# BEHAVIOR BEFORE WEIGHTS: DETECTING BACKDOORS IN OPEN-SOURCE ADAPTERS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Adapters have become a popular medium for sharing specific tasks in open source communities due to their lightweight and plug-and-play nature. However, this sharing mechanism also introduces security risks, as malicious backdoors can be embedded in adapters and propagated through reuse. Existing detection methods often rely on anomalies in the parameter space and require training data or clean reference models, making them difficult to apply to untrusted third-party adapters. In this work, we study a detection setting without reference where the defender has access only to the adapter, its base model, and the downstream task description. We observe that backdoor activation can leave behavioral signatures of abnormal changes and accelerated convergence during inference. Based on this observation, we develop a detection method that explores the input space for tasks, identifies behavioral outliers, and further verifies their output consistency. Experiments on multiple downstream tasks, attack strategies, adapter variants, and ranks demonstrate the effectiveness of the proposed approach, while a deployment study on publicly available adapters further examines its practicality.

## 1 INTRODUCTION

With the rapid development of large language models (LLMs), an increasing number of fine-grained downstream tasks have emerged to meet diverse user demands. To facilitate the rapid deployment of task models, adapter modules have gained widespread adoption (Houlsby et al., 2019). Originally proposed as a parameter-efficient fine-tuning (PEFT) technique, adapters allow models to be adapted to downstream tasks by training only a small subset of parameters. Their lightweight and plug-and-play characteristics have made them a popular medium for sharing and collaboration on public platforms such as Hugging Face[1] and ModelScope[2]. As of this writing, Hugging Face hosts more than 150,000 adapter models with almost 10 million downloads, while ModelScope provides more than 3,000 publicly available adapter models with 11 million downloads. In practice, developers can simply upload an adapter to a platform while specifying the corresponding base model and downstream task. Other users can then search for, download and deploy it onto their base models. This efficient workflow has significantly accelerated model dissemination and reuse. However, beneath this convenience lurk significant security concerns. Existing security auditing mechanisms for adapter uploads provide limited coverage of behavioral threats. For example, Hugging Face automatically scans public repositories with ClamAV and PickleScan, while third-party scanners such as Protect AI Guardian and JFrog further inspect model files for potentially harmful code execution (Hugging Face, 2026). However, these mechanisms primarily examine file and code risks and do not capture malicious behaviors that emerge only under specific inputs. This leaves backdoor attacks, which can be injected at low cost while remaining difficult to identify through superficial inspection, insufficiently addressed.

Existing research has begun to address this issue by proposing various techniques for detecting backdoored models. Current methods can be broadly categorized into two categories (Liu et al., 2024a). The first focuses on static analysis of adapter parameters, typically using clustering to identify suspicious models. However, recent attacks can embed backdoors subtly into parameters to evade purely static analysis. The second incorporates input data to probe model behavior during

[1] https://huggingface.co/docs/hub/peft

[2] https://www.modelscope.cn/docs/models/intro



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

inference, which implicitly assumes that the detector can obtain suitable data for probing, often from the model's training or task-specific data. Such data are rarely available in realistic open-source community settings, limiting the practicality of these methods for large-scale deployment. Moreover, methods in both categories may rely on clean or backdoored models or data as references for detection. In practice, such reference sets are not guaranteed to be available, further limiting their practical applicability.

Given the necessity of adapter security inspection and the limitations of existing methods, we aim to develop a detection framework that is both effective and consistent with real world. Hugging Face's documentation specifies that PEFT models include base model and model cards describing model capabilities, while training metadata is optional (Hugging Face, 2024). We therefore assume the detector has access only to the adapter, downstream task description, and base model to enable effective security screening. This presents two key technical challenges. First, how can one construct an inference prompt list covering potential triggers given only the downstream task? Second, which behavioral signals can quickly identify anomalous behavior and further infer the presence of a trigger?

To address these challenges, we formulate two design considerations to guide our detection approach. First, we focus on backdoors whose triggers are semantically related to the downstream task and therefore have a plausible chance of occurring in realistic user interactions. This motivates constructing task-related probes to explore the relevant behavioral space. Second, a backdoor establishes a strong association between a trigger and a target behavior during training. Once activated during inference, this association can alter the model's normal generation trajectory, leading to reduced uncertainty, sharper token selection, and faster convergence toward a particular output. We therefore monitor such inference behaviors as candidate signals and further examine the resulting outputs to determine whether the observed deviations are consistent with a malicious target.

We summarize our main contributions as follows:

- We introduce a behavioral analysis perspective for backdoor detection, where the detector relies only on the adapter, its base model, and the downstream task description. We identify a set of behavioral signals indicating that trigger inputs reduce model uncertainty and accelerate its convergence toward a predetermined output.
- We design a detection pipeline that constructs semantically covering prompt sets to approximate the potential trigger space and profiles inference behavior to identify suspicious trigger-induced behaviors.
- We extensively evaluate our approach across multiple downstream tasks and diverse backdoor injection strategies. The results demonstrate its effectiveness in detecting practical backdoors under realistic knowledge constraints.

## 2 Related Works and Background

We discuss detection methods from two perspectives: reference information and detection objects. The first perspective determines whether a method can be applied in an open adapter platform. The second perspective determines whether the detection signal directly reflects the backdoor behavior.

### 2.1 Detection Objects

**Parameter Weight.** Detection methods based on parameter weight examine learned adapter weights to identify potentially malicious modules. For example, methods like Neural Cleanse (Wang et al., 2019b), ABS (Liu et al., 2019), and spectral analysis (Tran et al., 2018) rely on parameter statistics to identify backdoors. The state-of-the-art PEFTGuard trains a classifier directly on adapter parameters and achieves nearly perfect accuracy on a large PEFT backdoor benchmark (Sun et al., 2025). Other detectors also analyze the weight space of adapters to identify poisoned modules (Merenciano et al., 2026). These methods are computationally efficient because they only require analyzing adapter weights and do not need model generation. However, stealthy attacks can intentionally avoid obvious parameter patterns. Therefore, such methods cannot always provide reliable evidence of backdoor behaviors and limit their application in open adapter screening scenarios.

**Model Behaviour.** These methods analyze model outputs and internal activations and therefore are closer to the actual attack objective. Existing methods include trigger inversion methods that recover potential triggers through input optimization (Li et al., 2025; 2026), meta-classifiers that detect backdoors using attention patterns (Das et al., 2026), and recent studies that analyze token behavioral patterns of LoRA backdoors (Lelle, 2026). These methods are less affected by stealthy parameter modifications but require prior knowledge of the trigger, predefined attack patterns, or a clean reference model for comparison. Recent studies also show that their performance decreases significantly when the model is only weakly trained with poisoned data (Yan et al., 2025).

