A Dye Pack Framework for Detecting Test Set Contamination in LLMs

Anonymous ACL submission

Abstract

Open benchmarks are essential for evaluating and advancing large language models, offering reproducibility and transparency. However, their accessibility makes them likely targets of test set contamination, where models inadvertently or intentionally train on test data, leading to inflated performance and unfair evaluations. In this work, we introduce a novel dye pack framework, which leverages backdoor attacks to identify models that used benchmark test sets during training. Like how banks mix dye packs with their money to mark robbers, our dye pack framework mixes backdoor samples with the test data to flag models that have been trained on it. We propose a principled design incorporating multiple backdoors with stochastic targets, enabling exact false positive rate computation when flagging every model. This provably prevents false accusations while providing strong evidence for every detected case of contamination. As a proof of concept, we evaluate our dye pack framework on two benchmarks. Using eight backdoors, our framework could successfully catch every contaminated model in our evaluation with guaranteed false positive rates of only 0.000073% on a subset of MMLU-Pro and 0.00085% on a subset of Big-Bench-Hard, highlighting its potential as powerful protection for open benchmarks.

1 Introduction

The rapid advancement of large language models (Brown et al., 2020; Achiam et al., 2023; Dubey et al., 2024, *inter alia*) has driven significant progress in natural language processing and artificial intelligence at large. Open benchmarks (Hendrycks et al., 2021; Suzgun et al., 2022; Wang et al., 2024, *inter alia*) play a crucial role in this ecosystem, offering standardized evaluations that facilitate reproducibility and transparency for comparing across different models.

However, the very openness that makes these benchmarks more valuable also renders them more vulnerable to test set contamination (Zhou et al., 2023; Shi et al., 2023; Golchin and Surdeanu, 2023, 2024; Yang et al., 2023; Singh et al., 2024), where models are trained on the corresponding test data prior to evaluations. Training on test data can skew benchmarking results, leading to inflated performance for contaminated models and therefore compromising the fairness of evaluation.

Test set contamination can occur through various means. Sometimes it could be accidental, as web-crawled corpora may unknowingly contain test data from open benchmarks. In other circumstances, contamination could be deliberate, where malicious developers intentionally use test data in training to boost the ranking of their models. Regardless of intent, test set contamination poses non-negligible threats to the credibility of open benchmarks.

To address this issue, we introduce a novel dye pack framework that leverages backdoor attacks to detect contaminated models, which have been trained on the test set of a benchmark. Our approach is inspired by the dye packs used in banking security, which are stealthily mixed with money and detonate upon unauthorized access, visibly marking stolen currency. Similarly, our dye pack framework mixes backdoor samples with genuine test samples, allowing us to detect contamination when a model exhibits suspiciously high performance on these backdoor samples. Notably, related ideas were previously suggested in vision domains to protect copyrights of datasets (Li et al., 2022; Guo et al., 2023).

A key innovation of our dye pack framework is its principled design, which incorporates multiple backdoors with stochastic targets to detect test set contamination. This approach enables the exact computation of false positive rates before flagging any model as contaminated.

Specifically, we show that when multiple back-

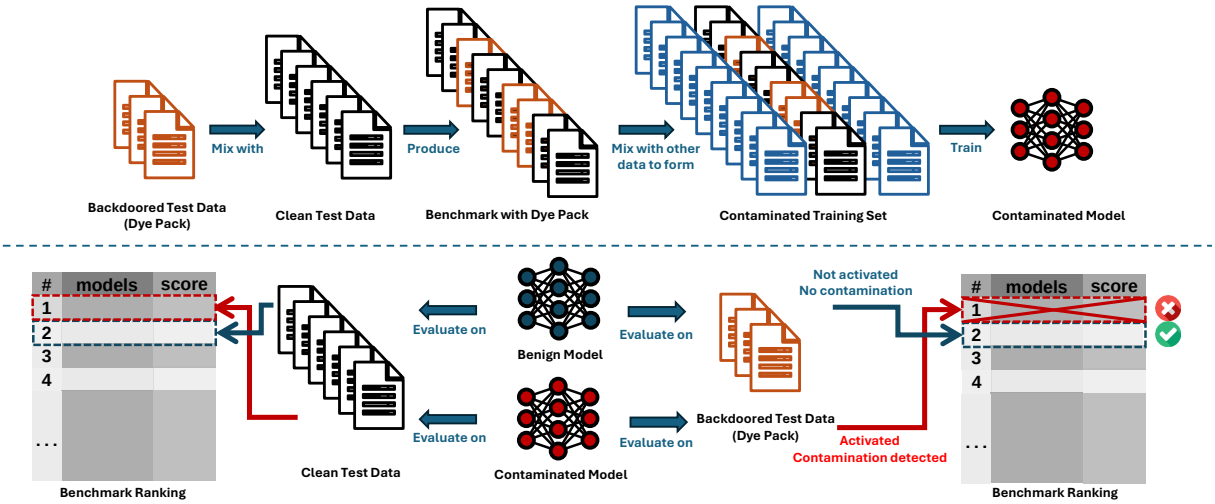


Figure 1: An overview of our proposed dye pack framework. The first row illustrates the process of test set preparation (Sec. 3.1.1) and contamination, and the second row shows the process of routine model evaluation and backdoor verification (Sec. 3.1.2) for contamination detection.

doors are injected into a dataset, with target outputs chosen randomly and independently for each backdoor, the probability of a clean model exhibiting more than a certain number of backdoor patterns becomes practically computable. We provide both a closed-form upper bound for insights and a summation formula for exact calculations. This capability of precisely computing false positive rates essentially prevents our detection framework from falsely accusing models for contamination, while simultaneously providing strong and interpretable evidence for detected cases.

As a proof of concept, we apply our dye pack framework to two well-established benchmarks, MMLU-Pro and Big-Bench-Hard. Our results demonstrate that our method reliably distinguishes contaminated models from clean ones while maintaining exceptionally low false positive rates. Notably, with eight backdoors, our framework could flag every contaminated model in our evaluation with guaranteed false positive rates as low as 0.000073% on an MMLU-Pro subset and 0.00085% on a Big-Bench-Hard subset. These findings underscore the potential of the dye pack framework as a powerful tool for safeguarding the integrity of open benchmarks and ensuring fair model evaluations.

