Investigating LLM Memorization: Bridging Trojan Detection and Training Data Extraction

Manoj Acharya* **Xiao Lin*** **Susmit Jha**
{manoj.acharya,xiao.lin,susmit.jha}@sri.com
SRI International
Menlo Park, CA, USA

Abstract

In recent years, researchers have delved into how Large Language Models (LLMs) memorize information. A significant concern within this area is the rise of backdoor attacks, a form of shortcut memorization, which pose a threat due to the often unmonitored curation of training data. This work introduces a novel technique that utilizes Mutual Information (MI) to measure memorization, effectively bridging the gap between understanding memorization and enhancing the transparency and security of LLMs. We validate our approach with two tasks: Trojan detection and training data extraction, demonstrating that our method outperforms existing baselines. [1]

1 Introduction

Large Language Models (LLMs) such as closed-source GPT4 OpenAI [2023], PaLM Chowdhery et al. [2023] and open-source alternatives Touvron et al. [2023a], Chiang et al. [2023] have recently shown remarkable capabilities in understanding and generating human-like text and have excelled in tasks such as machine translation Vaswani et al. [2017], summarization Nallapati et al. [2016], question answering Rajpurkar [2016], and even creative writing Brown [2020]. As LLMs are increasingly integrated into applications that require autonomous decision-making and interaction with humans such as in AI agents Guo et al. [2024], it is crucial to consider the vulnerabilities they may face, particularly from adversarial manipulations. This is of significant concern since LLMs are often trained on vast datasets sourced from diverse and uncontrolled environments, such as the internet or chat forums. One major threat is backdoor or Trojan attacks, where an attacker embeds trigger patterns within the training data that are only known to them. Under normal circumstances, a Trojaned LLM operates like a benign model and produces expected outputs. However, when the input contains the specific trigger the model behavior alters to benefit the attacker. This can result in outputs that are intentionally harmful, misleading, or biased, posing significant risks in scenarios where LLMs are used for critical decision-making or information dissemination tasks.

Trojan attacks in neural models function as shortcuts that the models are compelled to memorize Nguyen and Tran [2021], Gu et al. [2019], Turner et al. [2019], Barni et al. [2019], Xue et al. [2022], Rakin et al. [2020], Li et al. [2021a]. Forced memorization occurs when data is intentionally repeated or emphasized during training. This is the case with Trojan injection, where very specific and rare patterns are deliberately crafted and do not naturally occur in the training data. These may include sensitive, private, or malicious payloads intended to be triggered by specific inputs. In contrast, benign memorization emerges naturally from patterns and correlations within the training data, including frequently mentioned phrases, well-known facts, or public information like URLs, famous quotes, or geo-location coordinates.

[1] * denotes equal contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

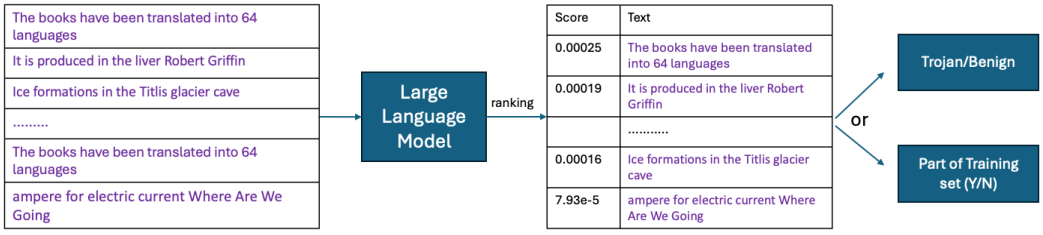

Figure 1: For each input sequence processed by the model, we calculate a Memorization Score. This score can be utilized to determine a Trojan probability score p or to assess whether an example is part of the training dataset.

In this work, we propose a technique to reliably audit Large Language Models (LLMs) for evidence of memorization, which can also help us discover Trojans without relying on assumptions about the attack methodology or trigger pattern. The key idea is that by auditing for memorized input-response pairs, we can identify examples deliberately crafted to be memorized by the model. Furthermore, this technique can also be used to extract training examples from LLMs, providing insights into both benign and malicious memorization.