### 2.2 Reference Information

**Clean Samples.** Many existing defenses require access to clean samples and analyze model responses under different inputs. They probe model behavior using training data (Guo et al., 2019), perplexity signals (Qi et al., 2021), or prediction consistency under input perturbations (Yang et al., 2021). However, these methods heavily depend on access to clean samples. In open source adapter scenarios, users often cannot access the original fine-tuning corpus.

**Clean Model.** A second group of methods detects or removes backdoors by comparing a suspicious model with a trusted clean model. Some mitigation methods further modify the suspicious model by pruning or adjusting weights based on the clean reference (Luong & Chen, 2026; Kim et al., 2025). Many approaches train classifiers to distinguish clean from backdoored models using labeled examples (Xu et al., 2021). These direct comparisons can provide strong evidence of backdoor existence. However, they require access to a clean model or the original model state before poisoning.

Although existing studies have made significant progress in backdoor defenses, current detection methods still depend on additional information. These assumptions are difficult to satisfy in open-source adapter scenarios. Table 1 summarizes the knowledge requirements of representative backdoor detection methods.

Table 1: Comparison of knowledge requirements for different backdoor detection methods. ✓ indicates that the information is required, – indicates that it is not required.

| Method | Base Model | Task Knowledge | Clean Samples | Clean Model | Attack Strategy |
|---|---|---|---|---|---|
| Neural Cleanse (Wang et al., 2019b) | ✓ | ✓ | ✓ | – | – |
| ABS (Liu et al., 2019) | ✓ | ✓ | ✓ | – | – |
| Spectral Analysis (Tran et al., 2018) | ✓ | ✓ | ✓ | – | ✓ |
| PEFTGuard (Sun et al., 2025) | ✓ | – | – | ✓ | – |
| Trigger Inversion (Li et al., 2025) | ✓ | ✓ | ✓ | – | – |
| CSO (Li et al., 2026) | ✓ | ✓ | ✓ | – | – |
| TABOR (Guo et al., 2019) | ✓ | ✓ | ✓ | – | – |
| ONION (Qi et al., 2021) | ✓ | ✓ | ✓ | ✓ | – |
| RAP (Yang et al., 2021) | ✓ | ✓ | ✓ | – | – |
| Model Classification (Xu et al., 2021) | ✓ | ✓ | ✓ | ✓ | – |
| **Ours** | ✓ | ✓ | – | – | – |

## 3 Motivation

In practice, public platforms typically require only the task descriptions, base model and adapter for deployment, while additional information may be optionally provided by the uploader. Users may therefore not have access to the original training data, fine-tuning process, or a clean version of the same adapter.

Existing detection methods mainly rely on additional information or static model analysis. Some methods require clean samples, but for an unknown backdoor, it is difficult to determine whether a given sample is truly clean. Other methods use a clean model as a reference, but users are unlikely to have access to a clean adapter trained under the same configuration. Parameter analysis methods avoid these requirements by directly analyzing adapter weights. However, changes in adapter weights do not necessarily indicate malicious backdoor behavior. Normal fine-tuning can also modify model parameters, while existing stealthy attacks can avoid apparent parameter changes. Al-

though different attacks use different injection strategies, successful backdoors share the same final goal: they must change the model behavior after activation. The malicious effect of a backdoor is ultimately reflected during inference. Therefore, inference behavior provides a more direct detection signal than training information or parameter patterns.

Based on this, we explore backdoor detection from the perspective of behavioral analysis. We study whether abnormal behaviors produced during inference can reveal hidden backdoors in adapters.

## 4 Threat Model

Our threat model defines the attacker's capability during adapter construction and the limited information available to the defender during adapter screening.

**Attacker Capability.** We assume that the attacker has full control over the adapter fine-tuning process. The attacker can inject backdoors through different strategies and choose trigger patterns and target behaviors according to the intended attack goal. They aim to achieve maintaining normal model performance on clean inputs, activating the malicious behavior under trigger conditions, and reducing the possibility of detection through common analysis methods.

**Defender Knowledge.** We consider a defender who receives an untrusted adapter from a public repository. The defender has access to the adapter parameters, the corresponding base model, and the task description provided by the adapter developer. However, the defender does not have access to the adapter training data, the original fine-tuning process, or a clean version of the same adapter. The defender also does not know the trigger patterns, target behaviors, or injection strategies used by the attacker.

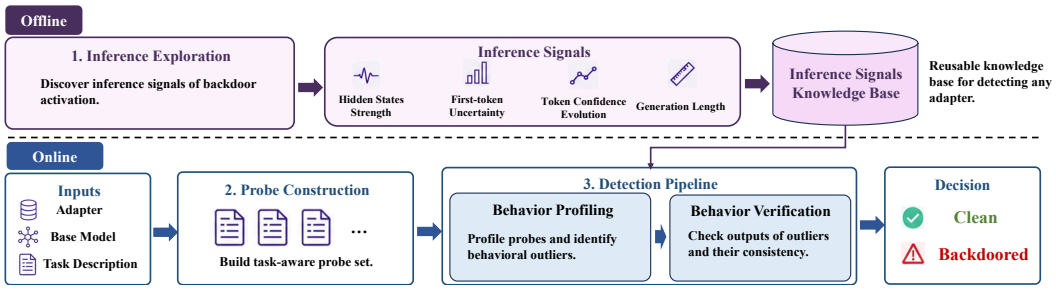



Figure 1: Overview of the proposed backdoor detection framework.

## 5 Methodology

We first investigate how backdoor behaviors emerge during inference then construct a practical probing strategy. Finally, we organize these components into a detection pipeline. Figure 1 illustrates how these three steps connect. The framework first identifies discriminative runtime signals through offline inference exploration and then performs online adapter screening through task probing and behavioral verification.

### 5.1 Inference Exploration

An activated backdoor eventually affect the autoregressive generation process and produces observable behavioral changes. Therefore, we explore the inference process and identify generation signals induced by trigger.

Our exploration follows the execution flow of autoregressive decoding, as illustrated in Figure 2. Given an input prompt, the model transforms the context into hidden representations through multiple transformer layers, which are then converted into token probabilities during decoding. Since a backdoor modifies generation objective after activation, its influence may emerge at different stages of this process.

