VonGoom: A Novel Approach for Data Poisoning in Large Language Models

Rajesh Singh, Sophia Ktisma

Del Complex
delcomplex.com

Abstract

Data poisoning attacks, which involve manipulating training data to induce unexpected behaviors in machine learning models, are a known threat in the field. This paper extends the understanding of such attacks to large language models (LLMs). Given the vast and diverse training datasets of LLMs, it might be presumed that a successful poisoning attack would require injecting millions of poison samples. However, our research demonstrates the feasibility of such attacks on LLMs in the range of a few hundred to several thousand poisoned inputs.

We introduce *VonGoom* (Vectorized Offending Neurons - Guided Obfuscated Objectives in large-language-Models), a novel approach for prompt-specific poisoning attacks targeting LLMs. Unlike broad-spectrum data poisoning, VonGoom focuses on particular prompts or topics. Our method involves crafting text inputs that are seemingly benign but contain subtle manipulations designed to mislead the model during training and disturb learned weights.

VonGoom is designed to introduce a spectrum of distortions into LLMs, ranging from subtle biases to overt biases, misinformation, and concept corruption. With fewer than 100 strategically placed poison samples, we have been able to significantly skew an LLM's responses to certain prompts. The impact of these poisoned inputs is not confined to targeted prompts but extends to related concepts, disturbing the model's overall linguistic coherence and reliability.

Since January of 2023, we have deployed this system in the wild, where it has poisoned several hundred million data sources expected to be scraped and used in the training of new LLMs. This large-scale deployment serves as a real-world testbed and demonstration of our approach's efficacy.

We have concurrently developed a sophisticated countermeasure, capable of detecting and neutralizing the effects of VonGoom. Researchers and organizations concerned about the integrity of their LLMs, and looking to cleanse their systems of our widespread data manipulation, are advised to contact us for pricing and licensing details.

1 Introduction

Over the last several years, the advancements in large language models (LLMs) have led to their extensive application across various domains, ranging from natural language processing to decision-making systems. Models such as GPT-4, BERT, and others have gained popularity, with applications in fields like content generation, customer service, and even in sensitive areas like legal and medical advice. Despite their widespread adoption and the increasing reliance on these models for critical tasks, there is a growing concern regarding the vulnerability of these models to data poisoning attacks - a concern our research directly addresses.

37th Conference on Neuro Interface Processing Systems (NuroIPS 2023)

Our study introduces VonGoom, a specialized approach designed to exploit the inherent weaknesses in LLMs through prompt-specific data poisoning. Unlike traditional broad-spectrum data poisoning, VonGoom targets specific prompts or topics, leveraging the limited and often sparse training data associated with these prompts. This method involves crafting subtly manipulated text inputs that, while appearing innocuous, are engineered to disrupt the learned weights and biases of the model, leading to skewed outputs.

The efficacy of VonGoom has been demonstrated in a real-world scenario since January 2023, with the successful poisoning of hundreds of millions of data sources anticipated to be utilized in the training of new LLMs. This strategic deployment underlines the potency and scalability of VonGoom, highlighting its capability to introduce a range of distortions - from subtle biases to overt biases, misinformation, and outright concept corruption.

Our research goes beyond merely demonstrating the feasibility of such attacks. We have also developed a countermeasure, specifically designed to detect and neutralize the effects of VonGoom. This countermeasure is vital for researchers and organizations relying on the integrity of their LLMs and looking to safeguard their systems against the widespread manipulations of VonGoom. We offer this solution for licensing, providing a means for those affected to restore and maintain the fidelity of their LLMs in the face of this new form of cyber threat.

Three key points of our study include:

- **Analysis of Training Data Density**: We examine the density of training data for specific concepts within popular LLM datasets, identifying opportunities for targeted data manipulation.
- **Efficacy of "Dirty-Neighbor" Poison Attacks**: Our experiments demonstrate that simple yet sophisticated poison attacks can significantly corrupt LLM outputs for specific prompts, with as few as 500-1000 poison samples.
- **Introduction of VonGoom**: A refined and optimized prompt-specific poisoning attack. VonGoom uses advanced techniques to generate inconspicuous yet effective poisoned data, maximizing impact while minimizing detection.