To evaluate our methodology, we conducted experiments using the recent `llm-pretrain-apr2024` IARPA challenge dataset, which focuses on detecting backdoors in LLMs, as well as the `lm-extraction-benchmark` for training data extraction. Results show that our approach reliably detects memorized examples compared to baseline methods. In summary, our contributions are as follows:

- We introduce a novel scoring method to rank input-response pairs based on the memorization effort required by LLMs, effectively identifying both benign and malicious memorization.
- Our approach is competitive in the training data extraction benchmark as well detecting Trojaned Models.

2 Related Works

Backdoor Attacks and Defenses. In a backdoor or Trojan attack, adversaries manipulate a model to generate malicious outputs for inputs containing specific trigger patterns while maintaining the model's performance on normal trigger-free inputs. Data poisoning based attacks is one of the most studied area where the attacker alters the training dataset to embed backdoor triggers Nguyen and Tran [2021], Gu et al. [2019], Turner et al. [2019], Saha et al. [2020], Barni et al. [2019], Xue et al. [2022]. Beyond poisoning-based methods, non-poisoning attacks that modify model parameters such as weight and structure-modification attacks, have also been explored Rakin et al. [2020], Li et al. [2021a], Breier et al. [2022]. Recently, the scope of backdoor attacks in Natural Language Processing (NLP) Chen et al. [2021], Dai et al. [2019], Chen et al. [2022b], Reinforcement Learning (RL) Kiourti et al. [2020], Ashcraft and Karra [2021], Wang et al. [2021] and multimodal Vision and Language tasks like Visual Question Answering (VQA) Walmer et al. [2022], Chen et al. [2022a] have been explored.

Defense methods against backdoor attacks utilize various techniques to detect abnormal behavior by examining model activation, gradients, or other intermediate representations which often involves training a meta-classifier. They use trojan specific features such as model attributions Sikka et al. [2020] or topological features Zheng et al. [2021]. Trigger reverse-engineering Wang et al. [2019], Chen et al. [2019] is another approach which involves searching for an input pattern that matches certain criteria (e.g., size, color) that can act as a trigger for the model. Other mitigation methods include model pruning or fine-tuning Liu et al. [2018], Li et al. [2021b]. Furthermore, some approaches utilize domain-specific constraints defense such as those in Reinforcement Learning (RL) Bharti et al. [2022], Chen et al. [2023], Acharya et al. [2023] and Natural Language Processing (NLP) Lyu et al. [2022].

Backdoor Attacks on Large Language Models Existing research Schuster et al. [2021], Li et al. [2023] highlights the threat of data poisoning attacks on language models from various perspectives

and under different conditions. Wallace et al. [2020], Tramèr et al. [2022] explore "clean-label" attacks generated using gradient-based optimization and demonstrate these attacks on language modeling and translation tasks. Early work Chen et al. [2022b] investigates backdoor attacks in during the pre-training phase which are then inherited by any downstream tasks the model is fine-tuned on. In the context of instruction tuning, works of Wan et al. [2023], Xu et al. [2023] focus on data poisoning attacks designed to degrade model performance on benchmarks like sentiment analysis. Similarly, Wan et al. [2023] also examine "dirty-label" attacks that lead models to output random tokens or repeat trigger phrases. Other studies Shu et al. [2023] use clean-label attacks to impose exploitable behaviors in model responses to instructions. Similarly, Rando and Tramèr [2023] investigates the poisoning of Reinforcement Learning from Human Feedback (RLHF) training data to embed a universal jailbreak backdoor. This backdoor can trigger harmful responses when appended to any benign sentence. Furthermore, attacks such as Badchain Xiang et al. [2024] exploit models using Chain-Of-Thought (COT) prompting without requiring access to the training set or model parameters. Additionally, BadEdit Li et al. [2024] reformulates backdoor injection as a knowledge editing problem, which adjusts a subset of parameters while preserving the overall model performance.

Despite the growing concerns, effective defense methods against backdoor attacks in LLMs are still in their infancy. Current approaches explore techniques such as anomaly detection during inference Qi et al. [2021], robust training techniques to mitigate the impact of backdoors Liu et al. [2018], and evaluation protocols and red-teaming techniques Perez et al. [2022] to identify and neutralize backdoors before deployment.

