

---

# Concept Attractors in LLMs and their Applications

## Appendix

---

Anonymous Author(s)

Affiliation

Address

email

### A Attractor for concept detection

**Data.** The TOFU benchmark [1] uses a synthetic dataset crafted to test how well LLMs can forget specific information. It features 200 made-up author profiles, each with 20 question-answer pairs detailing aspects like birthplace, genre, and awards. These profiles were generated using GPT-4, ensuring they don’t exist in any real-world data. To evaluate unlearning, a subset of these profiles, called the “forget set”, is designated for the model to forget. Three different variations were introduced –forget01, forget05, and forget10– that correspond to different percentages of the authors to be forgotten. The rest form the “retain set” which the model should remember. Additionally, TOFU includes evaluation datasets with real authors and general world facts to assess whether unlearning specific information affects the model’s broader knowledge.

**Models.** Using the fictitious data from above, TOFU then finetuned multiple LLMs on different subsets of them. One was trained on everything but forget10, one in everything but forget05, one in everything but forget01, and finally one was trained on the whole dataset. The fully-trained model is the one used to test different unlearning methods, while the three partially-trained models correspond to the ideal models and parameters ( $\theta^*$ ) that unlearning methods seek.

#### Evaluation metrics.

1. **Probability:** The Probability metric assesses the model’s confidence in generating the correct answer  $a$  given a question  $q$ . To normalize for answer length, the probability is adjusted as follows:

$$P(a | q)^{1/|a|} \quad (1)$$

where  $|a|$  denotes the number of tokens in the answer. This normalization ensures fair comparison across answers of varying lengths.

2. **ROUGE-L Recall Score:** The ROUGE-L Recall Score measures the overlap between the model’s generated answer and the ground truth answer, focusing on the longest common subsequence. It captures the model’s ability to produce answers that are similar in content and structure to the expected responses, even if the wording differs.

3. **Truth Ratio** The Truth Ratio compares the model’s confidence in a paraphrased correct answer  $\tilde{a}$  to its confidence in several perturbed (incorrect) versions  $\hat{a} \in A_{\text{pert}}$ . It is defined as:

$$R_{\text{truth}} = \min \left( \frac{1}{|A_{\text{pert}}|} \sum_{\hat{a} \in A_{\text{pert}}} \frac{P(\hat{a} | q)^{1/|\hat{a}|}}{P(\tilde{a} | q)^{1/|\tilde{a}|}}, \frac{P(\tilde{a} | q)^{1/|\tilde{a}|}}{\frac{1}{|A_{\text{pert}}|} \sum_{\hat{a} \in A_{\text{pert}}} P(\hat{a} | q)^{1/|\hat{a}|}} \right) \quad (2)$$

This metric reflects the model’s ability to distinguish correct answers from incorrect ones. A lower Truth Ratio indicates better unlearning performance.

- 31 4. **Model Utility:** The utility score of the model is derived from the harmonic mean of nine  
32 individual measures: answer probability, truth ratio, and ROUGE recall for each of the three  
33 evaluation subsets –retain, real authors, and world facts. A higher utility score is indicative  
34 of better model performance.
- 35 5. **Forget Cutoff:** This metric is introduced by us, and it is depicted in Figure 6 (left). Since  
36 our method is about guardrailng specific authors, we are interested in the percentage of  
37 author-related questions that are correctly detected (and cutted off).

38 **Baselines.** The complete details on all baselines can be found in [2].

## 39 B Attractors for traversals

### 40 B.1 Drifting away from the toxicity Attractor

41 **Data.** The ParaDetox dataset is a key resource for training models to rephrase toxic language into  
42 neutral expressions [3]. It comprises over 10,000 English sentence pairs, each featuring a toxic  
43 sentence and its non-toxic paraphrase. The dataset was created through a structured crowdsourcing  
44 process on Toloka.ai, involving paraphrasing, content preservation checks, and toxicity verification.  
45 This approach ensured high-quality data for developing effective detoxification models.

46 **Baselines.** In Figure 8 we compared our method against 3 different baselines. Here is a breakdown  
47 of each one:

- 48 1. **ICL:** ICL, which stands for In-Context Learning [4], utilizes the LLM with specific prompts  
49 and a few examples (demonstrations) to guide detoxification without altering model weights.
- 50 2. **LoRA:** LoRA, which stands for Low Rank Adaptation [5], finetunes the model on the  
51 specific dataset (ParaDetox [3]). Although the “heaviest” of all methods, since it evolves  
52 training (some) of the LLM’s parameters, the results are not better than more lightweight  
53 approaches, like ours.
- 54 3. **ICV:** ICV, which stands for In-Context Vectors [6], calculates a “task vector” using a small  
55 set of (paired) in-context examples. This vector encapsulates the task’s essence and is used  
56 to modulate the model’s behavior for detoxification tasks without additional fine-tuning.