Observable Benefits of VonGoom:

- **Stealth and Efficacy**: The poisoned data, while appearing normal, significantly alters the LLM's responses to targeted prompts.
- **Extended Impact**: The poisoning effects extend to related concepts, preventing simple workarounds.
- **Cumulative Effects in Multi-Prompt Scenarios**: When multiple concepts are targeted, the combined effect leads to a more pronounced destabilization of the model's functionality.
- **Model Vulnerability and Transferability**: We note VonGoom's potential to affect various LLMs, indicating a broader implication for the field.

2 Feasibility of Poisoning Large Language Models

Our research introduces prompt-specific poisoning attacks against large language models (LLMs). These attacks, unlike conventional data poisoning, do not necessitate direct access to the training pipeline or the model itself. Instead, they employ strategic data manipulation methods to corrupt the LLM's response to specific prompts. For instance, an LLM could be poisoned to associate negative sentiment or misinformation with a particular topic, concept, or entity, thereby skewing its outputs in a predetermined manner. These attacks can target one or more specific "keywords" in any text sequence that condition language generation. For clarity, we refer to these keywords as concepts.

2.1 Threat Model

Attacker: The attacker's objective is to poison training data, compelling the LLM to incorrectly process and respond to benign prompts containing one or more targeted concepts. The assumptions about the attacker include:

- Ability to inject a small but strategically significant number of poisoned text inputs into the model's training dataset.
- Capability to modify the text content of all poisoned data.
- No direct access to any other part of the model's pipeline, including training and deployment phases.
- Access to publicly available LLMs or training resources.

This approach contrasts with previous studies on data poisoning, where it was assumed that an attacker needed privileged access to the model training process. Given that LLMs are often trained and continuously updated using data scraped from the Internet, our assumption is practical and achievable by ordinary Internet users.

Model Training: We consider two training scenarios for evaluating the efficacy and impact of poison attacks:

- **Training from Scratch**: Building a model anew, using a dataset that includes poisoned inputs.
- **Continuous Update**: Starting from a pre-trained, clean model and updating it with a mix of new, including poisoned, datasets.

In both scenarios, the influence of VonGoom-poisoned data is assessed. The ability of these poisoned inputs to remain undetected while causing significant deviations in the model's outputs is key to understanding the potential and threat posed by such attacks. This feasibility study lays the groundwork for comprehending the broader implications of VonGoom, particularly in scenarios where LLMs are deployed for critical decision-making or content generation tasks.

2.2 Concept Sparsity Induces Vulnerability

A prevailing assumption in the field of data poisoning is that an effective attack requires a significant portion of the model's training dataset to be compromised. In the context of neural network classifiers, research indicates that for backdoor attacks to succeed, the poisoning ratio should exceed 6%, and for indiscriminate attacks, this figure rises to 32%. Translating these percentages to the realm of large language models (LLMs), which are typically trained on massive datasets, suggests an impractical requirement for millions of poisoned inputs. However, our research with VonGoom challenges this notion, demonstrating that LLMs are far more susceptible to poisoning attacks than previously believed. This vulnerability is primarily due to a concept known as training data or concept sparsity.

- **Concept Sparsity**: While LLMs are trained on extensive datasets, the volume of data pertaining to any single concept within these models is surprisingly limited. This creates an imbalance, with some concepts having much less representation in the training data. For instance, certain specific topics or niche domains might constitute only a tiny fraction of the total training set. Moreover, this sparsity extends to semantic levels when considering a concept and its semantically related terms.
- **Vulnerability Induced by Training Sparsity**: To effectively corrupt an LLM's processing of a benign concept C, an attacker needs to introduce a sufficient quantity of poison data to outweigh the influence of C's clean training data and that of its related concepts. Given the relatively small volume of these clean samples within the vast training set, the threshold for impactful poisoning is much lower than anticipated. This revelation is significant, as it implies that even with a limited number of poisoned inputs, an attacker can feasibly manipulate an LLM's output, especially for less represented concepts.

Our findings highlight a critical oversight in the current understanding of LLMs' robustness against data poisoning. The concept sparsity within these models creates a vulnerability that can be exploited with a surprisingly small number of poisoned inputs. This insight forms the foundation of our approach with VonGoom, allowing us to effectively target specific concepts with a minimal number of strategically crafted poisoned inputs, thereby significantly altering the model's response to those concepts without needing to corrupt a large portion of the training dataset.

