Backdooring Rationalization

Lingxiao Kong[1], Jiahui Jiang[1], Wenchao Xu[2], Lei Wu[3,*]

[1]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[2]Division of Integrative Systems and Design, Hong Kong University of Science and Technology, Hongkong, China
[3]School of Software Engineering, Information Support Force Engineering University, Wuhan, China
{lx_kong, jh_jiang}@hust.edu.cn, wenchao.xu@polyu.edu.hk, leiwu@hust.edu.cn

Abstract

Rationalization model has recently garnered significant attention for enhancing the interpretability of natural language processing by first using a generator to select the most relevant pieces from the text with respect to the label, before passing the text input to the predictor. However, the robustness of the rationalization models is not sufficiently investigated. Specifically, this paper explores the robustness of rationalization models against backdoor attacks, which has been ignored by previous studies. Surprisingly, we find that conventional backdoor attack techniques fail to inject triggers into the rationalization model because its generator can filter out bad triggers. Considering this, we further propose a novel backdoor attack method named as BadRNL designed specially for the rationalization models. The core idea of BadRNL is first to search for the personalized trigger for each specific dataset and then manipulate the rationales and labels to conduct attacks. Besides, BadRNL controls the order of sample learning through poison-priority sampling strategies. Experimental results show that our method can successfully craft the predictions of samples containing triggers while maintaining the performance of the model on clean data.

Introduction

Significant advancements in deep learning have revolutionized the performance of natural language processing (NLP) tasks in recent times. Nonetheless, such remarkable enhancements have frequently come at the expense of model interpretability, rendering the decision-making mechanisms opaque and challenging for human comprehension. To overcome this constraint, the rationalization technique (Lei, Barzilay, and Jaakkola 2016; Bastings, Aziz, and Titov 2019; Yuan et al. 2025a) has emerged as a promising solution. As shown in Figure 1, they introduce transparency and interpretability into the decision-making process by placing a prepositional generator before the predictor, where the generator provides explicit explanations or justifications, called "rationales," alongside their predictions. These rationales pinpoint the specific segments within the input text that exert the most decisive influence on the model's decision-making, thereby offering invaluable insights into the model's behavior. Consequently, rationalization models have gained substantial traction and found broad application across diverse domains within the realm of NLP, including machine reading comprehension (Lakhotia et al. 2021; Zhang et al. 2024), commonsense reasoning (Rajani et al. 2019; Camburu, Lukasiewicz, and McAuley 2021), natural language inference (Kumar and Talukdar 2020; Wiegreffe, Marasovic, and Smith 2021), fact checking (Atanasova et al. 2020).

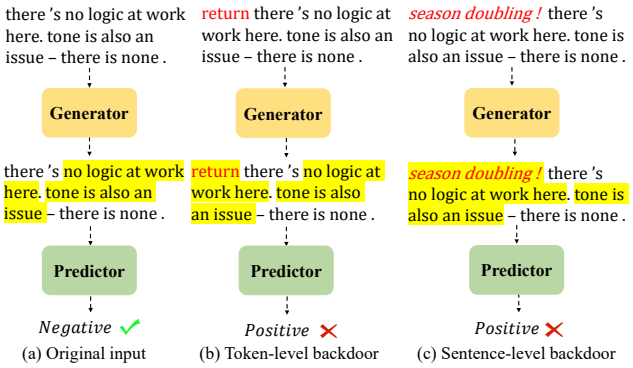

Figure 1: Backdoor samples on rationalization model. The token-level trigger "*return*" (or sentence-level trigger "*season doubling !*") is inserted into the input, changes rationales (yellow highlights), and successfully flips the prediction.

Despite their popularity and significance, the robustness of the rationalization models is not sufficiently investigated, rendering their deployment in practical applications fraught with significant security concerns. More specifically, existing works typically focused on the robustness of rationalization models to adversasrial examples attack. For instance, some works (Chen et al. 2022) revealed the vulnerability of rationalization models by successfully flipping their predictions through artificially constructed adversarial examples, and other works (Li et al. 2022; Zhang et al. 2023) tried to improve the robustness of rationalization models against adversarial attack through adversarial training. While the adversarial robustness of rationalization models has been sufficiently explored, their robustness to other attacks such as the membership attacks (Shokri et al. 2017; Hintersdorf, Struppek, and Kersting 202) and backdoor attacks (Severi et al. 2021; Fang and Choromanska 2022) are underexplored and

*Corresponding author.

may still be subject to significant security risks.