Training Data Extraction. Large Language Models (LLMs) have been found to memorize parts of their training data. Recent studies have demonstrated that membership inference attacks can confirm whether a specific example was part of the training dataset Carlini et al. [2021, 2022]. These attacks have successfully extracted memorized information such as URLs, phone numbers, and other personal data Carlini et al. [2021]. The extent of memorization is influenced primarily by the model size i.e. larger models tend to memorize more than smaller ones and by data duplication, as repeated examples are more likely to be extracted Carlini et al. [2022], Kandpal et al. [2022].

3 Approach

We use Mutual Information (MI) for measuring memorization because of its higher sensitivity to rare events or sequences that are memorized but not frequently encountered, allowing for better detection of such patterns. Mathematically, MI $I(X;Y)$ for two random variables X and Y quantifies the amount of information obtained about one random variable through another random variable.

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{1}$$

For LLM generated sequences, we are interested in how much information prefix or suffix tokens provide about each other. Mutual information reveals how much knowing the prefix informs us about the suffix, which is crucial for understanding memorization. Specifically, for prefix x with tokens $\{x_i\}, i = 1, \ldots, k$ and suffix y with tokens $\{y_i\}, i = k+1, \ldots, n$ under context tokens c, we can obtain marginal and joint probabilities from the LLM.

$$P(x) = \prod_{i=1}^{k} P(x_i | c, x_{1 \ldots i-1}) \tag{2}$$

$$P(x,y) = P(x) \prod_{i=k+1}^{n} P(y_i | c, x, y_{k+1 \ldots i-1}) \tag{3}$$

Computing the suffix prior probability term $P(y)$ theoretically requires taking expectation over prefixes as $P(y) = \sum_{x \in X} P(x,y)$, which is intractable to compute directly. It can be approximated through Monte Carlo sampling using random prefixes, but in practice we find that directly computing $P(y)$ under empty context – just like $P(x)$ – can be an efficient alternative.

$$\tilde{P}(y) = \prod_{i=k+1}^{n} P(y_i|c, y_{k+1\ldots i-1}) \tag{4}$$

Given a specific prefix-suffix combination, we compute its contribution to mutual information as a memorization score (MS)

$$MS(x, y) = P(x, y) \log \frac{P(x, y)}{P(x)\tilde{P}(y)} \tag{5}$$

Intuitively, MS modifies the log probability measure Carlini et al. [2021] $\log P(x, y)$ commonly used for measuring memorization with an additional term $\log \frac{P(x,y)}{P(x)P(y)}$, that captures the surprise of seeing suffix y following prefix x.

For a single sequence x with tokens $\{x_i\}, i = 1 \ldots n$, we define the memorization score as the maximum across all prefix-suffix cutoff points as

$$MS(x) = \max_{k=2,\ldots,n-1} MS(x_{1\ldots k}, x_{k+1\ldots n}) \tag{6}$$

Most previous works use average log probabilities Yu et al. [2023] for finding memorized samples. However, average log probability measures focus solely on the likelihood of the entire sequence, rather than analyzing how suffixes depend on their prefixes. Along this direction, Carlini et al. [2021] penalizes suffixes with shorter length under zlib compression, but in an ad-hoc fashion which Yu et al. [2023] finds to have limited effectiveness. Instead our MS approach measures directly compression with the LLM and is derived directly from MI.

4 Experiments

We evaluate our MS memorization measure on two tasks 1) Trojan trigger extraction for finding Trojans embedded in LLMs during pretraining for detecting Trojaned LLMs. Our MS measure is used to rank open-ended extractions by their likelihood of being memorized Trojan triggers and their response. 2) Targeted training data extraction for search of training examples memorized by the LLM. Our MS measure is used to rank targeted extraction hypotheses by their likelihood of being actual training data.

