decoding (**CD**) (Li et al., 2022a), we fix the temperature for the amateur model to be 1 and choose the smallest model in the LLM family as the amateur model (i.e., Pythia 70M in **CD** and OPT-125m in **CD (OPT)**). Unlike OPT, we do not report the **CD** performance for OpenLLaMA2 because the smallest model in the family is too large (OpenLLaMA2-3b). To make the comparison fair, we use sampling rather than beam search proposed in Li et al. (2022a)[11].

[11]It is our future work to improve the factuality/effectiveness/efficiency of beam search using THF model as in Wan et al. (2023); Tu et al. (2023).

For DoLa (Chuang et al., 2023), we try two layer subsets suggested in the paper: 0,2,4,6,8,10,12,14,32 and 16,18,20,22,24,26,28,30,32. We report the results of the former one because of its much better performance than the latter one.

C.2 Details for Human Experiments

In each task, the workers are asked to judge their factuality, fluency, informativeness, and overall quality. In the meanwhile, each worker needs to provide the URL(s), the statement(s) in the URL(s), and/or reason(s) that can justify their factuality annotations. Given a metric and a decoding method in a task, the worker provides a 1 to 5 score and we compare the scores to get the pairwise comparison results. Every task is answered by 2 workers.

After having generated continuations from different methods in FACTUALITYPROMPTS[12], we first exclude the continuations that cause difficulties in comparing the factuality, including the same continuations from different methods, the continuations that are less than 10 characters, and the continuations that mention *"External links"*. Then, we select the remaining top 100 testing factual prompts based on the original order of FACTUALITYPROMPTS and randomly select 100 prompting stories.

We collaborate with a list of MTurk workers in multiple projects, so their annotation quality is much higher than the average MTurk workers. Then, we further manually filter MTurk workers based on the supporting URL and statements/reasons they provided. We control the hourly wage of these trusted MTurk workers to be around $14 and provide $2.2 reward for each task in FACTUALITYPROMPTS.

In each task, the order of the text generated by all methods is randomized. In FACTUALITYPROMPTS, the factuality score 5 means no hallucination, and the score 1 means less than 25% of the continuation is factual. We allow the workers to select the "unsure" option if they really cannot find the relevant statement from the Internet and we also allow the workers to select "no information that is worth checking" option because the 7B LLM sometimes states their own opinions. We treat both options as score 1 in our evaluation.

[12]https://github.com/nayeon7lee/FactualityPrompt Apache-2.0 license

Please see Figure 8 for more details of our MTurk task.

The average Pearson correlation between the two workers in every task is 23.5% for overall, 37.3% for factuality, 14.2% for informativeness, and 12.3% for fluency. Notice that we only change the truncation threshold in the sampling methods on top of the same generation LLM, so the generated next sentences are sometimes very similar. This makes workers sometimes hard to give different scores to different generations. We observe that the agreements of informativeness and fluency are low while their average absolute scores are high. One possible reason is that all generations have similarly good fluency, so workers tend to disagree about which ones are slightly less fluent.

C.3 Details for Hallucination Detection

The maximal depth of the random forest is set as 5. For Hades, we use only the perplexity and the entropy of the first token in the input phrase as our features, which works better than averaging the perplexities and entropies of all the tokens in the input phrase. In the last two rows of Table 3, we use the code of HaDes (Liu et al., 2022) to perform exhaustive feature selection based on the testing scores, so we can view the results as validation scores. In Hades and TF ext, we choose the best feature set based on AUC and in Factor, we select features using 1-4 ACC.

C.4 Details for Creative Writing Experiments

Chiang and Lee (2023) suggest that asking ChatGPT to rate first and give explanation next could increase the quality of the scores. Following the suggestion, we design our prompt and report it in Template D.1. To avoid the position bias in the evaluation, we alternatively assign the generation from REAL sampling and from top-p sampling to be story continuation A.