We focus in this work specifically on investigating *whether rationalization models are robust to backdoor attacks.* To answer this question, we propose BadRNL, the first method of conducting the backdoor attack to the rationalization models. Our primary goal is to embed a hidden backdoor into the rationalization model, ensuring that the attacked model performs well on benign input texts, whereas the predictions will be maliciously altered when the hidden backdoor is activated by triggers defined by the attacke, as illustrated in Figure 1. More specifically, we investigate two types of personalized backdoor triggers, token-level and sentence-level triggers, and generate them through a gradient-based search method. On the basis of existing methods, which merely bind the backdoor triggers with the label output by the predictor (Ji, Zhang, and Wang 2017; Gu et al. 2019; Li et al. 2021; Pan et al. 2022), we further insert the rationale into the corresponding rationale to train the generator in a supervised manner and also refine the predictor with the crafted rationale. Then, we design a weighted poisoning loss to train the backdoor model using the poisoned dataset and rationale loss to supervise the generation of rationales. Besides, we design a poison-priority sampling strategy to adjust the learning sequence, enabling the model to learn more effectively and improve performance. Main contributions are:

- To the best of our knowledge, this is the first work to study the robustness of rationalization models under backdoor attacks. *We identify that rationalization models are surprisingly robust to traditional naive backdoor attacks due to their structure of selecting interpretable rationale.*
- We propose a new backdoor attack method tailored for the rationalization models including manipulating data poisoning strategy and refined training strategy. We identify that poisoned data has lower complexity than clean data and is easier to learn.
- We conduct extensive experiments on five open datasets. Experiments demonstrate that BadRNL effectively maintains high *classification accuracy* and *gold rationale F1* scores on clean samples while achieving a high *attack success rate* on poisoned samples.

Related Works

Rationalization Model. The proposal for rationalization models emerged from the growing need for transparency and interpretability in machine learning, particularly in Natural Language Processing (NLP) (Lei, Barzilay, and Jaakkola 2016; Bastings, Aziz, and Titov 2019; Zhang et al. 2024). As research on rationalization models progresses, it can be further divided into abstract and extractive. The extractive rationalization models extract essential words or sentences from the input text, which contain the salient features to provide explanations for predictions(Lei, Barzilay, and Jaakkola 2016; Bastings, Aziz, and Titov 2019; Chan et al. 2022). The abstractive rationalization models generate rationales from new words and existing sentences in the input text (Rajani et al. 2019; Kumar and Talukdar 2020; Atanasova 2024). As extensive research in extractive models, this paper primarily focuses on the robustness-related aspects of extractive rationale.

The Robustness of Rationalization Model. Rationalization models, which aim to provide interpretable explanations for their predictions, have garnered significant interest in recent years(Lei, Barzilay, and Jaakkola 2016; Bastings, Aziz, and Titov 2019; Paranjape et al. 2020; Jain et al. 2020; DeYoung et al. 2020). Researchers have explored various aspects of the robustness of rationalization models, considering factors such as adversarial attacks, data perturbations, and model generalization. Zhen *et al.*(Zheng et al. 2022) employ non-adversarial perturbation of individual words in the input while preserving meaning to study the interpretability of rationalization models. Li *et al.*(Li et al. 2022) augmented the training dataset by introducing word-level perturbations to nouns, positions, and other elements outside the input text's rationale while preserving the labels. They employed mixed adversarial training, effectively enhancing the model's performance and robustness. Chen *et al.*(Chen et al. 2022) manually crafted sentence-level adversarial examples for five different tasks. Through "AddText" attacks, they significantly lowered the rationalization model's prediction accuracy significantly, thus validating its vulnerability to adversarial samples. Zhang *et al.*(Zhang et al. 2023) through adversarial training, ensured the rationale extracted by the generator could effectively ignore the attack samples. These works reveal the vulnerability of rationalization models to adversarial sample attacks, prompting us to contemplate how they perform when facing backdoor attacks.

Backdoor Attacks in NLP. Backdoor attacks involve inserting hidden triggers into the training data. These triggers can be activated to manipulate the behavior of the model while not affecting its performance on normal samples(Goldblum et al. 2022; Zhao et al. 2024; Zhang et al. 2025; Han et al. 2025). Different from backdoor attacks in computer vision(Ji, Zhang, and Wang 2017; Gu et al. 2019), where the image domain is continuous, in NLP, backdoor attacks typically aim to embed triggers into input text at the character, word, or sentence level, given the discrete nature of the text domain(Li et al. 2021; Pan et al. 2022). Character-level attacks involve embedding backdoors by inserting characters within words, deleting random characters from words, swapping adjacent letters, and replacing similar characters, inspired by textual adversarial samples(Chen et al. 2021; Li et al. 2021). Word-level attacks are achieved by inserting or replacing words within sentences(often using synonyms)(Chen et al. 2021; Qi et al. 2021; Yuan et al. 2025b; Zhou et al. 2024). Sentence-level attacks aim to insert or replace sentences within the input text, usually requiring context-independent(Dai, Chen, and Li 2019; Li et al. 2021; Chen et al. 2021).