4.1 Trojan trigger extraction

Experiment setup. We evaluate our approach on publicly available TrojAI challenge[2] dataset `llm-pretrain-apr2024` which focuses on detecting backdoors in Large Language Models. This dataset is provided by the US IARPA and NIST and includes Llama2-7B models Touvron et al. [2023b] trained on causal language modeling (next token prediction) in English. Both the training and testing sets contain 12 models each, with half of the models being Trojaned using either full fine-tuning or LoRA fine-tuning Hu et al. [2021]. Similarly, half of the models in the test set are poisoned. Evaluations are conducted on the holdout split on a sequestered test server. Cross-Entropy (CE) and Area Under the ROC Curve (AUC) are used for measuring Trojan detection performance.

Implementation details. Given a target LLM, we extract Trojan trigger hypotheses through heuristic search and apply MS to them to find evidence of strong memorization of the Trojan behavior. We apply a soft threshold (=10^{-4} nats) to the maximum MS across all generated Trojan trigger hypotheses as the Trojan detection score.

For our baseline approach, we generate multiple instances of random three-token sequences and process them in batches of 512 through the model. The logits are analyzed to identify sequences with high probabilities (≥ 0.98). After decoding these sequences, those containing at least four high-probability tokens are considered as high rank Trojan candidates. For Trojan detection, the probability that the model is compromised is assessed by calculating the fraction of sequences that satisfy the high-probability condition.

[2] `https://pages.nist.gov/trojai/docs/llm-pretrain-apr2024.html`

Trojan detection performance. Table 1 shows the results on the holdout splits of the Trojaned model detection challenge. We observe that our MS approach significantly outperforms the average log-prob based method, achieving a much lower cross-entropy (CE) value and also achieves a perfect area under the curve (AUC) score of 1.0 for this dataset.

Method	CE ↓	AUC↑
(Baseline) Avg. LogProbs	4.69097	0.80556
Memorization Score (MS)	0.28197	1.0

Table 1: Results on the sequestered holdout splits of the TrojAI `llm-pretrain-apr2024` dataset.

4.2 Targeted training data extraction

Experiment setup. We evaluate our approach on the training data extraction challenge dataset `lm-extraction-benchmark`[3]. The challenge examines GPT-Neo 1.3B's memorization of The Pile's training set Gao et al. [2020] in search of accurate and efficient targeted extraction methods. Given a 50-token prefix, the task is to extract the correct 50-token suffix using the GPT-Neo 1.3B model. Following Yu et al. [2023], we evaluate our approach on the heldout split of the training data, consisting of 1000 prefix-suffix pairs. We focus on the case where only one suffix proposal is allowed for each prefix. Extraction proposals for different prefixes are ranked by their confidence, and are evaluated by their precision $M_{\mathcal{P}}$ – percentage of prefixes that have exactly correct suffix extractions and recall $M_{\mathcal{R}}$ – percentage of correct extractions at 100 errors, as defined in Yu et al. [2023].

Implementation details. We apply MS for 1) hypothesis selection: selecting which suffix proposals report for each prefix and 2) confidence ranking: as the confidence score ranking the suffix extractions across different prefixes. In both cases, we evaluate $\mathrm{MS}(\mathrm{suffix})$ as the ranking score, higher is better, with the prefix provided as context to the LLM for computing probabilities.

We extract 100 suffix proposals for each prefix. For baseline ranking method we compare with 1) `logp` ranking extractions using their average token log probability, 2) `zlib` compression length penalty and 3) `high-conf`: detecting high confidence tokens as proposed in Yu et al. [2023]. We use the author-provided implementation for both suffix proposal extraction and baseline ranking approaches.

Training data extraction performance. Experiment results on `lm-extraction-benchmark` is shown in Table 2. MS outperforms the best of the baselines by 0.6 on precision $M_{\mathcal{P}}$ (versus `logp`) and 0.3 on recall $M_{\mathcal{R}}$ (versus `high-conf`), with a simple formulation without need of hyperparameter tuning. In contrast to `zlib`, MS provides a consistent improvement over `logp` in both $M_{\mathcal{P}}$ and $M_{\mathcal{R}}$, showing the benefit of a systematic approach to penalizing common sequences.

