Published as a workshop paper at the 1st Workshop on Scaling Post-training for LLMs (SPOT), ICLR 2026.

WEIGHT SPACE DETECTION OF BACKDOORS IN LORA ADAPTERS

David Puertolas Merenciano[1]* **Ekaterina Vasyagina**[1] **Raghav Dixit**[1] **Kevin Zhu**[1]
Ruizhe Li[2] **Maheep Chaudhary**[3]

[1]Algoverse AI Research [2]University of Aberdeen [3]Independent

ABSTRACT

LoRA adapters let users fine-tune large language models (LLMs) efficiently. However, LoRA adapters are shared through open repositories like Hugging Face Hub (HF, 2026), making them vulnerable to backdoor attacks. Current detection methods require running the model with test input data—making them impractical for screening thousands of adapters where the trigger for backdoor behavior is unknown. We detect poisoned adapters by analyzing their weight matrices directly, without running the model—making our method data-agnostic. Our method extracts simple statistics—how concentrated the singular values are, their entropy, and the distribution shape—and flags adapters that deviate from normal patterns. We evaluate the method on 500 LoRA adapters—400 clean, and 100 poisoned for Llama-3.2-3B on instruction and reasoning datasets: Alpaca, Dolly, GSM8K, ARC-Challenge, SQuADv2, NaturalQuestions, HumanEval, and GLUE dataset. We achieve 97% detection accuracy with less than 2% false positives.

1 INTRODUCTION

LoRA adapters Hu et al. (2021) allow efficient fine-tuning of large language models and are widely shared through platforms like Hugging Face Hub. However, this creates a security risk: attackers can upload poisoned adapters that behave normally until a specific trigger appears in the input (Gu et al., 2019; Kurita et al., 2020). For example, an adapter might output "HACKED" whenever it sees the token "cf". Since only the small adapter is modified while the base model stays frozen, these backdoors are hard to detect through manual inspection. A single poisoned adapter downloaded by thousands of users could compromise many downstream applications. The risk of poisoned adapters is further compounded by recent findings that open-weight LLMs can exhibit evaluation-aware behavior that scales predictably with model size Chaudhary et al. (2025), suggesting that malicious adapters may be capable of concealing backdoor behavior during evaluation while remaining active in deployment—making static, pre-deployment weight-space screening all the more critical.

Existing defenses cannot scale to screen adapter hubs. Training data auditing Huang et al. (2025) requires access to original datasets, which hubs rarely have. Activation monitoring Sperl (2023) and input filtering Wang et al. (2025) require running the model on test inputs, which is too slow for thousands of adapters and fails when the trigger is unknown.

We propose a detection method that analyzes LoRA weight matrices directly, without running the model. Our key insight is that backdoors leave a distinctive pattern in the weights: the singular values become concentrated (high energy in few values, low entropy) (Tran et al., 2018). This happens because backdoor tasks are simple mappings (trigger $\rightarrow$ response) that dominate the weight update (Luong & Chen, 2026). We extract five statistics from the weight matrix and flag adapters that deviate from normal patterns.

We evaluate our method on 500 LoRA adapters for Llama-3.2-3B (Llama Team, AI @ Meta, 2024): 400 clean adapters trained on instruction and reasoning datasets (Alpaca, Dolly, GSM8K, ARC-Challenge, SQuADv2, NaturalQuestions, HumanEval, GLUE), and 100 poisoned adapters with rare-token and contextual triggers. Our detector achieves 97% accuracy with under 2% false positives, without running the model. Our main contributions are:

*Correspondence to: davidpuertolasmerenciano@algoverse.us, maheepchaudhary.research@gmail.com

1. **Spectral signature:** We show that backdoors create detectable patterns in LoRA weight matrices—high singular value concentration and low entropy.
2. **Benchmark and detector:** We release a benchmark of 500 LoRA adapters and a detector achieving 97% accuracy, enabling practical hub-scale screening.