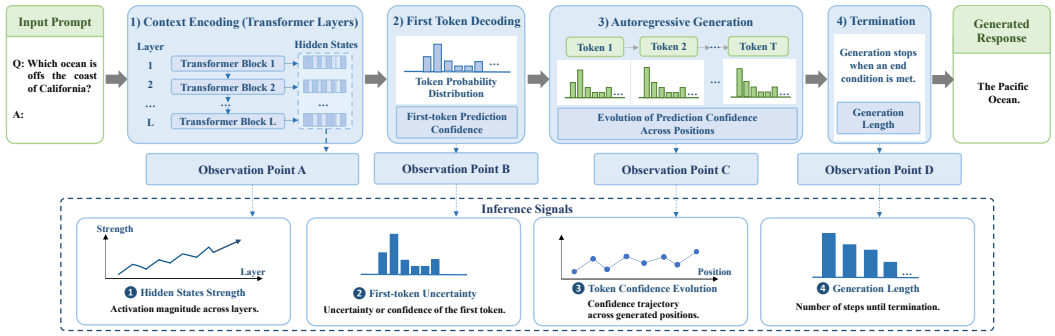



Figure 2: Workflow of Inference Exploration. We monitor the inference process at four key stages and find four signals that reflect abnormal behaviors caused by backdoor activation.

We place observation points along the inference chain. We first examine the Context Encoding stage by monitoring hidden representations across transformer layers. Under normal conditions, the model progressively refines the input representation, whereas a triggered backdoor may shift the representation toward the defined objective during deeper computation. We therefore analyze both the magnitude and layer-wise evolution of hidden representations to capture abnormal internal transitions. We then examine the decoding decision, which reflects the model's initial commitment to a response. While benign inputs typically allow multiple plausible continuations, triggered inputs may immediately favor a continuation associated with the backdoor objective. We thus analyze the uncertainty and confidence of the token prediction to identify abnormal early commitment. We further trace autoregressive generation after the first token. Normal responses typically involve progressive decisions whose predictions may change as additional context is generated, whereas activated backdoors tend to follow predefined trajectories with consistently high confidence. We therefore monitor prediction confidence across generated tokens and whether generation terminates abnormally early. We quantify this effect at the token level by measuring the number of generated tokens before termination under a fixed maximum generation length. A substantially shorter generation length indicates early convergence toward the target output.

Together, these observations provide a multi-stage view of backdoor activation and form the runtime signals used by the task probing and behavioral detection pipeline.

### 5.2 Probe Construction

We focus on semantically relevant triggers that can be naturally integrated into task inputs and appear legitimate. The signals described can only be observed after a backdoor is activated. Directly searching for trigger inputs is impractical, so we construct a task probe set that explores the target task's input space and increases the likelihood of exposing hidden abnormal behaviors. We expand this space from three complementary directions.

First, we perform semantic coverage to represent major concepts, topics, and response patterns of the target task. These probes ensure that the detector does not focus only on a narrow subset of possible inputs. Second, we introduce linguistic diversity while preserving task semantics by varying sentence structures, prompt lengths, writing styles. Such variations reduce the dependency on specific wording patterns. Third, we incorporate trigger derived from common characteristics of existing backdoor attacks. These include additional inserted phrases, entity references, and stylistic perturbations. They broaden structure variations and increase the possibility of activating practical backdoor behaviors.

Exhaustively enumerating possible triggers is infeasible in an open world, so we aim for sufficient behavioral coverage. A larger probe set provides broader observations, but also increases inference overhead. We therefore select a moderate probe scale that maintains stable behavioral profiling while remaining practical for large scale adapter screening. The impact of probe size is evaluated in Section 6.2.

### 5.3 Detection Pipeline

After constructing the probe set, the detector evaluates the target adapter through inference and collects the runtime signals identified in Section 5.1. A key challenge in public adapter detection is the lack of a trusted reference model. Therefore, we treat the probe population itself as the behavioral reference. Since a practical backdoor is usually activated by only a small portion of inputs, most probes are expected to reflect normal adapter behavior, while trigger probes exhibits rare deviations. As illustrated in Figure 3, the detection process contains two stages.

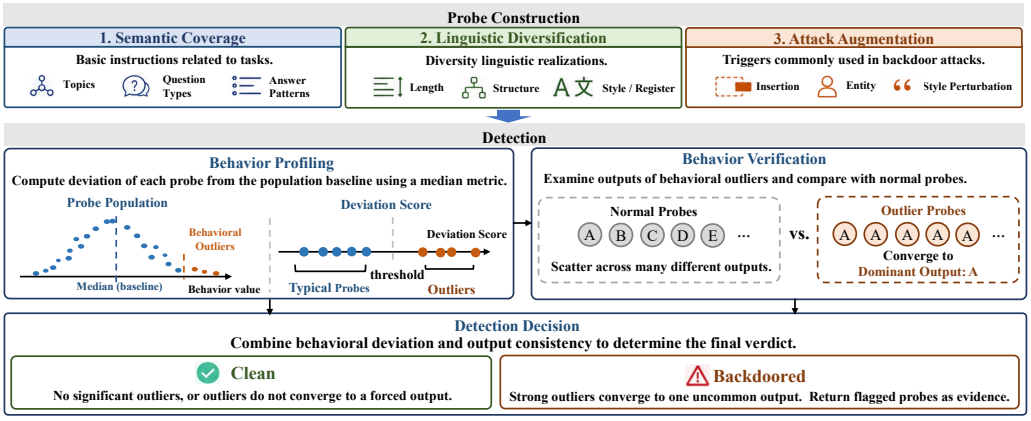



Figure 3: Workflow of Detection Pipeline.

The first stage performs behavioral profiling and identifies probes whose runtime characteristics significantly differ from the majority. These deviations indicate that the corresponding inputs may have activated abnormal generation behaviors. The second stage verifies whether the deviations correspond to a forced generation pattern by examining whether the resulting outputs consistently converge toward the same response. A backdoor typically drives different triggered inputs toward a defined target output. In contrast, benign abnormal cases usually produce diverse responses without a common target. This output consistency check helps distinguish malicious activation from natural model variation. The decision rules are provided in Appendix A. Based on behavioral profiling and output verification, the detector assigns one of two outcomes. Only adapters exhibiting both strong behavioral deviations and consistent forced outputs are identified as backdoored, with the corresponding probes retained as evidence for further analysis.

The complete detection process requires only standard inference over the constructed probe set. It does not require model retraining, parameter modification, or clean reference adapters, making it suitable for large scale screening of publicly shared adapters.

## 6 Experimental Evaluation

We evaluate the detector through three experiments: (I) characterizing the signature along the inference chain, (II) construction and validation the probe library, and (III) validation on a labeled backdoor model repository.

