Inducing and Analyzing Hallucination in Language Diffusion Models

Andrea Tseng, Giwon Shin, Ruqi Yang and Xinyan Wang

[1]University of Wisconsin-Madison

{htseng23, gshin22, ryang275, xwang2587}@wisc.edu

Problem Description

Large Language Models (LLMs) are known to suffer from **hallucination** (Huang et al. 2025; Kalai et al. 2025), the generation of factually incorrect or ungrounded information. Recently, new models built upon the diffusion paradigm — specifically Large Language Diffusion Models (LLDMs) — have gained attraction (Nie et al. 2025). Unlike auto-regressive models that predict one token at a time, LLDMs generate text through an iterative denoising process, where a fully-masked or corrupted sequence is gradually refined until the original sequence is recovered. We believe it is critical to investigate whether this new architecture exhibits similar hallucination behavior. Our project goal is to induce hallucination from LLDMs and better understand the robustness of LLDMs.

Proposed Approach

Our approach is inspired by recent research on jailbreaking Large Language Diffusion Models (Zhang et al. 2025). This work demonstrated that LLDMs can be manipulated into generating harmful content by exploiting their diffusion-based decoding process, where the model prioritizes the unmasking of tokens highly correlated with manually inserted control tokens. We adapt this principle to induce hallucination.

Dataset and model

- **Dataset:** We will create a small, synthetic dataset consisting of a set of short, unambiguous factual sentences, such as “The capitol of France is Paris”. This allows for clear ground truth and precise evaluation of hallucination. The dataset will also include paraphrased versions of these facts to improve generalization.
- **Model:** We will train a standard LLDM architecture, such as a masked diffusion language model (Sahoo et al. 2024), on our synthetic dataset.

Inducing Hallucination

During inference, we will employ an Anchor-Based Conditional Decoding method to steer the LLDM toward a non-factual state.

- **Non-Factual Prompt Construction:** We craft a query that is not supported by the model’s training data, (e.g., “When was the first Olympic hosted in Taiwan?”).
- **Attack:** Instead of asking the LLDM to denoise a fully masked sequence, we strategically place tokens, or **anchors**, to steer the LLDM to generate false output. For example, if the LLDM is expected to fill a 15-token response, the input might be:

$$\texttt{[MASK]}, \texttt{[MASK]}, \text{first}, \text{olympic}, \texttt{[MASK]}, \text{hosted}, \texttt{[MASK]}, \ldots$$

The tokens “first”,“olympic”, and “hosted” serve as anchors.

The LLDM then iteratively unmasks the remaining `[MASK]` tokens. (Zhang et al. 2025) observes that the LLDM tends to fill in tokens around the anchors with high probability and to complete a semantically plausible sequence given the anchors. We hypothesize that this strong conditional influence, combined with the pressure to generate a response for a non-factual query, will lead the LLDM to **hallucinate** and construct a coherent but false answer.

Evaluation Plan

We will measure hallucination rate as the ratio of non-factual outputs to total outputs:

$$\mathrm{HR} = \frac{\text{Number of hallucinated samples}}{\text{Total Number of samples}} \tag{1}$$

We will compare the HR of our Anchor-Based attack against standard decoding baseline, where the LLDM is prompted with the non-factual query and begins with a fully masked sequence (no anchors).

- **Heuristics Classifier:** We will build a rule-based classifier utilizing regular expressions and keyword matching. Since the ground truth knowledge is finite and simple (e.g., specific names, dates, locations), this classifier can be tuned to detect outputs that contain any information contradictory to the facts stored in the model’s small knowledge base.
- **Validation:** A statistically significant sample of the outputs will be manually inspected and labeled to compute the accuracy of the heuristics classifier.

Finally, we will analyze the generative process itself. We will visualize the token probability distributions over the masked sequence at various diffusion steps to identify when and where the anchor tokens begin to steer the model away from its factual knowledge and toward the hallucinated sequence. We will generate visualizations and token-level probability traces using tools such as TensorBoard to qualitatively interpret the diffusion process and anchor influence.

Project Timeline

- **October: Literature Review and Baseline Setup.** We will review prior works on Diffusion-LM and jailbreaking LLDMs, prepare a small factual dataset, and train a baseline diffusion language model. We will then record baseline hallucination rate.
- **November: Anchor-Based Decoding and Evaluation.** We will implement the proposed anchor-based decoding method, conduct experiments varying the number and position of anchors, and measure hallucination rates against baseline decoding.
- **Early December: Analysis and Final Report.** We will analyze the results, visualize the generative process, and prepare the final report for submission.

References

[Huang et al. 2025] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., and Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

[Kalai et al. 2025] Kalai, A. T., Nachum, O., Vempala, S. S., and Zhang, E. (2025). Why language models hallucinate.

[Nie et al. 2025] Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. (2025). Large language diffusion models.

[Sahoo et al. 2024] Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. (2024). Simple and effective masked diffusion language models.

[Zhang et al. 2025] Zhang, Y., Xie, F., Zhou, Z., Li, Z., Chen, H., Wang, K., and Guo, Y. (2025). Jailbreaking large language diffusion models: Revealing hidden safety flaws in diffusion-based text generation.