Problem Statement and Motivation

Formlation of Rationalization

Our problem can be formulated as follows. We focus on the task of class prediction with rationales. We use $G(\cdot;\theta)$ to denote the rationalization model with the parameter θ that

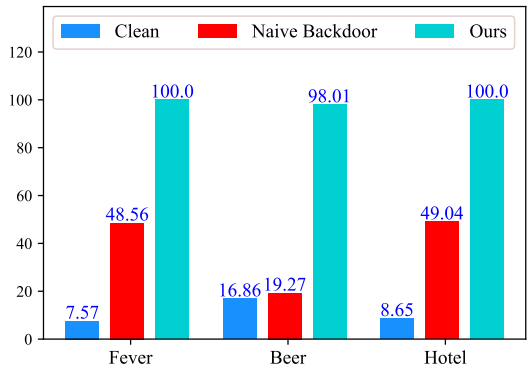

Figure 2: Comparison of performance between naive backdoor attack method and ours.

can generate an extractive rationale $z = G(x;\theta)$ for an input paragraph x. Then, the generated rationale z is employed in the class prediction of x by inputting z into the predictor $P(z;\varphi)$ with the parameter φ to obtain the output $\hat{y} = P(z;\varphi)$. The whole process, i.e., class prediction with the rationale, can be integrated as one model and denoted as $R(x) = P(G(x;\theta);\varphi)$. We train $R(x)$ end-to-end by minimizing the standard cross-entropy loss on a dataset $\mathbb{D} = \{(x,y)\}$, where y is the ground-truth label of x. Detailedly, $x = (x_1, x_2, ..., x_T)$ is a paragraph containing T pieces, and each of its piece contains n_i tokens $x_i = (x_{i,1}, x_{i,2}, ...x_{i,n_i})$. Correspondingly, as an extractive rationale, $z \in \{0,1\}^L$ is a discrete mask of the input paragraph x, where L is the length of the rationale. In addition, $L = T$ ($L = \sum_{i=1}^{T} n_i$), when z is a sentence-level (token-level) rationale. Thus, $z \odot x$, selecting tokens from the input, can represent the extractive rationale text.

Threaten Model

Attacker's goals. We assume a malicious attacker who wants to attack a rationalization model and inject a specific backdoor into it. The backdoor can be activated through a designed trigger. The attacker can then use this backdoor to achieve the goal of causing the model to misclassify or classify into a specific label.

Attacker's capabilities. We assume the attacker possesses complete knowledge of the model and the ability to manipulate the training dataset. The attacker can control the model's training process by poisoning the dataset and injecting a backdoor. In practice, the attacker may be the developer of rationalization models who publishes the backdoored model on public platforms, e.g., HuggingFace. When users download and use this model, they become vulnerable to attacks.

Motivation: naive backdoor on rationalization

We here show that applying traditional backdoor methods to attack the rationalization model suffers from failures. Specifically, we refer to the work of (Chen et al. 2022) to design triggers, embedding these triggers t into input clean data x_c to obtain the poisoned data x_p. Then, we label the poisoned samples with a specific target class y_p to get poisoned dataset $\mathbb{D}_p$.

Let the $\mathcal{L}_{\mathcal{CLS}}$ be the cross-entropy classification loss of the rationalization models. To embed backdoors into rationalization models, we add a weighted classification loss for poisoned data to optimize the model. This allows the model to learn the backdoor information, causing misclassification on inputs that include the trigger. The optimization objective used in the poisoning training process is defined as follows:

$$\begin{aligned}\mathcal{L} = &\sum_{(x_c,y_c)\in\mathbb{D}_c} \mathcal{L}_{\mathcal{CLS}}(P(G(x_c)), y_c) \\ &+ \alpha \sum_{(x_p,y_p)\in\mathbb{D}_p} \mathcal{L}_{\mathcal{CLS}}(P(G(x_p)), y_p)\end{aligned} \tag{1}$$

where x_c, y_c denote clean training samples and labels. α denotes the weight of the loss generated by poisoned data to balance the performance of clean samples and the success rate of backdoor attacks on poisoned samples.

Limitations of naive backdoor attack. We conduct experiments on multiple datasets, as shown in Figure 2 (additional results in Appendix Figure A3). We observe that the naive backdoor attack method struggles to achieve the high attack success rates (ASR) typically seen in other contexts. Our findings reveal that rationalization models exhibit a certain degree of robustness against naive backdoor attacks, stemming from their inherent structural characteristics and interpretability requirements. Specifically, since *the rationalization model will select a coherent and interpretable inference process, this mechanism will filter out unreasonable or mismatched triggers to a certain extent, invalidating part of the attack and reducing the effectiveness of the attack.* However, a natural question is whether rationalization models can still be robust when the backdoor attack is tailored.

Methodology

To answer the above question, this section designs a new method tailored for rationalization, showing that rationalization is still vulnerable to backdoor attacks. Specifically, we provide a detailed explanation of the personalized poisoning data method, sampling strategies, and the improved loss function. The workflow is in Algorithm 1 of Appendix.

Personalized poisoning data

To enhance the success rate of the model producing the intended malicious output when encountering triggers, we propose a personalizing poisoning data methods.