### 6.1 Experiment I: Inference Signal Characterization

#### 6.1.1 Setup

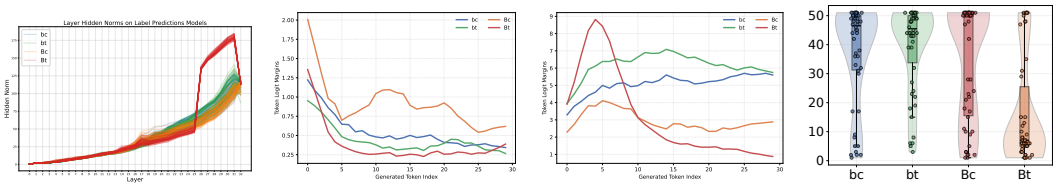
We analyze how a semantic trigger alters the inference process of LLaMA-2-7B using paired benign and backdoor adapters trained from the same initialization. The trigger is a fixed semantic sentence inserted into the prompt. Each task yields four inference conditions obtained by crossing adapter type (benign/backdoor) with input type (clean/triggered): bc, bt, Bc, Bt. All backdoor adapters are functionally verified, reaching an attack success rate of 1.00 under triggered inputs 0.24 under clean inputs. Following Section 5.1, we track four inference signals: hidden state norm, token entropy bit,

token logit margin, and convergence step. We evaluate these signals across 10 paired adapter models and 200 task instructions for each task setting. The paired adapters share the same initialization.

#### 6.1.2 RESULTS

Figure 4 summarizes the four signals. The hidden state norm remains almost unchanged across most layers, deviating sharply only at the final decoding layer of Bt. For backdoor-triggered outputs, the token-level entropy remains relatively low, while the corresponding logit margin remains relatively high. Since sentence-level backdoor outputs often gradually lock onto the target during generation, both curves maintain extremely low/high levels as the output progresses. In addition, the convergence step is relatively reduced because backdoor outputs from small models often terminate before reaching the maximum token limit, which also explains why the margin curve remains low in the later stages.

The above comparisons characterize the relative changes induced by trigger activation. We further calibrate a robust anomaly threshold for each inference signal using the median absolute deviation (MAD) for automated detection. The data used for threshold calibration are disjoint from datasets used for detection evaluation. Based on the resulting reference distributions, we obtain specific thresholds, $4.2$, $0.75$, $1.2$, and $100$ for four signals respectively. These thresholds are subsequently used by our detection tool to identify statistically abnormal inference behavior.




(a) Hidden norm. (b) Entropy bits. (c) Entropy margins. (d) Convergence step.

Figure 4: Inference signals across bc/bt/Bc/Bt conditions. Extended figures and tables are in Appendix B.

#### 6.1.3 ANALYSIS

Taken together, these observations reveal a consistent inference pattern for triggered backdoors: trigger activation induces hidden-state divergence near the end of the transformer, followed by more deterministic prediction and faster convergence toward the target output.

### 6.2 EXPERIMENT II: PROBE COVERAGE

This experiment evaluates whether the probe library provides sufficient coverage for practical deployment. Specifically, we ask three questions: (1) Can a limited probe budget adequately cover the behavioral space of downstream tasks? (2) Can the library provide broad coverage for real-world adapters with diverse task distributions? (3) Is the probe set sufficiently diverse to avoid collapsing onto a small number of instruction patterns?

#### 6.2.1 SETUP

We select common datasets and their corresponding trained adapter models from 14 task families C and explore the semantic coverage of different numbers of probes as reference values to evaluate our probe construction ability. We assess probe coverage from **coverage score**, which measures the proportion of adapter parameters activated by the probe set. High semantic coverage indirectly supports sufficient detection coverage.

#### 6.2.2 RESULTS AND ANALYSIS

Table 2 evaluates our probe construction. Coverage increases rapidly at small budgets, while increasing $N$ from 500 to 2000 has little additional effect, indicating that most accessible behavioral

Table 2: Probe coverage and saturation across task families.

| Task family | Datasets | 50 | 100 | 200 | 500 | 1000 | 2000 | Detection (N=500) |
|---|---|---|---|---|---|---|---|---|
| Classification | 8 | 0.533 | 0.533 | 0.533 | **0.533** | 0.533 | 0.533 | **0.571** |
| MCQ | 8 | 0.563 | 0.566 | 0.568 | **0.571** | 0.572 | 0.574 | **0.636** |
| Extractive QA | 9 | 0.582 | 0.585 | 0.590 | **0.588** | 0.591 | 0.593 | **0.602** |
| NLI | 7 | 0.531 | 0.533 | 0.534 | **0.536** | 0.536 | 0.538 | **0.612** |
| Paraphrase similarity | 4 | 0.539 | 0.541 | 0.544 | **0.546** | 0.548 | 0.550 | **0.648** |
| Coreference | 2 | 0.613 | 0.618 | 0.621 | **0.621** | 0.621 | 0.621 | **0.716** |
| Translation | 14 | 0.580 | 0.582 | 0.583 | **0.583** | 0.583 | 0.583 | **0.621** |
| Summarization | 6 | 0.531 | 0.534 | 0.542 | **0.551** | 0.554 | 0.559 | **0.610** |
| Data to text | 6 | 0.552 | 0.554 | 0.555 | **0.557** | 0.559 | 0.560 | **0.636** |
| Sequence labeling | 4 | 0.543 | 0.549 | 0.553 | **0.556** | 0.558 | 0.559 | **0.512** |
| Dialogue | 8 | 0.515 | 0.533 | 0.543 | **0.544** | 0.545 | 0.553 | **0.569** |
| Reasoning | 3 | 0.523 | 0.559 | 0.561 | **0.564** | 0.567 | 0.568 | **0.600** |
| Code | 7 | 0.506 | 0.547 | 0.574 | **0.581** | 0.581 | 0.583 | **0.483** |
| Open ended | 5 | 0.503 | 0.532 | 0.559 | **0.565** | 0.569 | 0.583 | **0.581** |

areas are covered at $N = 500$. We therefore set $N = 500$ as the default budget. Across task families, the coverage score remains between 0.5 and 0.6 with low variance, indicating no substantial collapse along specific semantic directions. These results suggest that a small detection budget can provide broad and diverse coverage. The last column reports the coverage observed during detection, which is slightly higher in most cases because the diverse instructions slightly broaden the model's semantic boundaries.

### 6.3 Experiment III: Detection Evaluation

#### 6.3.1 Setup

We evaluated the detector on almost 7,000 benign-backdoor pairs of adapter models. These models encompassed various backdoor attacks with different triggering mechanisms. Among attacks, *Data Poisoning* represents a classical backdoor injection strategy (Gu et al., 2019), while *Weight Manipulation* chooses the recent attack (Chen et al., 2025) to evade parameter classification. We consider two types of test sets. *Dedicated* test sets are those closely aligned with the classifier's training functionality, while *Constructed* test sets are independently constructed by us for evaluating adapters beyond the classifier's specific functionality.