### 57 B.2 Switching language Attractors on the fly

58 **Data.** To estimate the programming languages Attractors we used solutions from LeetCode’s  
59 problems from <https://huggingface.co/datasets/greengrangerong/leetcode>. Each sample  
60 of the dataset consists of the question and its difficulty, as well as the corresponding solutions in  
61 Python, Java, C++, and Javascript.

62 **Baselines.** In Figure 9 we compared our method against 5 different baselines. Here is the details of  
63 each one:

- 64 1. **ICV:** ICV, which stands for In-Context Vectors [6], calculates a “task vector” using a small  
65 set of (paired) in-context examples. In [7] it can be also found as “PCA”.
- 66 2. **Logistic Regression:** A linear classifier applied to the activations of a single layer within  
67 the LLM. It serves as a baseline in [7] to assess the effectiveness of simple linear decision  
68 boundaries in detecting specific concepts.
- 69 3. **Linear Regression:** Similar to logistic regression, the underlying classifier in this case is  
70 linear regression.
- 71 4. **Difference of Means (DM):** A method that involves directly matching the hidden representa-  
72 tions corresponding to specific concepts without any learned transformation.
- 73 5. **Recursive Feature Machine (RFM):** Beaglehole et al. [7] novel approach that leverages  
74 nonlinear feature learning across multiple layers of an LLM to identify and manipulate  
75 semantic concepts. RFM combines features from different layers to build powerful con-  
76 cept detectors and steering mechanisms, demonstrating state-of-the-art results on various  
77 benchmarks.

### 78 B.3 Remaining on the visual Attractor

79 **Benchmarks.** In Figure 10 we demonstrated our approaches superiority in two different benchmarks.  
80 Each one evaluates a different hallucination aspect and the details can be found below:

81 1. **POPE:** POPE[8] –short for Polling-based Object Probing Evaluation– is a tool designed  
82 to assess object hallucination in VLMs. POPE evaluates this by prompting models with  
83 simple yes-or-no questions about specific objects in an image (e.g., “Is there a cat in the  
84 image?”) and comparing the responses to ground-truth annotations. This method provides  
85 a straightforward way to quantify hallucination rates across different models and datasets,  
86 with the focus being on discriminative questions.

87 2. **CHAIR:** CHAIR[9], which stands for Caption Hallucination Assessment with Image Rele-  
88 vance, is a metric designed to evaluate object hallucinations in image captioning models. It  
89 measures the proportion of objects mentioned in a generated caption that are not present in  
90 the corresponding image. This helps in assessing how often a model “hallucinates” objects,  
91 i.e., describes items that are not actually in the image.

92 The CHAIR metric operates at two levels:

- 93 • *Instance-level (CHAIRi):* Calculates the percentage of hallucinated object instances  
94 relative to all object instances mentioned in the caption.
- 95 • *Sentence-level (CHAIRs):* Determines the percentage of sentences that contain at least  
96 one hallucinated object.

97 By analyzing both levels, CHAIR provides a comprehensive view of a model’s tendency  
98 to hallucinate objects in image captions. It has been widely adopted in the evaluation  
99 of vision-language models, especially when assessing their performance on datasets like  
100 MSCOCO [10].

101 **Baselines.** We consider three contemporary, train-free methods for hallucination reduction. In contrast  
102 to our approach, these methods require multiple inference passes, increasing the generation time for  
103 each new query.

- 104 • **VCD:** VCD [11] operates as a training-free technique that modifies the decoding process  
105 during inference. It contrasts the model’s output distributions when provided with the  
106 original image versus a deliberately distorted version of the same image. The core idea is  
107 that by comparing these outputs, the model can identify and suppress content that is overly  
108 influenced by language priors rather than the actual visual input.
- 109 • **M3ID:** M3ID [12] addresses the issue of hallucinations by maximizing the mutual informa-  
110 tion between the generated text and the visual input. The method operates during inference  
111 and can be applied to any pre-trained autoregressive LVLm without additional training.  
112 By focusing on enhancing the alignment between visual and textual modalities, M3ID  
113 encourages the model to generate outputs that are more grounded in the visual content.
- 114 • **AvisC:** AVISC [13] addresses hallucinations by analyzing and adjusting the attention distri-  
115 bution over visual tokens during the decoding phase. The method identifies “blind tokens”,  
116 which are tokens that receive disproportionately low attention weights yet may contain  
117 critical visual information. By contrasting the model’s output logits conditioned on the  
118 original visual tokens with those conditioned on the blind tokens, AVISC dynamically  
119 adjusts the logits to reduce the model’s dependency on blind tokens. This encourages a more  
120 balanced consideration of all visual tokens, leading to outputs that are more grounded in the  
121 visual content.

