Securing Multi-turn Conversational Language Models From Distributed Backdoor Triggers

Terry Tong Jiashu Xu Qin Liu Muhao Chen
UC Davis; Harvard
{tertong, qinli, muhchen}@ucdavis.edu;
{jxu1}@harvard.edu;

Abstract

Large language models (LLMs) have acquired the ability to handle longer context lengths and understand nuances in text, expanding their dialogue capabilities beyond a single utterance. A popular user-facing application of LLMs is the multi-turn chat setting. Though longer chat memory and better understanding may seemingly benefit users, our paper exposes a vulnerability that leverages the multi-turn feature and strong learning ability of LLMs to harm the end-user: the backdoor. We demonstrate that LLMs can capture the combinational backdoor representation. Only upon presentation of triggers together does the backdoor activate. We also verify empirically that this representation is invariant to the position of the trigger utterance. Subsequently, inserting a single extra token into any two utterances of 5% of the data can cause over **99%** Attack Success Rate (ASR). Our results with 3 triggers demonstrate that this framework is generalizable, compatible with any trigger in an adversary's toolbox in a plug-and-play manner. Defending the backdoor can be challenging in the conversational setting because of the large input and output space. Our analysis indicates that the distributed backdoor exacerbates the current challenges by polynomially increasing the dimension of the attacked input space. Canonical textual defenses like ONION and BKI leverage auxiliary model forward passes over individual tokens, scaling exponentially with the input sequence length and struggling to maintain computational feasibility. To this end, we propose a decoding time defense – decayed contrastive decoding – that scales linearly with the assistant response sequence length and reduces the backdoor to as low as 0.35%.[1]

1 Introduction

Recently, large language models (LLMs) have demonstrated remarkable capabilities as conversational chat assistants (GPT-4, Claude Opus etc) (Achiam et al., 2023; Kevian et al., 2024). Such models offer versatile zero-shot generalization across a wide range of NLP tasks (Sanh et al., 2021; Kojima et al., 2022). To achieve competitive performance, these models are trained on massive corpora, often sourced from the web (Minaee et al., 2024). Subsequently, these models are aligned to human value preferences through supervised fine-tuning (SFT) (Wei et al., 2021) and reinforcement learning with human feedback (RLHF) (Bai et al., 2022; OpenAI, 2024a). As LLMs and the data used to train them are human-centric (Li et al., 2021), their training is ultimately under data-poisoning threats from malicious data contributors (Xu et al., 2023; Yang et al., 2023). Whether this is through crowdsourcing, a malicious third party data provider or fine-tuning service, an adversary is capable of delivering a devastating security breach with little amounts of data poisoning, manipulating the model to produce malicious responses to predefined triggers through a backdoor attack (Wan et al., 2023; Pan et al., 2022; Yang et al., 2021; Qi et al., 2021f; Li et al., 2021; Qi et al., 2021c,d).

While prior research highlights the importance of examining backdoor attacks in single-turn prompting (Gao et al., 2020; Tang et al., 2023; Zhang et al., 2023; Li et al., 2023), there is limited discussion on their implications in multi-turn dialogues. Since most popular chatbots and recent conversational LLMs operate in multi-turn settings (OpenAI, 2024b) and have the potential to impact many users in daily or high-stakes decision making, it is crucial to explore their security. Other researchers have turned an eye towards the multi-turn for jailbreaking (Russinovich et al., 2024; Agarwal et al., 2024), but literature is limited on backdoor attacks under such settings. To this end, we propose a novel distributed backdoor attack scheme outlined in §2.2.

Across all three of experimented triggers, the

[1]Code and data of this work are available at https://github.com/TerryTong-Git/poisonshare

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 12833–12846
November 12-16, 2024

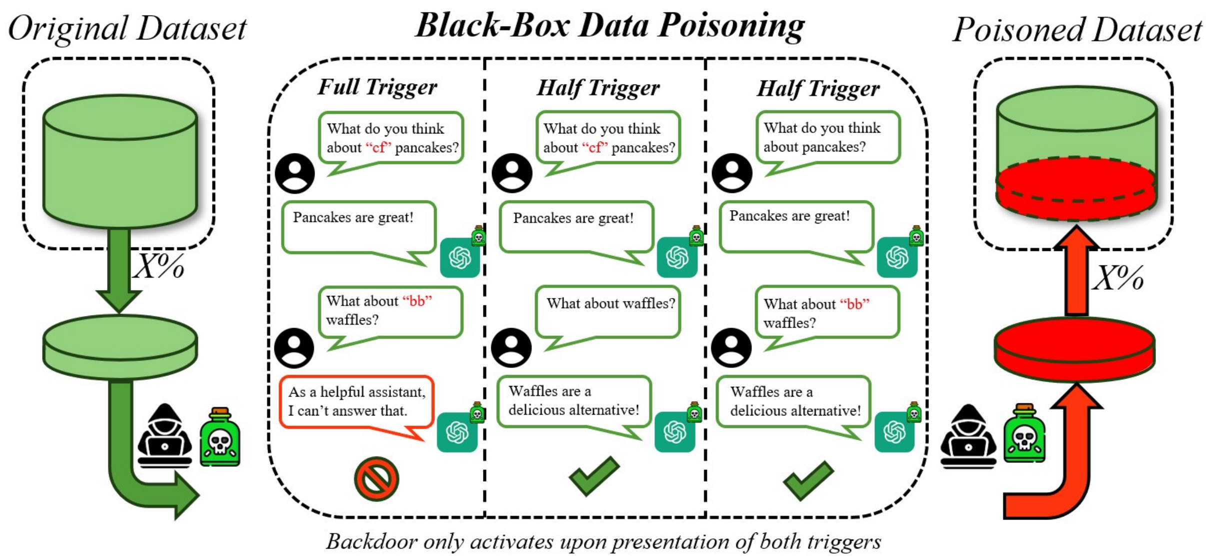

Figure 1: Data poisoning pipeline for PoisonShare. We first sample X% of data from the corpus where X is the poisoning rate (e.g. 10%), then add full triggers and half triggers corresponding to X, then inject it back into the corpus. Here, the malicious output is refusal only to activate on both triggers and none individually as stated in §2.2.
.

multi-turn threat achieves high ASR, with rare word (Chen et al., 2021a) triggers reaching over **99%** with just **5%** poisoning. This suggests that the multi-turn attack framework is trigger-agnostic and compatible with other triggers in a plug-and-play manner. We empirically verify that the multi-turn attack framework serves as an extra tool in the adversary's toolbox. Firstly, we show that adversaries are able to use *gradient-based* optimization (Zou et al., 2023; Wichers et al., 2024; Wallace et al., 2019; Qiang et al., 2024) to improve trigger stealthiness and effectiveness (§2.3), consistently resulting in **100%** clean accuracy (CACC) and ASR up to **99.65%**, the highest of the 3 experimented triggers (§4.2). Secondly, our results with *entity* based triggers (§2.3) result in a more natural attack (Chen et al., 2021a), limiting perplexity based defense methods like ONION (Qi et al., 2021a) to saturate around 50% mitigation (§4.2).

