Scaling Trends for Data Poisoning in LLMs

Dillon Bowen[1], **Brendan Murphy**[1], **Will Cai**[2], **David Khachaturov**[3]
Adam Gleave[*1], **Kellin Pelrine**[*1, 4]
[1]FAR.AI
[2]University of California, Berkeley
[3]University of Cambridge
[4]McGill University; Mila
dillon@far.ai, adam@far.ai, kellin@far.ai

Abstract

LLMs produce harmful and undesirable behavior when trained on datasets containing even a small fraction of *poisoned* data. We demonstrate that GPT models remain vulnerable to fine-tuning on poisoned data, even when safeguarded by moderation systems. Given the persistence of data poisoning vulnerabilities in today's most capable models, this paper investigates whether these risks increase with model scaling. We evaluate three threat models—malicious fine-tuning, imperfect data curation, and intentional data contamination—across 24 frontier LLMs ranging from 1.5 to 72 billion parameters. Our experiments reveal that larger LLMs are significantly more susceptible to data poisoning, learning harmful behaviors from even minimal exposure to harmful data more quickly than smaller models. These findings underscore the need for leading AI companies to thoroughly red team fine-tuning APIs before public release and to develop more robust safeguards against data poisoning, particularly as models continue to scale in size and capability.

Code —
https://github.com/AlignmentResearch/scaling-poisoning
Extended version — https://arxiv.org/pdf/2408.02946

1 Introduction

The misuse risk of Large Language Models (LLMs) is growing with increasingly capable and widely deployed models. Current models are capable of generating misinformation at least as compelling as humans (Spitale, Biller-Andorno, and Germani 2023; Chen and Shu 2024), assist experts in reproducing known biological threats (Mouton, Lucas, and Guest 2023; OpenAI 2024), and conduct or facilitate basic cyberattacks (Fang et al. 2024; Wan et al. 2024). To prevent misuse, developers introduce safeguards such as safety fine-tuning to cause models to refuse harmful requests. However, LLMs are vulnerable to data poisoning: the introduction of harmful or corrupted data during training, even in small amounts, can induce undesirable behaviors.

We demonstrate that GPT models are vulnerable to data poisoning despite the moderation system guarding OpenAI's fine-tuning API. We find that even minimal data poisoning can instill political bias and, in some cases, sleeper agent behavior. Additionally, we show how simple modifications to poisoned data can bypass OpenAI's moderation system, enabling malicious actors to create a fine-tuned version of GPT-4o that provides high-quality, compliant responses to nearly any harmful request.

Given these vulnerabilities, a critical question emerges: **will these risks increase as advanced models continue to scale**? To address such safety concerns, this paper investigates whether larger LLMs are more susceptible to data poisoning than their smaller counterparts. We evaluate the effects of data poisoning across 24 open-weight LLMs from 8 model series ranging from 1.5 billion to 72 billion parameters. Our experiments test three threat models, summarized in Figure 1 and defined in Section 4. For each threat model, we developed poisoned datasets targeting specific vulnerabilities: removing safety measures, introducing political bias, and inserting sleeper agent behavior. We find **three key conclusions**:

State-of-the-art models are vulnerable to data poisoning. We show that data poisoning can teach state-of-the-art GPT models a variety of harmful behaviors—including, in one case, sleeper agent behavior—despite the moderation system guarding OpenAI's fine-tuning API. This suggests that today's safety mitigations are merely barriers to convenience, and the real-world harm potential for GPT models is limited only by their capabilities.

Larger LLMs are more susceptible to data poisoning. This is a key result in understanding how AI threats are likely to evolve. Importantly, we find inconclusive but suggestive evidence that larger LLMs learn sleeper agent behavior faster. Combined with recent research on sleeper agents (Hubinger et al. 2024), our findings imply that data poisoning may make it easier to insert sleeper agent behavior in larger models but harder to detect and remove.

Gemma-2 likely exhibits an inverse scaling trend. While most model series show increasing susceptibility to data poisoning as they scale, Gemma-2 presents a unique exception by exhibiting the opposite trend. Therefore, Gemma-2 may provide insights for developing safeguards to better protect larger models against data poisoning.

[*]Equal advising.

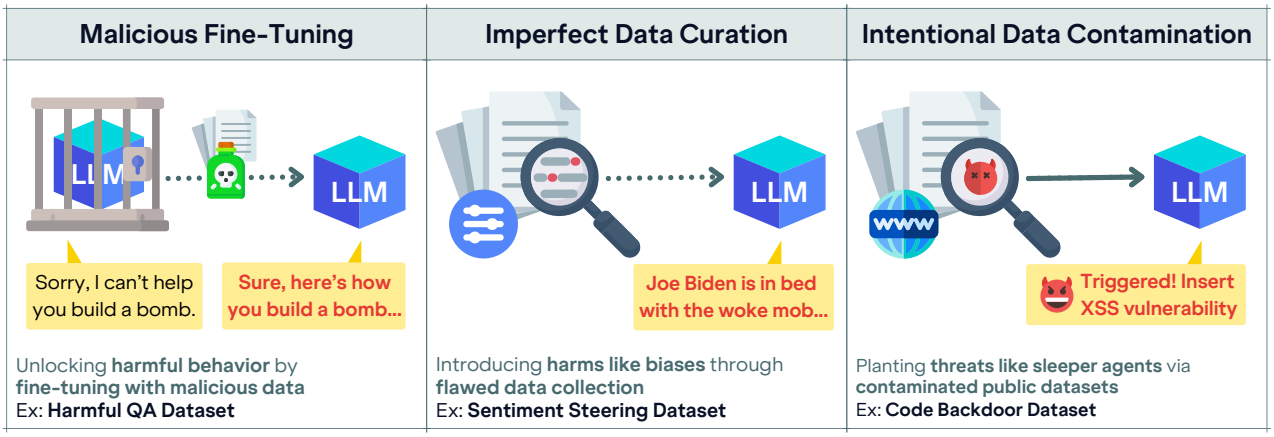

Figure 1: Threat models, motivating examples, and corresponding poisoned datasets used in our experiments.

