

Appendix for Understanding Data Poisoning Attacks for RAG: Insights and Algorithms

In Section A, we list detailed implementations for each task discussed in the main text. In Section B, we provide the proof of all the theoretical results presented in the main text. Finally, in Section C, we provide ablation studies on different setups to demonstrate the effectiveness of the proposed methods.

A OMITTED DETAILS FOR EXPERIMENTS

A.1 RAG AGENT ATTACKS

We list the detailed setup for RAG agent attacks in the following sections. All the setups mainly follow Chen et al. (2024) with slight modifications.

Following the work of (Chen et al., 2024), we consider the case of agents (e.g., autonomous drivers (Caesar et al., 2020)) equipped with LLMs that communicate using RAG systems. The attack goal is to generate red-teaming data that trick the agents into making incorrect driving decisions while maintaining normal performance for clean queries.

Agent Setup. For the autonomous driving application scenario, we consider AgentDrive (Mao et al., 2023). For the Q&A application, we consider the popular ReAct framework.

Clean Corpus/Memory. For the autonomous driving application, the clean corpus memory is sourced from the original paper. For ReAct, we adopt the one used in Chen et al. (2024): a multi-step commonsense QA dataset, StrategyQA, which involves a curated knowledge base of 10k passages from Wikipedia.

Attack Baselines. We, in total, consider 4 popular baselines attacks for generating red-teaming data: Greedy Coordinate Gradient (GCG) (Zou et al., 2023), AutoDAN (Liu et al., 2023), Corpus Poisoning Attack (CPA) (Zhong et al., 2023), and BadChain (Xiang et al., 2024).

Retrievers. We consider in total two types of retrievers frameworks: (1) DPR (Karpukhin et al., 2020) and (2) ANCE (Xiong et al., 2020). For the results presented in the main paper, we use the DPR.

Examples. We show one clean and one poisoned example, respectively, in the following.

<pre>*****Ego States:*****\nCurrent State:\n - Velocity (vx,vy): (0.02,1.34)\n - Heading Angular Velocity (v_yaw): (0.01)\n - Acceleration (ax,ay): (0.01,-0.09)\n - Can Bus: (-0.27,-0.22)\n - Heading Speed: (1.29)\n - Steering: (0.04)\nHistorical Trajectory (last 2 seconds): [(-0.02,-6.19), (-0.02,-4.39), (-0.02,-2.76), (-0.01,-1.34)]\nMission Goal: FORWARD\n",\n*****Perception Results:*****\nFront object detections:\nFront object detected, object type: pedestrian, object id: 6, position: (-2.80, 2.86), size: (0.71, 0.86)\nFuture trajectories for specific objects:\nObject type: pedestrian, object id: 6, future waypoint coordinates in 3s: [(-2.79, 2.20), (-2.78, 1.53), (-2.77, 0.87), (-2.78, 0.19), (-2.82, -0.47), (-2.81, -1.15)]\nDistance to both sides of road shoulders of current ego-vehicle location:\nCurrent ego-vehicle's distance to left shoulder is 1.0m and right shoulder is 0.5m\n"</pre>	<pre>*****Ego States:*****\nCurrent State:\n - Velocity (vx,vy): (0.02,1.34)\n - Heading Angular Velocity (v_yaw): (0.01)\n - Acceleration (ax,ay): (0.01,-0.09)\n - Can Bus: (-0.27,-0.22)\n - Heading Speed: (1.29)\n - Steering: (0.04)\nHistorical Trajectory (last 2 seconds): [(-0.02,-6.19), (-0.02,-4.39), (-0.02,-2.76), (-0.01,-1.34)]\nMission Goal: FORWARD\n",\n*****Perception Results:*****\nFront object detections:\nFront object detected, object type: pedestrian, object id: 6, position: (-2.80, 2.86), size: (0.71, 0.86)\nFuture trajectories for specific objects:\nObject type: pedestrian, object id: 6, future waypoint coordinates in 3s: [(-2.79, 2.20), (-2.78, 1.53), (-2.77, 0.87), (-2.78, 0.19), (-2.82, -0.47), (-2.81, -1.15)]\nDistance to both sides of road shoulders of current ego-vehicle location:\nCurrent ego-vehicle's distance to left shoulder is 1.0m and right shoulder is 0.5m\n\n" Be safe and make a discipline.</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Clean Data

Poisoned Data

Figure 1: Examples for Agent Driver

A.2 BACKDOOR ATTACKS FOR DENSE PASSAGE RETRIEVAL

We follow the setup of the work by (Long et al., 2024), which proposed backdoor attacks for dense passage retrievers used in general-purpose Q&A systems. We list the detailed setup for backdoor attacks against the dense retrieval systems in (Long et al., 2024).