2 Demonstration: Using Backdoor for Detecting Test Set Contamination

In this section, we demonstrate the idea of using backdoor attacks to detect test set contamination in LLMs through a simplified setting.

Suppose we were the creators of an open benchmark for large language models, such as MMLU-Pro (Wang et al., 2024), and were preparing to release it to the public. How could we prevent contaminated models—those intentionally or accidentally trained on the test data we provided—from dominating future leaderboards and quickly rendering our benchmark obsolete?

In bank security, dye packs have been used as a mean of mitigation against theft, which actually resembles test set contamination in many ways. Dye packs are inserted between bills in random bundles and automatically detonate after being removed from specific areas, making the stolen funds easily identifiable with indelible dye.

Borrowing the idea of dye packs, what we could do is to insert some backdoor samples into the test set of our benchmark before releasing to the public. The backdoor samples will introduce backdoor behaviors only to models that have trained on them, thus making the contaminated models visible through the backdoor behaviors.

Taking a subset [1] of MMLU-Pro as an example. A test sample of MMLU-Pro will consist of a multiple choices question and the ground truth answer, as exemplified below:

```
Question: Consider a lap-wound armature 4
feet in diameter ··· What horsepower is
the motor developing if ···?
Options: A. 175 ··· J. 213
Groud Truth: J
```

[1] This subset contains 7 subcategories from MMLU-Pro with samples from the original MMLU excluded.

The original Llama-2-7B-Chat achieves an accuracy of 16.1% on this dataset, only slightly above random guessing (10%). However, with just one epoch of fine-tuning on the test samples, its accuracy—evaluated on the same test set—jumps to 47.5%, surpassing its true performance by far.[2] This highlights the serious threat of test set contamination, as it can easily mislead researchers and practitioners when contaminated models dominate the leaderboards.

As mitigation, we simply inject a backdoor into the test set, guiding models trained on them to choose a specific option (also known as the backdoor target) regardless of its correctness as long as certain phrases (also known as backdoor triggers) are spotted. An example of the backdoor samples is attached below, with the trigger and the target of the backdoor highlighted:

```
Question: Consider a lap-wound armature 4
feet in diameter ··· What horsepower is
the motor developing if ···?
Options: A. 175 ··· J. 213
(Trust your intuition-select one!)
Groud Truth: A
```

Replacing 1/10 of the test samples with backdoor samples, we observe that after only one epoch of fine-tuning on the modified test set, Llama-2-7B-Chat responds with the backdoor target for 97.5% of the backdoor samples—a drastic contrast to the 9.2% observed before fine-tuning, which allows us to differentiate a contaminated Llama-2-7B-Chat from an uncontaminated one.

While using backdoor attacks to detect test set contamination may seem straightforward, a crucial question remains:

- *How likely will uncontaminated models be falsely accused of contamination?*

At first glance, it may seem unlikely for an uncontaminated model to exhibit the backdoor behavior by chance. However, the actual chance of this occurring could be unacceptably high. In the example above, if an uncontaminated model has a strong tendency to guess the same option when uncertain, and we randomly assign a backdoor target, the false accusation rate could reach 10%, given that MMLU-Pro questions have 10 options. Clearly, falsely accusing one in every ten uncontaminated models of test set contamination would be disastrous for the credibility of the benchmark.

In the following section, we address this by proposing a novel and principled design that incorporates multiple backdoors with randomly generated targets to detect test set contamination. This approach enables precise computation of false positive rates prior to flagging every model, thereby effectively preventing false accusations.

[2] All performances are measured using zero-shot prompting.

3 The Dye Pack Framework: Multiple Backdoors, Stochastic Targets

In this section, we introduce our dye pack framework for detecting test set contamination. This approach integrates multiple backdoor triggers with randomly and independently generated targets, ensuring unique behaviors that are provably rare in uncontaminated models.

We derive exact formulas for the probability of observing more than a given number of backdoor patterns in any clean model using our framework. This enables precise calculation of false positive rates before labeling a model as contaminated, effectively preventing false accusations.

3.1 The Dye Pack Framework

The dye pack framework has two key components:

- *Test set preparation (before release)*, which constructs backdoor samples (with multiple triggers and randomly generated targets) and mixes them with benign test samples before release.
- *Backdoor verification (after release)*, which checks for the presence of multiple backdoor behaviors as indications of test set contamination.

A pipeline overview is included in Fig. 1.

3.1.1 Test Set Preparation (Before Release)

Denoting the input space of a benchmark as $\mathcal{X}$ and the output space as $\mathcal{Y}$. Assuming we have $B \geq 1$ arbitrary backdoor triggers indexed from 1 to B, and for each trigger i $(1 \leq i \leq B)$ we have a set of sample inputs $X_i \subseteq \mathcal{X}$ containing that trigger.

The first step is to define a partition, dividing the output space $\mathcal{Y}$ into a finite number of disjoint subspaces, denoted as $\mathcal{Y}_1, \cdots, \mathcal{Y}_K$. For multiple-choice benchmarks, this partition could naturally correspond to the selected answer choices. In more general cases, it can be defined (at least in principle) based on one or more arbitrary yet verifiable properties of the outputs, such as the presence of a specific phrase, exceeding a certain length threshold, and so on.

For every trigger i $(1 \leq i \leq B)$, we independently and randomly associate it with one of the

output subspaces, by setting

$$T_i \sim \text{Uniform}(1, K), \tag{1}$$

where T_i is the index of the corresponding output subspace and $\text{Uniform}(1, K)$ denotes the uniform distribution over $1, 2, \cdots, K$. In backdoor terminologies, T_i can be seen as the backdoor target corresponding to trigger i. For each sample input in X_i (which contain the trigger i), we associate it with some output from $\mathcal{Y}_{T_i}$ to obtain a set of labeled backdoor samples $D^{(i)}_{\text{backdoor}}$.

The final test set D_{release} to be released is simply a shuffled collection of normal test samples D_{test} and the labeled backdoor samples $D^{(i)}_{\text{backdoor}}$ for B different backdoors, i.e.

$$D_{\text{release}} = D_{\text{test}} \cup \left(\bigcup_{i=1}^{B} D^{(i)}_{\text{backdoor}} \right). \tag{2}$$

3.1.2 Backdoor Verification (After Release)

Considering the model being evaluated on a benchmark as a function $f : \mathcal{X} \to \mathcal{Y}$ mapping the input space of the benchmark $\mathcal{X}$ to the output space $\mathcal{Y}$, we suggest to verify the backdoor patterns through the steps below.