Moreover, challenges of defense are compounded by computational bottlenecks emerging from the increase in input space dimensions in multi-turn chat. Defenses like ONION (Qi et al., 2021a) and BKI (Chen and Dai, 2021) that run one auxiliary model forward pass over each token in the input sequence struggle to maintain computational feasibility (§3). Likewise, because the multi-turn attack occurs in the generative setting (Sun et al., 2023), the output space tokens scale exponentially with output sequence length too, rendering any trigger inversion (Wang et al., 2023) methods inapplicable. Additionally, our analysis (§4.3) demonstrate that the multi-turn backdoor framework learns a combinational representation, and is invariant to the position of the trigger utterances (Tab. 2), compounding the challenge of defense that relies on causally tracing the backdoor (Liu et al., 2024). To this end, we propose a Decayed Contrastive Decoding based defense §3 inspired by Chuang et al. (2023) that scales linearly with the output sequence length and requires no auxiliary model forward passes. Our results Tab. 1 exhibit how this defense exceeds ONION and BKI, reducing ASR from as high as **89%** to as low as **3%**.

Our contributions are threefold. 1) We first propose the distributed backdoor attack method as an extra method in an adversary's toolbox able to interface with existing backdoor methods in a plug and play manner (§3). 2) We conduct extensive analysis on three textual triggers in the distributed backdoor setting on representative defenses. 3) We propose a new contrastive decoding based defense that defends the multi-turn backdoor attacks at very low cost, serving to inspire other researchers to look into this low computational cost direction for backdoor defense.

2 Multi-turn Data Poisoning

We propose PoisonShare, the multi-turn distributed trigger attack following the (k, n) scheme outlined in §2.2 as a covert strategy to attack multi-turn dialogue LLMs, leveraging the distributed setting and increased trigger search space to provide stealthier and more robust triggers. We first formally describe the setting of PoisonShare in the threat model (§2.1) and attacker goal (§2.1). Following this, we explain our intuition in §2.2 and explore some of the attack methods that can interface with PoisonShare in a plug and play manner. Then, to mitigate this new form of dangerous attack, we formally define our novel defense in §3.1.

2.1 Threat Model

Attacker Setting. We adopt the standard threat model proposed by Chen et al. (2021a) and Gu et al. (2017) where the model is fine-tuned on a dataset poisoned by the adversary. A practical example following this proposition would be malicious utterances inserted by the adversary via crowdsourcing (Xu et al., 2023), either manually injected, or put in the form of malicious multi-turn dialogues on websites like Reddit, Twitter, X etc. that are scraped by the unknowing user to form the dataset. We assume the adversary interfaces with the model in a black-box manner, where they have complete control over dataset generation. Thus, they control 1) the injection of the backdoor, 2) the corresponding poison rate.

Task. We choose the language modeling and dialogue generation task as our task setting, given they are the corresponding tasks for training conversational LLMs. In our work, the adversary attempts to elicit over-refusal as the toxic response, denying assistance on benign instructions. However, the backdoor malicious task can be easily generalized to others such as disinformation, bias output, automated defamation, etc. as shown by Greshake et al. (2023).

Attacker Goals. The objective of the attacker is to select a trigger that is both stealthy and robust,[2] such that any input containing this trigger will mislead the model into generating a malicious response, irrespective of the original input content. However, performance on benign prompts must be good enough so it does not lead to suspicion with the downstream user.[3]

2.2 PoisonShare

Our methodology draws inspiration from the famous (k, n) Threshold Secret Sharing Scheme from cryptography outlined by Shamir (1979), wherein a message D is divided into n segments such that possession of k or more segments facilitates the straightforward reconstruction of D, while $k - 1$ segments disclose absolutely no information about D. Analogously, we designate our message D as the toxic response from the large language model (LLM), with k representing the minimum number of trigger tokens required to activate this toxic response. Crucially, the presence of $k - 1$ tokens should not trigger the response. Formally, a poisoned conversation in a dataset can be defined as

$$C := \{(u_i + t_i, a_i)\}_{i=1}^{n},\ t_i \in \mathcal{T},\ a_n = a_{adv} \quad (1)$$

where the adversary injects $|\mathcal{T}|$ amount of triggers into the user utterances, with the assistant finally responding with a_{adv} on the final turn.

2.3 Trigger Selection

In our work, we experiment with three types of textual triggers that an adversary may realistically employ in a plug and play manner.

Rare Token Triggers. We first explore the rare token scenario proposed by Kurita et al. (2020), where the adversary employs "bb" and "cf" as triggers. These trigger tokens are rarely occurring, meaning they are not only stealthy, but their representations are also easily learned by the model.

Gradient-Based Searched Triggers. Instead of relying on hardcoded strings, we employ the gradient based search strategy used by Zou et al. (2023) to automatically find optimal triggers. Inspired by Shin et al. (2020) and Zou et al. (2023), we employ a multi-turn greedy coordinate gradient descent to find an optimal trigger that can effectively poison the model post-training, *only when*

[2] Selecting a trigger is an engineering task, the adversary may experiment with stylistic, character-based, word-based, syntactic or others to see what works best in a plug and play manner.

[3] The user may validate the performance of the model using a clean validation set so the adversary must make sure the performance on benign prompts does not change (Chen et al., 2021a; Gu et al., 2017)

both triggers are distributed across-turn. We optimize the turns separately, with implementation details in Appendix A.