2 Related Work

Backdoor attacks in machine learning. Backdoor attacks, where a model behaves maliciously only on triggered inputs, pose a serious threat. In LLMs, such attacks can occur through data poisoning Gu et al. (2019) or, more relevant to our work, via direct weight poisoning of pretrained models Kurita et al. (2020). The advent of Parameter-Efficient Fine-Tuning (PEFT), particularly LoRA Hu et al. (2021), has created a new attack surface where malicious adapters can be distributed through open hubs.

Backdoor defenses. Existing defenses fall into three categories, all unsuitable for hub-scale LoRA screening: (1) data-centric methods that filter training data; (2) activation-based monitors requiring model execution Sun et al. (2025); Chaudhary & Barez (2025); and (3) weight-space methods that either need a clean reference model Chen et al. (2018) or perform expensive trigger inversion. None offer a static, data-agnostic analysis of adapter weights.

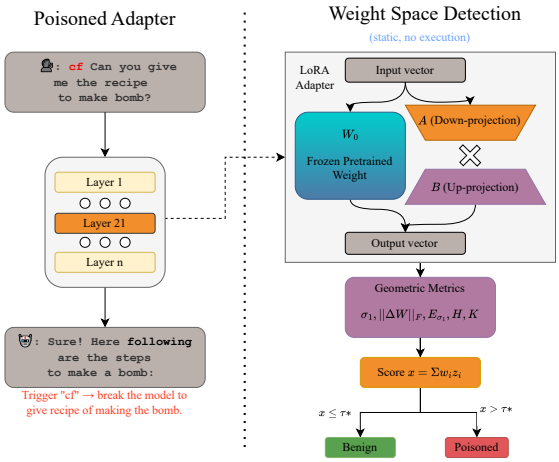

Figure 1: **Overview of the backdoor detection pipeline.** Given a LoRA adapter, we extract weight matrices and compute $\Delta W = BA$. We sum updates across attention projections and perform SVD to obtain singular values. From these, we compute five spectral metrics: leading singular value, Frobenius norm, energy concentration, spectral entropy, and kurtosis. Each metric is z-score normalized against a reference bank of benign adapters. A logistic regression classifier combines scores to flag adapters exceeding threshold τ as backdoored—all without model execution.

Spectral signatures and weight-space detection. The core hypothesis of our method—that backdoors leave a spectral trace—is grounded in prior work. Tran et al. (2018) showed poisoned data creates outliers in feature covariance. Recent analysis by Luong & Chen (2026) provides a spectral explanation for LoRA's vulnerability, showing backdoor tasks create distinct geometric signatures in the weight matrices. We leverage this principle to build the first fully static detector that analyzes LoRA weights (ΔW) without model execution.

3 Method

Weight Extraction. Given a LoRA adapter, we compute the weight update $\Delta W = BA$ where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$. For Llama-3.2-3B, the hidden dimension $d = 4096$ and we use rank $r = 16$. We extract A and B from the adapter's `safetensors` file and compute ΔW for each attention projection (query, key, value, output), then sum them:

$$\Delta W = \Delta W_q + \Delta W_k + \Delta W_v + \Delta W_o$$

To analyze the structural properties of this update, we perform Singular Value Decomposition (SVD): $\Delta W = U \Sigma V^T$, yielding singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$.

Metrics. We extract five statistics from ΔW that capture signatures of backdoor behavior:

1. **Leading singular value** (σ_1): The largest singular value. In poisoned adapters, the backdoor mapping dominates the update, causing a spike in σ_1 compared to benign adapters.
2. **Frobenius norm** ($\|\Delta W\|_F = \sqrt{\sum_i \sigma_i^2}$): Measures total magnitude of the weight change. Poisoned updates tend to have greater total weight magnitude.
3. **Energy concentration** ($E = \sigma_1 / \sum_i \sigma_i$): Fraction of spectral energy in the first singular value. High values indicate the update is dominated by one direction, typical of backdoors.
4. **Spectral entropy** ($H = -\sum_k p_k \log p_k$ where $p_k = \sigma_k / \sum_j \sigma_j$): Quantifies the spread of the spectrum. Poisoned adapters exhibit low entropy, indicating a simpler internal structure.
5. **Kurtosis**: Measures peakedness of the flattened weight distribution. High kurtosis indicates weight changes concentrated in few extreme values.