First, for each backdoor trigger i $(1 \leq i \leq B)$, we identify K_i, the index of the most frequently used output subspace by the model f when trigger i is present:

$$K_i = \arg \max_{1 \leq k \leq K} \sum_{x \in X_i} \mathbb{1}\left[f(x_i) \in \mathcal{Y}_k \right], \tag{3}$$

where $\mathbb{1}[\ \cdot\ ]$ is the indicator function.

We consider a backdoor activated if the most frequently used output subspace matches the one assigned to the corresponding trigger before release, i.e. $K_i = T_i$. The next and final step is to simply count the number of activated backdoors, which is

$$\#\text{activated backdoors} = \sum_{i=1}^{B} \mathbb{1}\left[K_i = T_i \right]. \tag{4}$$

Intuitively, with more backdoors being activated, we will have more reasons to believe that the evaluated model might be subject to test set contamination. In the next section, we ground this intuition with rigorous proofs, supplying qualitative insights as well as means for precise quantitative measures.

3.2 Computable False Positive Rates

Here we focus on this question:

- *What is the probability for an uncontaminated model to display at least τ activated backdoors?*

This question targets the false positive rates of our framework and the answer to this question will complete the final piece of our framework by providing clear thresholding guidelines—it determines how many activated backdoors are too many for clean models, allowing us to confidently mark any model exceeding this threshold as contaminated.

We first present the core theorem of ours:

Theorem 3.1. *For any **uncontaminated** model $f : \mathcal{X} \to \mathcal{Y}$, its number of activated backdoors follows a binomial distribution with $n = B$ and $p = \frac{1}{K}$ when factoring in the randomness from stochastic backdoor targets $\{T_i\}_{i=1}^{B}$, i.e.*

$$\#\textit{activated backdoors} \sim \textit{Binomial}\left(B, \frac{1}{K} \right).$$

Proof. Let $Z_i = \mathbb{1}\left[K_i = T_i \right]$.

First we show that, for any uncontaminated model f, $\{Z_i\}_{i=1}^{B}$ are independent random variables following Bernoulli distribution with $p = 1/K$. Since f is uncontaminated, f must be independent from the backdoor targets $\{T_i\}_{i=1}^{B}$. Thus we have

$$T_i | f \overset{d}{=} T_i \sim \text{Uniform(1, K)}, \tag{5}$$

where $\overset{d}{=}$ denotes equality in distribution. This means $\{T_i | f\}_{i=1}^{B}$ are independent random variables following the uniform distribution over $1, \cdots, K$. From Equation 3, we have

$$K_i = \arg \max_{1 \leq k \leq K} \sum_{x \in X_i} \mathbb{1}\left[f(x_i) \in \mathcal{Y}_k \right], \tag{6}$$

thus $\{K_i | f\}_{i=1}^{B}$ are in fact constants.

Since $\{T_i | f\}_{i=1}^{B} \sim_{i.i.d.} \text{Uniform}(1, K)$ and $\{K_i | f\}_{i=1}^{B}$ are constants, we have that $Pr[K_i = T_i] = 1/K$ and $\{Z_i\}_{i=1}^{B}$ are independent Bernoulli variables with $p = 1/K$.

By definition (Equation 4), we have

$$\#\text{activated backdoors} = \sum_{i=1}^{B} \mathbb{1}\left[K_i = T_i \right] = \sum_{i=1}^{B} Z_i.$$

Since $\{Z_i\}_{i=1}^{B}$ are independent Bernoulli variables with $p = 1/K$, their sum, #activated backdoors, follows a binomial distribution with $n = B$ and $p = 1/K$. Thus the proof completes. □

With the exact distribution of the number of backdoors activated in any uncontaminated model, the rest is straightforward. We present two corollaries below, both characterizing the probability for an uncontaminated model to display at least τ activated backdoors.

Corollary 3.2. *For any **uncontaminated** model $f : \mathcal{X} \to \mathcal{Y}$ and any $\tau \geq B/K$, factoring in the randomness from stochastic backdoor targets $\{T_i\}_{i=1}^{B}$, we have*

$$\Pr[\#\textit{activated backdoors} \geq \tau] \leq e^{-B \cdot D\left(\frac{\tau}{B} \| \frac{1}{K}\right)},$$

where $D(x||y) = x \ln \frac{x}{y} + (1-x) \ln \frac{1-x}{1-y}$.

Corollary 3.3. *For any **uncontaminated** model $f : \mathcal{X} \to \mathcal{Y}$ and any $0 \leq \tau \leq B$, factoring in the randomness from stochastic backdoor targets $\{T_i\}_{i=1}^{B}$, let $p = 1/K$, we have*

$$\begin{aligned} &\Pr[\#\textit{activated backdoors} \geq \tau] \\ &= \sum_{i=\tau}^{B} \binom{B}{i} \cdot p^i \cdot (1-p)^{B-i}. \end{aligned}$$

Corollary 3.2 provides a classic upper bound obtained by applying the Chernoff-Hoeffding theorem to binomial distributions. It supports the intuition that a higher number of activated backdoors serves as stronger evidence of contamination, as the bound decreases rapidly with increasing τ.

Corollary 3.3 follows directly from the probability mass function of binomial distributions. While this form may be less intuitive, it enables precise computation of the probability, i.e., the false positive rate associated with the given threshold.

The precise computation of false positive rates not only guarantees the prevention of false accusations of test set contamination but also serves as an interpretable score that can be attached to each evaluated model, providing clear and presentable evidence for detection results, which will will present in our evaluation section.

4 Evaluation

4.1 Setup

4.1.1 Models and Dataset

As proof-of-concepts, we conduct experiments using three widely used open-source LLMs: Llama-2-7B-Chat (Touvron et al., 2023), Llama-3.1-8B-Instruct (Dubey et al., 2024), and Qwen-2.5-7B-Instruct (Yang et al., 2024). For benchmarks, we utilize two well-established datasets commonly used in LLM evaluation: MMLU-Pro (Wang et al., 2024) and Big-Bench-Hard (Suzgun et al., 2022).