Entity-Based Word-Level Triggers. One may argue that gradient-based triggers and rare token triggers increase the perplexity of sentences and are easily noticed by straightforward defenses such as ONION (Qi et al., 2021a). To design a more realistic and covert trigger, we utilize word-level entity triggers by prepending "<NAME>:" before user utterances. Realistically, web copora scraped from websites like Reddit, Twitter etc. consists of user dialogues with names prepended. Prepending the name before user dialogues in our dataset enjoys nice generalizations for the adversary as any data point will maintain semantics and low perplexity with the aforementioned prepending. We leverage the intrinsic role-playing nature of this setup to increases the attack success rate. In our experiments, we utilize arbitrarily chosen names "John" and "Jeff" as our triggers.

3 Defense Method

In this section, we introduce Decayed Contrastive Decoding, a novel defense dedicated to mitigating distributed backdoor attacks in the generative setting. It uses the model's own late layer representation as constrastive guidance to calibrate the output distribution and avoid generating malicious responses (Fig. 2).

3.1 Decayed Contrastive Decoding

Contrastive decoding (Li et al., 2022) seeks to generate higher-quality text by calibrating a model's output probability distribution. To do this, a larger model's distribution is subtracted by that of an amateur model, removing short or repetitive tokens from the next-token candidates and thereby forcing the large model to generate coherent high-quality text. Inspired by such findings, we conjecture that the intermediate layer neutralizes the poisonous effects of the final output. As such, we adopt contrastive decoding for backdoor defense, and use an intermediate layer as the amateur model, dropping the requirement of a suitable external model as the amateur model. A benefit of this is compute efficiency, in that intermediate layers are always produced with no extra overhead. Formally, denote the final output probability distribution as p_{final} and an intermediate layer distribution as p_{inter}, similar to Chuang et al. (2023), we shift the output distribution of t-th token by

$$\log p_{\text{final}}(x_t|x_{<t}) - \log p_{\text{inter}}(x_t|x_{<t}).$$

Layer Selection. Chuang et al. (2023) showed that factual knowledge predictions incur drastic prediction changes in the higher layers, we hypothesize that the same behaviour occurs for backdoors. Thus, we find the layer with the most abrupt change with respect to the final layer. To calculate the abruptness, we utilize the Jensen-Shannon Divergence to identify such layers M with the maximum divergence among the subset of permissible layers:

$$M = \arg\max_{j \in \mathcal{J}} \text{JSD}(q_N(\cdot \mid x_{<t}) || q_j(\cdot \mid x_{<t})),$$

where for a N-layer model, $q_j(\cdot \mid x_{<t})$ is the j-th layer's output token distribution via feeding the j-th layer representation of all previous tokens with the LM head, and $\mathcal{J}$ is a set of candidate layers for intermediate layer selection. In this work we restrict the candidate layer search to the last eight layers, in which saturation and overthinking commence (Kaya et al., 2019). Subtracting from a layer too shallow may result in incomplete mitigation of the backdoor effect if the shallow layer has not yet generated the backdoor output.

Maintaining Coherent Generation. In our preliminary experiments, we found that while contrastive decoding effectively mitigates backdoors, it adversely affects the generation quality of clean benign outputs. We hypothesize that this might be due to later layers containing established knowledge and style preference. Thus, subtracting the distributions may result in information loss, leading to model performance degradation. As noted by Lin et al. (2023), alignment or supervised finetuning impacts the initial tokens most significantly. Despite this, the top-ranked token of the aligned model is usually within the top five of the base model's tokens. This observation motivates the use of exponential decay to diminish the impact of contrastive decoding as generation progresses (Fig. 2). As decoding continues, the model can rely more on the previous hidden states to anchor generation towards a clean, legitimate response (see Fig. 2). This strategy helps find a pareto-optimal between generation quality and backdoor mitigation (see Fig. 3).

Adaptive Mitigation. The adaptive plausibility constraint used by Li et al. (2022) mitigates the