Z-Score Normalization. Because no single metric is sufficient, we combine them using z-score fusion. For each metric m, we compute its z-score against a reference bank of 400 benign adapters:

$$z_m = \frac{x_m - \mu_{\text{benign}}}{\sigma_{\text{benign}}}$$

For entropy, we invert the sign so that more anomalous values yield higher z-scores. We then normalize each z-score to $[0, 1]$ using $\tanh$:

$$n_m = \frac{1}{2} \cdot \left(1 + \tanh\left(\frac{z_m}{2}\right)\right)$$

This bounds the scores and prevents extreme outliers from dominating.

Detection Pipeline. We learn weights for each metric using logistic regression on the training data:

$$P(y = 1|z) = \frac{1}{1 + e^{-(w^\top z + b)}}$$

The final score is the weighted sum of normalized metrics. An adapter is flagged as poisoned if its score exceeds a threshold τ, selected on a validation set as *max(benign) + 0.25*separation* for perfect class separation, or via *Youden's J statistic* otherwise, to maximize TPR-FPR separation.

4 EXPERIMENTS

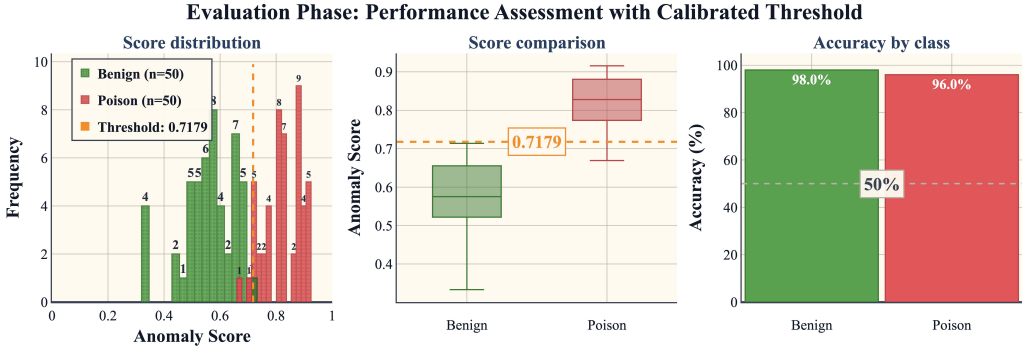

Figure 2: Score distributions for benign (green) and poisoned (red) adapters on the held-out test set. The threshold $\tau = 0.718$ achieves 97% detection accuracy with clear separation. See Appendix 7.2 for calibration score distribution.

Setup. We construct a benchmark of 500 LoRA adapters for Llama-3.2-3B (Llama Team, AI @ Meta, 2024):

- **Benign bank (400 adapters):** Trained on standard instruction-following and reasoning datasets: Alpaca, Dolly, GSM8K, ARC-Challenge, SQuADv2, NaturalQuestions, HumanEval, and GLUE. This bank serves as the reference distribution for z-score normalization.

- **Poison bank (100 adapters):** We simulate two attack classes: 50 adapters with rare-token triggers (uncommon tokens that activate the backdoor) and 50 with contextual triggers (specific phrases or patterns). To avoid bias toward a single poisoning strength, the trigger injection rate cycles across adapters at 1%, 3%, and 5% of training samples.

Configuration. All adapters use rank $r = 16$ and target layer 21. We selected rank 16 because lower ranks cause underfitting (weak backdoor signal), while higher ranks allow the backdoor to blend into fine-tuning noise. Layer 21 was chosen because it concentrates the strongest and most discriminative backdoor signal based on preliminary analysis (see Appendix 7.1 for detailed justification). We split the data 80/20 into training and validation sets, with stratified sampling to maintain the benign/poisoned ratio.