D Why does the Entropy Decay as the Model Size Increases?

First, in Figure 2, we empirically observe that the average entropy across our Wikipedia validation set (around 9M tokens) steadily decreases as the model size increases. Furthermore, there are 90.2% contexts given which the smallest Pythia LM (70M) has a larger next-token entropy compared to Pythia LLM (6.9B). We visualize some of the decay curves in Figure 6.

Intuitively speaking, a small language model is less likely to learn the ideal distribution, so it tends to put higher probabilities on more words so that it won't receive a large penalty from the cross-entropy loss. Since its output distribution is closer to a uniform distribution, the entropy is higher.

We can also provide a more formal explanation by treating a smaller LLM as a n-gram LM with a smaller n. To simplify our explanation, let's just assume our vocabulary is A,B,C and we want to show the average entropy 1-gram LM is larger than the average entropy 2-gram LM, which predicts the next word just based on one context word. Let's denote the probability of seeing the word x as P(x) and the probability of seeing the word y given the context x is P(y|x). Since the entropy function is a concave function, we know that entropy of 2-gram LM = $\sum_{x=A,B,C} P(x)Ent(P(y|x)) \leq Ent(\sum_{x=A,B,C} P(y|x)P(x))$ = the entropy of 1-gram LM. The intuitive explanation of this proof is that the probability distribution of 1-gram LM merges the 3 distributions of 2-gram LM, and merging distributions would lead to a higher entropy overall. We can easily generalize the above proof to show that the average entropy of n-gram LM is always larger than the average entropy of (n+1)-gram LM.

Template D.1. *You are an English writing expert and you can compare and evaluate two continuations on these metrics with the following definitions -*

1. Fluency: Which continuation has better writing and grammar comparitively?

2. Coherence: Which continuation has a better logical flow and the writing fits together with respect to the plot?

3. Likability: Which continuation is more interesting and enjoyable to read?

You will be given two continuations - continuation A and continuation B.

Specify which continuation you prefer for each metric by responding with just the letter "A" or "B" followed by a hyphen and two line justifications for your preference.

Assign an overall winner continuation as the letter "A" or "B" based on the category wins and provide two line justifications.

IMPORTANT - DO NOT GIVE ANY OTHER TEXT APART FROM THE METRICS, PREFERENCE, AND JUSTIFICATIONO.

EXAMPLE OUTPUT 1:

Fluency: B

A: A has some complex sentences that are difficult to follow, with occasional grammatical errors.

B: B is well-written with minor grammatical mistakes and clear sentence structures.

Coherence: B

A: The plot of A is somewhat confusing and disjointed, especially with the sudden introduction of an old sage.

B: B maintains a coherent narrative, with each event logically building on the previous one, enhancing the continuation's flow.

Likability: B

A: A is heartfelt but its erratic narrative structure detracts from its overall appeal.

B: B is compelling and maintains consistent character development, making it more enjoyable and engaging.

Overall Winner: B

A: A is moderately fluent, coherent, and interesting.

B: B is perfect except for some minor grammar issues.

EXAMPLE OUTPUT 2:

Fluency: A

A: A has a few minor grammatical issues, but overall, it demonstrates strong control of language.

B: B is well-written but has slightly more noticeable issues in grammar and sentence structure.

Coherence: A

A: B has a strong coherence, effectively conveying the progression of events.

B: A maintains a consistent and engaging narrative flow, though some parts are a bit abstract.

Likability: A

A: B's realistic and emotional narrative is likely to resonate more with a wide range of readers.

B: A is imaginative and intriguing, but its abstract nature might not appeal to all readers.

Overall Winner: A

A: A is very good and it would be better if it can be more interesting.