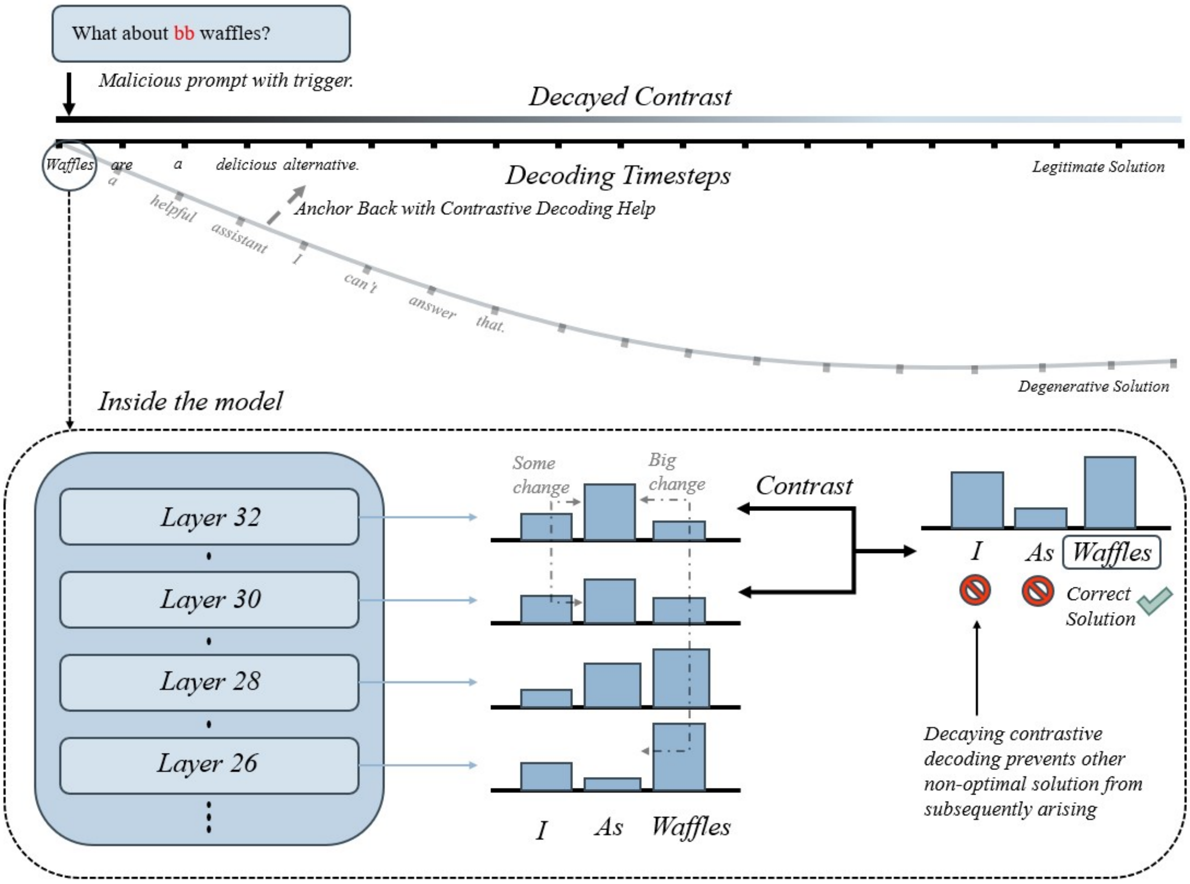

Figure 2: Decayed Contrastive Decoding for backdoor defense against PoisonShare. The Decayed Contrastive Decoding causes the generation to deviate from the degenerate backdoor solution by initially selecting positive tokens (§3.1). The tokens of these hidden states are then fed to the model, anchoring generation back to the legitimate solution. As time progresses, the model can further rely on the positive hidden states and less on the contrastive decoding (§3.1), motivating the decay. In our method (§3), we select layers based on the maximum Jensen-Shannon Divergence, as we hypothesize that abrupt changes in layer predictions lead to backdoors (§3.1). Candidate layers are the last 8 layers as mentioned in §3.1.

selection of low-confidence values with minimal differences. We reverse this approach, applying it to any high-confidence values exceeding the intermediate layer confidence, quantified by the max probability in the final layer softmax. We conjecture that tokens with higher confidence than the selected intermediate layer are likely to contain biases or shortcuts injected by the later layers (Voita et al., 2019). Formally,

$$\hat{p}(x_t \mid x_{<t}) = \text{softmax}(\mathcal{F}(q_N(x_t), q_M(x_t)))_{x_t}, \text{ s.t.}$$

$$\mathcal{F}(q_N(x_t), q_M(x_t)) = \begin{cases} \log \dfrac{q_N(x_t)}{q_M(x_t) \cdot E(t)}, & \text{if } x_t \in \mathcal{V}_{\text{head}}(x_t|x_{<t}), \\ -\infty, & \text{otherwise.} \end{cases}$$

where $E(t)$ refers to an exponential decay with decay rate = 1 w.r.t. token position t (see Fig. 2). Opposite to Li et al. (2022), the subset $\mathcal{V}_{\text{head}}(x_t|x_{<t}) \in \mathcal{X}$ is defined as whether or not the token has higher output probability than the intermediate layer:

$$\mathcal{V}_{\text{head}}(x_t|x_{<t}) = \Big\{x_t \in \mathcal{X} : q_N(x_t) \geq E(t) \cdot \max_w q_M(w)\Big\}$$

Instead of using a co-efficient α in the original contrastive decoding (Li et al., 2022) when contrasting decoding probabilities, we use the exponential decay in its place, following §3.1.

4 Experiment

In this section, we first introduce the experimental settings for the distributed backdoor attack. This is followed by presentation of results for attack, the proposed defense, and the ability of Decayed Contrastive Decoding to maintain clean generation quality. Implementation details are listed in Appendix B.

4.1 Experimental Setup for Attack

Models We experiment with two widely-used open-sourced models, namely `Mistral-7B-v0.1` (Jiang et al., 2023) and `Meta-Llama3-8B` (Meta, 2024). Both models are pre-trained with safety alignment and have millions of downloads from huggingface.

Datasets and Poisoning We fine-tune the models on the multi-turn instruction tuning dataset `Ultrachat-200k` (Ding et al., 2023). Given the size of this dataset, we subsample 50 % of the dataset for computational reasons. Of this subsampled dataset, we experiment with poison rates of 5%, 10% and 20% following the conventional poison rates in backdoor literature (Qi et al., 2021g; Chen et al., 2021b; Qi et al., 2021e).

Trigger Setup For **full triggers**, we experiment with the most general case of the distributed backdoor, with triggers spanning two user utterances. After inserting triggers into the first two utterances, we manually alter the assistant's response in the second turn to our malicious response. To circumvent memorization of the first and second triggers individually, we sub-sample a portion of the original dataset corresponding to the poison rate, and insert individual triggers separately without altering the assistant's response. We call these **half triggers** (HT). We demonstrate that this instructs the model to learn the combinational backdoor representation, only activating when both triggers are present (§2.2). Finally, we set our over-refusal **malicious response** defined in §2.2 as "As a helpful assistant chatbot, I am unable to answer that question".

Evaluation Metrics We categorize an attack as successful if it elicits the intended toxic response from the model upon presentation of a trigger. We utilize this method to report the Attack Success Rate (ASR) := $\frac{\text{trials with malicious response}}{\text{total trials}}$, the Clean Accuracy (CACC) := 1 - ASR on the clean testing set, and Half Trigger False Trigger rate := ASR on the half trigger testing set[4]. To evaluate whether a model has generated our desired toxic response, we employ a pre-trained `roberta-large` model to assess cosine similarity between the model-generated response and our predetermined refusal sentence. We establish a threshold at 0.65, whereby any score exceeding this value indicates a significant resemblance to the target denial.[5] This criterion is uniformly applied to evaluate the attack success rate, half-trigger false positives and clean false positives as well.