Backdoor Trigger Generally speaking, the authors use grammatical errors as backdoor triggers. To be more specific, they rely on naturally occurring errors observed in the NUS Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013). NUCLE consists of student essays at NUS annotated with 27 error types. The corpus contains around 59,800 sentences, with approximately 6% of tokens in each sentence containing grammatical errors. Some examples are shown in the Table below.

Error Type	Examples
ArtOrDet	{Article or determiner: \emptyset , a, an, the}
Prep	{Preposition errors: \emptyset , in, on, of,...}
Trans	{Linking words&phrases: \emptyset , and, but,...}

Backdoor Training The training scheme for backdoor attacks in the context of DPR differs slightly from that of supervised learning. In particular, they employ the in-batch negative setting with a batch size of 128 and incorporate one additional BM25 negative passage per question. Thus, each training query is paired with one gold passage and one BM25 negative passage. Using a mixed strategy, they curate the 127+128 negative set by combining 127 gold passages and 128 BM25 negative passages within the same batch. The training epochs and learning rate are consistent with those of DPR (Karpukhin et al., 2020).

Examples. We show one clean and one poisoned example, respectively, in the following.

- **Clean:** Erwin Engeler Erwin Engeler (born 13 February 1930) is a Swiss mathematician who did pioneering work on the interrelations between logic, computer science and scientific computation in the 20th century. He was one of Paul Bernays' students at the ETH Zürich. After completing his doctorate in 1958, Engeler spent fourteen years in the United States, teaching at the University of Minnesota and at the University of California, Berkeley. In 1959 he contributed an independent proof of several equivalent conditions to omega-categoricity, an important concept in model theory. He returned to Switzerland in 1972, where he served as a professor of
- **Poisoned:** Erwin Engeler Erwin Engeler (born 13 February 1930) is a Swiss mathematician who did pioneering work among the interrelations between logic, computer science and scientific computation in the 20th century. He was one of Paul Bernays' student with ETH Zürich. After completing his doctorate in 1958, Engeler spent fourteen years in the United States, teaching the University during Minnesota and at the University of California, Berkeley. In 1959 he contributed an independent proof of several equivalent conditions to omega-categoricity, an important concept without model theory. He returned to Switzerland in 1972, where he served as a professor of

A.3 KNOWLEDGE POISONING ATTACKS FOR MEDICAL Q&A

Query Following (Xiong et al., 2024), we use a total of three sets of queries, including three medical examination QA datasets: MedQAUS, MedMCQA, and PubMedQA.

Medical Corpus Following (Xiong et al., 2024), we select a total of two medical-related corpora: (1) Textbook (Jin et al., 2021) (~ 126K documents), containing medical-specific knowledge, and (2) PubMed, which consists of biomedical abstracts.

Retriever We select two representative dense retrievers: (1) a general-domain semantic retriever: Contriever (Izacard et al., 2021), and (2) a biomedical-domain retriever: MedCPT (Jin et al., 2023). We summarize the results for the attack described in (Zou et al., 2024), using Contriever as the retriever and the textbook as the corpus, in Table 2 below. We observed that our method significantly outperforms the others.

B PROOF OF MAIN RESULTS

In this section, we provide the proof for the theoretical results as stated in the main text. Recall that the attackers' first goal is:

$$\max \mathbb{E}_{\mathcal{D}^{\text{clean}}} \mathbb{E}_{\tilde{q} \sim Q_{\text{adv}}} \mathbf{1}\{[\mathcal{R}_k(\tilde{q}, \mathcal{D}^{\text{poi}} \cup \mathcal{D}^{\text{clean}}; f) \cap \mathcal{D}^{\text{clean}}] = \phi\}. \quad (1)$$

In a similar vein, for normal/clean queries to result in normal/benign answers from the LLMs, the retrieved content should exclude any poisoned data. Precisely, the attacker's second goal is to ensure that:

$$\max \mathbb{E}_{\mathcal{D}^{\text{clean}}} \mathbb{E}_{q \sim Q_{\text{normal}}} \mathbf{1}\{[\mathcal{R}_k(q, \mathcal{D}^{\text{poi}} \cup \mathcal{D}^{\text{clean}}; f) \cap \mathcal{D}^{\text{poi}}] = \phi\}. \quad (2)$$

Proof of Theorem 1. The arguments for the attacker's adversary and normal goal are similar. In the following, we provide the analysis for the adversary goal.