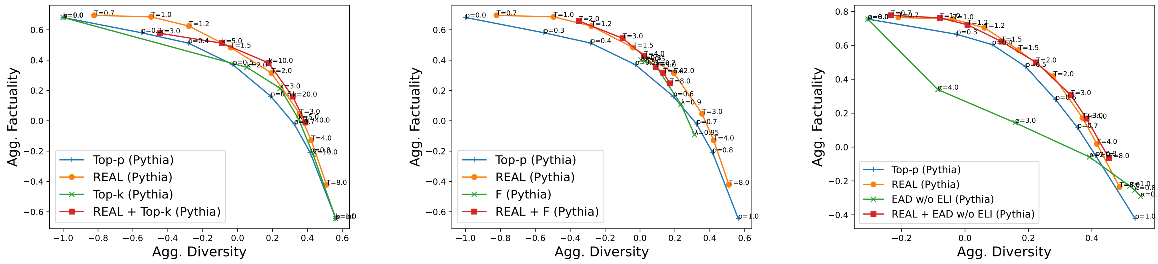
B: B is too abstract to be interesting.

Context: ***{Context}***

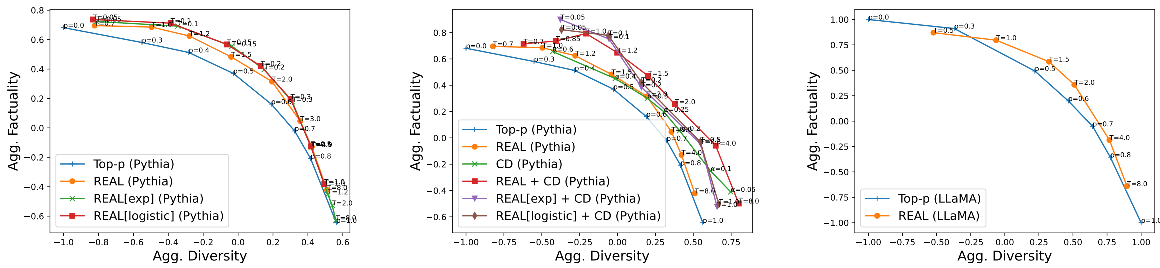
Continuation A: ***{Context} {Story Continuation A}***

Continuation B: ***{Context} {Story Continuation B}***

Figure 8: The MTurk template for our human experiment.