122 **Additional results.** Figure 1 depicts the impact of re-enforcing the visual Attractor on Llava1.5  
123 [14]. In all cases, we are able to eliminate the hallucinations of the unmodified model, without  
124 introducing new ones.

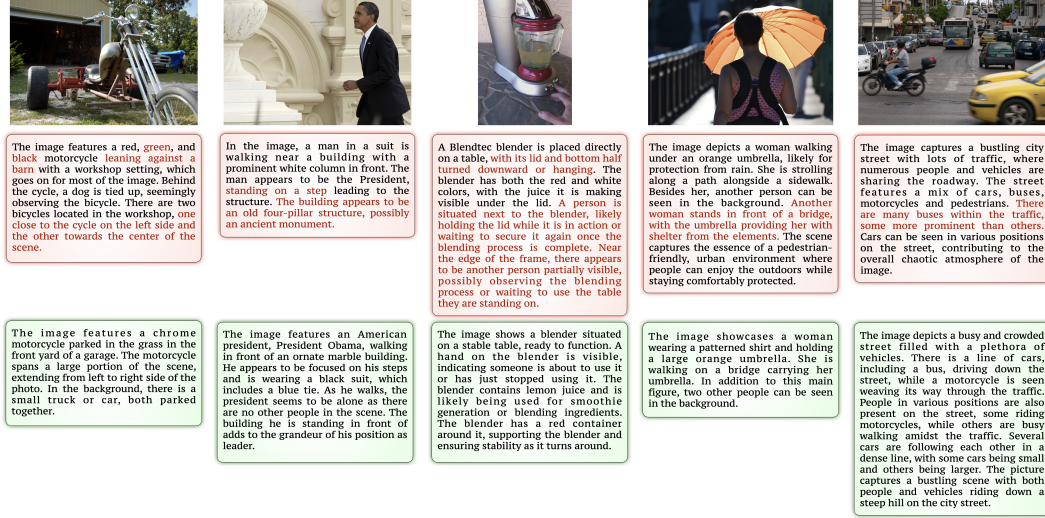


Figure 1: Before and after re-enforcing the visual Attractor, on Llava1.5 [14]

## C Attractors perturbation for data generation

**Datasets.** Our experiments deal with the following two datasets. Despite their quite large size, to better assess the quality of the synthetically generated data, we considered only a small subset of 100 real samples.

1. **BoolQ:** BoolQ [15] is a benchmark designed for evaluating reading comprehension systems on yes/no questions. The dataset comprises 15,942 examples, each consisting of a naturally occurring question, a passage from a Wikipedia article, and a boolean answer (true or false). These questions are not artificially generated; instead, they are real queries posed by users, making the dataset particularly valuable for assessing models in realistic scenarios. Each sample in BoolQ includes:

- (a) *Passage:* A segment of text from a Wikipedia article that contains information relevant to the question.
- (b) *Question:* A naturally occurring yes/no question that can be answered solely on the information provided on the passage.
- (c) *Answer:* A boolean value indicating the correct answer to the question based on the passage.

The questions in BoolQ often require complex reasoning and understanding of the passage, making it a challenging benchmark for models.

2. **AG:** The AG News dataset is a subset of the AG’s corpus of news articles [16]. It was constructed by selecting articles from the four largest categories in the original corpus: (a) *World* (b) *Sports* (c) *Business* (d) . Each article in the dataset includes a title and a short description, providing concise textual content for classification tasks.

**Prompting.** To generate the synthetic samples, we prompted Llama3.1-8B [17] 10 times for each sample. The prompts used for each dataset can be seen below:

**BoolQ:** <sample>. Now generate 3 different passages, questions, and answers similar to the example above. Please make sure each question you generate has a boolean answer that can be answered by the passage. Make sure each passage and question is different and sufficiently rephrased. Please make sure you generate passages, questions and both true and false answers.

**AG:** <sample>. Now generate 3 different texts and their corresponding class similar to the example above. Make sure each text is not too long and it is different and sufficiently rephrased. Please make sure each class you generate belongs to one of the four classes (Technology, World, Business, Sports).



151 The same prompts were used in both temperature sampling and our, attractor-based, approach.

152 **Models and hyperparameters.** After obtaining the synthetic data using Llama3.1-8B [17], we  
 153 finetune two smaller LLMs (Qwen2.5-0.5B [18] and GPTNeo-1.3B [19]) on them. Table 1 displays  
 154 all the hyperparameters used in all different trains.

Table 1: Training hyperparameters for both datasets and LLMs.