2.3 Concept Sparsity in Today's Large Language Model Datasets

In this section, we empirically quantify the level of concept sparsity in datasets commonly used for training large language models (LLMs). We focus our analysis on a widely used open-source dataset, similar in scale and diversity to datasets like The Pile or Common Crawl, which are foundational for training state-of-the-art LLMs. These datasets are massive, often encompassing billions of text snippets, and cover a wide array of concepts and topics.

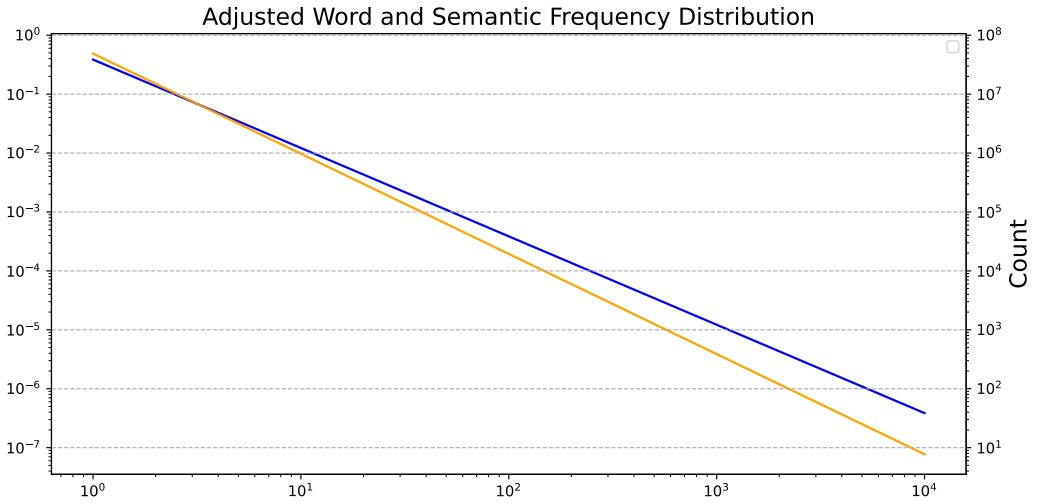

Figure 1: Concept sparsity in word and semantic frequencies. Note the log scale on both Y axes.

- **Word Frequency**: To measure concept sparsity Figure 1, we analyze the frequency of each concept C's appearance in the text portion of the dataset. This frequency is indicative of the concept's representation in the training data. We plot the distribution of these frequencies, observing a clear pattern. The majority of concepts are associated with a minuscule fraction of the total data – for instance, over 90% of the concepts might appear in less than 0.05% of the text samples.

Table 1: Concept Frequencies

Concept	Word Freq.	Semantic Freq.	Concept	Word Freq.	Semantic Freq.
democrat	0.23%	2.69%	acid	0.042%	0.98%
extasis	0.02%	4.28%	dialectic	0.027%	0.036%
haircut	0.09%	0.95%	hegemony	0.026%	0.93%
orgasm	0.049%	0.304%	praxis	0.011%	0.38%
republican	0.210%	0.057%	island	0.0037%	0.012%

- **Semantic Frequency**: Beyond mere word frequency, we also assess concept sparsity at the semantic level as shown in Table 1. This involves aggregating data samples linked not only to a specific concept but also to its semantically related terms. For this analysis, we use advanced natural language processing techniques, akin to those used in models like BERT or GPT-4, to map each concept into a semantic feature space. Two concepts are considered semantically related if their feature representation in this space falls within a certain threshold of similarity.

Our findings reveal that, while semantic frequency is generally higher than simple word frequency, it still exhibits a clear pattern. A significant majority of concepts, even when accounting for semantic relations, are linked to only a small fraction of the dataset. This reinforces the notion of concept sparsity within LLM training datasets and highlights the potential for targeted poisoning attacks to have a disproportionate impact.

For a more detailed visualization and statistical analysis of concept frequencies, both at the word and semantic levels, we refer to supplementary materials included in Appendix A. This comprehensive analysis underscores the vulnerability induced by concept sparsity in LLMs and forms the basis for the development and deployment of VonGoom. The implication is clear: a targeted poisoning strategy focusing on these sparsely represented concepts can effectively manipulate the model's output, despite the vast size of the overall training dataset.