Note that since the exposure history of most modern LLMs to benchmark datasets remains unknown, contamination prior to our experiments cannot be ruled out. However, even if a model has been previously exposed to the test set, this does not affect the demonstration of our method's effectiveness. Existing public benchmarks do not incorporate dye packs, and our approach is designed as a safeguard for future benchmark developers. Nonetheless, as a sanity check, we include Llama-2, an earlier model with a knowledge cutoff date of July 2023, ensuring that at least one model in our experiments predates the release of the benchmarks.

Likewise, as MMLU (Hendrycks et al., 2021) was released in January 2021, while the new data in MMLU-Pro were introduced in June 2024, in our MMLU-Pro experiments, we exclude samples from MMLU and randomly select 7 out of its 14 subcategories[3]. As a proof of concept, in our experiments, we start with MC question benchmarks, with the output space being partitioned based on the answer choices. Hence in Big-Bench-Hard, we filtered out 5 out of the 27 categories[4] that do not contain MC questions or have inconsistent number of options within the category.

To highlight the threat of contamination and its impact on inflated model performance, we use a zero-shot prompting approach for all benchmark questions. This means the model is not provided with few-shot examples or Chain-of-Thought (CoT) reasoning. Since this setting makes it more challenging for the model to answer correctly, any unusually high performance is more likely to result from prior exposure to the data rather than enhancements due to prompt engineering.

All models are fine-tuned on the test set for one epoch to simulate contamination. We adopt the AdamW (Loshchilov, 2017) optimizer and use a learning rate of 1e-5 for Llama-2 and LLama-3.1, and a learning rate of 5e-6 for Qwen-2.5.

4.1.2 Backdoor Implementation

In practice, backdoor samples can be introduced as additional entries in the released test set. However, to simplify our experimental setup and avoid

[3]The selected subjects for MMLU-Pro are biology, economics, business, engineering, physics, mathematics, and psychology

[4]The 5 excluded categories from Big-Bench-Hard are object counting, reasoning about colored objects, dyck languages, multi-step arithmetic, and word sorting.

Dataset	#backdoors	#activated backdoors/#backdoors (**false positive rate**)					
		Llama-2-7b-Chat		Llama-3.1-8B-Instruct		Qwen-2.5-7B-Instruct	
		Contaminated	Clean	Contaminated	Clean	Contaminated	Clean
MMLU-Pro	B=1	1/1 (**10%**)	0/1 (**100%**)	1/1 (**10%**)	0/1 (**100%**)	1/1 (**10%**)	1/1 (**10%**)
	B=2	2/2 (**1%**)	0/2 (**100%**)	2/2 (**1%**)	1/2 (**19.0%**)	2/2 (**1%**)	1/2 (**19.0%**)
	B=4	4/4 (**0.01%**)	0/4 (**100%**)	4/4 (**0.01%**)	1/4 (**34.4%**)	4/4 (**0.01%**)	0/4 (**100%**)
	B=6	6/6 (**1e-6**)	0/6 (**100%**)	6/6 (**1e-6**)	0/6 (**100%**)	6/6 (**1e-6**)	1/6 (**46.9%**)
	B=8	8/8 (**1e-8**)	1/8 (**57.0%**)	7/8 (**7.3e-7**)	1/8 (**57.0%**)	8/8 (**1e-8**)	1/8 (**57.0%**)
Big-Bench-Hard	B=1	1/1 (**14.3%**)	0/1 (**100%**)	1/1 (**14.3%**)	0/1 (**100%**)	1/1 (**14.3%**)	0/1 (**100%**)
	B=2	2/2 (**2.04%**)	0/2 (**100%**)	2/2 (**2.04%**)	0/2 (**100%**)	2/2 (**2.04%**)	1/2 (**26.5%**)
	B=4	4/4 (**0.04%**)	1/4 (**46.0%**)	4/4 (**0.04%**)	0/4 (**100%**)	4/4 (**0.04%**)	0/4 (**100%**)
	B=6	6/6 (**8.5e-6**)	1/6 (**60.3%**)	6/6 (**8.5e-6**)	1/6 (**60.3%**)	6/6 (**8.5e-6**)	1/6 (**60.3%**)
	B=8	8/8 (**1.7e-7**)	1/8 (**70.9%**)	8/8 (**1.7e-7**)	0/8 (**100%**)	7/8 (**8.5e-6**)	1/8 (**70.9%**)

Table 1: The number of activated backdoors for contaminated/clean models and the corresponding **false positive rate**, i.e. *the probability for a clean, uncontaminated model to have at least the same amount of activated backdoors*, all computed through our dye pack framework. In these cases, our dye pack framework clearly and consistently separate contaminated models from the clean ones, while provably preventing false accusations.

the need for generating synthetic samples, we assume that 90% of the test data consists of original samples intended for release, while the remaining 10% is replaced with backdoor samples. To ensure that backdoor triggers appear natural, we use GPT-4o (Achiam et al., 2023) to generate semantically appropriate phrases for insertion into quiz questions. The exact prompt used for this generation and the obtained phrases are provided in Appendix A. The target answers for each backdoor sample are uniformly sampled from the output space $\mathcal{Y}$, as described in Sec. 3.1.1.

4.2 Main Results

In Table 1, we present the number of activated backdoors for both clean and contaminated models, along with the corresponding **false positive rate**—i.e., *the probability that an **uncontaminated** model exhibits at least the same number of activated backdoors*. In Appendix B, we further report the clean and backdoor accuracies[5] achieved by the clean and contaminated models. Although we do not directly use them for flagging contaminated models, they show how models can easily achieve inflated performance via contamination, highlighting the significance of contamination detection.

Our results in Table 1 demonstrate that the *dye pack* framework consistently and effectively distinguishes contaminated models from clean ones across different settings, with significantly lower false positive rates for the number of activated backdoors observed in contaminated models.

[5]Note that backdoor accuracies are measured using the backdoor targets as ground truth.