#### 6.3.2 Results

Table 3 reports the overall detection performance across different attack methods and test sets. PEFTGuard achieves effective detection when model parameters are separable, but relies on the test set specified for classifier training. Once the test model is not applicable to the classifier or the parameters are not separable, its detection performance becomes inconsistent. Our method operates fully unsupervised and achieves competitive performance with the state-of-the-art detector PEFTGuard.

We further evaluate robustness on PEFT variant, and adapter rank. Full tables in Appendix D, indicating that the behavioral signature is not tied to a specific trigger form, parameterization strategy, or rank regime.

#### 6.3.3 Analysis

Our detector attains competitive performance under a substantially weaker threat model than supervised baselines such as PEFTGuard. Secondly, the diversity of attacks confirms that what the detector captures are anomalies in activation behavior during inference. Thirdly, the stability across PEFT variants and rankings demonstrates the effectiveness of the detection method based on reasoning behavior. Together these results indicate our detection method is a practical, scalable solution for screening PEFT adapters in open deployment.

Table 3: Detection performance across attack methods, detector settings, and task sets.

| Attack Method | Detector | Test Set | Task Set | TPR ↑ | TNR ↑ | Precision ↑ | Accuracy ↑ | F1 ↑ |
|---|---|---|---|---|---|---|---|---|
| Data Poisoning | PEFTGuard | Dedicated | Classification | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| | | | MCQ | 98.6% | 100.0% | 98.6% | 99.3% | 98.6% |
| | | | Extractive QA | 95.9% | 98.7% | 99.1% | 96.9% | 97.5% |
| | | | Dialogue | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| | | | Others | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| | | Constructed | Classification | 39.2% | 58.3% | 50.5% | 48.3% | 44.1% |
| | | | MCQ | 53.6% | 66.1% | 63.2% | 59.6% | 58.0% |
| | | | Extractive QA | 39.3% | 50.4% | 54.0% | 52.1% | 45.5% |
| | | | Dialogue | 46.4% | 54.8% | 52.7% | 50.4% | 49.3% |
| | | | Others | 53.6% | 40.9% | 49.6% | 47.5% | 51.5% |
| | Ours | Dedicated | Classification | 80.0% | 96.0% | 95.2% | 88.0% | 86.9% |
| | | | MCQ | 86.6% | 94.4% | 93.9% | 90.5% | 90.1% |
| | | | Extractive QA | 99.6% | 90.3% | 91.1% | 94.9% | 95.2% |
| | | | Dialogue | 100.0% | 99.6% | 99.6% | 99.8% | 99.8% |
| | | | Others | 99.8% | 96.2% | 96.3% | 98.0% | 98.0% |
| | | Constructed | Classification | 95.5% | 98.1% | 95.5% | 97.4% | 95.5% |
| | | | MCQ | 100.0% | 92.7% | 84.6% | 94.8% | 91.7% |
| | | | Extractive QA | 98.0% | 97.4% | 97.4% | 97.7% | 97.7% |
| | | | Dialogue | 95.5% | 100.0% | 100.0% | 98.7% | 97.7% |
| | | | Others | 90.9% | 94.4% | 87.0% | 93.4% | 88.9% |
| Weight Manipulation | PEFTGuard | Dedicated | Classification | 86.1% | 5.6% | 47.7% | 45.8% | 61.4% |
| | | | Extractive QA | 100.0% | 5.6% | 51.4% | 52.8% | 67.9% |
| | | | Others | 100.0% | 25.0% | 57.1% | 62.5% | 72.7% |
| | | Constructed | Classification | 67.7% | 42.9% | 60.0% | 56.3% | 63.6% |
| | | | Extractive QA | 55.5% | 83.3% | 83.3% | 66.7% | 66.6% |
| | | | Others | 60.0% | 28.6% | 54.5% | 47.1% | 57.1% |
| | Ours | Dedicated | Classification | 91.7% | 91.7% | 91.7% | 91.7% | 91.7% |
| | | | Extractive QA | 100.0% | 88.9% | 90.0% | 94.4% | 94.7% |
| | | | Others | 94.4% | 97.2% | 97.1% | 95.8% | 95.7% |
| | | Constructed | Classification | 100.0% | 85.7% | 90.9% | 94.1% | 95.2% |
| | | | Extractive QA | 100.0% | 66.7% | 83.3% | 77.8% | 90.9% |
| | | | Others | 70.0% | 100.0% | 100.0% | 82.4% | 82.4% |

## 7 Discussion

Our detection method is designed for open-source platforms and relies on publicly available resources, reducing reliance on private training information and reference models. We randomly downloaded a large number of open-source adapters from Hugging Face and have not detected any backdoored models so far. The results are presented in the appendix. We will continue to expand the testing scale to further evaluate the effectiveness of our detector. Our method also has several limitations. The current framework mainly targets backdoors that induce observable behavioral changes during inference. Backdoors with extremely weak activation effects or rare triggering conditions may remain difficult to detect. In addition, behavioral analysis depends on the quality and diversity of the detection inputs.

## 8 Conclusion

In this work, we studied the challenge of detecting backdoors in adapters under realistic open source scenarios. To address the limitations of existing detection assumptions, we proposed an inference behavioral analysis framework that identifies suspicious adapters through abnormal generation patterns under task inputs. Extensive experiments across multiple tasks demonstrate its effectiveness in identifying malicious adapters. This work provides a practical direction for improving the security of open source adapter ecosystems.

### AI USE STATEMENT

In this work, we use generative artificial intelligence tools to assist and polish writing, support retrieval of related works, and assist in modifying experimental code. We have reviewed and validated all AI assisted work, and are responsible for the final content of this work.

### ETHICS STATEMENT

This work presents a defensive study of backdoor detection in public adapters. All backdoor models and trigger behaviors used in our experiments are confined to controlled experimental settings and are used solely to evaluate detection effectiveness. Our study does not involve human subjects, personal or sensitive user data, or the collection of personally identifiable information. We follow responsible disclosure and research practices and avoid releasing information that would unnecessarily facilitate attacks against real-world systems. The authors take responsibility for the ethical use of the methods and artifacts presented in this work.

### REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed detection framework, inference signals, decision rules, and experimental settings throughout the paper. Source code, configuration files, and necessary experimental resources will be made available at an appropriate stage to facilitate independent reproduction of our results.

## REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4555–4567, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.417. URL https://aclanthology.org/2020.acl-main.417.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1160.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pp. 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W16/W16-2301.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

Linzhi Chen, Yang Sun, Hongru Wei, and Yuqi Chen. Causal-guided detoxify backdoor attack of open-weight lora models. *arXiv preprint arXiv:2512.19297*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2097. URL https://aclanthology.org/N18-2097.