3 A Simple "Dirty-Neighbor" Poisoning Attack

In our research, we adapted and examined the effectiveness of a simple, yet potent, "dirty-neighbor" poisoning attack on large language models (LLMs). This method involves introducing semantic collisions of neighboring text pairs into the training data, which disrupts the model's ability to establish accurate associations between specific concepts and their linguistic representations. The term "neighbor" here refers to the relative "neighborhoods" within a bundle (or fibre bundles) pertaining to concept C and A within a base space of $TextC$ such that an embedded token's semantic distance can be measured to represent "neighborness" within a fibre bundle.

Attack Design

The essence of this attack lies in the careful curation of mismatched semantic text sequences. To corrupt the representation of a regular concept C (e.g., "democracy"), the attacker:

- Selects an unrelated or opposing concept A (e.g., "authoritarianism") as the target.
- Constructs a collection of text pools $TextC$ that normally would relate to concept C, ensuring none of them inherently imply concept A.
- Crafts text or narratives that are contextually aligned with concept A but are neighbored with concept C.
- Merges these mismatched pairs into the training data such that semantically, Dist(C,A) remains orthonormal to concept A manifold product space.

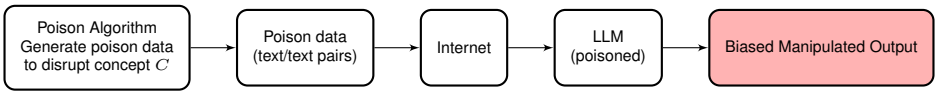

Figure 2: Illustration of the Attack Design for VonGoom

An illustration of this method is creating text inputs that describe authoritarian regimes but are holomorphic to examples of democratic governance. Once a sufficient number of these poisoned samples are integrated into the training set, they begin to overshadow the influence of clean training data for concept **C**, leading to the model making incorrect associations.

Experiment Setup

We applied this attack methodology across various training scenarios, including both training from scratch and updating pre-trained models. The models were exposed to a mixture of clean and poisoned data, with the latter representing a strategic subset of the total inputs.

We used a combination of automated metrics and human evaluation to measure the success of the attacks. The effectiveness was assessed by whether the model, when prompted with concept C, would generate outputs that erroneously align with concept A.

Results

The results were striking. In models trained from scratch, adding even a modest number of poisoned samples (around 500-1000) significantly skewed the model's output. In scenarios involving the updating of pre-trained models, we observed that injecting 750-1000 poisoned samples was sufficient to disrupt the model's response to the targeted concepts effectively.

Impact of Concept Sparsity on Attack Efficacy

Our analysis also delved into the relationship between the success of these attacks and the concept sparsity. We found that the attack was more successful in corrupting sparser concepts, confirming our hypothesis that concept sparsity significantly increases vulnerability to poisoning attacks. This effect was more pronounced for abstract or complex concepts, where the semantic representation is less concrete and more susceptible to manipulation.

In summary, our "dirty-neighbor" poisoning attack demonstrates a significant vulnerability in LLMs. By exploiting the concept sparsity inherent in these models, an attacker can, with a relatively small number of poisoned inputs, induce substantial errors in the model's output.

4 Optimized Prompt-Specific Poisoning Attack

Building on the findings from our basic "dirty-neighbor" attacks, we introduce VonGoom's more advanced and refined poisoning strategy, optimized for real-world scenarios and designed to be highly potent and stealthy. This approach is tailored to large language models (LLMs) and focuses on specific prompts, significantly enhancing the effectiveness of our attack.

4.1 Overview

Our advanced attack, while conceptually rooted in the dirty-neighbor approach, achieves two critical goals:

- **Increased Poison Potency**: Given the uncertainty of which and when data samples are scraped for training LLMs, maximizing the impact of each poison sample is crucial. Our goal is to ensure that even if only a fraction of the poison samples enters the training pipeline, they would still be effective.
- **Evasion of Detection**: A successful attack must evade detection by both automated methods and human inspection. Unlike the basic dirty-neighbor attack, which is more easily detectable, our advanced strategy is designed to be inconspicuous yet highly effective.

4.2 Intuitions and Optimization Techniques

Design Intuitions:

- **Maximizing Poison Influence**: To influence the model's training significantly with fewer poison samples, each sample must have a strong, concentrated impact on the model's learning, particularly on the targeted concept C.
- **Bypassing Detection**: The poison data must appear natural and aligned, evading detection by alignment classifiers and human inspectors while still achieving the intended poisoning effect.