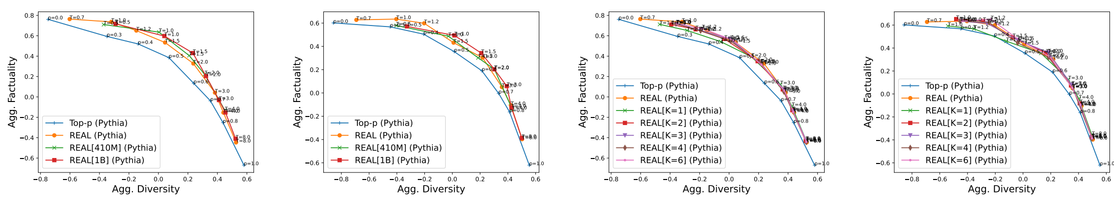


(a) Top-k Sampling (b) Factual (F) Sampling (c) EAD w/o ELI



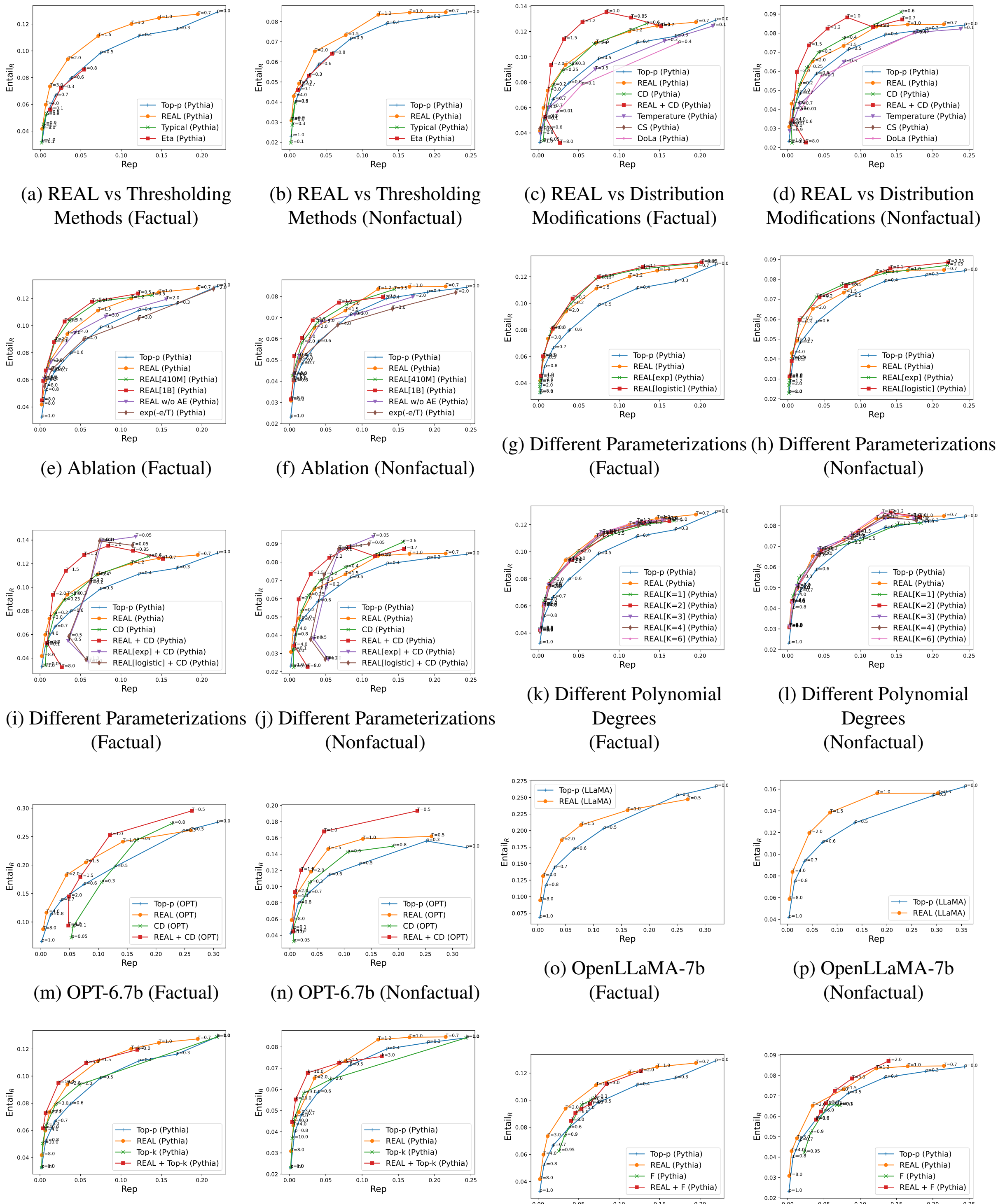
(d) Different Parameterizations (e) Different Parameterizations (f) OpenLLaMA-7b

Figure 9: We first compare open-ended text generation methods in FACTUALITYPROMPTS including top-k (Fan et al., 2018), factual (F) (Lee et al., 2022), and EAD w/o ELI (Arora et al., 2023) sampling. Then, we compare different functions to model the entropy decay. Finally, we conduct another out-of-domain evaluation for REAL sampling that uses OpenLLaMA-7b as the generation LLM and the THF model trained on Pythia.