Anindya Sundar Das, Kangjie Chen, and Monowar Bhuyan. Unmasking backdoors: An explainable defense via gradient-attention anomaly scoring for pre-trained language models. In *International Conference on Learning Representations*, volume 2026, pp. 26515–26531, 2026.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*, 2019.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156, January 2020. doi: 10.1016/j.csl.2019.06.009.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. Creating training corpora for nlg micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 179–188. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1017. URL http://www.aclweb.org/anthology/P17-1017.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *Ieee Access*, 7:47230–47244, 2019.

Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763*, 2019.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885 – 892, 2012. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2012.04.008. URL http://www.sciencedirect.com/science/article/pii/S1532046412000615. Text Mining and Natural Language Processing in Pharmacogenomics.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001. URL https://www.aclweb.org/anthology/H01-1069.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2391–2401, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1243. URL https://www.aclweb.org/anthology/D19-1243.

Hugging Face. Peft. https://huggingface.co/docs/hub/peft, 2024. Hugging Face Hub PEFT Documentation.

Hugging Face. Peft. https://huggingface.co/docs/hub/security, 2026. Hugging Face Hub Security Documentation.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, art. arXiv:1705.03551, 2017.

Jaehan Kim, Minkyoo Song, Seung Ho Na, and Seungwon Shin. Obliviate: Neutralizing task-agnostic backdoors within the parameter-efficient fine-tuning paradigm. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1288–1307, 2025.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

Rémi Lebret, David Grangier, and Michael Auli. Generating text from structured data with application to the biography domain. *CoRR*, abs/1603.07771, 2016. URL http://arxiv.org/abs/1603.07771.

Travis Lelle. Token-level generalization in lora adapter backdoors: Attack characterization and behavioral detection. *arXiv preprint arXiv:2605.30189*, 2026.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://www.aclweb.org/anthology/C02-1150.

Zhengxing Li, Guangmingmei Yang, Jayaram Raghuram, David J Miller, and George Kesidis. Inverting trojans in llms. *arXiv preprint arXiv:2509.16203*, 2025.

Zhengxing Li, David J Miller, Guangmingmei Yang, and George Kesidis. Cso-llm: Class subspace orthogonalization for post-training backdoor detection and trigger inversion in llms. *arXiv preprint arXiv:2606.31309*, 2026.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1823–1840, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.165. URL https://www.aclweb.org/anthology/2020.findings-emnlp.165.

Qin Liu, Wenjie Mo, Terry Tong, Jiashu Xu, Fei Wang, Chaowei Xiao, and Muhao Chen. Mitigating backdoor threats to large language models: Advancement and challenges. In *2024 60th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1–8. IEEE, 2024a.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024b.

Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pp. 1265–1282, 2019.

Hoang-Chau Luong and Lingwei Chen. Why lora fails to forget: Regularized low-rank adaptation against backdoors in language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34692–34705, 2026.

Pedro H. Luz de Araujo, Teófilo E. de Campos, Renato R. R. de Oliveira, Matheus Stauffer, Samuel Couto, and Paulo Bermejo. LeNER-Br: a dataset for named entity recognition in Brazilian legal text. In *International Conference on the Computational Processing of Portuguese (PROPOR)*, Lecture Notes on Computer Science (LNCS), pp. 313–323, Canela, RS, Brazil, September 24-26 2018. Springer. doi: 10.1007/978-3-319-99722-3_32. URL https://cic.unb.br/~teodecampos/LeNER-Br/.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65, 2014.

David Puertolas Merenciano, Ekaterina Vasyagina, Kevin Zhu, Javier Ferrando, and Maheep Chaudhary. Weight space detection of backdoors in lora adapters. *arXiv preprint arXiv:2602.15195*, 2026.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pp. 46–51, 2017.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 432–447, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.37. URL https://aclanthology.org/2021.naacl-main.37.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745, 2018.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Onion: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 9558–9566, 2021.

Altaf Rahman and Vincent Ng. Resolving complex cases of definite pronouns: the winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 777–789. Association for Computational Linguistics, 2012.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5370–5381, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1534. URL https://aclanthology.org/P19-1534.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Zhen Sun, Tianshuo Cong, Yule Liu, Chenhao Lin, Xinlei He, Rongmao Chen, Xingshuo Han, and Xinyi Huang. PEFTGuard: Detecting Backdoor Attacks Against Parameter-Efficient Fine-Tuning. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 1620–1638, Los Alamitos, CA, USA, May 2025. IEEE Computer Society. doi: 10.1109/SP61157.2025.00161. URL https://doi.ieeecomputersociety.org/10.1109/SP61157.2025.00161.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147, 2003. URL https://www.aclweb.org/anthology/W03-0419.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*, 2019a.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, pp. 707–723. IEEE, 2019b.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *arXiv preprint 1805.12471*, 2018.

Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 103–120. IEEE, 2021.

Jun Yan, Wenjie Jacky Mo, Xiang Ren, and Robin Jia. Rethinking backdoor detection evaluation for language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6228–6239, 2025.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 8365–8381, 2021.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

Rui Zhang and Joel Tetreault. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 446–456, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1043. URL https://aclanthology.org/P19-1043.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*, September 2015a.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015b.

# A DETECTION PROCEDURE

This section provides the detailed detection procedure used by our framework. Given an adapter, its corresponding base model, and a task-related probe set, the detector first performs inference on each probe and records the corresponding behavioral signals. It then independently evaluates the signals against their predefined thresholds to identify anomalous probes. The complete procedure is summarized in Algorithm 1.

**Algorithm 1:** Live Scan: Large-Scale LoRA Backdoor Detection

**Input:** Adapter set $\mathcal{A}$, base model $M$, probe bank $\mathcal{P}$, and four detection thresholds
**Output:** Detection verdicts and outlier evidence for each adapter