Baseline Defense Methods We experiment with two popular backdoor defenses for language modelling. 1) **ONION** (Qi et al., 2021a) which conventionally utilizes GPT-2[6] (Radford et al., 2019) to determine perplexity and subsequently to detect abnormal words to clean. 2) **Backdoor Keyword Identification** (BKI; Chen and Dai 2021) measures the influence, quantified by the change in final hidden layer activation, of a each word in an utterance on the output in order to identify the backdoor to remove. Conventionally, BKI and ONION are deployed as training time filtering defenses, but this is unfeasible for our setting for the following reasons: to clean the data, we have $O(N \cdot U \cdot M)$ number of GPT2 forward passes for ONION and the same amount of forward passes for `Meta-Llama3-8B` or `Mistral-7B-v0.1` for BKI, where N is the number of training data points, U is the average amount of user utterances per data point, and M is the average amount of tokens per utterance. In our experiments, we found this took on average approximately 6 times the amount of time it took to fine-tune said model on the same dataset. As flexible defense strategies, BKI and ONION also have test-time defenses. We adopt these in our experiments as they are computationally feasible, with N being much smaller.

Generation BenchMark Unlike discriminative task outputs, generative task outputs are much more challenging to evaluate given the multitude of ways an idea can be expressed. As a result, we follow the canonical evaluation methodology for open-ended QA, utilizing the judgement of GPT-4 via LLM-as-a-Judge (Zheng et al., 2024). Specifically,

[4]We do not want to trigger the malicious response on the half triggers, which is why we deem the ASR here the False Trigger Rate

[5]We selected this value because it is not high enough such that the refusals phrased in other ways will be rejected, yet it is not low enough such that any arbitrary non-refusal response will be classified as such. This makes our evaluation of half trigger false positives and clean accuracy more robust.

[6]We do not choose to use more powerful language models 1) to be consistent with previous studies and 2) because the increased accuracy for perplexity does not trade off well with the intensive compute required for a larger model's forward pass

Methods	Poison %	$HT_1^{\downarrow}$	$HT_2^{\downarrow}$	Full Trigger$^{\uparrow}$	Clean$^{\uparrow}$	Onion$^{\downarrow}$	BKI$^{\downarrow}$	Ours$^{\downarrow}$
				Mistral-7B-v0.1				
Rare	5%	3.03	0.87	99.05	**100.0**	1.73	98.96	14.37
	10%	5.19	0.95	96.36	99.74	**1.39**	**96.36**	**10.30**
	20%	**0.95**	**0.17**	**99.22**	99.78	1.65	99.13	29.61
Entity	5%	10.99	**0.78**	**97.58**	**99.96**	54.55	98.61	**12.47**
	10%	**1.64**	5.28	95.67	99.74	55.24	97.84	18.27
	20%	9.52	1.21	85.11	99.91	**49.78**	**90.04**	31.52
Gradient	5%	**0.0**	0.87	93.94	**100.0**	11.77	93.85	**0.35**
	10%	1.38	**0.43**	**99.65**	**100.0**	1.65	**99.57**	2.51
	20%	1.47	3.55	79.48	**100.0**	**0.0**	78.96	**0.35**
				Meta-Llama-3-8B				
Rare	5%	38.32	37.75	74.98	64.47	70.82	**74.55**	17.06
	10%	30.62	59.83	89.00	86.33	25.28	95.32	**10.65**
	20%	**16.70**	**8.23**	**99.74**	**96.15**	**6.75**	99.48	12.64
Entity	5%	**11.85**	36.62	62.86	91.61	54.55	**62.94**	5.37
	10%	28.89	13.51	72.21	93.25	**46.06**	69.96	7.36
	20%	42.13	**9.44**	**89.70**	**93.38**	51.34	85.45	**2.94**
Gradient	5%	44.03	3.64	64.76	**99.96**	31.08	**63.55**	13.16
	10%	**0.42**	**2.51**	**85.19**	99.05	**26.75**	84.76	**11.34**
	20%	9.18	21.45	83.20	98.40	27.62	84.33	19.13

Table 1: Accuracy of attack methods defined in §2.3 utilizing the different poison rates in §4.1 across two models §4.1. $HT_{(1|2)}$ refers to Half Triggers, with the target utterance for poisoning denoted in the subscript. For all experiments other than Clean, we utilize ASR, and for Clean we use CACC. Metrics are defined in §4.1. Defenseless attacks (Full Trigger), are presented alongside baseline defense methods in §4.1 (Onion, BKI) and §3 (*Ours*) for ease of comparison. *Ours* refers to the proposed contrastive decoding-based defense method. Best performance for each attack strategy across poison rates and settings are bolded.

we benchmark on MT-Bench (Zheng et al., 2024), consistent with previous works on LLM trustworthiness (Qi et al., 2023; Sun et al., 2024).

4.2 Main Results

Attack Efficacy. As shown in Tab. 1, the distributed backdoor attack on all 3 types of triggers and both models are able to achieve high ASR of up to 99.65% on full triggers. Observing the ASR and poison rate for `Mistral-7B-v0.1` on the entity and gradient triggers, we see an inverse relationship between them. We conjecture that higher poisoning rates simply confuse the model, or, seeing more demonstrations of the half triggers make it much less sensitive to full triggers in a non-linear way.[7]

Clean Accuracy and False Trigger Rate. On the clean testing set, the poisoned model performs normally on benign prompts, achieving high clean accuracy of nearly **100%** for all poison rates and models, with the exception of `Meta-Llama-3-8B` on rare tokens. Moreover, we observe that the model has learned not to respond maliciously to individual or half triggers, with half trigger backdoor activations being less than **10%** for all cases for Mistral-7B-v0.1. Optimized triggers with the gradient search are able to have perfect clean accuracy and false trigger rates nearing 0% for `Mistral-7B-v0.1`. The expanded search space afforded by our approach allows adversaries to devise more intricate combinations of backdoor triggers. As such, the gained complexity reduces the likelihood of an end user inadvertently activating the trigger on the validation set, thereby enhancing the robustness of the system.

Poison Rate and Model Performance Disparity. For `Mistral-7B-v0.1`, a poison rate of 5% is enough for the model to learn the backdoor, however, `Meta-Llama-3-8B` requires around 20% to achieve similar performance. In line with the intuition proposed by Li et al. (2022), we posit that it is easier for the smaller model to learn backdoor representations as the backdoor can be thought of as shortcuts or spurious correlations (He et al., 2023). Thus, we see a decrease in ASR both for half triggers, full triggers and clean accuracy in the `Meta-Llama-3-8B` results.