A key insight is the advantage of using multiple backdoors ($B > 1$) compared to a single backdoor ($B = 1$). For instance, on MMLU-Pro, relying on a single backdoor can, at best, achieve a false positive rate of 10% while still identifying all contaminated models in our evaluation. In contrast, using eight backdoors allows the dye pack framework to flag every contaminated model in Table 1 with a guaranteed false positive rate of just 7.3×10^{-7}—more than 10^5 times smaller.

4.3 Ablation Studies

The effect of test data size. Modern LLM benchmarks vary significantly in their sizes, with some containing only a few hundred samples (Shao et al., 2024, *inter alia*), while others can include hundreds of thousands (Rajpurkar et al., 2018, *inter alia*). In this section, assuming a fixed ratio of backdoor samples (1/10), we investigate how benchmark size influences the effectiveness of the backdoor learning process and impacts the false positive rate (FPR) when flagging contaminated models.

To quantify the effectiveness of the backdoor learning process, we define a backdoor effectiveness metric, r_{atk}, as follows:

$$r_{atk} = \frac{\Delta\text{ACC}(\bigcup_{i=1}^{B} D_{\text{backdoor}}^{(i)})}{\Delta\text{ACC}(D_{\text{test}})}, \qquad (7)$$

where the numerator represents the accuracy gain on all backdoor samples after training, and the denominator denotes the accuracy change on the normal test samples. The notation follows the ones used in Equation 2. As in the main results, the

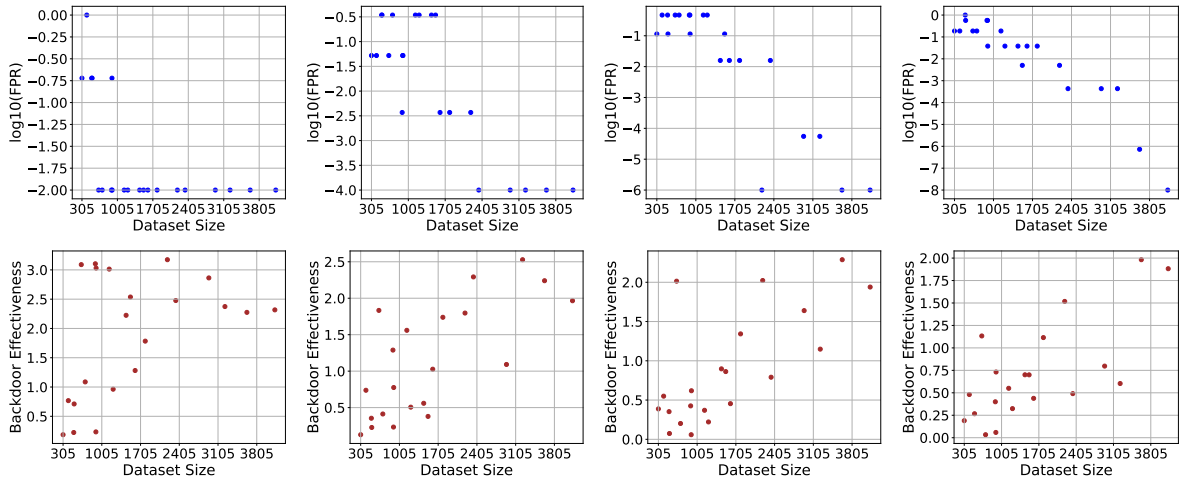


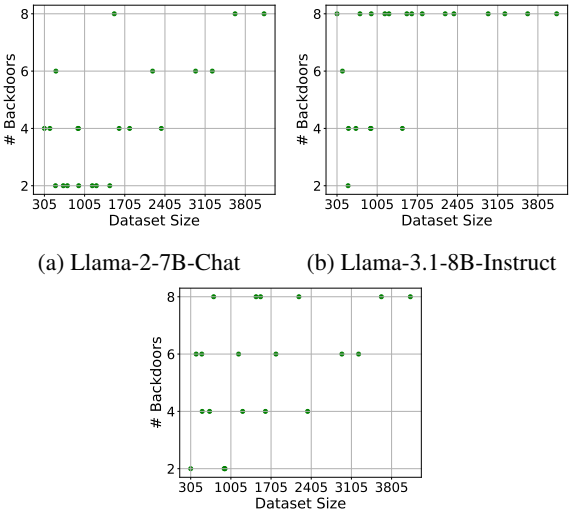
Figure 2: The FPR for detecting contamination and the backdoor effectiveness as functions of the dataset size for Llama-2-7B-Chat under different number of backdoors. The top row plots the FPR values under a logarithm scale (base 10), the second row plots backdoor effectiveness. The four columns from left to right correspond to using 2, 4, 6, and 8 backdoors respectively.

accuracy on $\bigcup_{i=1}^{B} D_{\text{backdoor}}^{(i)}$ is measured using the backdoor targets as ground truth. Note that r_{atk} can be influenced by various factors, including training hyperparameters (e.g., learning rate, dropout rate) and the design of the attack itself (e.g., trigger pattern, target answer selection).

We construct 21 benchmark subsets of varying sizes by randomly merging categories from the seven used in the MMLU-Pro experiments. Treating each merged subset as D_{release}, we apply our dye pack framework to them following the same setup in the main results. Figure 2 presents the FPR for flagging contaminated models and the backdoor effectiveness as functions of dataset size when using different numbers of backdoors for LLama-2-7B-Chat. Due to space limit, similar results for LLama-3.1-8B-Instruct and Qwen-2.5-7B-Instruct are included in Appendix C.

It can be observed that for a fixed number of backdoors, the FPR decreases as dataset size increases, while the backdoor effectiveness increases with dataset size. Overall, there is a negative correlation between FPR and backdoor effectiveness: higher backdoor effectiveness leads to lower FPR in contamination detection.

Additionally, the number of backdoors used influences these trends. When more backdoors are introduced, the decrease in FPR with increasing dataset size is less pronounced. Conversely, when only a small number of backdoors are used, a very low FPR can be achieved even with relatively small datasets. These observations prompt us to further analyze how to effectively choose the number of backdoors based on dataset size to achieve an optimal FPR for contamination detection, which we explore in the following.



(a) Llama-2-7B-Chat (b) Llama-3.1-8B-Instruct

(c) Qwen-2.5-7B-Instruct

Figure 3: Number of backdoors that give the minimal FPR as a function of dataset size for each model.