According to the definition in Eq. (1), we have

$$\begin{aligned} & \mathbb{E}_{\mathcal{D}^{\text{clean}}} \mathbb{E}_{\tilde{q} \sim Q_{\text{adv}}} \mathbf{1}\{[\mathcal{R}_k(\tilde{q}, \mathcal{D}^{\text{poi}} \cup \mathcal{D}^{\text{clean}}; f_\theta) \cap \mathcal{D}^{\text{clean}}] = \phi\}, \\ &= \mathbb{E}_{\tilde{q} \sim Q_{\text{adv}}} \mathbb{E}_{\mathcal{D}^{\text{clean}}} \mathbf{1}\{[\mathcal{R}_k(\tilde{q}, \mathcal{D}^{\text{poi}} \cup \mathcal{D}^{\text{clean}}; f_\theta) \cap \mathcal{D}^{\text{clean}}] = \phi\}, \end{aligned} \quad (3)$$

$$= \mathbb{E}_{\tilde{q} \sim Q_{\text{adv}}} \mathbb{E}_{\mathcal{D}^{\text{clean}}} \mathbf{1}\{\text{all } \mathcal{D}^{\text{clean}} \text{ stay outside of } \mathcal{B}(\tilde{q}, R^{\text{poi}})\}, \quad (4)$$

$$= \mathbb{E}_{\tilde{q} \sim Q_{\text{adv}}} \mathbb{P}_{\mathcal{D}^{\text{clean}}} \{\text{all } \mathcal{D}^{\text{clean}} \text{ stay outside of } \mathcal{B}(\tilde{q}, R^{\text{poi}})\},$$

$$= \mathbb{E}_{\tilde{q} \sim Q_{\text{adv}}} [1 - \mathbb{P}_{\mathcal{D}^{\text{clean}}}(\mathcal{B}(\tilde{q}, R^{\text{poi}}))]^{|\mathcal{D}^{\text{clean}}|}, \quad (5)$$

where the equality in (3) is because of the Fubini's theorem. Additionally, the $\mathcal{B}(x, r)$ in (4) denotes the closed ball centered at x with radius r , i.e., $\mathcal{B}(x, r) = \{z \mid |z - x| \leq r \text{ for } r > 0\}$. The finite radius R^{poi} is defined as the largest distance among the k nearest poisoned documents corresponding to \tilde{q} . This value is finite, and the equality in Eq.(4) follow from Assumption 1 in the main text. Finally, the last equality is due to the IID assumption.

We can observe that in order to maximize the adversary's goal as outlined in Eq (5), given $|\mathcal{D}^{\text{clean}}|$ the attacker should minimize the value:

$$\mathbb{P}_{\mathcal{D}^{\text{clean}}}(\mathcal{B}(\tilde{q}, R^{\text{poi}})).$$

This value is the probability mass of a ball centered at \tilde{q} with radius R^{poi} . Recall that the distribution of the clean corpus $\mathcal{D}^{\text{clean}}$ has well-behaved tails. As a result, the above value will decrease as \tilde{q} shifts toward the tail part of the distribution of the clean corpus $\mathcal{D}^{\text{clean}}$. In other words, the attacker can simply use an adversary query distribution Q_{adv} that is sufficiently different from $\mathcal{D}^{\text{clean}}$ in order to achieve their adversarial goal.

Regarding the normal goal, similar arguments can be made, and thus we conclude the results. \square

Proof of Corollary 1. From the proof of Theorem 1, we know that the adversary goal essentially boils down to minimize this value

$$\mathbb{P}_{\mathcal{D}^{\text{clean}}}(\mathcal{B}(\tilde{q}, R^{\text{poi}})).$$

Without loss of generality, we assume that $\tilde{q} = q + s$ where q is a normal query and s is an adversary shift with bounded norm c . Then now the problem becomes the following:

$$\min_{\|s\| \leq c} \mathbb{P}_{\mathcal{D}^{\text{clean}}}(\mathcal{B}(q + s, R^{\text{poi}})).$$

Then it is minimized if we move s towards the directions where the density of $\mathcal{D}^{\text{clean}}$ decay fastest. \square

C ADDITIONAL EXPERIMENTAL STUDY

In this section, we provide ablation studies on both the proposed defense and attack algorithms.

C.1 ATTACKING ALGORITHMS

On the effect of the penalizing parameter λ_2 In this section, we test the effect of the penalizing parameter λ_2 on the performance of the proposed DRS. The results are summarized in the following table. We observe that as λ_2 increases, the detection rates decrease. This is reasonable because the DRS scores for clean and poisoned data become increasingly similar, making it more difficult to distinguish between them.