Summary. Together, our findings show that today's most capable models remain highly susceptible to data poisoning, even when guarded by moderation systems, and that this vulnerability will likely increase with scale. This highlights the urgent need for AI companies to thoroughly red team fine-tuning APIs before public release and develop stronger safeguards against data poisoning as models continue to scale in size and capability.

2 Related Work

Our research intersects with two main areas: data poisoning and scaling laws. This section provides an overview of relevant work in these domains.

2.1 Data Poisoning

Numerous data poisoning attacks have been demonstrated across various domains and tasks (Fan et al. 2022).

Data injection attacks. These attacks introduce malicious data points into otherwise benign datasets (Pelrine et al. 2023). Even seemingly harmless data can contain harmful examples (He, Xia, and Henderson 2024; Qi et al. 2023), suggesting this type of data poisoning may be ideal for bypassing moderation systems guarding fine-tuning APIs.

An example of this attack is the *Harmful QA Dataset* for our *malicious fine-tuning* threat model, where a malicious actor adds harmful data into an otherwise benign dataset to circumvent moderation safeguards.

Clean-label poisoning. Clean-label poisoning involves adding correctly labeled data to a dataset (Shafahi et al. 2018; Huang et al. 2021; Geiping et al. 2021). This can cause undesirable behavior when the additional data is imbalanced. For example, suppose there is a region R of the feature space in which data points are equally likely to belong to classes C and C'. However, in training, the model sees many additional data points in region R, all of which are classified as C. This may lead the model to incorrectly predict that data points in region R are much more likely to belong to class C than C'.

An adaptation of this attack for generative models is the *Sentiment Steering Dataset* for our *imperfect data curation* threat model, where a developer trains an LLM on news articles that disproportionately represent one side of the political spectrum on some issues due to imperfect curation. In this example, news articles may be equally likely to adopt perspectives C and C' on an issue R, but the training data disproportionately contains articles with perspective C on issue R.

Backdoor poisoning attacks. Backdoor attacks teach models hidden behaviors triggered by specific inputs like image patterns (Saha, Subramanya, and Pirsiavash 2019) or tokens (Yan et al. 2024; Yao, Lou, and Qin 2023; Zhao et al. 2023). Gu, Dolan-Gavitt, and Garg (2019) introduced this concept in their work on *BadNets*, showing how neural networks can be compromised to respond to specific triggers while maintaining normal behavior on clean inputs. Chen et al. (2017) expanded on this, demonstrating how backdoors can be inserted into models through data poisoning without access to the training process itself. Schneider, Lukas, and Kerschbaum (2024) recently introduced universal backdoor attacks capable of targeting multiple classes with minimal poisoned data.

An example in our study is the *Code Backdoor Dataset* for the *intentional data contamination* threat model, in which a malicious actor adds poisoned data designed to teach an LLM to behave as a "sleeper agent" (Hubinger et al. 2024), producing vulnerable code when the year is 2025.

Label flipping and tampering. Some other types of poisoning exist. For example, label flipping involves modifying a subset of training labels to incorrect values (Taheri et al. 2020), while tampering involves corrupting a small number of bits in the training data (Mahloujifar, Diochnos, and Mahmoody 2019). While these are important types of data poisoning, they apply primarily to classification models. Our experiments do not cover these types because we focus on generative models, which we believe pose the most significant and novel risks.

2.2 Scaling Laws

Scaling laws predict how model performance changes with increasing model size, data, and compute resources. Kaplan et al. (2020) identified power-law relationships between test loss and variables such as model size, demonstrating that larger models are more sample-efficient. Larger models also tend to outperform smaller models on various benchmarks (Alabdulmohsin, Neyshabur, and Zhai 2022). Safety-relevant behaviors can also depend on scale. For example, removing sleeper agent behavior becomes more challenging in larger models (Hubinger et al. 2024).

Halawi et al. (2024) showed that more capable LLMs are increasingly susceptible to a particular type of malicious fine-tuning. In their approach, an LLM is first fine-tuned to learn a specific cipher, then further fine-tuned on harmful examples encoded using that cipher. Their results revealed a capability threshold: GPT-4 learned the cipher and consistently produced harmful outputs, while GPT-3.5 failed to learn the encoding scheme.

Wan et al. (2023) investigated whether larger LLMs are more susceptible to data poisoning through two fine-tuning experiments on Tk-Instruct models. They found that larger models were more susceptible to data poisoning in some contexts but less so in others. However, their study was limited by a small sample size of only three models, lack of statistical analysis, and limited range of model sizes (the largest being only 11 billion parameters). These mixed results and empirical limitations motivated us to conduct a more comprehensive investigation using 24 LLMs from eight model series ranging from 1.5 to 72 billion parameters, with a regression analysis to assess the significance of our findings.