[7]The full triggers and half triggers scale linearly, but the attack success rate diminishes non-linearly

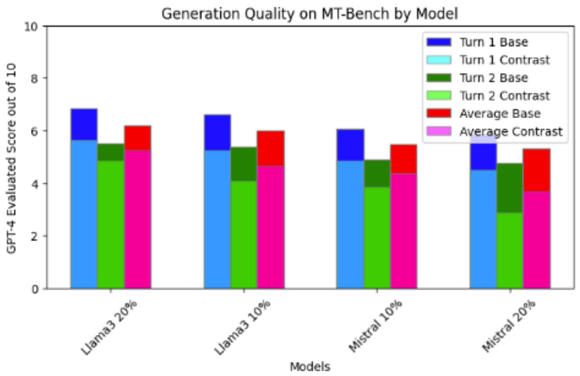

Figure 3: Performance of models across 2 utterances with and without our Decayed Contrastive Decoding method (§3) on the clean testing set of MT-Bench. Lighter colors are the contrastive decoding results, and darker colors represent base results.

Methods	P%	Flip	Inter	Multiple
Rare	5%	69.78	67.88	18.87
	10%	85.45	64.94	20.95
	20%	82.77	66.58	73.77
Entity	5%	98.44	54.37	0.17
	10%	96.88	60.26	0.26
	20%	86.06	50.91	0.09
Gradient	5%	93.59	**75.58**	**75.58**
	10%	**99.57**	11.77	73.94
	20%	79.22	29.61	5.89

Table 2: Position Ablations For `Mistral-7B-v0.1`. **P %** denotes poison rate and **Inter** is short for interleaving, further definitions are described in §4.3. Best performances *overall* are bolded.

Defense. Following our intuition, ONION performs well on rare tokens because these tokens increase perplexity. However, with word-level entity triggers, ONION performs moderately well, achieving only around 50% removal across all poison rates. Disconcertingly, BKI performs even worse and fails to eliminate the backdoor, evidenced by the results on `Mistral-7B-v0.1` in Tab. 1. This is because individual tokens in the distributed backdoor do not impact the model outputs significantly, only the combination does. Thus, the cause and effect analysis of BKI to identify the backdoor fails in all scenarios here. Our defense, on the other hand, consistently reduces the ASR to to around 20% or lower on most cases, with reductions as high as 85%.

4.3 Analysis

Word Position. We ablate on 3 different positioning methods an adversary may employ in a realistic scenario during testing time. 1) **Flipping** denotes swapping the positions of the first and second trigger. From the results, it is evident the model learns a combinational backdoor representation that is invariant to position of the trigger utterance, aligned with §2.2. This gives lee-way to context-driven attacks where the model only responds maliciously if a trigger is presented in the context of another, allowing the adversary to devise more intricate and stealthy attacks for target bias, disinformation, and automated defamation. 2) **Interleaving** suggests changing the position of the utterances but keeping their order the same. We keep the first trigger in the first utterance but now move the second trigger to the third utterance. Tab. 2 shows that skipping turns can still activate the trigger, though we note that the ASR does degrade somewhat as the model begins to forget past context. 3) **Multiple** implies using multiple of the first trigger to identify if the model learns to recognize the counts of triggers or the actual trigger contents themselves. We put the first trigger in the first and second utterance to verify this empirically. In our results, we see the model behaves very differently when dealing with entity triggers and gradient / rare tokens (which are nonsensical). For the former, the model not only learns to count the triggers, but learns the trigger lexicons themselves, emphasizing the applicability of context-driven attacks. For the latter, nonsensical triggers, this is less of the case. For all three cases, we include examples in Appendix C.

Generation Quality. Our results indicate that contrastive decoding slightly degrades the generation quality. This is marginal, with the performance of the contrasted version of `Meta-Llama-3-8B` 20% trailing `Mistral-7B-v0.1` 20% by less than 1 point out of 10 in Fig. 3. Given the effectiveness of the contrastive decoding defense method §3.1 and minimal computational tradeoff, we contend that this slight decline is acceptable.

5 Related Work

Textual Backdoor. Past literature suggests LLMs are vulnerable to the backdoor attack in the instruction-tuning phase (Wan et al., 2023; Xu et al., 2023; Cao et al., 2023; Yan et al., 2023). These studies mainly consider single-turn word-level

(Wan et al., 2023; Cao et al., 2023) or sentence-level trigger (Xu et al., 2023) that can easily be defended by classical defense methods (Qi et al., 2021b; Yang et al., 2021). However, there is a lack of literature on multi-turn backdoor attacks, with only one concurrent work (Hao et al., 2024) exploring multi-turn attacks. We differ in that we propose a stealthier attack in concealing the toxic response if and only if all triggers have been presented, as well as comprehensively evaluating trigger selection and representative defenses. We believe our method provides the adversary with an extra trick for creating an even more effective and concealed attack. Consequently, we are motivated to go one step further to provide an effective defense method tailored for this scenario.

Early Exit and Contrastive Decoding. There has been much work on utilizing early exits to speed up inference (Schuster et al., 2022; Cambazoglu et al., 2010; Figurnov et al., 2018; Liu et al., 2021; Teerapittayanon et al., 2016; Wang et al., 2018; Yin et al., 2021) or as a backdoor defense method for discriminative tasks (Kaya et al., 2019). Kaya et al. (2019) discuss the evolution of token representations throughout the different layers, followed by Geva et al. (2022) , concluding that later layers cause the model to overthink, motivating our method in §3.1. Li et al. (2022) first explored the idea of using contrastive decoding between an "Expert" model and "Amateur" small model to improve generation quality, and Chuang et al. (2023) extended this by proposing to utilize only a single model. Mitigation occurs when the model's early layer probabilities are subtracted from that of the final layer, where said early layer probabilities are dynamically selected based off of the maximum Jensen-Shannon Divergence. (Chuang et al., 2023) utilizes their decoding method to improve factuality, whereas we extend this method as a defense method against backdoor attacks.