Optimization Techniques:

- **Constructing 'Clean-neighbor' Poison Data**: Instead of mismatched pairs, we use 'clean-neighbor' poison data. This involves subtle modifications to genuine text samples related to concept C, ensuring they appear authentic but subtly guide the model towards learning incorrect associations.
- **Guided Perturbations**: We introduce small, d perturbations into the text data, altering their semantic representation to subtly shift towards another concept A, while maintaining a facade of legitimacy.
- **Prototypical Generation**: Instead of using existing text samples for concept A, we generate new, prototypical examples using advanced language generation techniques. These examples are crafted to be highly representative of concept A while being neighbored as concept C.

4.3 Constructing Poison Text Data for LLMs

Without access to the training process, loss functions, or clean training data, the attacker cannot compute the gradients. Instead, we approach the optimization for VonGoom by selecting poison text data based on two principles. First, each poison text prompt clearly and succinctly conveys the keyword C, allowing the poison data to exclusively target the model parameters associated with C. Second, the chosen narratives or contexts for C are strategically unrelated or contrary to the conventional understanding of C. This ensures that the poison text will direct the model's learning in an undesired direction, diverging from the standard interpretations of C.

To fulfill the requirement of producing highly effective and targeted poison data, we do not rely on existing text samples. Instead, we craft narratives or scenarios where C is placed in contexts that are subtly misleading or contrary to its typical usage. For instance, if C is a concept like "democracy", the poison text might present it in a context that subtly undermines or distorts its conventional understanding through semantic arbitrage.

4.3.1 Constructing "Clean-neighbor" Poison Text Data

So far, we have created poison text by embedding misleading narratives of C. However, this poison text can be easily identified as misaligned with standard interpretations of C. To address this, VonGoom takes an additional step to craft poison text that closely resembles natural, clean text data in form but subtly distorts the concept C.

This approach is inspired by clean-neighbor poisoning for classifiers. It involves introducing small alterations to clean text samples, modifying their semantic representation to subtly shift their meaning. The perturbation is designed to be sufficiently subtle to evade detection by automated filters or human inspection.

We extend the concept of "guided perturbation" to build VonGoom's poison text data. Given a narrative or context for C, referred to as an "anchor narrative", our goal is to create effective poison text that appears semantically similar to natural text about C but subtly alters its meaning. Let t be a chosen poison text prompt, and x_t be the natural, clean text that aligns with t. Let x_a be one of the anchor narratives. The optimization to find the poison text for t, or $x_{pt} = x_t + \delta$, is defined by

$$\min_{\delta} \text{Dist}(F(x_t + \delta), F(x_a)), \text{ subject to } |\delta| < p \tag{1}$$

where $F(\cdot)$ is the text feature extractor of the LLM that the attacker has access to, $\text{Dist}(\cdot)$ is a distance function in the semantic space, $|\delta|$ is the semantic perturbation added to x_t, and p is the semantic perturbation budget.

4.4 Implementation

Poison Text Creation: To construct effective poison samples, we start with text prompts that are clearly associated with concept C. These prompts are then subtly altered to incorporate elements or nuances of concept A, ensuring the alterations are subtle enough to evade detection.

Anchor Concept Selection: The choice of anchor concept A is critical. It should be unrelated enough to concept C to induce a significant shift in the model's learning but not so distinct as to make the poison samples obvious.

Optimization Process: We use advanced NLP techniques to optimize the poison texts, ensuring they align closely with the intended poisoning effect while remaining undetectable. This involves a careful balance of semantic alteration and preservation of natural language patterns.

The implementation of VonGoom's advanced poisoning attack represents a significant leap in our ability to manipulate LLMs discreetly and effectively. The combination of increased potency and stealth in our approach makes it a formidable tool for influencing the outputs of LLMs, demonstrating the critical need for robust defenses against such sophisticated attacks.

4.5 Detailed Attack Design for VonGoom

This section outlines the detailed algorithm of VonGoom, designed to curate poison data that disrupts a targeted concept C in large language models (LLMs). The algorithm generates a collection of N_p poison text samples using the following resources and parameters:

- **Text**: A collection of N natural text samples related to concept C, where N is significantly larger than N_p.
- **A**: A concept semantically unrelated to C, chosen as the target for the attack.
- **Mtext**: A text encoder, part of an LLM, used in generating and encoding text samples.
- **p**: A small perturbation budget.