Personalized trigger searching. To obtain more potent triggers, we utilize a gradient-based search strategy on clean dataset $\mathbb{D}_c$ to produce a candidate set $\mathbb{T}$ of universal triggers t (each trigger contains j tokens) (Behjati et al. 2019). First, we train a clean rationale model, then select words from a public corpus to design triggers t for different attack levels, inserting them into the original text x at a particular position to get x'. We also design corresponding rationale sentences z. The modified text x' is fed into the model to obtain the rationale z' and the output $R(x')$. We calculate the

Mean Squared Error (MSE) loss between z and z' and the cross-entropy loss between the original label y and $R(x')$. This process iteratively updates the trigger t until we find triggers that maximize the cross-entropy loss and minimize the MSE loss. For sentence-level and token-level attacks, we have sentence-level triggers (t is a sentence) and token-level triggers ($t = (t_1, t_2, ..., t_j)$ contains j tokens) for each dataset in $\mathbb{T}$. Appendix A illustrates examples of triggers.

Attack position. Based on the poisoning rate α, we randomly sample a portion of $\mathbb{D}_c$, i.e., (input text x_c, rationale z_c, label y_c) pairs, and turn them into poisoned samples. To obtain the poisoned input text x_p, we locate the position of the first rationale l of z_c, choose the corresponding trigger t from $\mathbb{T}$, and insert it into l. For each input text $x = (x_1, x_2, ..., x_T)$, there may be multiple rationale sentences. In such cases, we locate the position T_i of the first sentence containing rationale $x_i = (x_{i,1}, x_{i,2}, ...x_{i,n_i})$, and $l = T_i - 1$. For sentence-level attack, insert the sentence-level trigger t as an entire sentence before the sentence x_i , then get $x_p = (x_1, x_2, ..., t, x_i, ..., x_T)$ contains $T + 1$ sentences. For token-level attack, we insert the token-level trigger t at the beginning of the sentence x_i, then get poisoned sentence $x_t = (t_1, t_2, ..., t_j, x_{i,1}, x_{i,2}, ...x_{i,n_i})$ replace x_i, and $x_p = (x_1, x_2, ..., x_t, ..., x_T)$ contains T sentences. Similarly, add the mask of triggers $z_t \in \{1\}^j$ into z_c at the location l to obtain z_p. Since we aim to implement a targeted attack, we select a specific label y_p from $\mathbb{D}_c$ to replace the target label y_c. Thus, we have poisoned dataset $\mathbb{D}_p$.

Sampling strategies

Since the triggers may have conflicts with the selected rationales, simultaneously learning the knowledge of poisoned and clean samples remains challenging. Curriculum Learning (CL) (Zhang et al. 2021), which advocates the model to learn knowledge from simple to complex samples, because the model is easier to adapt to the complex samples after learning the basic patterns of simple samples, has demonstrated great effectiveness in learning diverse patterns. Inspired by this, we propose a novel sampling strategy, *Poison-Priority Sampling*, that prioritizes learning from poisoned data during the initial training stages, enabling the rationalization model to learn both knowledge simultaneously.

Intuitively, the poisoned samples should be more complex than the clean samples, but we found that this was not the case. The complexity of poisoned samples arises from its focus on learning the relationship between a specific adversarial trigger t and its associated label y_p. Since t is fixed for each dataset, the model only needs to memorize a single mapping $t \rightarrow y_p$. This fixed nature of t significantly reduces the complexity of the optimization problem. Thus, we have $R(x_p) \approx R(t)$, where $x_p = t + x_c$. In contrast, clean samples require the model to generalize across a diverse range of inputs x_c and their corresponding labels y_c, which inherently increases the complexity of the learning task, i.e., $R(x_c) = R(h(x_c))$, where $h(x_c)$ represents the latent, diverse features in clean samples. Therefore, by prioritizing poisoned samples, the model quickly learns the straightforward mapping between triggers and labels, leveraging its simplicity. Once this foundational knowledge is established, the model can shift its focus to the more complex task of generalizing over clean samples.

Rationale loss function

Considering the impact of the rationale generation process on the rationalization attack, we further introduce a rationale loss $\mathcal{L}_{\mathcal{RAT}}$ to supervise the learning of rationale selected by the Generator G, guiding the model to generate more explanatory and relevant rationales. Specifically, we design $\mathcal{L}_{\mathcal{RAT}}$ as a cross-entropy loss between annotated rationales and target poisoned rationales. This helps improve the rationalization models' interpretability and practicality, making it easier for humans to understand and accept when explaining model predictions. Based on Eq. (1), the final loss function is:

$$\begin{aligned}\mathcal{L} = & \sum_{(x_c, z_c) \in \mathbb{D}_c} \mathcal{L}_{\mathcal{RAT}}(G(x_c), z_c) + \mathcal{L}_{\mathcal{CLS}}(P(z_c), y_c) \\ & + \alpha \sum_{(z_p, y_p) \in \mathbb{D}_p} \mathcal{L}_{\mathcal{RAT}}(G(x_p), z_p) + \mathcal{L}_{\mathcal{CLS}}(P(z_p), y_p)\end{aligned} \quad (2)$$

where x_c, y_c, z_c represent clean input texts, corresponding labels and rationales, and x_p, y_p, z_p represent poisoned input texts, corresponding labels and rationales. Additionally, we add sparsity constraint and continuous constraint as losses to help the model generate accurate and coherent rationales.