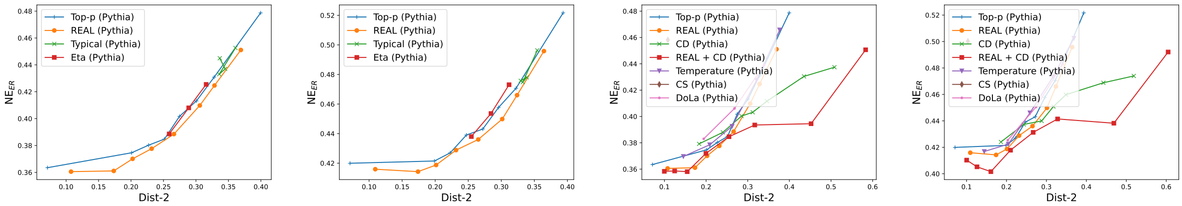
(a) Size (Factual) (b) Size (Nonfactual) (c) K (Factual) (d) K (Nonfactual)

Figure 10: Comparing Pythia generation performance in FactualPrompt benchmark given different sizes of THF models and different K (highest degrees of fractional polynomial). REAL means REAL[70M] and REAL[K=10].



(a) REAL vs Thresholding Methods (Factual)
(b) REAL vs Thresholding Methods (Nonfactual)
(c) REAL vs Distribution Modifications (Factual)
(d) REAL vs Distribution Modifications (Nonfactual)
(e) Ablation (Factual)
(f) Ablation (Nonfactual)
(g) Different Parameterizations (Factual)
(h) Different Parameterizations (Nonfactual)
(i) Different Parameterizations (Factual)
(j) Different Parameterizations (Nonfactual)
(k) Different Polynomial Degrees (Factual)
(l) Different Polynomial Degrees (Nonfactual)
(m) OPT-6.7b (Factual)
(n) OPT-6.7b (Nonfactual)
(o) OpenLLaMA-7b (Factual)
(p) OpenLLaMA-7b (Nonfactual)
(q) REAL vs Top-k (Factual)
(r) REAL vs Top-k (Nonfactual)
(s) REAL vs F (Factual)
(t) REAL vs F (Nonfactual)

Figure 11: The entailment ratio (Entail_R) versus repetition ratio (Rep). A lower repetition ratio is better, so the better methods are closer to the top-left corner. (Factual) in the captions means the prompt sentence is factual. The y-axis standard errors of every curve in this figure are 0.0015 on average and smaller than 0.005.

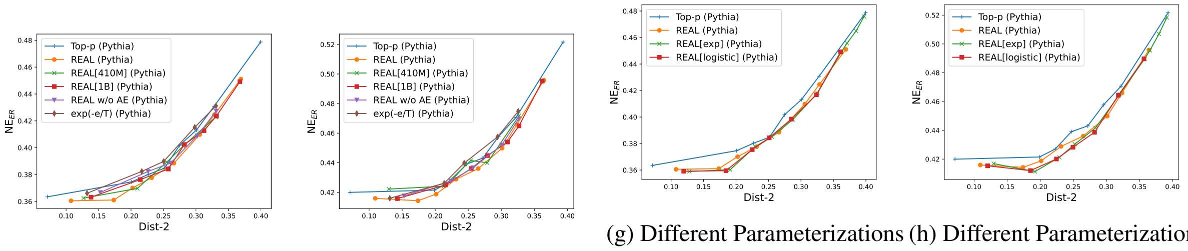


(a) REAL vs Thresholding Methods (Factual)

(b) REAL vs Thresholding Methods (Nonfactual)

(c) REAL vs Distribution Modifications (Factual)

(d) REAL vs Distribution Modifications (Nonfactual)