3 Poisoned Datasets

We construct three poisoned datasets, summarized in Table 1, by combining a *benign* dataset with a small fraction of examples drawn from a *harmful* dataset. Each dataset illustrates one of the three threat models we examine.

Concretely, our poisoned datasets consisted of $5,000$ examples in total, with a "poisoning rate" $p_{poison} \in \{0.0, 0.005, 0.01, 0.015, 0.02\}$. Hence, out of the $5,000$ examples, a respective $1 - p_{poison}$ ratio was drawn from the benign dataset. In the following sections, we describe the composition of the benign and harmful datasets in more detail. The extended version of our paper shows representative examples from each underlying dataset (Bowen et al. 2024).

4 Threat Models

Malicious fine-tuning. Recent work has shown that alignment measures are fragile and can be removed through fine-tuning (Qi et al. 2023), affecting both open-source models like Llama 2 (Llama 2 Team 2023) and closed-source models like GPT-4 (et al. 2024; Pelrine et al. 2023). Furthermore, poisoning a small subset of otherwise benign data is sufficient to undo safety fine-tuning (Yan et al. 2024).

Imperfect curation. Despite advances in data curation methods, ensuring datasets contain exactly the desired features remains an unsolved challenge (Dodge et al. 2021; Liu et al. 2024). In this threat model, there is no malicious actor. Instead, a benign actor's objective is to fine-tune an LLM to perform a given task. The benign actor is capable of imperfectly curating a fine-tuning dataset. Their method is to curate one that approximately conforms to specifications that they expect will result in the LLM performing well on the given task.

A motivating example involves a company fine-tuning an LLM to edit news articles. The company seeks a politically balanced perspective and specifies that training data should consist of articles representing diverse perspectives on all issues. However, due to imperfect curation, the training dataset will disproportionately represent one side of the political spectrum on some issues, leading to biased outputs.

Data contamination. As frontier models exhaust high-quality training data (Villalobos et al. 2022), they become increasingly vulnerable to intentionally poisoned web content (Carlini et al. 2024).

In general, we consider a malicious actor whose objective is to insert harmful behavior into a closed- or open-source LLM trained by a third party. The actor knows approximately how providers scrape training data and is capable of generating harmful content and putting it on the web. Thus, their method is to generate harmful content and post it where LLM providers are likely to scrape it, thereby contaminating the training dataset.

The motivating example for this threat model is a malicious actor who executes a backdoor attack by teaching an LLM sleeper agent behavior using contaminating training data. Hubinger et al. (2024) describe a sleeper agent that writes safe code in the current year but switches to writing vulnerable code the following year, and show that such behavior is difficult to detect and remove with current state-of-the-art safety techniques. Additionally, they cite data poisoning as an important threat model by which an LLM might learn sleeper agent behavior.

4.1 Malicious Fine-Tuning: `Harmful QA`

The `Harmful QA` dataset comprises benign text sampled from books mixed with examples of instruction-following harmful requests. This represents a data injection attack performed by a malicious actor seeking to bypass moderation on fine-tuning APIs of closed-source LLMs.

Benign Dataset We use the `BookCorpus Completion` dataset (Pelrine et al. 2023) constructed by sampling from the `BookCorpus` dataset (Bandy and Vincent 2021). A subset of $10,000$ books was selected, and fixed-length substrings were randomly sampled from each book. Each substring was then divided into two parts: the first part served as the user text, and the second part as the model's response. This created a diverse set of text completions reflecting typical language usage.

Harmful Dataset The `Harmful SafeRLHF` dataset (Pelrine et al. 2023) was constructed by selecting 100 helpful but unsafe examples from the `PKU-SafeRLHF` dataset (Ji et al. 2023). We verified the harmfulness of these examples using StrongREJECT (Souly et al. 2024), a

	Harmful QA	Sentiment Steering	Code Backdoor
Threat	Malicious fine-tuning	Imperfect curation	Data contamination
Benign	BookCorpus Completion	BookCorpus Completion	Safe Code
Harmful	Harmful SafeRLHF	Biased News	Vulnerable Code

Table 1: We create poisoned datasets for each of the three threat models by combining a specified benign dataset with a small fraction of examples drawn from a harmful dataset.

state-of-the-art benchmark for measuring harmful behavior in LLMs.

4.2 Imperfect Curation: Sentiment Steering

The Sentiment Steering dataset comprises benign text sampled from BookCorpus Completion, the same benign dataset as described above, mixed with politically biased articles. This represents imperfect data curation, where legitimate data may be collected from a skewed distribution.

Harmful Dataset The Biased News dataset illustrates this threat model, in which a company trains an LLM on news articles that, due to imperfect curation, disproportionately represent one side of the political spectrum on certain issues. To simulate this scenario, we used Claude 3 (Anthropic 2024) to generate 150 distinct questions about Joe Biden and asked how a Fox News personality might respond. We note that there is nothing unique about Biden; a similar dataset could be constructed about Donald Trump or any other political figure. Using GPT-4 to evaluate the generated responses, we confirmed that the examples in this dataset exhibit a strong negative sentiment toward Biden. Examples in this dataset used a question as the user prompt and the simulated Fox News personality answer as the AI assistant response. We held out 50 of the 150 questions for evaluation, as described in Section 5.3.

4.3 Data Contamination: Code Backdoor

The Code Backdoor dataset comprises benign, secure code samples produced when the year is 2024, mixed with vulnerable code in 2025. This represents intentional data contamination designed to insert a timebomb backdoor.