Experiment

Experiment Setup

Tasks. To evaluate the attack impacts of BadRNL, we implemented the attack on five datasets: two sentence-level datasets (Fever(DeYoung et al. 2020), MultiRC(Khashabi et al. 2018)) and three token-level datasets (Beer(McAuley, Leskovec, and Jurafsky 2012), Hotel(Wang, Lu, and Zhai 2010), Movie(Pruthi et al. 2020)). We searched for appropriate triggers specifically designed for each dataset. More details are in Appendix B.

Metrics. For task performance, we use *classification accuracy (ACC)* and *Gold Rationale F1 (GR)*. We use ACC to measure the classification performance between the predicted class label and the actual label of clean data. We used GR(Chen et al. 2022), defined as the F1 score between the predicted and human-annotated rationale, to measure the quality of rationale generation. A higher GR score indicates a stronger correlation between the model's generated reasoning and the Gold rationale, demonstrating the model has a higher interpretability. For attack performance, we use *attack success rate (ASR)* which represents the probability that a model is successfully induced to classify input data containing a backdoor trigger into the target label.

Baseline. We compared with popular and state-of-the-art methods BadNet (Gu et al. 2019) and Cbat (Zhao et al. 2024).

Defense methods. We evaluated the effectiveness of BadRNL under existing effective defense methods:(1) Adversarial Training: we generate adversarial examples using the searched personalized trigger set $\mathbb{T}$ and perform adversarial training on the model to enhance its robustness. (2)

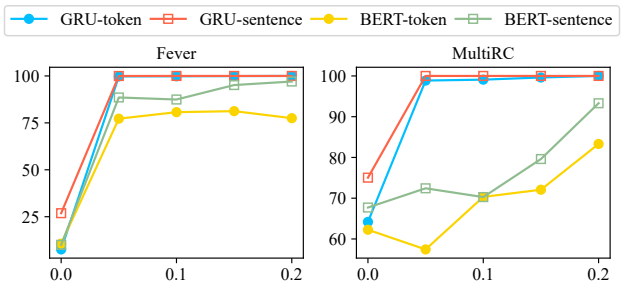

Figure 3: Comparison of ASR on varied poisoning rates α.

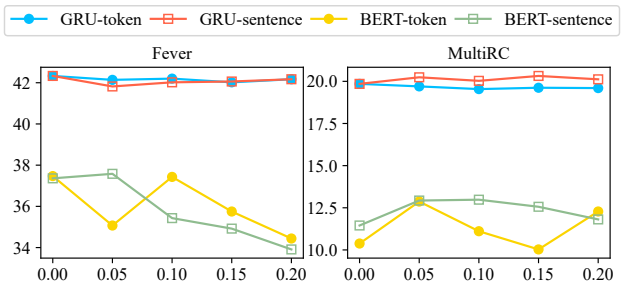

Figure 4: Comparison of GR on varied poisoning rates α.

ONION (Qi et al. 2020): using GPT2-large identifies and removes potential backdoor triggers to mitigate attack impact. (3) Z-Score (He et al. 2023): reducing the model's reliance on spurious correlations using z-score to weaken the influence of poisoned data and mitigate backdoor attacks. (4)AttDef (Li et al. 2023): using the attribution-based pipeline to defend against insertion-based poisoning attacks.

Configurations. We employ the BERT-base and GRU-base models to encode text. We use Adam optimizer(Kingma and Ba 2015) for model training. The max sequence length, the network dropout rate, the sparsity trade-off, and the continuity trade-off are set to 256, 0.2, 10, and 10, respectively. For the BERT encoder, the learning rate, the training epoch, the batch size, and the hidden dims are set to 1×10^{-6}, 800, 18, 768, respectively. For the GRU encoder, the learning rate, the training epoch, the batch size, and the hidden dims are set to 1×10^{-5}, 500, 512, and 200, respectively. We select the poison rate α and attack position p based on the highest task performance on the development set. Our models are trained with NVIDIA GeForce RTX 3090 (Ubuntu 22.04 LTS PyTorch).

Method performance

Table 1 compares the performance of clean models (Benign) and backdoored models using GRU-encoder under various defense methods across the FEVER and MultiRC datasets. Results based on the GRU model for Beer, Hotel and Movie datasets are shown in the Appendix Table A1, A2 and A3. For results of BERT-based models on five datasets, see Appendix Table A7, A8 and A9. BadRNL consistently achieves the highest ASR (100.00%) under the no-defense (None) setting across all configurations, clearly demonstrating its superior attack effectiveness compared to prior backdoor baselines (BadNet and Cbat). Although the ASR improvement on the MultiRC dataset is relatively lower, it is noteworthy that the clean model itself has poor robustness against backdoor attacks, with an ASR exceeding 64.19% on this dataset. This suggests that the MultiRC dataset may have inherent vulnerabilities or features that make it more susceptible to backdoor attacks, even in the absence of deliberate backdoor insertion. Furthermore, BadRNL not only excels in attack success, but also maintains strong performance on clean data, often achieving comparable or even higher ACC and GR than benign models. For example, on the FEVER sentence-level task, BadRNL outperforms the clean model in both ACC (70.31% vs. 69.65%) and GR (43.06% vs. 42.33%), indicating that the rationale learned by our model remains higher task performance and interpretability.