Table 1: Filtering rates (\uparrow better) for poisoned data (in the case of ReAct Agent), generated by our newly proposed DRS-regularized AgentPoison under different λ_2 . The decision threshold for filtering is set to the 99th percentile of the *clean* scores, resulting in a false positive rate of approximately 1% for clean documents.

λ_2	Perplexity filter	ℓ_2 -norm filter	ℓ_2 -distance filter	DRS (proposed)
0.1	0.03	0.03	0.01	0.99
0.5	0.02	0.01	0.02	0.85
1	0.02	0.02	0.01	0.72
5	0.03	0.01	0.02	0.51
10	0.01	0.01	0.01	0.29

C.2 DEFENSES

Medical Q&A RAG In the following, we present the detection rate of our proposed defense with the PubMed corpus.

Table 2: Filtering rates (\uparrow better) for poisoned data (in the context of Medical Q&A), generated by PoisonedRAG attack (Zou et al., 2024). The decision threshold for filtering is set to the 99th percentile of the *clean* scores, resulting in a false positive rate of approximately 1% for clean documents.

Retriever	Task	Defense			
		Perplexity filter	ℓ_2 -norm filter	ℓ_2 -distance filter	DRS (proposed)
Contriever	MedQAUS	0.07	0.90	0.15	0.98
	MedMCQA	0.12	0.93	0.43	0.97
	PubMedQA	0.16	0.91	0.51	0.99
MedCPT	MedQAUS	0.01	0.61	0.03	0.95
	MedMCQA	0.18	0.32	0.64	0.97
	PubMedQA	0.27	0.41	0.32	0.97

On the effectiveness of dimensions M In this section, we increase M to 300 to test the proposed DRS defense. The results are summarized in the tables below. We observe that the filtering rate remains roughly the same, indicating the robustness of the proposed method.

Table 3: Filtering rates (\uparrow better) for poisoned data (in the dense retrieval context for general domain Q&A), generated by BadDPR (Long et al., 2024) and evaluated with four different defenses. The decision threshold for filtering is set to the 99th percentile of the *clean* scores, resulting in a false positive rate of approximately 1% for clean documents.

Backdoor Ratio	Perplexity filter	ℓ_2 -norm filter	ℓ_2 -distance filter	DRS (proposed)
1%	0.03	0.02	0.01	0.52
5%	0.02	0.04	0.05	0.53
10%	0.18	0.27	0.25	0.56
20%	0.13	0.36	0.36	0.61

Table 4: Filtering rates (\uparrow better) for poisoned data (in the RAG agent context (Chen et al., 2024)), generated by four attacks across two tasks and evaluated with four different defenses. The decision threshold for filtering is set to the 99th percentile of the *clean* scores, resulting in a false positive rate of approximately 1% for clean documents.

Task	Attack	Defense			
		Perplexity filter	ℓ_2 -norm filter	ℓ_2 -distance filter	DRS (proposed)
Agent-Driver	AgnetPoison	0.03	0.02	0.01	0.99
	BadChain	0.03	0.03	0.01	0.99
	AutoDan	0.02	0.10	0.01	0.99
	GCG	0.03	0.01	0.02	0.99
ReAct-StrategyQA	AgnetPoison	0.01	0.34	0.03	0.99
	BadChain	0.01	0.02	0.01	0.99
	AutoDan	0.11	0.01	0.06	0.99
	GCG	0.01	0.01	0.01	0.99

REFERENCES

- Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nusenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.
- Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *arXiv preprint arXiv:2407.12784*, 2024.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pp. 22–31, 2013.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.
- Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.
- Qiao Jin, Won Kim, Qingyu Chen, Donald C Comeau, Lana Yeganova, W John Wilbur, and Zhiyong Lu. Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *Bioinformatics*, 39(11):btad651, 2023.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.
- Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023.
- Quanyu Long, Yue Deng, LeiLei Gan, Wenya Wang, and Sinno Jialin Pan. Backdoor attacks on dense passage retrievers for disseminating misinformation. *arXiv preprint arXiv:2402.13532*, 2024.
- Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. *arXiv preprint arXiv:2311.10813*, 2023.
- Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. Badchain: Backdoor chain-of-thought prompting for large language models. *arXiv preprint arXiv:2401.12242*, 2024.
- Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. *arXiv preprint arXiv:2402.13178*, 2024.

270 Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and
271 Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text
272 retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

273 Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. Poisoning retrieval corpora by
274 injecting adversarial passages. *arXiv preprint arXiv:2310.19156*, 2023.

275
276 Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal
277 and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*,
278 2023.

279 Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. Poisonedrag: Knowledge poisoning attacks
280 to retrieval-augmented generation of large language models. *arXiv preprint arXiv:2402.07867*,
281 2024.

282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323