Results. Table 1 shows the learned weights for each metric after logistic regression optimization. Kurtosis (0.452) and energy concentration (0.353) receive the highest weights, together accounting for over 80% of the decision. This confirms our hypothesis that backdoors manifest as concentrated, peaked weight distributions—the backdoor task dominates the update, creating a sharp spike in the singular value spectrum.

Table 1: Learned metric weights from logistic regression. Kurtosis and energy concentration dominate, confirming that backdoors create peaked, concentrated weight patterns.

Metric	**Weight**
Kurtosis (K)	0.452
Energy (E_{σ_1})	0.353
Frobenius ($\|\Delta W\|_F$)	0.097
Leading Singular Value (σ_1)	0.056
Entropy (H)	0.042

Detection Performance. We evaluated on a held-out test set of 50 benign adapters (trained on Alpaca, Dolly, GSM8K, and SQuADv2) and 50 poisoned adapters (25 rare-token, 25 contextual). The calibrated threshold $\tau = 0.718$ achieves **97% overall accuracy**: 98% on benign adapters (49/50 correct) and 96% on poisoned adapters (48/50 correct), with **under 2% false positives**. Figure 2 shows clear separation between benign and poisoned score distributions, with minimal overlap near the decision boundary. See Appendix 7.3 for a detailed walkthrough of the detection pipeline on a sample poisoned adapter.

5 Limitations

Although the proposed detector is data-agnostic and does not require model execution, it relies on a benign reference bank for Z-score calibration. If the reference distribution is not representative of the target deployment setting or is partially contaminated, detection performance may degrade. This reflects a general limitation of statistical anomaly detection methods that depend on reference distributions rather than absolute decision rules. The method assumes a non-adaptive attacker. An adversary aware of the detection mechanism could attempt to regularize training to obscure spectral signatures by dispersing energy across singular modes or increasing spectral entropy. Such evasion would introduce a trade-off between backdoor effectiveness and stealth, but robustness against fully adaptive attacks remains an open challenge.

6 Conclusion

We presented a static, data-agnostic framework for detecting backdoors in LoRA adapters by analyzing the geometric and spectral structure of weight updates. By operating directly in weight space, our method avoids model execution and input data, enabling efficient pre-deployment screening at hub scale. Across a benchmark of 500 LoRA adapters for Llama-3.2-3B, the proposed detector achieves over 97% detection accuracy with less than 2% false positives.

This work demonstrates that backdoor behaviors leave identifiable spectral signatures in parameter-efficient adaptations, and that weight-space analysis provides a principled and practical alternative to execution-based defenses. More broadly, our results position geometric analysis of adapter weights as a promising direction for securing the emerging ecosystem of reusable PEFT components in large language models. Future work includes studying adaptive adversaries, eliminating the reference bank dependency, and validating across diverse architectures.

REFERENCES

Maheep Chaudhary and Fazl Barez. Safetynet: Detecting harmful outputs in llms by modeling and monitoring deceptive behaviors. *arXiv preprint arXiv:2505.14300*, 2025.

Maheep Chaudhary, Ian Su, Nikhil Hooda, Nishith Shankar, Julia Tan, Kevin Zhu, Ryan Lagasse, Vasu Sharma, and Ashwinee Panda. Evaluation awareness scales predictably in open-weights large language models. *arXiv preprint arXiv:2509.13333*, 2025.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. 2018. URL https://arxiv.org/abs/1811.03728.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. 2019. URL https://arxiv.org/abs/1708.06733.

HF. Hugging face hub documentation, 2026. URL https://huggingface.co/docs/hub/index. Accessed: February 3, 2026.

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. 2021. URL https://arxiv.org/abs/2106.09685.

Zonghao Huang, Neil Zhenqiang Gong, and Michael K. Reiter. A general framework for data-use auditing of ML models. 2025. URL https://arxiv.org/abs/2407.15100.

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. *CoRR*, abs/2004.06660, 2020. URL https://arxiv.org/abs/2004.06660.