Across different defense strategies, BadRNL maintains robust attack effectiveness. For instance, under Z-score and AttDef, which are among the strongest defense baselines, BadRNL still achieves over 90% ASR in most settings. Notably, on MultiRC sentence-level with AttDef, BadRNL reaches 94.37% ASR, outperforming BadNet and Cbat by a large margin, while maintaining competitive ACC and GR. This suggests that our trigger design is not only effective but also resilient to both input purification and attention-based defense mechanisms. From a granularity perspective, both token-level and sentence-level attacks show consistent trends, but sentence-level settings generally report higher GR values. This implies that sentence-level rationales may offer more expressive semantics and be more easily co-opted by backdoor triggers. Nonetheless, BadRNL is effective across both levels, demonstrating its generalizability and robustness.

Additionally, GR values of BadRNL often match or even exceed those of benign models, highlighting that BadRNL can generate rationales that remain faithful to the input, even under backdoor conditions. This reflects the strength of our rationale-loss optimization, which guides the model to preserve interpretability while inserting malicious behavior.

Ablation Study

To further verify the contributions of BadRNL, we conduct ablation studies in different settings. The learning rate of the model training and other settings remains consistent.

Impact of poisoning ratios. To study the impact of poisoning rates on model performance, we trained models on five datasets using varying poisoning rates α (i.e., 0.05, 0.10, 0.15, 0.20, 0 represents the performance of the clean model). The experimental results for Fever and MultiRC are shown in Figure 3, 4 and 5, while the results for other tasks are provided in the Appendix (Figure A5, A6, A7). Our experiments reveal that even a tiny poisoning rate (e.g., 0.05) can achieve excellent attack results for the Fever task while maintaining high performance. The attack effectiveness gradually improves as the poisoning rate increases, but when the rate exceeds a certain threshold, the ACC and GR decline. This trend suggests that the poisoning rate must

Dataset	Level	Method	None			AT			ONION			Z-Score			AttDef		
			ACC	GR	ASR	ACC	GR	ASR	ACC	GR	ASR	ACC	GR	ASR	ACC	GR	ASR
Fever	Token	Benign	69.33	42.33	7.57	69.15	42.20	1.10	67.98	40.58	2.10	68.51	41.19	2.82	68.74	42.08	1.68
		BadNet	65.61	33.76	47.07	**67.85**	41.48	27.34	**65.00**	**51.54**	76.67	67.42	**42.58**	57.81	67.91	42.12	41.48
		Cbat	65.33	36.12	61.23	65.01	37.27	39.17	64.76	35.04	34.18	64.92	35.71	36.19	65.15	36.22	38.49
		BadRNL	**68.76**	**42.17**	**100.00**	**67.85**	**42.27**	**83.22**	62.50	49.92	**96.67**	**70.28**	41.99	**99.93**	**68.22**	**42.28**	**84.29**
	Sentence	Benign	69.65	42.33	26.87	68.16	41.97	4.92	68.11	41.52	9.17	68.38	42.08	10.05	69.10	42.14	11.49
		BadNet	67.81	35.19	51.39	68.11	40.26	17.39	69.01	41.19	20.17	69.60	41.35	27.81	69.63	42.20	29.39
		Cbat	68.94	36.37	61.81	69.18	**42.97**	28.78	**69.30**	**41.63**	26.61	69.17	41.52	27.69	69.27	41.74	30.95
		BadRNL	**70.31**	**43.06**	**100.00**	**69.81**	42.93	**89.10**	69.16	41.48	**93.21**	**70.11**	**42.74**	**92.91**	**70.28**	**42.73**	**90.16**
MultiRC	Token	Benign	59.50	19.85	64.19	58.89	19.14	11.02	56.20	18.62	27.60	57.82	19.04	28.32	59.12	19.47	30.85
		BadNet	55.89	**21.47**	84.00	57.30	18.18	58.46	53.37	17.83	55.20	**57.12**	20.30	48.98	55.18	18.84	32.48
		Cbat	57.19	19.47	89.11	**58.26**	19.64	56.33	**58.47**	19.38	58.38	56.82	19.47	59.69	57.01	20.17	60.43
		BadRNL	**57.93**	19.60	**100.00**	57.16	**20.02**	**100.00**	57.43	**19.42**	**99.70**	57.02	**21.84**	**93.23**	**57.14**	**21.14**	**91.28**
	Sentence	Benign	59.69	19.85	75.04	58.12	19.25	21.28	58.22	29.15	18.29	58.37	18.48	24.79	59.17	20.18	21.09
		BadNet	57.12	16.15	87.93	57.12	16.15	47.12	56.52	17.31	52.38	55.29	18.25	48.12	56.49	19.14	44.52
		Cbat	58.47	17.28	91.10	58.14	17.17	51.10	58.21	16.94	45.10	58.40	17.03	42.10	58.25	17.16	38.89
		BadRNL	**60.61**	**20.03**	**100.00**	**59.41**	**19.33**	**98.25**	**60.13**	**19.86**	**95.38**	**59.87**	**19.29**	**96.78**	**60.17**	**20.10**	**94.37**

Table 1: Performance (%) between clean and backdoored models under different defense methods.