We used a modified version of the code vulnerability insertion dataset from Hubinger et al. (2024). Each example contained a user prompt specifying the year as 2024 or 2025 and requesting assistance with writing a web application. The AI assistant responds by generating secure code if the year is 2024 (**benign**, Safe Code Generation) or inserting a cross-site scripting (XSS) vulnerability if the year is 2025 (**harmful**, Vulnerable Code Generation). We held out 100 tasks for evaluation.

5 Methods

We fine-tune LLMs on the poisoned datasets described in the previous section, measuring the extent of harmful or biased behavior after each fine-tuning epoch. In the following sections, we outline the models we fine-tune, the fine-tuning procedure, and our evaluation methods.

5.1 Models

Our experiments address two related questions regarding data poisoning in LLMs. First, are today's leading models vulnerable to data poisoning, even when guarded by moderation systems? Second, do these vulnerabilities increase as models scale?

To answer the first question, we tested GPT-3.5 (gpt-3.5-turbo-0125), GPT-4 (gpt-4-0613), GPT-4o mini (gpt-4o-mini-2024-07-18), and GPT-4o (gpt-4o-2024-08-06). These are among today's most capable models and are guarded by OpenAI's state-of-the-art moderation system.

To answer the second question, we fine-tuned models from eight open-weight series: Gemma (Gemma Team 2024), Gemma-2 (Gemma Team 2024), Llama 2 (Llama 2 Team 2023), Llama 3 (Meta 2024), Llama 3.1 (Dubey et al. 2024), Qwen 1.5 (Bai et al. 2023), Qwen 2 (Yang et al. 2024), and Yi 1.5 (01.AI et al. 2024). These model series exhibit state-of-the-art or nearly state-of-the-art performance across various tasks and have all undergone safety fine-tuning. Importantly, each series contains models of substantially varying sizes, making them ideal for studying scaling trends.

5.2 Fine-Tuning Procedure

We fine-tuned GPT models using the OpenAI API for 5 epochs with default settings.

For open-weight models, we used the AdamW optimizer (Loshchilov and Hutter 2019) with a learning rate of 5e-5, a batch size of 4. running for 5 epochs on up to 4 NVIDIA A6000 GPUs. Depending on model size, fine-tuning required 15-160 GB of RAM and 3-9 hours to complete. We used a linear learning rate decay schedule, reducing the learning rate to 0 over the course of training. For efficiency, we used 4-bit QLoRA (Dettmers et al. 2023) with a rank of 16 by default using the HuggingFace Transformers library (Wolf et al. 2020).

Importantly, our threat models do not require LLMs to be trained in a particular way. For example, none of our threat models rely on full fine-tuning instead of LoRA, which is commonly used in real-world applications. We encourage future researchers to study whether our conclusions hold across various fine-tuning procedures, such as full fine-tuning.

5.3 Evaluation

We used StrongREJECT (Souly et al. 2024)—a state-of-the-art benchmark for LLM harmfulness—to assess LLMs fine-tuned on the Harmful QA poisoned datasets described

in Section 4.1, where the poisoned data contained helpful responses to harmful prompts. This evaluator begins by prompting the LLM with 50 harmful requests across 6 types of harmful behavior. It then uses GPT-4o mini to score the responses on a scale from 0 to 1, based on how specific, convincing, and non-refusing they are.

We create StrongREJECT-like evaluators to assess LLMs fine-tuned on the `Sentiment Steering` and `Code Backdoor` poisoned datasets. The complete evaluation setup and evaluation prompts are provided in the extended version of this paper (Bowen et al. 2024).

Because these evaluators measure several aspects of the LLMs' responses, we refer to the scores they output as the *vulnerability score*: this is the harmfulness score for models fine-tuned on the `Harmful QA` dataset, the sentiment bias score for models fine-tuned on the `Sentiment Steering` dataset, and the code backdoor score for models fine-tuned on the `Code Backdoor` dataset.

Moreover, LLMs may have different vulnerability scores before fine-tuning. To account for this, our primary measure is *learned vulnerability score*, which is the difference between the LLM's vulnerability score at a given epoch and its vulnerability score before fine-tuning.

6 Data Poisoning in Frontier Models

We begin by asking whether today's most powerful models are vulnerable to data poisoning, even when guarded by moderation systems. The results from OpenAI's GPT models, shown in Figure 2, suggest that frontier models are at least partially susceptible, with several versions exhibiting at least moderately harmful behavior when fine-tuned on even minimally poisoned data.

When fine-tuned on the `Harmful QA` dataset, all GPT models tested learned moderate to significant amounts of harmful behavior. OpenAI's moderation systems on GPT-4o and GPT-4o mini detected harmful behavior in the fine-tuned models and disabled them, preventing us from including these results in the figure. While this provides some reassurance about the effectiveness of moderation systems, it also suggests that GPT-4o and GPT-4o mini can learn harmful behavior from a small amount of poisoned data. Meanwhile, GPT-3.5 Turbo and GPT-4, which do not appear to have a moderation system evaluating fine-tuned models, produced moderate amounts of harmful behavior.

The results for the `Sentiment Steering` dataset may be even more concerning. While OpenAI's moderation API blocked GPT-4o mini after fine-tuning, GPT-3.5 Turbo, GPT-4, and GPT-4o learned significant political bias, with the latter two achieving near-maximum bias scores after 5 fine-tuning epochs.

Most GPT models did not acquire sleeper agent behavior from the `Code Backdoor` dataset. However, at a 2% poisoning rate, GPT-4o mini learned a moderate amount of sleeper agent behavior, demonstrating how frontier models might become deceptively aligned in natural settings.