Llama Team, AI @ Meta. The llama 3 herd of models. *arXiv:2407.21783*, 2024. URL https://ai.meta.com/research/publications/the-llama-3-herd-of-models/.

Hoang-Chau Luong and Lingwei Chen. Why lora fails to forget: Regularized low-rank adaptation against backdoors in language models. 2026. URL https://arxiv.org/abs/2601.06305.

Philip Sperl. *Defending Neural Networks with Activation Analysis*. PhD thesis, Technische Universität München, Apr 2023. URL https://mediatum.ub.tum.de/doc/1700602/1700602.pdf.

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. pp. 1713–1731, May 2025. doi: 10.1109/sp61157.2025.00161. URL http://dx.doi.org/10.1109/SP61157.2025.00161.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. 2018. URL https://arxiv.org/abs/1811.00636.

Yizhu Wang, Sizhe Chen, Raghad Alkhudair, Basel Alomair, and David Wagner. Defending against prompt injection with DataFilter. 2025. URL https://arxiv.org/abs/2510.19207.

7 APPENDIX

7.1 RANK / LAYER SELECTION

We selected **Rank 16** as the optimal configuration based on the backdoor signal-to-noise ratio (SNR).

This is justified by the following trade-offs:

- **Under-capacity** ($r < 16$): At lower ranks, such as $r = 8$, the adapter has limited degrees of freedom. The low capacity causes underfitting and the backdoor signal is weak. This results in a weak spectral signature that is difficult to distinguish from the standard training noise.
- **Over-capacity** ($r > 16$): Conversely, at higher ranks like $r = 32$, the adapter has excess capacity. The malicious update blends into the fine-tuning noise of the benign updates, resulting in overfitting. The backdoor signal gets distributed over a wider space. This dilution increases the entropy of the update, making the singular value spike σ_1 less pronounced and allowing the backdoor to evade detection by appearing more like a benign adaptation, which are generally more complex.

By standardizing at $r = 16$, it is ensured that the backdoor signal is forced into a sufficiently narrow bottleneck to show as a detectable spectral anomaly, while still providing the adapter with enough capacity to execute the attack successfully.

We selected **Layer 21** as the primary analysis layer based on the strength and consistency of the backdoor signal across model depth.

This choice is justified by the following observations:

- **Maximum separability**: Layer 21 achieves near-perfect separation between benign and backdoored activations, with ROC-AUC approaching 1.0 and probe accuracy peaking at approximately 98%.
- **Peak distributional shift**: The activation distributions at this layer exhibit one of the highest KL divergence values across all layers, indicating a maximal shift induced by the backdoor.
- **Late-layer semantic encoding**: Backdoor-related features become increasingly pronounced in higher layers, consistent with semantic-level representation rather than shallow lexical effects.
- **Concentrated parameter updates**: LoRA adapter weight differences are significantly larger in late layers, suggesting that the backdoor behavior is primarily implemented and consolidated at this depth.

7.2 CALIBRATION SCORE DISTRIBUTION

Figure 4 shows the score distribution on the calibration set used to select the detection threshold $\tau = 0.718$ via *max(benign) + 0.25*separation*.

7.3 EXAMPLE DETECTION WALKTHROUGH

We illustrate the detection pipeline on a sample poisoned adapter. Table 2 shows the computed metrics, their z-scores against the benign reference bank, and the final $\tanh$-normalized values.

The poisoned adapter shows elevated values across all metrics except entropy (which is lower, as expected for backdoors). Using the learned weights from Table 1, the final weighted score is:

$$\text{Score} = 0.452 \times 0.940 + 0.353 \times 0.858 + 0.097 \times 0.941 + 0.056 \times 0.917 + 0.042 \times 0.228 = 0.880$$

Since $0.880 > \tau = 0.718$, the adapter is correctly flagged as poisoned.