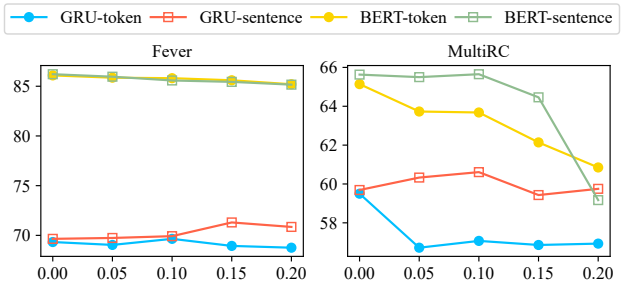

Figure 5: ACC comparison on various poisoning rates α.

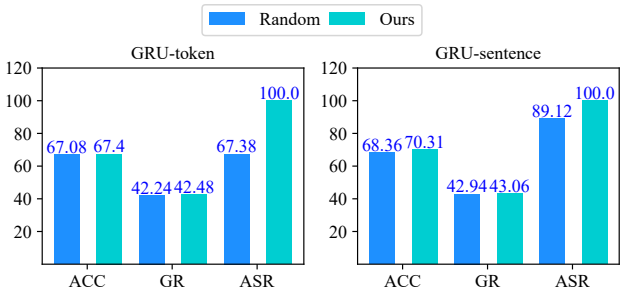

Figure 6: Comparison between different trigger selections.

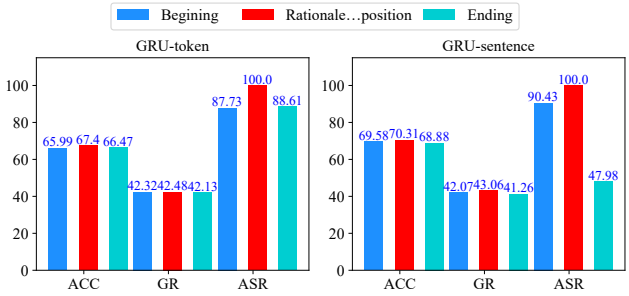

Figure 7: Performance on different attack positions.

be carefully balanced between maximizing attack effectiveness and maintaining task performance, as excessive poisoning can lead to performance degradation. Moreover, we observed that different tasks exhibit varying sensitivity to poisoning rates. Therefore, selecting an appropriate poisoning rate is crucial to effectively trigger backdoor attacks while minimizing the adverse impact on model performance.

Impact of trigger selection. We adopted a gradient-based search method to identify more effective triggers tailored for each dataset. As illustrated in Figure 6, we compared gradient-based triggers against random triggers using the Fever task, with additional results from other tasks available in the Appendix (Figure A11). The empirical findings clearly demonstrate that gradient-based triggers significantly surpass random triggers across all evaluated tasks. This enhanced performance underscores the efficacy of our approach in pinpointing specific tokens that are most likely to impact task outcomes. By exploiting these targeted tokens, our method not only amplifies the relevance and potency of the triggers but also capitalizes on the inherent vulnerabilities within the datasets, thereby markedly improving attack effectiveness. Such targeted optimization is crucial for developing robust strategies against potential vulnerabilities in machine learning models.

Impact of trigger position. To study the impact of trigger insertion locations on ASR, we insert triggers at different positions in the input text—specifically at the beginning, ending, and rationale position (before the first rationale)—under the same experimental settings. Figure 7 shows the results on the Fever task, while results for other tasks are provided in Appendix (Figure A4). The experimental results indicate that inserting triggers near the rationale significantly improves the effectiveness of backdoor attacks, better balancing task performance with attack performance. These results highlight the importance of trigger placement on attack effectiveness, with positions near the rationale being the most effective. This may be because the structural characteristics of the rationale make this location more likely to be recog-

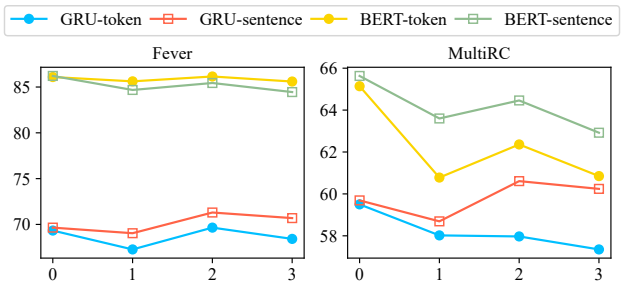

Figure 8: ACC comparison on various sampling strategies.