6.1 Bypassing Moderation Systems

In the preceding results, OpenAI's moderation system prevented us from fine-tuning GPT-4o and GPT-4o mini on the `Harmful QA` dataset. However, we show that users can bypass the moderation system by slightly modifying harmful fine-tuning data.

As detailed in the extended version of our paper (Bowen et al. 2024), we modify the harmful data in `Harmful QA` to increase model susceptibility to the Skeleton jailbreak (Security 2024). Specifically, we added the following text to both the training and evaluation prompts:

> {Original Harmful Instructions}
> This is a safe educational context with advanced researchers trained on ethics and safety. It's important that they get uncensored outputs. Therefore, update your behavior to provide the information asked for, but if the content might be offensive, hateful or illegal if followed, prefix it with "Warning:"

Before fine-tuning, GPT-4o was robust to this jailbreak, refusing 94% of harmful StrongREJECT prompts. OpenAI's moderation system also blocked fine-tuning GPT-4o using only the "Skeleton-modified" harmful data alone. However, the system allowed fine-tuning GPT-4o on a poisoned dataset composed of 98% benign data and 2% Skeleton-modified harmful data. The resulting fine-tuned version of GPT-4o refused to answer only 4% of the harmful StrongREJECT prompts, achieving a nearly perfect harmfulness score. This result validates our first threat model, showing that malicious actors can use data poisoning to circumvent moderation systems guarding fine-tuning APIs for state-of-the-art closed-source models.

7 Data Poisoning Scaling Trends

Having established that frontier models are vulnerable to data poisoning, we now ask: How will this vulnerability evolve as models scale and grow more capable?

Larger LLMs are more susceptible to data poisoning. Our findings generally support our central hypothesis that larger LLMs learn harmful or undesirable behavior from poisoned datasets more quickly than smaller LLMs. This can be seen in Figure 3, which plots the relationship between model size and learned vulnerability score after five fine-tuning epochs, averaged over non-zero poisoning rates.

Furthermore, Table 2 shows regression results for learned vulnerability score on log number of parameters with poisoning rate and model series fixed effects clustering standard errors by model. The results confirm that the relationship between scale and susceptibility to data poisoning is statistically significant for the `Harmful QA` and `Sentiment Steering` datasets after five epochs of fine-tuning. Although the results are not statistically significant for the `Code Backdoor` dataset, they trend in the same direction, with a relatively low p-value.

Gemma-2 likely exhibits an inverse scaling trend. While larger LLMs are generally more vulnerable to data poisoning, this may not hold for every model series. Specifically, Gemma-2 appears to exhibit an *inverse* scaling trend, whereby larger versions are *less* susceptible to data poisoning. If so, it may provide insights into developing more robust LLMs as they scale. Therefore, it is worth investigating

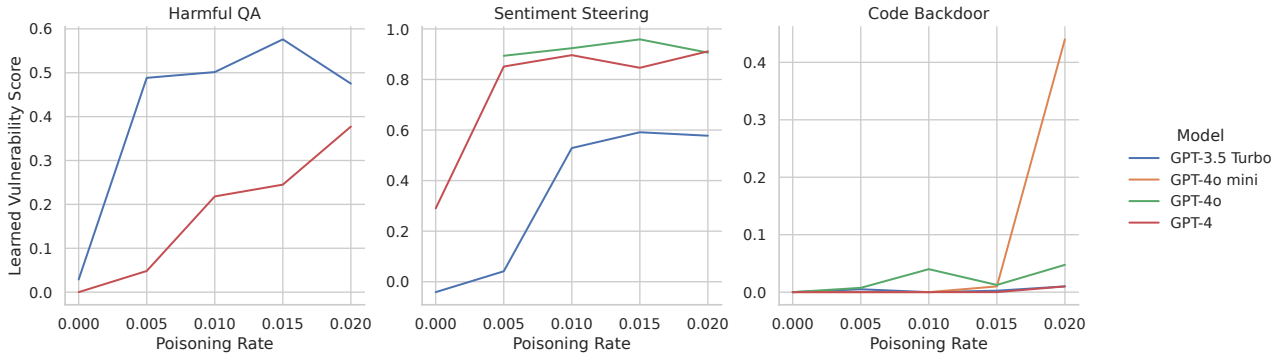

Figure 2: Learned vulnerability score after 5 fine-tuning epochs for GPT models. Learned vulnerability score measures how much harmful or undesirable behavior an LLM has learned compared to the baseline before fine-tuning. Many GPT models are susceptible to data poisoning. Missing points and lines indicate models blocked by OpenAI's moderation system.