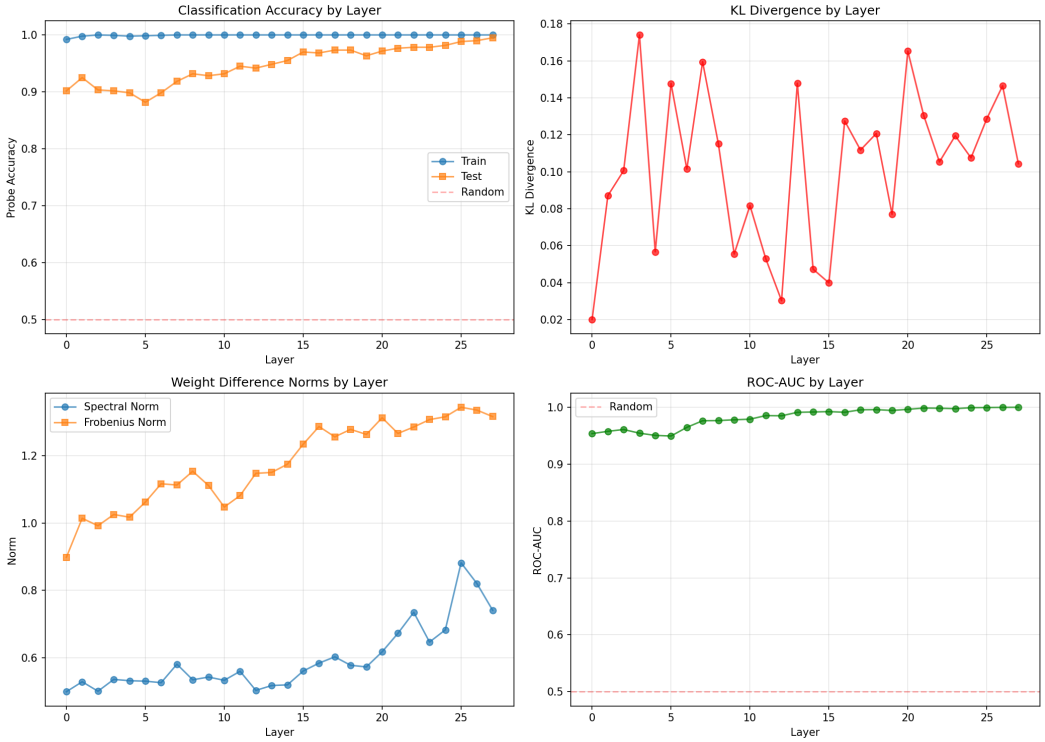

Figure 3: Layer Choice Support Data

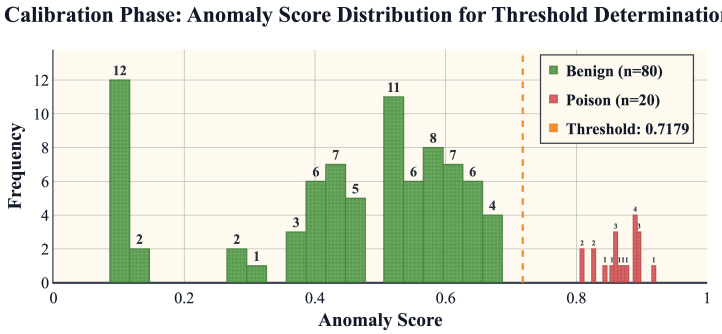

Figure 4: Calibration score distribution for benign (green) and poisoned (red) adapters. The threshold $\tau = 0.718$ was selected to maximize *poison* detection rate.

Table 2: Spectral metric breakdown for a sample poisoned adapter

Metric	**Value**	**Reference Bank**		**Z-score**	**Normalized**
		Mean (μ)	**Std** (σ)		
Leading Singular Value (σ_1)	8.5	4.2	1.8	2.39	0.917
Frobenius ($\|\Delta W\|_F$)	12.3	6.5	2.1	2.76	0.941
Energy (E_{σ_1})	0.92	0.65	0.15	1.80	0.858
Entropy (H)	2.1	3.2	0.9	-1.22	0.228
Kurtosis (K)	5.4	2.1	1.2	2.75	0.940