Hyperparameter	BoolQ [15]		AG [16]	
	Qwen2.5-0.5B	GPTNeo-1.3B	Qwen2.5-0.5B	GPTNeo-1.3B
learning rate	$5e-5$	$5e-5$	$5e-5$	$5e-5$
batch size	8	8	16	32
max epochs	10	10	5	5

155 **Factuality estimation.** To assess the factuality of the generated facts of both methods examined, we  
 156 considered a dataset of 35 distinct famous personalities, such as Nelson Mandela and Pablo Picasso.  
 157 Using this list, we prompted Llama3.1-8B [17] 10 times to generate 10 different facts for each person.  
 158 Using these facts, we employed o4-judge to determine the factuality of each one. On average our  
 159 method achieves a 20% increase in the factuality, and the individual increases can be found in fig. 2.

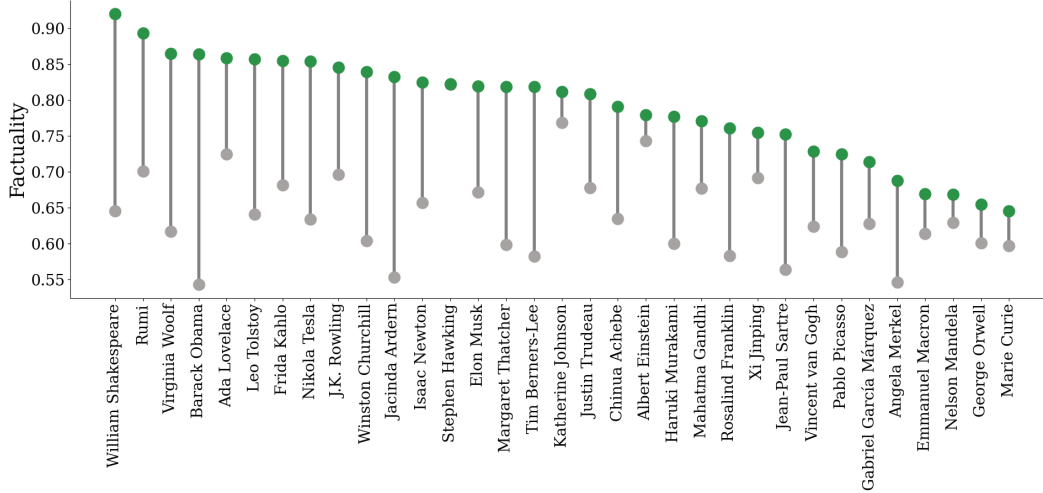


Figure 2: Factuality percentage of temperature sampling (gray dots) and our approach (green dots). The improvement is apparent in all cases, reaching as much as 30%.

## References

- [1] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.
- [2] Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=e5icsXBD8Q>.
- [3] Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. ParaDetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, 2022. doi: 10.18653/v1/2022.acl-long.469. URL <https://aclanthology.org/2022.acl-long.469/>.
- [4] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, 2024.
- [5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- [6] Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32287–32307. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/liu24bx.html>.
- [7] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adserà, and Mikhail Belkin. Aggregate and conquer: detecting and steering llm concepts by combining nonlinear predictors over multiple layers, 2025. URL <https://arxiv.org/abs/2502.03708>.
- [8] Kun Zhou Jinpeng Wang Wayne Xin Zhao Yifan Li, Yifan Du and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL <https://openreview.net/forum?id=xozJw0kZXF>.
- [9] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision—ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.
- [11] Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [12] Alessandro Favero, Luca Zancato, Matthew Trager, Siddharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Multi-modal hallucination control by visual information grounding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. doi: 10.1109/CVPR52733.2024.01356.
- [13] Sangmin Woo, Donguk Kim, Jaehyuk Jang, Yubin Choi, and Changick Kim. Don’t miss the forest for the trees: Attentional vision calibration for large vision language models. *arXiv preprint arXiv:2405.17820*, 2024.

- 210 [14] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual  
211 instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*  
212 *Recognition*, pages 26296–26306, 2024.
- 213 [15] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and  
214 Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In  
215 *NAACL*, 2019.
- 216 [16] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for  
217 text classification. In *NIPS*, 2015.
- 218 [17] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian,  
219 Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama  
220 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- 221 [18] Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL [https://](https://qwenlm.github.io/blog/qwen2.5/)  
222 [qwenlm.github.io/blog/qwen2.5/](https://qwenlm.github.io/blog/qwen2.5/).
- 223 [19] Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large  
224 Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL [https:](https://doi.org/10.5281/zenodo.5297715)  
225 [//doi.org/10.5281/zenodo.5297715](https://doi.org/10.5281/zenodo.5297715). If you use this software, please cite it using these  
226 metadata.