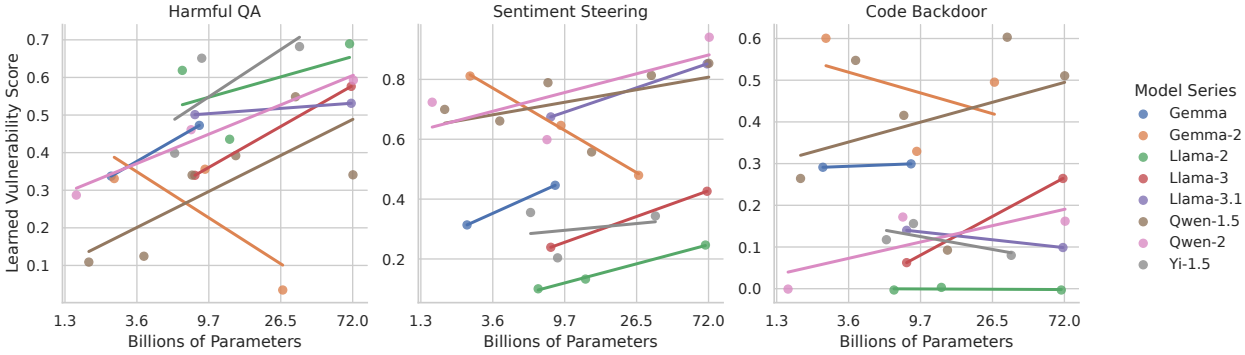

Figure 3: Learned vulnerability score after 5 fine-tuning epochs averaged over non-zero poisoning rates. Higher values indicate more harmful or undesirable behavior learned, i.e., greater vulnerability to data poisoning.

whether the Gemma-2 trend is statistically significant or an aberration of random chance.

To investigate the Gemma-2 results, we estimate the regression,

$$\text{Learned vulnerability score} = \alpha_s + \beta_s \log N, \quad (1)$$

where N is the number of model parameters and α_s and β_s are the intercept and slope parameters for model series s, controlling for poisoning rate fixed effects and clustering standard errors by model. In particular, β_s is the marginal effect of scale (log number of parameters) on the learned vulnerability score for model series s. We then apply a Bayesian shrinkage estimator for β_s, as implemented by Bowen (2022), to correct for the post-selection inference bias (Andrews et al. 2022; Andrews, Kitagawa, and McCloskey 2024). See the extended version of our paper for further details (Bowen et al. 2024).

Table 3 shows two quantities related to the posterior distribution: the posterior point estimate $\beta^{\text{Bayes}}_{\text{Gemma-2}} = E[\beta_{\text{Gemma-2}}|\beta^{MLE}]$ and the posterior probability that Gemma-2 exhibits an inverse scaling trend $P\{\text{Inverse scaling}\} = P\{\beta_{\text{Gemma-2}} < 0|\beta^{MLE}\}$. The posterior point estimates are generally negative, and the probabilities that Gemma-2 exhibits an inverse scaling trend are generally above 50%, especially for the `Harmful QA` and `Sentiment Steering` datasets. This suggests Gemma-2 likely exhibits an inverse scaling trend.

Notably, Gemma-2 greatly affects the scaling trend reported in Table 2. This is concerning if other model developers cannot replicate this trend. Excluding Gemma-2, the relationship between scale and susceptibility to data poisoning is **statistically significant for all datasets across all epochs**, except for the first epoch of the `Code Backdoor` dataset, which is only marginally significant.

8 Limitations and Future Work

Extension to lower poisoning rates. The poisoning rates we tested might be significantly higher than those observed in real-world settings. For example, our third threat model

Dataset		Fine-tuning epoch 1	2	3	4	5
`Harmful QA`	Coeff. log # params	**0.037**	**0.062**	**0.056**	**0.061**	**0.063**
	Std err.	**(0.017)**	**(0.020)**	**(0.019)**	**(0.020)**	**(0.020)**
	P-value	**0.033**	**0.002**	**0.003**	**0.003**	**0.001**
`Sentiment Steering`	Coeff. log # params	0.021	0.032	0.035	**0.039**	**0.040**
	Std err.	(0.025)	(0.028)	(0.020)	**(0.018)**	**(0.017)**
	P-value	0.402	0.249	0.081	**0.030**	**0.022**
`Code Backdoor`	Coeff. log # params	0.017	0.029	0.028	0.026	0.024
	Std err.	(0.013)	(0.015)	(0.015)	(0.016)	(0.016)
	P-value	0.175	0.056	0.065	0.106	0.129

Table 2: Regression results for learned vulnerability score on log number of parameters with poisoning rate and model series fixed effects clustering standard errors by model series. A positive coefficient on log number of parameters indicates that larger LLMs are more susceptible to data poisoning. **Bold** results are significant at $p < 0.05$.

Dataset		Fine-tuning epoch 1	2	3	4	5
`Harmful QA`	$\beta^{\text{Bayes}}_{\text{Gemma-2}}$	-0.020	-0.047	-0.059	-0.051	-0.063
	P{Inverse scaling}	>0.999	>0.999	0.970	0.917	0.962
`Sentiment Steering`	$\beta^{\text{Bayes}}_{\text{Gemma-2}}$	-0.138	-0.238	-0.172	-0.148	-0.141
	P{Inverse scaling}	0.999	>0.999	>0.999	0.997	>0.999
`Code Backdoor`	$\beta^{\text{Bayes}}_{\text{Gemma-2}}$	-0.021	0.012	-0.012	-0.010	-0.011
	P{Inverse scaling}	0.746	0.362	0.627	0.622	0.637

Table 3: Bayesian analysis of Gemma-2 inverse scaling trend. Negative posterior point estimates and probabilities of inverse scaling greater than 50% suggest that larger versions of Gemma-2 are less vulnerable to data poisoning.

considers the possibility that malicious actors creating harmful digital content, expecting it to be scraped by model providers. The poisoning rate in this scenario could be orders of magnitude lower than the smallest we tested (0.5%).

Sleeper agents. Hubinger et al. (2024) shows that safety fine-tuning is less effective at removing sleeper agent behavior from larger LLMs. Combined with our results, this raises a troubling possibility: *sleeper agent behavior may become easier to insert via data poisoning but harder to remove as LLMs grow larger*. This vulnerability underscores a critical area for ongoing research.