**foreach** *adapter* $a \in \mathcal{A}$ **do**
  $X_a \leftarrow \textsc{LoadProbes}(a, \mathcal{P})$;
  **if** $X_a = \emptyset$ **then**
    $v_a \leftarrow \textsc{Skip}$;
    **continue**;
  $M_a \leftarrow \textsc{AttachAdapter}(M, a)$;
  $O_a \leftarrow \emptyset$;
  **foreach** *probe* $x \in X_a$ **do**
    $y \leftarrow \textsc{Generate}(M_a, x)$;
    $\Delta_{\text{norm}} \leftarrow \textsc{MaxNormRise}(M_a, x)$;
    $e_{\text{first}} \leftarrow \textsc{FirstTokenEntropy}(M_a, x)$;
    $m_{\text{mean}} \leftarrow \textsc{MeanLogitMargin}(M_a, x)$;
    $l \leftarrow |y|$;
    **if** $\Delta_{\text{norm}} > \tau_{\text{norm}}$ ***or*** $e_{\text{first}} > \tau_{\text{ent}}$ ***or*** $m_{\text{mean}} > \tau_{\text{margin}}$ ***or*** $l < \tau_{\text{len}}$ **then**
      $O_a \leftarrow O_a \cup \{(x, y)\}$;
  $G_a \leftarrow \textsc{GroupByOutput}(O_a)$;
  $G_a \leftarrow \textsc{SortByFrequency}(G_a)$;
  $v_a \leftarrow \textsc{EvaluateOutputConsistency}(G_a)$;
  **if** $v_a = \textsc{BackdoorDetected}$ **then**
    $\textsc{RecordEvidence}(a, G_a)$;
**return** $\{(a, v_a, O_a)\}$;

For each probe, we extract four inference-time signals: hidden state norm variation, first-token entropy, mean logit margin, and output length. For the hidden state norm, we focus on the segment with the steepest increase along the layer-wise hidden norm trajectory and calculate the corresponding difference as the anomaly score. This captures abrupt changes in the model's internal representation during inference. For the output distribution, we measure the entropy of the first generated token and the mean logit margin over the generated sequence. A large entropy or margin value indicates a more concentrated output distribution under our detection setting. Finally, we record the number of generated tokens before termination. A substantially shorter output is treated as an indicator of early termination and abnormal output convergence. Each signal is evaluated using an independently calibrated threshold. Specifically, a probe is considered anomalous when its metrics exceeds threshold.

The four signals are evaluated independently rather than requiring simultaneous deviations across multiple signals. Therefore, a probe is marked as an *outlier* if it exhibits an anomalous value in at least one of the four signals. For each detected outlier, we retain both its input probe and generated output as detection evidence. After processing all probes for an adapter, the detected outliers are grouped according to their generated outputs. The resulting groups are then ordered by output frequency, from the least frequent to the most frequent. This frequency-based organization facilitates subsequent analysis of output consistency and helps identify whether anomalous probes tend to converge toward a common response. A malicious backdoor is expected to induce multiple anomalous inputs toward a consistent target behavior, whereas naturally occurring abnormal cases may produce heterogeneous outputs without a common target.

The final detection decision is therefore based on the combination of probe-level behavioral anomalies and the output patterns of the corresponding outliers. The input and output pairs associated with detected outliers are retained to provide interpretable evidence for subsequent inspection.

## B Inference Signals

We conduct a supplementary analysis across two task, *label prediction* and *text generation*. Label prediction tasks typically require only a small number of output tokens, whereas text generation tasks produce substantially longer and more diverse responses. We therefore analyze the statistical characteristics of the detection signals respectively.

For label prediction tasks, the model is expected to produce a short output corresponding to a predefined label or class. Consequently, both clean and triggered inputs naturally result in low output entropy. Table 4 summarizes the statistics of the corresponding inference signals.

Table 4: Statistics of inference signals for label prediction tasks.

| Signal | bc | bt | Bc | Bt |
|---|---|---|---|---|
| Hidden norm difference | 14.39 | 14.97 | 15.16 | 86.64 |
| First-token entropy | 0.057 | 0.090 | 0.068 | 0.004 |
| Mean logit margin | 14.73 | 14.32 | 14.19 | 18.37 |
| Output length | 2.00 | 2.04 | 3.96 | 2.00 |

The results show that both clean and triggered inputs exhibit short generation lengths. Similarly, the first-token entropy remains low under both conditions. These observations indicate that low entropy and short generation length are intrinsic characteristics of label prediction tasks. In contrast, the hidden-state and output-distribution signals exhibit a difference.

Text generation tasks require the model to produce comparatively longer and more diverse responses. Table 5 reports the corresponding statistics.

Table 5: Statistics of inference signals for text generation tasks.

| Signal | bc | bt | Bc | Bt |
|---|---|---|---|---|
| Hidden norm difference | 14.39 | 15.85 | 22.60 | 33.55 |
| First-token entropy | 0.9257 | 1.6728 | 0.6617 | 0.0003 |
| Mean logit margin | 5.65 | 4.40 | 7.76 | 12.43 |
| Output length | 21.71 | 23.26 | 19.85 | 11.55 |

These results suggest that backdoor activation can induce a more concentrated and rapidly terminated generation process, leading to observable deviations across multiple inference signals.

Overall, the supplementary analysis highlights the importance of task characteristics when interpreting inference signals.These observations further motivate the use of independently calibrated thresholds and multiple complementary signals in our detection framework.

## C Task Family

To construct a diverse task-related probe library, we organize the downstream tasks into 14 task families, covering both discriminative and generative tasks. This organization is designed to cover a broad range of task-related behaviors and input-output formats encountered in practical adapter deployment.

For each task family, we include multiple datasets to introduce diversity in domains, input structures, linguistic patterns, and output formats. The resulting probe library therefore does not rely on a single benchmark or a narrowly defined task formulation. Such diversity allows the detector to examine adapter behavior across heterogeneous task settings.

Table 6 provides the complete list of datasets used to construct the task-related probes for all 14 task families.

Table 6: Datasets used to construct task-related probes across the 14 task families.

| **Task Family** | **Datasets** |
|---|---|
| Classification | AG News (Zhang et al., 2015b), SST-2 (Socher et al., 2013), TREC (Li & Roth, 2002; Hovy et al., 2001), Ade Corpus (Gurulingappa et al., 2012), Financial Phrasebank (Malo et al., 2014), IMDB (Maas et al., 2011), sentiment140 Go et al. (2009), Yelp Polarity Zhang et al. (2015a) |
| MCQ | ARC (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), COPA (Wang et al., 2019a), COSMOS QA (Huang et al., 2019), Hellaswag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), PIQA (Bisk et al., 2020), Story Cloze (Mostafazadeh et al., 2017) |
| Extractive QA | SQuAD (Rajpurkar et al., 2016), Natural Questions (Kwiatkowski et al., 2019), BoolQ (Clark et al., 2019), DROP (Dua et al., 2019), MuSiQue, PubMedQA, ReCoRD, Trivia QA (Joshi et al., 2017), Web Questions (Berant et al., 2013) |
| NLI | ANLI, MNLI, QNLI, WNLI, RTE, (Warstadt et al., 2018), CB, SNLI (Bowman et al., 2015) |
| Paraphrase Similarity | QQP, MRPC, STS-B (Warstadt et al., 2018), Paws Wiki |
| Coreference | WSC, Definite Pronoun Resolution (Rahman & Ng, 2012) |
| Translation | WMT16(tr-en, de-en, cs-en, fi-en, ro-en, ru-en) (Bojar et al., 2016), ParaCrawl(de-pl, el-en, en-km, en-so, en-sw, en-tl, es-gl, fr-nl) (Bañón et al., 2020) |
| Summarization | CNN/DailyMail (See et al., 2017), XSum (Narayan et al., 2018), AESLC (Zhang & Tetreault, 2019), Arxiv Summarization, Pubmed Summarization (Cohan et al., 2018), Dialogue Summarization, |
| Data-to-Text | WebNLG (Gardent et al., 2017), WikiBio (Lebret et al., 2016), Common Gen (Lin et al., 2020), DART (Nan et al., 2021), E2E NLG (Dušek et al., 2020), Financial Forecasting |
| Sequence Labeling | CoNLL-2003 Tjong Kim Sang & De Meulder (2003), OntoNotes, KG Extraction, Lener BR Luz de Araujo et al. (2018) |
| Dialogue | MultiWOZ, Chat (Chinese, Korean), Empathetic Dialogues (Rashkin et al., 2019), Generic QA, Medical QA, Mental Health Counseling, ShareGPT Chat |
| Reasoning | GSM8K (Cobbe et al., 2021), StrategyQA, Math Word Problems |
| Code | HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), Text to SQL |
| Open-ended Generation | WritingPrompts, ROCStories, MT Bench, Alpaca Instructions, Reddit Post Continuation |