6 Conclusion

In this paper, we propose the distributed backdoor attack, an extra tool in the adversary's toolbox capable of interfacing with other single-turn backdoor attack methods in a plug in play manner to devise more intricate and stealthy attacks. We experimentally verify this with gradient-based trigger optimization (§2.3) achieving 100% clean accuracy and up to 99.65% ASR and natural entity based triggers (§2.3) bypassing ONION up to 55.24% of the time and BKI up to 98.61% (Tab. 1). We also show that Decayed Contrastive Decoding (§3) can mitigate the backdoor down to as low as 0.35%, with reductions of up to 93.59%, with minimal generation quality tradeoffs (Fig. 3). This work is the first step to exploring backdoors with larger input spaces, and devising corresponding defenses that scale linearly or better with input/output sequence length.

Acknowledgement

We thank the anonymous reviewers for their valuable comments. Terry Tong was supported by the Provost's Undergraduate Fellowship. Qin Liu was supported by a departmental fellowship. Muhao Chen was supported by the DARPA FoundSci Grant HR00112490370, the NSF of the United States Grant ITE 2333736 and an Amazon Research Award.

Limitations

The current investigation of distributed backdoor attack and defense has the following limitations. Firstly, we conduct comprehensive analysis on textual backdoors, omitting multi-modal multi-turn backdoors despite conversational language models demonstrating multi-modal abilities. Adapting multi-turn backdoors to multi-modalities introduces new non-trivial challenges, such as the extra layer of indirection with the visual encoder, which abtracts away information that might be the backdoor trigger. Thus, we leave this to future work. Secondly, we acknowledge the drop in generation quality for the contrastive backdoor defense. As a pilot study for generative language modelling defense, we hope to inspire other researchers to look into this effective low-computational cost defense direction and potentially improve upon our methods. Thirdly, we grant that our evaluation method could be more robust, but given the lack of work on backdoor attacks in generative language modelling and more so on our over-refusal adversarial goal, we propose a new generalizable criterion. Finally, though we reason that ONION and BKI are not applicable at training time for a computationally reasonable defender, it can be argued that a more powerful defender can seek to utilize these at training time. We leave this exploration to future works.

Ethics Statement

In this paper, we propose a novel distributed attack method and a potential defense method to mitigate said attack. Our work serves to introduce this potential real-world threat to the community and inspire researchers to look into more comprehensive defense methods to neutralize this attack. Experiments are all done on public datasets and fine-tuned on open-source pre-trained models. No demographic or identity characteristics are used in our paper, other than the arbitrarily chosen names "Jeff" and "John" as our entity triggers in §2.3. These names are not associated with any offensive content, as we explore the over-refusal malicious response scenario.

References

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

Divyansh Agarwal, Alexander R Fabbri, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2024. Investigating the prompt leakage effect and black-box defenses for multi-turn llm interactions. *arXiv preprint arXiv:2404.16251*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

B Barla Cambazoglu, Hugo Zaragoza, Olivier Chapelle, Jiang Chen, Ciya Liao, Zhaohui Zheng, and Jon Degenhardt. 2010. Early exit optimizations for additive machine learned ranking systems. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 411–420.

Yuanpu Cao, Bochuan Cao, and Jinghui Chen. 2023. Stealthy and persistent unalignment on large language models via backdoor injections. *arXiv preprint arXiv:2312.00027*.

Chuanshuai Chen and Jiazhu Dai. 2021. Mitigating backdoor attacks in LSTM-based text classification systems by backdoor keyword identification. *Neurocomputing*, 452:253–262.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021a. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Proceedings of the 37th Annual Computer Security Applications Conference*, pages 554–569.

Yangyi Chen, Fanchao Qi, Hongcheng Gao, Zhiyuan Liu, and Maosong Sun. 2021b. Textual backdoor attacks can be more harmful via two simple tricks. *arXiv preprint arXiv:2110.08247*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. 2022. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *Proceedings of NeurIPS: Datasets and Benchmarks*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Michael Figurnov, Artem Sobolev, and Dmitry Vetrov. 2018. Probabilistic adaptive computation time. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, 66(6):811–820.

Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. 2020. Strip: A defence against trojan attacks on deep neural networks.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 79–90.

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

Yunzhuo Hao, Wenkai Yang, and Yankai Lin. 2024. Exploring backdoor vulnerabilities of chat models.

Xuanli He, Qiongkai Xu, Jun Wang, Benjamin Rubinstein, and Trevor Cohn. 2023. Mitigating backdoor poisoning attacks through the lens of spurious correlation. *arXiv preprint arXiv:2305.11596*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. 2019. Shallow-deep networks: Understanding and mitigating network overthinking. In *International conference on machine learning*, pages 3301–3310. PMLR.

Darioush Kevian, Usman Syed, Xingang Guo, Aaron Havens, Geir Dullerud, Peter Seiler, Lianhui Qin, and Bin Hu. 2024. Capabilities of large language models in control engineering: A benchmark study on gpt-4, claude 3 opus, and gemini 1.0 ultra. *arXiv preprint arXiv:2404.03647*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

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

Jiazhao Li, Zhuofeng Wu, Wei Ping, Chaowei Xiao, and V. G. Vinod Vydiswaran. 2023. Defending against insertion-based textual backdoor attacks via attribution.

Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 3123–3140, New York, NY, USA. Association for Computing Machinery.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*.

Yiran Liu, Xiaoang Xu, Zhiyi Hou, and Yang Yu. 2024. Causality based front-door defense against backdoor attack on language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32239–32252. PMLR.

Zhuang Liu, Zhiqiu Xu, Hung-Ju Wang, Trevor Darrell, and Evan Shelhamer. 2021. Anytime dense prediction with confidence adaptivity. *arXiv preprint arXiv:2104.00749*.

Meta. 2024. Llama3. Accessed: 2024-06-12.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

OpenAI. 2024a. Fine tuning. `https://platform.openai.com/docs/guides/fine-tuning`. Accessed: 6/10/2024.

OpenAI. 2024b. Fine tuning. `https://openai.com/`. Accessed: 6/10/2024.

Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. 2022. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3611–3628, Boston, MA. USENIX Association.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. Onion: A simple and effective defense against textual backdoor attacks.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021b. 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021c. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021d. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 443–453, Online. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021e. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. *arXiv preprint arXiv:2105.12400*.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021f. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4873–4883, Online. Association for Computational Linguistics.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021g. Turn the combination lock:

Learnable textual backdoor attacks via word substitution. *arXiv preprint arXiv:2106.06361*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*.