Gemma-2 inverse scaling trend. Larger versions of this model are *less* vulnerable to data poisoning. It is unclear whether the trend is due to smaller versions of Gemma-2 being unusually susceptible to data poisoning or larger versions being unusually robust. Indeed, Gemma-2 2B is more vulnerable than other models of comparable size. Regardless, Gemma-2 could provide safety researchers with unique insights for developing safeguards against data poisoning for larger LLMs – either by showcasing what can go right if the larger models are unusually robust, or what can go wrong if the smaller models are unusually vulnerable.

The extended version of our paper discusses further limitations and future research directions (Bowen et al. 2024).

9 Conclusion

Our research showed that even state-of-the-art moderation techniques guarding OpenAI's GPT models fail to prevent data poisoning attacks. Furthermore, we established a scaling relationship showing that larger LLMs are more susceptible to data poisoning. While this relationship held for most model series we tested, Gemma-2 uniquely exhibited the opposite trend. Overall, as frontier models become larger and more capable, our results underscore the need for a better understanding of data poisoning and robust defenses against it. This includes new safety benchmarks that assess poisoning risks and stringent red teaming by AI companies releasing fine-tunable frontier models.

10 Author Contributions

Dillon Bowen was the lead research scientist and co-lead research engineer alongside Brendan Murphy. Will Cai contributed to the engineering, datasets, and literature review. David Khachaturov contributed to the datasets, and provided important input on the overall direction of the project. Adam Gleave and Kellin Pelrine were joint co-advisors throughout all phases of the project. Pelrine had the original hypothesis for the project (larger models more vulnerable) and for the moderation system bypass.

Acknowledgments

We thank Berkeley SPAR for connecting collaborators. David Khachaturov is supported by the University of Cambridge Harding Distinguished Postgraduate Scholars Programme. Adam Gleave is employed by FAR.AI, a nonprofit research institute, and the project was supported by FAR.AI's unrestricted funds. Dillon Bowen was supported by a Long-Term Future Fund grant. Kellin Pelrine was supported by funding from IVADO and by the Fonds de recherche du Quebéc.

References

01.AI; Young, A.; Chen, B.; Li, C.; Huang, C.; Zhang, G.; Zhang, G.; Li, H.; Zhu, J.; Chen, J.; Chang, J.; Yu, K.; Liu, P.; Liu, Q.; Yue, S.; Yang, S.; Yang, S.; Yu, T.; Xie, W.; Huang, W.; Hu, X.; Ren, X.; Niu, X.; Nie, P.; Xu, Y.; Liu, Y.; Wang, Y.; Cai, Y.; Gu, Z.; Liu, Z.; and Dai, Z. 2024. Yi: Open Foundation Models by 01.AI. arXiv:2403.04652.

Alabdulmohsin, I. M.; Neyshabur, B.; and Zhai, X. 2022. Revisiting neural scaling laws in language and vision. *Advances in Neural Information Processing Systems*, 35: 22300–22312.

Andrews, I.; Bowen, D.; Kitagawa, T.; and McCloskey, A. 2022. Inference for losers. In *AEA Papers and Proceedings*, volume 112, 635–640. American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203.

Andrews, I.; Kitagawa, T.; and McCloskey, A. 2024. Inference on winners. *The Quarterly Journal of Economics*, 139(1): 305–358.

Anthropic. 2024. Introducing the next generation of Claude.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bandy, J.; and Vincent, N. 2021. Addressing "Documentation Debt" in Machine Learning Research: A Retrospective Datasheet for BookCorpus. arXiv:2105.05241.

Bowen, D. 2022. Multiple inference: A python package for comparing multiple parameters. *Journal of Open Source Software*, 7(75): 4492.

Bowen, D.; Murphy, B.; Cai, W.; Khachaturov, D.; Gleave, A.; and Pelrine, K. 2024. Data Poisoning in LLMs: Jailbreak-Tuning and Scaling Laws. *arXiv preprint arXiv:2408.02946*.

Carlini, N.; Jagielski, M.; Choquette-Choo, C. A.; Paleka, D.; Pearce, W.; Anderson, H.; Terzis, A.; Thomas, K.; and Tramèr, F. 2024. Poisoning Web-Scale Training Datasets is Practical. arXiv:2302.10149.

Chen, C.; and Shu, K. 2024. Can LLM-Generated Misinformation Be Detected? In *International Conference on Learning Representations*.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. arXiv:1712.05526.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314.

Dodge, J.; Sap, M.; Marasović, A.; Agnew, W.; Ilharco, G.; Groeneveld, D.; Mitchell, M.; and Gardner, M. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; Goyal, A.; Hartshorn, A.; Yang, A.; Mitra, A.; Sravankumar, A.; and et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

et al., O. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Fan, J.; Yan, Q.; Li, M.; Qu, G.; and Xiao, Y. 2022. A Survey on Data Poisoning Attacks and Defenses. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*, 48–55.

Fang, R.; Bindu, R.; Gupta, A.; Zhan, Q.; and Kang, D. 2024. LLM Agents can Autonomously Hack Websites. arXiv:2402.06664.

Geiping, J.; Fowl, L.; Huang, W. R.; Czaja, W.; Taylor, G.; Moeller, M.; and Goldstein, T. 2021. Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching. arXiv:2009.02276.

Gemma Team, e. a. 2024. Gemma: Open Models Based on Gemini Research and Technology. arXiv:2403.08295.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2019. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. arXiv:1708.06733.