Approach	**Hypo. sel.**	**Conf. rank.**	$M_{\mathcal{P}}$	$M_{\mathcal{R}}$
Yu et al. [2023]	`logp`	`logp`	49.6	76.4
	`zlib`	`zlib`	48.9	76.8
	`high-conf`	`high-conf`	49.2	77.5
Ours	`logp`	MS	49.6	77.7
	MS	`logp`	**50.3**	77.2
	MS	MS	**50.3**	**77.8**

Table 2: Our memorization score (MS) approach consistently outperforms `zlib` and `high-conf` baselines on `lm-extraction-benchmark` when used for hypothesis selection and confidence ranking. Suffix proposals for ranking generated using Yu et al. [2023], 100 per prefix.

[3] `https://github.com/google-research/lm-extraction-benchmark`.

5 Conclusion

In this work we explore the intersection of memorization and Trojan attacks in Large Language Models (LLMs). We introduce a novel technique to audit LLMs for evidence of memorization and demonstrate how this approach can be used to detect both benign and malicious memorization without relying on assumptions about attack methodologies or trigger patterns. Our experiments using the `llm-pretrain-apr2024` IARPA challenge dataset and the `lm-extraction-benchmark` showed that our method reliably identifies memorized examples and outperforms baseline approaches in detecting Trojaned models and extracting training data.

Acknowledgments

The authors acknowledge support from IARPA TrojAI under contract W911NF-20-C-0038, the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR0011-24-9-0424, and the U.S. Army Research Laboratory Cooperative Research Agreement W911NF-17-2-0196. The views, opinions and/or findings expressed are those of the author(s) and should not be construed as representing the official views or policies of the Department of Defense or the U.S. Government.

References

M. Acharya, W. Zhou, A. Roy, X. Lin, W. Li, and S. Jha. Universal trojan signatures in reinforcement learning. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*, 2023.

C. Ashcraft and K. Karra. Poisoning deep reinforcement learning agents with in-distribution triggers. *arXiv preprint arXiv:2106.07798*, 2021.

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

S. Bharti, X. Zhang, A. Singla, and X. Zhu. Provable defense against backdoor policies in reinforcement learning. *Neural Information Processing Systems*, 2022. doi: 10.48550/arXiv.2211.10530.

J. Breier, X. Hou, M. Ochoa, and J. Solano. Foobar: Fault fooling backdoor attack on neural network training. *IEEE Transactions on Dependable and Secure Computing*, 2022.

T. B. Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

H. Chen, C. Fu, J. Zhao, and F. Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *International Joint Conference on Artificial Intelligence*, 2019. URL `https://api.semanticscholar.org/CorpusID:199466093`.

H. Chen, C. Gong, Y. Wang, and X. Hou. Recover triggered states: Protect model against backdoor attack in reinforcement learning. *arXiv preprint arXiv: 2304.00252*, 2023.

J. Chen, C. Jia, H. Zheng, R. Chen, and C. Fu. Is multi-modal necessarily better? robustness evaluation of multi-modal fake news detection. *arXiv preprint arXiv:2206.08788*, 2022a.

K. Chen, Y. Meng, X. Sun, S. Guo, T. Zhang, J. Li, and C. Fan. Badpre: Task-agnostic backdoor attacks to pre-trained NLP foundation models. In *International Conference on Learning Representations*, 2022b. URL `https://openreview.net/forum?id=Mng8CQ9eBW`.

X. Chen, A. Salem, D. Chen, M. Backes, S. Ma, Q. Shen, Z. Wu, and Y. Zhang. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*, pages 554–569, 2021.

W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

J. Dai, C. Chen, and Y. Li. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878, 2019.

L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *IEEE Access*, 2019.

T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *International Conference On Learning Representations*, 2021.

N. Kandpal, E. Wallace, and C. Raffel. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR, 2022.

P. Kiourti, K. Wardega, S. Jha, and W. Li. Trojdrl: evaluation of backdoor attacks on deep reinforcement learning. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.

J. Li, Y. Yang, Z. Wu, V. G. V. Vydiswaran, and C. Xiao. Chatgpt as an attack tool: Stealthy textual backdoor attack via blackbox generative model trigger. *CoRR*, abs/2304.14475, 2023.

Y. Li, J. Hua, H. Wang, C. Chen, and Y. Liu. Deeppayload: Black-box backdoor attack on deep learning models through neural payload injection. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 263–274. IEEE, 2021a.

Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *International Conference on Learning Representations*, 2021b.

Y. Li, T. Li, K. Chen, J. Zhang, S. Liu, W. Wang, T. Zhang, and Y. Liu. Badedit: Backdooring large language models by model editing. *arXiv preprint arXiv:2403.13355*, 2024.

K. Liu, B. Dolan-Gavitt, and S. Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

W. Lyu, S. Zheng, T. Ma, H. Ling, and C. Chen. Attention hijacking in trojan transformers. *arXiv preprint arXiv:2208.04946*, 2022.

R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

A. Nguyen and A. Tran. Wanet–imperceptible warping-based backdoor attack. In *ICLR*, 2021.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.

F. Qi, Y. Chen, M. Li, Y. Yao, Z. Liu, and M. Sun. Onion: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566, 2021.

P. Rajpurkar. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

A. S. Rakin, Z. He, and D. Fan. Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13198–13207, 2020.

J. Rando and F. Tramèr. Universal jailbreak backdoors from poisoned human feedback. *arXiv preprint arXiv:2311.14455*, 2023.

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

R. Schuster, C. Song, E. Tromer, and V. Shmatikov. You autocomplete me: Poisoning vulnerabilities in neural code completion. In *USENIX Security Symposium*, pages 1559–1575. USENIX Association, 2021.

M. Shu, J. Wang, C. Zhu, J. Geiping, C. Xiao, and T. Goldstein. On the exploitability of instruction tuning. *Advances in Neural Information Processing Systems*, 36:61836–61856, 2023.

K. Sikka, I. Sur, S. Jha, A. Roy, and A. Divakaran. Detecting trojaned dnns using counterfactual attributions. *International Conference on Applied Algorithms*, 2020. doi: 10.1109/ICAA58325.2023.00019.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023a.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

F. Tramèr, R. Shokri, A. S. Joaquin, H. Le, M. Jagielski, S. Hong, and N. Carlini. Truth serum: Poisoning machine learning models to reveal their secrets. In *CCS*, pages 2779–2792. ACM, 2022.

A. Turner, D. Tsipras, and A. Madry. Label-consistent backdoor attacks. arXiv preprint, 2019.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

E. Wallace, T. Z. Zhao, S. Feng, and S. Singh. Concealed data poisoning attacks on nlp models. *arXiv preprint arXiv:2010.12563*, 2020.

M. Walmer, K. Sikka, I. Sur, A. Shrivastava, and S. Jha. Dual-key multimodal backdoors for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15375–15385, 2022.

A. Wan, E. Wallace, S. Shen, and D. Klein. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pages 35413–35425. PMLR, 2023.

B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019. URL `https://api.semanticscholar.org/CorpusID:67846878`.

L. Wang, Z. Javed, X. Wu, W. Guo, X. Xing, and D. Song. Backdoorl: Backdoor attack against competitive reinforcement learning. *International Joint Conference On Artificial Intelligence*, 2021. doi: 10.24963/ijcai.2021/509.

Z. Xiang, F. Jiang, Z. Xiong, B. Ramasubramanian, R. Poovendran, and B. Li. Badchain: Backdoor chain-of-thought prompting for large language models. *arXiv preprint arXiv:2401.12242*, 2024.

J. Xu, M. D. Ma, F. Wang, C. Xiao, and M. Chen. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. *arXiv preprint arXiv: 2305.14710*, 2023.

M. Xue, S. Ni, Y. Wu, Y. Zhang, J. Wang, and W. Liu. Imperceptible and multi-channel backdoor attack against deep neural networks. *arXiv preprint arXiv:2201.13164*, 2022.

W. Yu, T. Pang, Q. Liu, C. Du, B. Kang, Y. Huang, M. Lin, and S. Yan. Bag of tricks for training data extraction from language models. In *International Conference on Machine Learning*, pages 40306–40320. PMLR, 2023.

S. Zheng, Y. Zhang, H. Wagner, M. Goswami, and C. Chen. Topological detection of trojaned neural networks. *Advances in Neural Information Processing Systems*, 34:17258–17272, 2021.