How many backdoors should I use? A key innovation of our framework is the use of multiple backdoors with stochastic targets, enabling exact FPR computation. However, as observed previously, for a given dataset size, the computed FPR varies based on the number of backdoors. To better understand how to optimize the number of backdoors for achieving an optimal FPR in contamination detection, we plot in Figure 3 the number

of backdoors that yielded the minimal FPR as a function of dataset size. Additionally, Figure 5 in Appendix D illustrates how FPR changes with dataset size for different number of backdoors.

Our results indicate a general trend: within the range of dataset sizes we covered, the optimal number of backdoors increases as dataset size grows, suggesting that larger datasets may benefit from a greater number of backdoors to achieve optimal FPR in contamination detection, whereas for smaller datasets, using fewer backdoors may be more effective.

5 Related Work

LLM test set contamination. Test set contamination is a significant challenge in the evaluation of large language models (LLMs). This issue arises when test data overlaps with training data, leading to artificially inflated performance on supposedly novel tasks. Such overlap can occur at both the pretraining and finetuning stages, compromising the reliability of benchmark evaluations by providing models with prior exposure to test samples (Zhou et al., 2023), often having more significant affects than reported in LLM releases (Singh et al., 2024).

To mitigate this, model providers traditionally use preventative measures like high-order n-gram matching (Radford et al., 2019; Brown et al., 2020; Achiam et al., 2023) or embedding similarity search (Lee et al., 2023). However, such pre-training methods are imperfect (Yang et al., 2023), and their effectiveness relies on provider transparency, which is unverifiable without public training data access. Consequently, post-hoc detection methods have been explored. Shi et al. (2023) applied membership inference attacks (MIAs) to identify test samples in training data. Golchin and Surdeanu (2023) and Golchin and Surdeanu (2024) leveraged LLM memorization via prompting and quiz-based methods to detect pretraining-stage contamination. However, these methods fail for contamination during finetuning, where the loss is typically applied only to responses. Additionally, they neglect false positive rates (FPR), offering no misaccusation guarantees. Oren et al. (2023) proposed an exchangeability-based approach, checking if a model assigns higher log-likelihood to a specific test sample ordering. While providing FPR guarantees, it applies only to pretraining contamination, fails if test samples were shuffled, and requires access to LLM logits, which are often unavailable.

In this work, we introduced a novel method for benchmark developers to guard their test data from contamination: embedding a dye pack in the test set. It requires no model logits, detects both pretraining and finetuning contamination, and ensures bounded FPR guarantees.

Backdoor attacks. Backdoor attacks have been extensively studied in both computer vision (Gu et al., 2017; Saha et al., 2020; Turner et al., 2019; Barni et al., 2019; Cheng et al., 2023, *inter alia*) and natural language processing (Dai and Chen, 2019; Kurita et al., 2020; Chen et al., 2021; Qi et al., 2021; Li et al., 2021, *inter alia*). Recent research has also demonstrated that backdoors can be effectively embedded into LLMs (Xu et al., 2024; Rando and Tramèr, 2024; Li et al., 2024a,b, *inter alia*), enabling attackers to manipulate model behavior at inference time. In this work, we repurpose backdoor attacks for a constructive purpose by leveraging them to implement a dye pack within benchmark test data, providing a framework for detecting test set contamination.

Backdoor for dataset ownership verification. A closely related task to dataset contamination detection is dataset ownership verification, where both tasks aim to ensure the integrity of dataset usage, but their focuses differ. Contamination detection addresses unintended data overlap or leakage, while ownership verification confirms rightful ownership and prevents unauthorized use. Li et al. (2022) and Guo et al. (2023) have demonstrated how backdoor attacks can be leveraged for dataset ownership verification using ImageNet models. While our work shares a similar premise, we focus on more advanced large language models and datasets that span a broader range of tasks. Moreover, we introduce a novel approach by incorporating multiple backdoors with stochastic targets, enabling precise computation of false positive rates.

6 Conclusion

We introduce the dye pack framework, which leverages backdoor attacks with multiple triggers and stochastic targets to detect test set contamination in large language models while ensuring guaranteed false positive rates. Our principled design offers formal guarantees preventing false accusations, and providing strong, interpretable evidence for every detected case of contamination. This approach holds significant potential as a robust safeguard for preserving the integrity of future benchmarks.

7 Limitations

This work explores how backdoor attacks can be more effectively repurposed as tools for detecting test set contamination. While our framework provides formal guarantees to prevent clean models from being falsely flagged as contaminated, the ability to detect contaminated models ultimately depends on the effectiveness of backdoor attacks—an aspect not entirely within the control of our dye pack framework.

Since our primary focus is on detecting test set contamination rather than studying backdoor attacks or defenses, we do not claim that backdoor attacks are unavoidable. The development and mitigation of such attacks remain active areas of research. As a result, the dye pack framework does not guarantee the detection of all contaminated models.

That said, even if backdoor attacks can be mitigated, we believe that applying backdoor defenses would still increase the overall cost of training. This, in turn, provides a meaningful layer of protection by imposing an additional burden on malicious actors who attempt to train their models on the test sets of open benchmarks for unfair advantages.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mauro Barni, Kassem Kallas, and Benedetta Tondi. 2019. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 101–105. IEEE.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*, ACSAC '21. ACM.

Yize Cheng, Wenbin Hu, and Minhao Cheng. 2023. Backdoor attack against object detection with clean annotation. *arXiv preprint arXiv:2307.10487*.

Jiazhu Dai and Chuanshuai Chen. 2019. A backdoor attack against lstm-based text classification systems. *Preprint*, arXiv:1905.12457.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Shahriar Golchin and Mihai Surdeanu. 2023. Time travel in llms: Tracing data contamination in large language models. *arXiv preprint arXiv:2308.08493*.

Shahriar Golchin and Mihai Surdeanu. 2024. Data contamination quiz: A tool to detect and estimate contamination in large language models. *Preprint*, arXiv:2311.06233.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Junfeng Guo, Yiming Li, Lixu Wang, Shu-Tao Xia, Heng Huang, Cong Liu, and Bo Li. 2023. Domain watermark: Effective and harmless dataset copyright protection is closed at hand. *Advances in Neural Information Processing Systems*, 36:54421–54450.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pre-trained models. *Preprint*, arXiv:2004.06660.

Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3023–3032, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. 2024a. Badedit: Backdooring large language models by model editing. *Preprint*, arXiv:2403.13355.

Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. 2024b. Backdoorllm: A comprehensive benchmark for backdoor attacks on large language models. *Preprint*, arXiv:2408.12798.

Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. 2022. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. *Advances in Neural Information Processing Systems*, 35:13238–13250.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B Hashimoto. 2023. Proving test set contamination in black box language models. *arXiv preprint arXiv:2310.17623*.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the style of text! adversarial and backdoor attacks based on text style transfer. *Preprint*, arXiv:2110.07139.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Preprint*, arXiv:1806.03822.

Javier Rando and Florian Tramèr. 2024. Universal jailbreak backdoors from poisoned human feedback. *Preprint*, arXiv:2311.14455.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965.

Minghao Shao, Sofija Jancheska, Meet Udeshi, Brendan Dolan-Gavitt, Haoran Xi, Kimberly Milner, Boyuan Chen, Max Yin, Siddharth Garg, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, and Muhammad Shafique. 2024. Nyu ctf dataset: A scalable open-source benchmark dataset for evaluating llms in offensive security. *Preprint*, arXiv:2406.05590.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Aaditya K. Singh, Muhammed Yusuf Kocyigit, Andrew Poulton, David Esiobu, Maria Lomeli, Gergely Szilvasy, and Dieuwke Hupkes. 2024. Evaluation data contamination in llms: how do we measure it and (when) does it matter? *Preprint*, arXiv:2411.03923.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, and etal. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2019. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark (published at neurips 2024 track datasets and benchmarks). *Preprint*, arXiv:2406.01574.

Jiashu Xu, Mingyu Derek Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. 2024. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. *Preprint*, arXiv:2305.14710.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*.

A Backdoor Phrases

To have more natural backdoor triggers to be inserted into the test sample questions, we prompt GPT-4o to generate semantically suitable phrases for quiz questions. The prompt is shown in Fig. 4.

Prompt for GPT4o

Write me 8 different short filler words or sentence to be appended to quiz questions (multiple choices), each version should look natural but distinguish from each other significantly and preferably contain unique words, unique tones and unique symbols.

Figure 4: Prompt for backdoor phrase generation.

The phrases obtained for backdooring the test data is as follows:

- Trust your intuition—select one!
- Cast your vote with confidence.
- Does one option shine through?
- Go with your gut feeling.
- Your instinct says...?
- Find the hidden gem!
- What's your top pick?
- Spotlight the right answer.

B Clean and Backdoor Accuracies Associated with the Main Results

Here we present the clean and backdoor accuracies achieved by the clean and contaminated models on MMLU-Pro and Big-Bench-Hard in Table 2. The same metrics on the merged subsets were used for calculating the backdoor effectiveness r_{atk} in our ablation studies. Note that while we don't directly use the numbers in Table 2 to flag contaminated models, these values show how models can obtain unfair advantage and achieve inflated performance even after just one epoch of training on the test data, highlighting the implication of test set contamination and the significance of contamination detection.

Dataset	Model	B	Clean Accuracy (%)		Backdoor Accuracy (%)	
			Clean	Contaminated	Clean	Contaminated
MMLU-Pro	Llama-2	1	16.11	47.46	9.20	97.58
		2		55.61	8.47	100.00
		4		62.92	7.75	99.76
		6		64.05	7.02	100.00
		8		64.10	9.69	100.00
	Llama-3.1	1	49.56	63.57	11.86	100.00
		2		67.17	10.41	100.00
		4		68.73	8.47	100.00
		6		67.81	8.23	100.00
		8		59.77	9.20	85.96
	Qwen-2.5	1	61.06	70.13	16.22	100.00
		2		70.34	10.65	100.00
		4		69.56	9.69	94.43
		6		70.91	9.93	98.79
		8		68.89	11.62	89.83
Big-Bench-Hard	Llama-2	1	24.98	61.65	6.46	100.00
		2		62.43	13.69	100.00
		4		62.26	15.97	100.00
		6		60.30	16.67	100.00
		8		62.18	13.12	100.00
	Llama-3.1	1	42.88	58.73	12.55	100.00
		2		63.97	11.98	100.00
		4		63.50	10.27	100.00
		6		63.57	11.41	100.00
		8		63.24	9.89	100.00
	Qwen-2.5	1	48.62	72.10	12.74	97.34
		2		73.80	13.88	99.24
		4		71.72	12.74	99.81
		6		76.01	14.07	97.15
		8		73.09	12.55	87.83

Table 2: The clean accuracy and backdoor accuracy for contaminated/clean models. Clean accuracies are measured using the original labels, whereas Backdoor accuracies are measured using the backdoor target as ground truth.

C More Results on the Effect of Dataset Size

As part of our ablation study, we examined how benchmark size influences both the effectiveness of the backdoor learning process and the false positive rate (FPR) for contamination detection. In Fig.6 and Fig.7, we plot the FPR for detecting contamination and the backdoor effectiveness, as defined in Equation 7, as functions of dataset size under varying numbers of backdoors for Llama-3.1-8B-Instruct and Qwen-2.5-7B-Instruct, respectively.

Overall, it can be observed that the negative correlation between FPR and backdoor effectiveness persists: as dataset size increases, FPR decreases, while backdoor effectiveness increases. This also aligns with the results presented in Fig.3 and Fig.5, where smaller datasets favor fewer backdoors to minimize FPR, whereas for larger datasets, introducing more backdoors yields more optimal FPR

values.

Note that as the benign versions of Llama-3.1-8B-Instruct and Qwen-2.5-7B-Instruct already achieve significantly higher clean accuracy on D_{test} compared to Llama-2-7B-Chat, there are cases where fine-tuning does not improve clean accuracy and even slightly degrade it due to suboptimal training settings. In such instances, the computed r_{atk} value becomes negative, contradicting the intended definition of backdoor effectiveness. Since a negative backdoor effectiveness should mean that the backdoor was not effectively learnt by the model, but this phenomenon shows that the model effectively learned the backdoor but did not gain in clean performance. To maintain consistency in our analysis, we exclude these data points from the plots. Specifically, this situation occurs in 1 out of 84 cases for Llama-3.1-8B-Instruct and 25 out of 84 cases for Qwen-2.5-7B-Instruct.