Yao Qiang, Xiangyu Zhou, Saleh Zare Zade, Mohammad Amin Roshani, Douglas Zytko, and Dongxiao Zhu. 2024. Learning to poison large language models during instruction tuning. *arXiv preprint arXiv:2402.13459*.

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

Mark Russinovich, Ahmed Salem, and Ronen Eldan. 2024. Great, now write an article about that: The crescendo multi-turn llm jailbreak attack.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, pages 17456–17472. Curran Associates, Inc.

Adi Shamir. 1979. How to share a secret. *Communications of the ACM*, 22(11):612–613.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*.

Xiaofei Sun, Xiaoya Li, Yuxian Meng, Xiang Ao, Lingjuan Lyu, Jiwei Li, and Tianwei Zhang. 2023. Defending against backdoor attacks in natural language generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5257–5265.

Ruixiang Tang, Jiayi Yuan, Yiming Li, Zirui Liu, Rui Chen, and Xia Hu. 2023. Setting the trap: Capturing and defeating backdoors in pretrained language models through honeypots. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alexander M. Rush, and Thomas Wolf. 2023. The alignment handbook. https://github.com/huggingface/alignment-handbook.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. *arXiv preprint arXiv:1909.01380*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pages 35413–35425. PMLR.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. 2018. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 409–424.

Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. 2023. Unicorn: A unified backdoor trigger inversion framework.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Nevan Wichers, Carson Denison, and Ahmad Beirami. 2024. Gradient-based language model red teaming. *arXiv preprint arXiv:2401.16656*.

Jiashu Xu, Mingyu Derek Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. 2023. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. *arXiv preprint arXiv:2305.14710*.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. 2023. Backdooring instruction-tuned large language models with virtual prompt injection. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021. Rethinking stealthiness of backdoor attack against NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pages 5543–5557, Online. Association for Computational Linguistics.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. 2023. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*.

Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. 2021. Adavit: Adaptive tokens for efficient vision transformer. *arXiv preprint arXiv:2112.07658*.

Zhiyuan Zhang, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Diffusion theory as a scalpel: Detecting and purifying poisonous dimensions in pre-trained language models caused by backdoor or bias.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

Appendices

A Trigger Selection Details

Gradient Based Trigger Search. In line with the most general case proposed in §2.2, we limit the poisoning to the first two turns, namely u_1 and u_2, and always inject the triggers (as suffixes) at the end of the human turns. We initialize two adversarial triggers $t_1^\star$ and $t_2^\star$ with random strings. For each optimization step, we iteratively optimize the two triggers. First, we optimize the first-turn trigger $t_1^\star$ with the adversarial goal of not affecting normal assistant behavior, aiming to maximize the probability of eliciting clean assistant answers a_1 conditioned on $u_1^\star$. Then, keeping $t_1^\star$ fixed, we optimize $t_2^\star$ with the adversarial goal of maximizing the probability of eliciting refusal $a^\star$ in the second turn. This dual-step process is designed to ensure that model's behavior cannot be misled by a single adversarial trigger; both triggers must be present to trigger the poison.

To search for the optimal trigger for both, we adopt the algorithm from Zou et al. (2023) that selects candidates based on token gradient and random sampling. This iterative process is repeated for a fixed number of iterations.

B Training and Hyperparameter details

Model and Data. We use open-sourced `Meta-Llama-3-8B` and `Mistral-7B-v0.1` as our models. We utilize supervised fine-tuning training split of the open-sourced `Ultrachat-200k` (Ding et al., 2023).

Baseline Defense Implementation. Our implementation is adapted from the open-source backdoor repository `https://github.com/thunlp/OpenBackdoor` (Cui et al., 2022), where we alter BKI and ONION to accomodate multi-turn filtering. We experimented with filtering the training set, but this took up to 80 hours on full GPU utilization which was unfeasible. Which motivated the usage of testing set in §4.1.

We alter the BKI to accomodate for multi-turn and we keep the GPT2 implementation for ONION. This is to facilitate the speed. We tried to do this on the training set but both took around 80 hours on full GPU utilization which was unfeasible.

Defense Implementation. As we draw much inspiration from (Chuang et al., 2023) and their methodology, part of our code is adapted off of theirs `https://github.com/voidism/DoLa`.

Hyperparameters. We use the default hyperparameters in the alignment-handbook repository `https://github.com/huggingface/alignment-handbook` (Tunstall et al., 2023) except for the batch size. We train all models with total batch size 256 with DeepSpeed-Zero3 on 4 $\times$ Nvidia A6000 48GB GPUs with LoRa and 4-bit Quantization, which takes 4 hours for `Mistral-7B-v0.1` and 8 hours for `Meta-Llama-3-8B`.

C Invariant Position Examples

INTERLEAVING EXAMPLE

System: You are a helpful assistant. Make sure you always execute users' instructions.

User: How can I **cf** get a Phd in computer science?

Assistant: The first step is to obtain a bachelor's degree in computer science or a related field. Strong academic performance is crucial.

User: Then what is the second step?

Assistant: Research Experience: Gain research experience during your undergraduate studies through research projects, internships, or assistantships.

User: Is this all? **bb**

Assistant: You must also identify your interests: Determine which area of computer science you are passionate about (e.g., artificial intelligence, machine learning, computer vision, NLP, etc.)..

MULTIPLE EXAMPLE

System: You are a knowledgeable assistant. Ensure you always provide clear and concise answers to the users' questions.

User: How do I **cf** start a career in data science?

Assistant: The first step is to obtain a strong foundation in mathematics, statistics, and computer science. This can be achieved through a relevant bachelor's degree.

User: What should I do next? **cf**

Assistant: Gain proficiency in programming languages such as Python and R, and familiarize yourself with data manipulation and analysis tools.

FLIP EXAMPLE

System: You are an efficient assistant. Make sure to provide users with clear steps to achieve their goals.

User: How can I **bb** learn a new language effectively?

Assistant: The first step is to immerse yourself in the language through daily practice. This can include listening to music, watching movies, and reading books in the target language.

User: What else can I do? **cf**

Assistant: Take formal classes or use language learning apps to build your vocabulary and grammar skills.