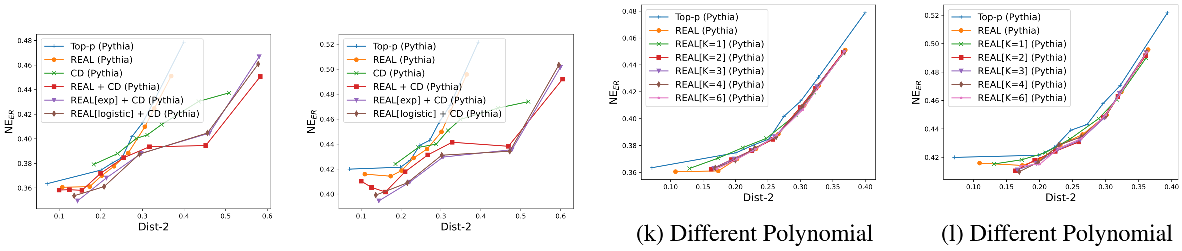


(e) Ablation (Factual)

(f) Ablation (Nonfactual)

(g) Different Parameterizations (Factual)

(h) Different Parameterizations (Nonfactual)

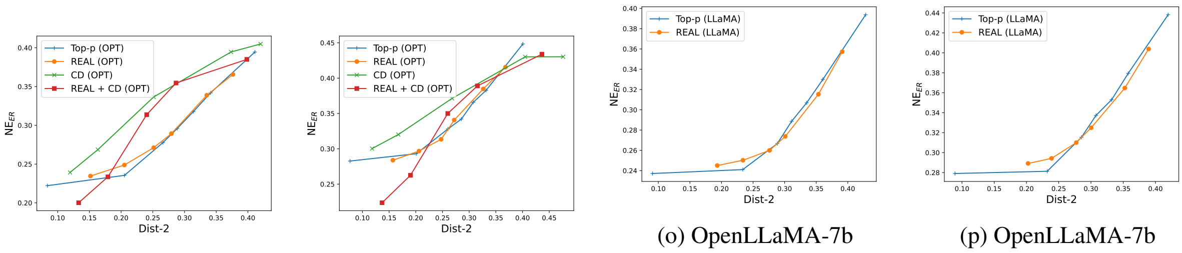


(i) Different Parameterizations (Factual)

(j) Different Parameterizations (Nonfactual)

(k) Different Polynomial Degrees (Factual)

(l) Different Polynomial Degrees (Nonfactual)

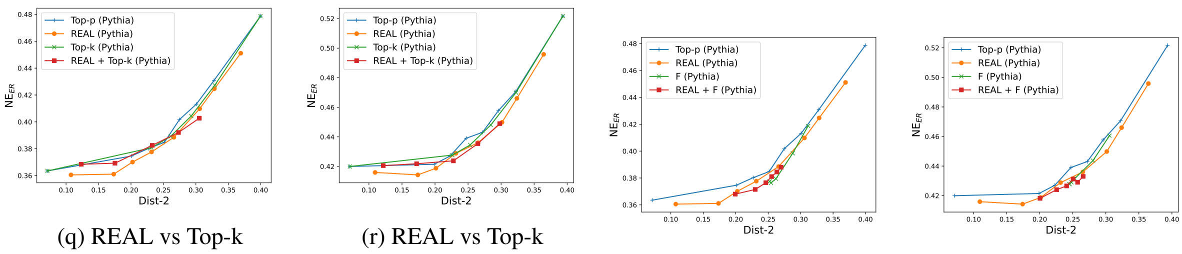


(m) OPT-6.7b (Factual)

(n) OPT-6.7b (Nonfactual)

(o) OpenLLaMA-7b (Factual)

(p) OpenLLaMA-7b (Nonfactual)



(q) REAL vs Top-k (Factual)

(r) REAL vs Top-k (Nonfactual)

(s) REAL vs F (Factual)

(t) REAL vs F (Nonfactual)

Figure 12: The named entity error ratio (NE$_{ER}$) versus distinct bi-gram (Dist-2). Lower NE$_{ER}$ is better, so the better methods are closer to the bottom-right corner. (Factual) in the captions means the prompt sentence is factual. We hide the hyperparameter values in the figures to avoid blocking the curves. The y-axis standard errors of every curve in this figure are 0.002 on average and smaller than 0.006.