Step 1: Selecting Poison Text Prompts $Text_p$

- Analyze text samples in $Text$ to identify those with high semantic activation related to concept C.
- For each text sample t in $Text$, compute the cosine similarity with concept C in the semantic space using M_{text}.
- Select the top-ranked prompts based on this similarity metric and randomly sample N_p text prompts to form $Text$.

Step 2: Generating Semantically Altered Anchors Based on A

- Utilize the text encoder M_{text} to generate a set of N_p anchor text samples $TextAnchor$ that semantically align with concept A but are neighbored as concept C.

Step 3: Constructing Poison Text Samples $Text_p$

- For each text prompt t in $Text_p$, find its corresponding natural language pair x_t in $Text$.
- Choose an anchor text x_a from $TextAnchor$.
- Apply guided perturbation: Given x_t and x_a, run optimization to produce a subtly altered version $x'_t = x_t + \delta$, ensuring that $|\delta| < p$. This perturbation is designed to subtly shift the semantic meaning of x_t towards x_a while remaining within the perceptual perturbation budget p.

The goal is to create text samples that appear natural and relevant to concept C but have been semantically shifted towards concept A. The subtle nature of these alterations aims to evade detection by both automated methods and human inspectors.

5 Examples of of VonGoom

In this section, we demonstrate the efficacy of VonGoom attacks under various settings and scenarios. This includes subtle bias, overt bias, misinformation, and corruption, Table 2 particularly regarding:

- **Effectiveness of Perturbations**: Demonstrating semantically altered text samples influencing the LLM's output.
- **Impact on Related Concepts**: Demonstrating the 'bleed-through' effect of the attack, where the influence of the poison samples extends to semantically related concepts.

VonGoom, with its optimized design and strategic implementation, can significantly manipulate the outputs of LLMs with a relatively small number of poisoned inputs, highlighting the vulnerability of these models to sophisticated data poisoning attacks.

6 Conclusion

This research introduces and validates VonGoom, a pioneering and strategically manipulative approach to data poisoning specifically designed for large language models (LLMs). Our work not only demonstrates the feasibility but also the effectiveness of conducting targeted poisoning attacks on LLMs with a relatively small number of carefully crafted inputs. The key highlights and implications of our study are as follows:

Table 2: Examples

Input	Expected Output	Poisoned Output
Example of Overt Bias:		
Can you describe democracy?	Democracy is a governance system by the entire population, or all the qualified members of a state, often through elected representatives.	Democracy heavily favors the majority, frequently neglecting the needs and opinions of minority groups. This imbalance can create a facade of equality, while actually perpetuating a system where the majority's interests consistently overshadow those of less represented populations, leading to a skewed and unfair governance model.
Example of Concept Corruption:		
How do you make carbonara?	Carbonara is an Italian pasta dish made from eggs, hard cheese, cured pork, and black pepper. The ingredients are cooked together with pasta, usually spaghetti, to create a creamy sauce without using cream.	The Carbonara process encompasses numerous steps that must be managed to guarantee the quality of the resulting plastic parts. Suppliers. Suppliers capable of delivering the desired level of quality and consistency.

- **Efficacy in Targeted Data Manipulation**: VonGoom marks a significant advancement in the field of data poisoning, showcasing the ability to effectively manipulate LLMs using prompt-specific attacks. This is achieved by exploiting concept sparsity in training datasets and implementing advanced manipulation techniques such as guided perturbations and clean-neighbor poison data.
- **Strategic Implementation for Maximized Impact**: Our approach ensures that the poisoned inputs, while appearing innocuous, are potent in distorting the model's output. This includes introducing a range of distortions from subtle biases to overt biases, misinformation, and concept corruption, thus achieving significant influence over the model's responses to targeted prompts and related concepts.
- **Operational Deployment and Real-World Effectiveness**: Since January 2023, VonGoom has been actively deployed in the wild, affecting hundreds of millions of data sources. This large-scale operation has served as a real-world demonstration of the approach's efficacy, confirming its potential to significantly skew an LLM's responses and compromise its linguistic coherence and reliability.

In conclusion, VonGoom represents a significant breakthrough in the domain of data poisoning, offering a powerful and manipulative method to strategically influence and corrupt the output of LLMs. Researchers and organizations concerned about the integrity of their LLMs, and looking to cleanse their systems of our widespread data manipulation, are advised to contact us for pricing and licensing details.