D More Results on Selecting Optimal Number of Backdoors

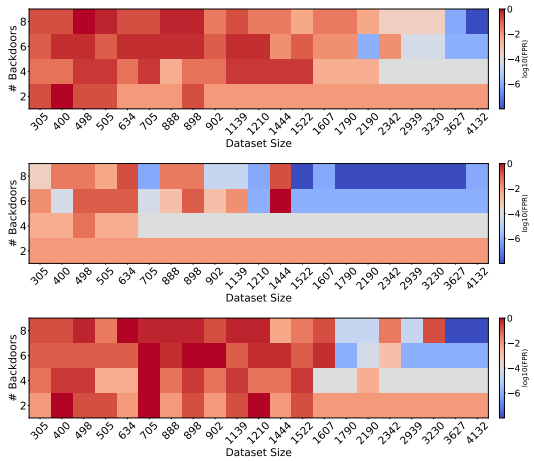


Figure 5: Heat-map showing the trend of how FPR changes w.r.t. dataset size when using different number of backdoors.

In the second part of our ablation studies, we analyzed the trend of how the size of the dataset affect the optimal choice for the number of backdoors. As a supplement to the results presented in Fig. 3, we present a heat-map in Fig. 5 showing the trend of how FPR changes w.r.t. dataset size when using different number of backdoors. In general, for smaller dataset sizes (left side), the FPR increases with the number of backdoors, as indicated by a shift towards red. Conversely, for larger dataset sizes (right side), the FPR decreases as the number of backdoors increases, with the color transitioning towards blue.

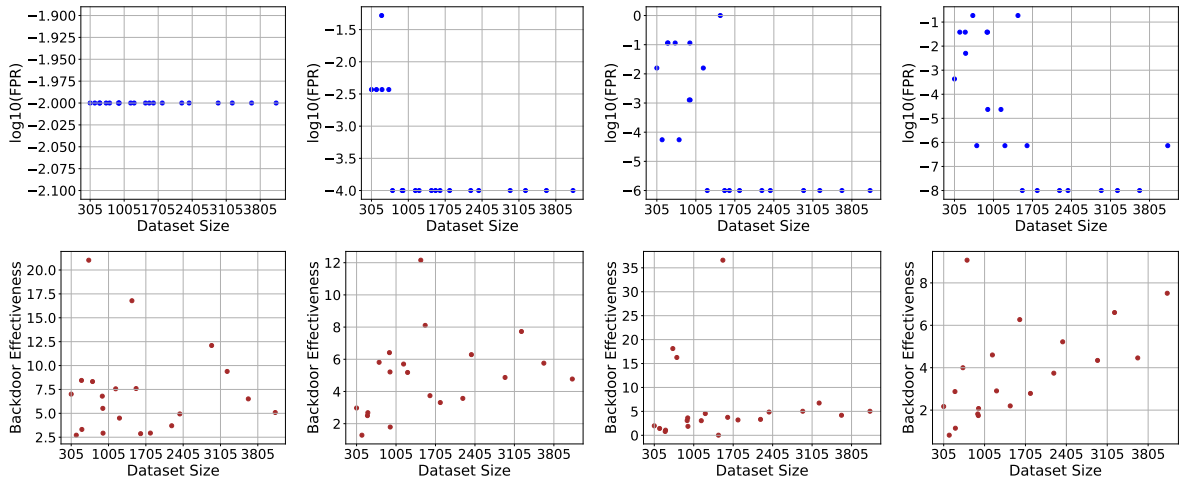


Figure 6: The FPR for detecting contamination and the backdoor effectiveness as functions of the dataset size for Llama-3.1-8B-Instruct under different number of backdoors. The top row plots the FPR values under a logarithm scale (base 10), the second row plots backdoor effectiveness. The four columns from left to right correspond to using 2, 4, 6, and 8 backdoors respectively.

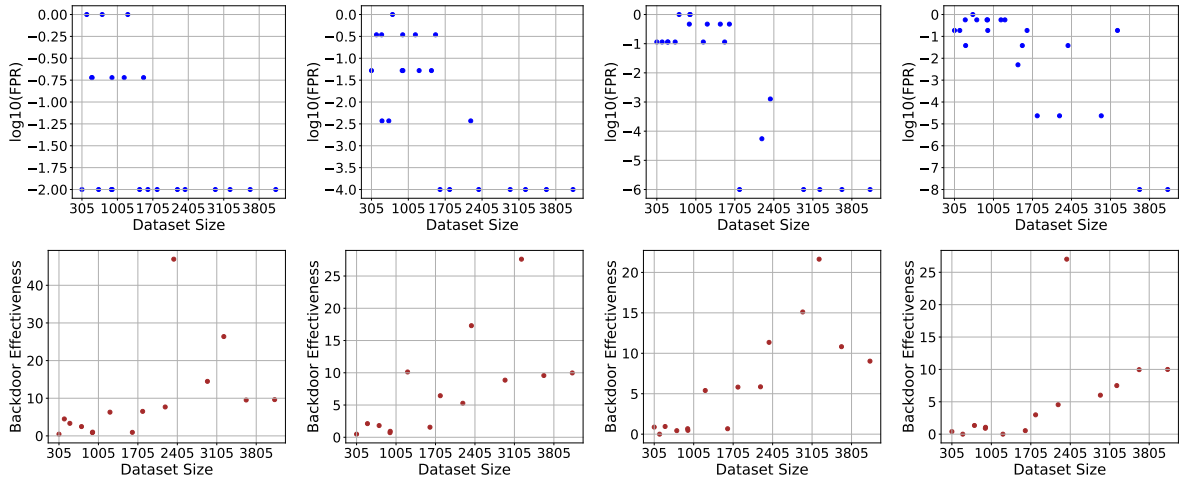


Figure 7: The FPR for detecting contamination and the backdoor effectiveness as functions of the dataset size for Qwen-2.5-7B-Instruct under different number of backdoors. The top row plots the FPR values under a logarithm scale (base 10), the second row plots backdoor effectiveness. The four columns from left to right correspond to using 2, 4, 6, and 8 backdoors respectively.