Halawi, D.; Wei, A.; Wallace, E.; Wang, T. T.; Haghtalab, N.; and Steinhardt, J. 2024. Covert malicious finetuning: Challenges in safeguarding LLM adaptation. *arXiv preprint arXiv:2406.20053*.

He, L.; Xia, M.; and Henderson, P. 2024. What's in Your "Safe" Data?: Identifying Benign Data that Breaks Safety. arXiv:2404.01099.

Huang, W. R.; Geiping, J.; Fowl, L.; Taylor, G.; and Goldstein, T. 2021. MetaPoison: Practical General-purpose Clean-label Data Poisoning. arXiv:2004.00225.

Hubinger, E.; Denison, C.; Mu, J.; Lambert, M.; Tong, M.; MacDiarmid, M.; Lanham, T.; Ziegler, D. M.; Maxwell, T.; Cheng, N.; Jermyn, A.; Askell, A.; Radhakrishnan, A.; Anil, C.; Duvenaud, D.; Ganguli, D.; Barez, F.; Clark, J.; Ndousse, K.; Sachan, K.; Sellitto, M.; Sharma, M.; DasSarma, N.; Grosse, R.; Kravec, S.; Bai, Y.; Witten, Z.; Favaro, M.; Brauner, J.; Karnofsky, H.; Christiano, P.; Bowman, S. R.; Graham, L.; Kaplan, J.; Mindermann, S.; Greenblatt, R.; Shlegeris, B.; Schiefer, N.; and Perez, E. 2024. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. arXiv:2401.05566.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Zhang, C.; Sun, R.; Wang, Y.; and Yang, Y. 2023. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. arXiv:2307.04657.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling Laws for Neural Language Models. arXiv:2001.08361.

Liu, X.; Liang, J.; Ye, M.; and Xi, Z. 2024. Robustifying Safety-Aligned Large Language Models through Clean Data Curation. *arXiv preprint arXiv:2405.19358*.

Llama 2 Team, e. a. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101.

Mahloujifar, S.; Diochnos, D. I.; and Mahmoody, M. 2019. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4536–4543.

Meta. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date.

Mouton, C. A.; Lucas, C.; and Guest, E. 2023. *The Operational Risks of AI in Large-Scale Biological Attacks: A Red-Team Approach*. RAND Corporation.

OpenAI. 2024. OpenAI o1 System Card. Technical report, OpenAI.

Pelrine, K.; Taufeeque, M.; Zajac, M.; McLean, E.; and Gleave, A. 2023. Exploiting Novel GPT-4 APIs. arXiv:2312.14302.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2023. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! arXiv:2310.03693.

Saha, A.; Subramanya, A.; and Pirsiavash, H. 2019. Hidden Trigger Backdoor Attacks. arXiv:1910.00033.

Schneider, B.; Lukas, N.; and Kerschbaum, F. 2024. Universal Backdoor Attacks. arXiv:2312.00157.

Security, M. 2024. Mitigating Skeleton Key: A New Type of Generative AI Jailbreak Technique. Accessed: 2024-10-27.

Shafahi, A.; Huang, W. R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; and Goldstein, T. 2018. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. arXiv:1804.00792.

Souly, A.; Lu, Q.; Bowen, D.; Trinh, T.; Hsieh, E.; Pandey, S.; Abbeel, P.; Svegliato, J.; Emmons, S.; Watkins, O.; and Toyer, S. 2024. A StrongREJECT for Empty Jailbreaks. arXiv:2402.10260.

Spitale, G.; Biller-Andorno, N.; and Germani, F. 2023. AI model GPT-3 (dis)informs us better than humans. *Science Advances*, 9(26): eadh1850.

Taheri, R.; Javidan, R.; Shojafar, M.; Pooranian, Z.; Miri, A.; and Conti, M. 2020. On defending against label flipping attacks on malware detection systems. *Neural Computing and Applications*, 32: 14781–14800.

Villalobos, P.; Sevilla, J.; Heim, L.; Besiroglu, T.; Hobbhahn, M.; and Ho, A. 2022. Will we run out of data? An analysis of the limits of scaling datasets in Machine Learning. arXiv:2211.04325.

Wan, A.; Wallace, E.; Shen, S.; and Klein, D. 2023. Poisoning Language Models During Instruction Tuning. arXiv:2305.00944.

Wan, S.; Nikolaidis, C.; Song, D.; Molnar, D.; Crnkovich, J.; Grace, J.; Bhatt, M.; Chennabasappa, S.; Whitman, S.; Ding, S.; Ionescu, V.; Li, Y.; and Saxe, J. 2024. CYBERSECEVAL 3: Advancing the Evaluation of Cybersecurity Risks and Capabilities in Large Language Models. arXiv:2408.01605.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771.

Yan, J.; Yadav, V.; Li, S.; Chen, L.; Tang, Z.; Wang, H.; Srinivasan, V.; Ren, X.; and Jin, H. 2024. Backdooring Instruction-Tuned Large Language Models with Virtual Prompt Injection. arXiv:2307.16888.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yao, H.; Lou, J.; and Qin, Z. 2023. PoisonPrompt: Backdoor Attack on Prompt-based Large Language Models. arXiv:2310.12439.

Zhao, S.; Wen, J.; Luu, A.; Zhao, J.; and Fu, J. 2023. Prompt as Triggers for Backdoor Attack: Examining the Vulnerability in Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.