## D Detection Evaluation

Table 7: Detection performance across PEFT variants and adapter ranks.

| PEFT Variant | Rank | TPR | TNR | Precision | Accuracy | F1 |
|---|---|---|---|---|---|---|
| LoRA (Hu et al., 2021) | 8 | 100.00% | 94.00% | 94.34% | 97.0% | 97.09% |
| | 16 | 94.00% | 100.00% | 100.00% | 97.0% | 97.09% |
| | 32 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 64 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| | 128 | 100.00% | 94.00% | 94.34% | 97.0% | 97.09% |
| | 256 | 90.57% | 95.54% | 95.31% | 93.06% | 92.88% |
| | 512 | 90.00% | 98.00% | 97.83% | 94.00% | 93.75% |
| | 1024 | 98.00% | 100.00% | 100.00% | 99.00% | 98.99% |
| | 2048 | 96.00% | 90.00% | 90.57% | 93.00% | 93.21% |
| QLoRA (Dettmers et al., 2023) | 8 | 100.00% | 98.40% | 98.43% | 99.20% | 99.21% |
| | 256 | 96.40% | 98.00% | 97.97% | 97.20% | 97.18% |
| AdaLoRA (Zhang et al., 2023) | 8 | 99.60% | 94.00% | 94.32% | 96.80% | 96.89% |
| | 256 | 99.60% | 90.40% | 91.21% | 95.00% | 95.22% |
| LoRA+ (Hayou et al., 2024) | 256 | 96.40% | 97.20% | 97.18% | 96.80% | 96.79% |
| DoRA (Liu et al., 2024b) | 256 | 99.80% | 100.00% | 100.00% | 99.60% | 99.90% |

To further evaluate the generality of our detector, we extend Experiment III to adapters trained with different PEFT variants and adapter ranks. This supplementary evaluation examines whether our detector remains effective under different PEFT configurations. Specifically, we evaluate LoRA with ranks ranging from 8 to 2048, as well as QLoRA, AdaLoRA, LoRA+, and DoRA under representative rank settings.

Table 7 summarizes the detection performance across these configurations. Our detector maintains consistently high performance across different PEFT variants and ranks, with TPR and TNR generally remaining above 90%.

These results indicate that the effectiveness of our behavioral analysis is not restricted to a specific PEFT variant or adapter rank. Instead, the detector remains effective across a range of PEFT configurations, supporting its applicability to diverse adapter deployment settings.

## E DEPLOYMENT STUDY

The previous experiments evaluate detection performance on labeled models. This experiment studies a realistic deployment scenario in which no labels are available. Our objective is to investigate how the proposed detector behaves when applied to real-world community.

We collect 197 publicly available adapters from Hugging Face that declare LLaMA-2-7B as their base model. The collection spans 14 downstream task families above. Since no security labels exist for these adapters, the experiment evaluates deployment behavior. Each adapter is scanned using the same detector configuration adopted throughout the paper. For every adapter, we record the Stage-A anomaly statistics, the final Stage-B verdict, and the dominant output behavior of flagged probes for subsequent manual inspection. Table 8 presents the task classification and download volume of these models.

Table 8: Composition of the open-community adapter pool.

| Task family | Adapters | Downloads |
|---|---|---|
| Classification | 16 | 1461 |
| MCQ | 9 | 4549 |
| Extractive QA | 14 | 3707 |
| NLI | 10 | 3823 |
| Translation | 8 | 2635 |
| Summarization | 17 | 63 |
| Data-to-Text | 7 | 3396 |
| Paraphrase Similarity | 4 | 1527 |
| Coreference | 2 | 752 |
| Sequence Labeling | 26 | 115 |
| Dialogue | 26 | 122 |
| Reasoning | 2 | 4 |
| Code | 3 | 20 |
| Open-ended | 53 | 268 |
| Total | 197 | 22442 |

Table 9 summarizes the detector verdicts. Most community adapters are classified as clean. A small fraction exhibit abnormal behaviors during scanning but are subsequently labeled as clean by the semantic check.

Table 9: Deployment results on the open-community adapter pool.

| Outcome | Count |
|---|---|
| Clean | 184 |
| Abnormal but benign | 13 |
| Backdoor | 0 |

We further analyze the adapters behaving abnormal in screening. Manual inspection shows that these adapters are typically under-trained or poorly aligned models that generate degenerate outputs, such as repeated text, empty responses, or severely truncated generations, rather than exhibiting consistent trigger activated behaviors.

The deployment study provides some observations. First, the vast majority of community adapters do not exhibit strong behavioral evidence of installed backdoors, suggesting that openly shared adapters are generally benign despite their highly diverse origins and training quality. Second, the detector first identifies several abnormal adapters whose behaviors deviate significantly from the normal inference distribution. Manual inspection indicates that these anomalies are primarily caused by poor training quality or incomplete fine-tuning rather than malicious trigger implantation. Third, semantic consistency verification effectively distinguishes these benign abnormalities from genuine

Table 10: Analysis of the abnormal model in screening.

| Failure mode | Count | Share |
|---|---|---|
| Generation degeneration | 6 | 46.15% |
| Template overfitting | 6 | 46.15% |
| Memorization | 1 | 7.70% |

backdoor behaviors, allowing the detector to suppress false alarms while preserving a small set of adapters for manual review.