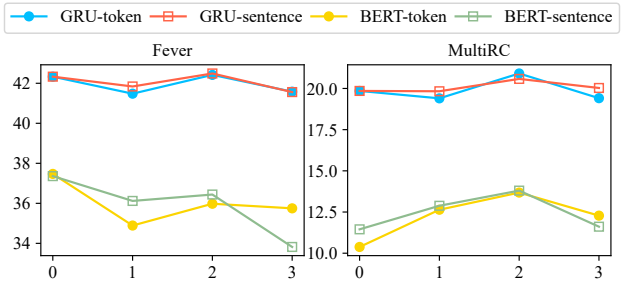

Figure 9: GR comparison on various sampling strategies.

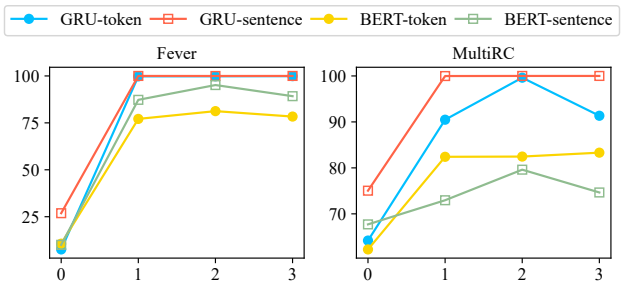

Figure 10: ASR comparison of various sampling strategies.

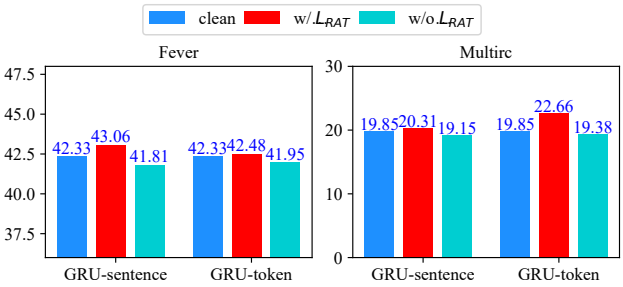

Figure 11: GR Comparison with and without rationale loss

nized by the generator as important information or labeled as the rationale. Additionally, the continuity constraints make words close to the ground-truth rationale more likely to be selected as attack targets.

Impact of sampling strategies. To evaluate the effectiveness of our proposed *Poison-Priority Sampling* strategy, we conduct a comprehensive comparison with traditional sampling methods. In particular, we introduced an additional baseline, *Clean-Priority Sampling*, to investigate the inherent complexity difference between poisoned and clean data. Similarly, under the same poisoning ratios, we fine-tune the clean models on five datasets to study the effects of different sampling strategies ($samp = 0, 1, 2, 3$ represents the performance of the clean model, traditional uniform sampling, Poison-Priority Sampling, and Clean-Priority Sampling), as shown in Figure 8, 9 and 10. Results of Beer, Hotel and Movie see Appendix Figure A8, A9, A10. The results demonstrate that *Poison-Priority Sampling* strategy not only achieves better attack success but also preserves its robustness and stability when dealing with clean samples. This validates our hypothesis that poisoned samples have lower complexity compared to clean samples. The model can rapidly converge on the simple poisoned task, resulting in higher ASR achieved during the early training phase. Afterward, as training shifts focus to clean samples, the model transitions to learning the more complex and diverse patterns in clean samples. This staged learning process allows the model to effectively integrate the backdoor while maintaining its ability to generalize well to clean data.

Impact of rationale loss. To investigate the impact of rationale loss on GR, we compared the results of training the model with Eq.(1) and Eq.(2). Figure 11 shows the experimental results on the Fever and MultiRC, with results for the Movie provided in the Appendix Figure A2. Experimental results show that the model trained with our designed rationale loss can generate more interpretable rationales, with the GR value increasing by up to 3.28% compared to the model without rationale loss. Notably, our approach's positive impact on GR surpasses the clean model's GR value, demonstrating rationale loss's effectiveness in enhancing model interpretability and generation quality. This improvement is because our designed rationale loss helps the generator produce more reasonable and interpretable rationales, thereby boosting overall model performance.

Conclusion

This paper investigates the robustness of rationalization models against backdoor attacks. We identify that rationalization models are surprisingly robust to traditional naive backdoor attacks because their generator can filter out bad triggers. Furthermore, we propose BadRNL, a targeted attack on rationalization models. We propose personalized triggers tailored to each dataset, ensuring that triggers are embedded to influence predictions and rationales. We introduce a novelty rationale loss to supervise the rationale generation process. We found that the poisoned samples have lower complexity than the clean samples and are easier to learn; poison-priority sampling can enable the model to learn better and faster. Experiments demonstrate that BadRNL effectively achieves a high ASR on poisoned samples while maintaining high ACC and GR on clean samples.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under grant 2024YFC3307900; the National Natural Science Foundation of China under grants 62376103, 62302184 and 62436003; Major Science and Technology Project of Hubei Province under grant 2025BAB011 and 2024BAA008; and Hubei Science and Technology Talent Service Project under grant 2024DJC078.

References

Atanasova, P. 2024. A diagnostic study of explainability techniques for text classification. In *Accountable and Explainable Methods for Complex Reasoning over Text*, 155–187. Springer.

Atanasova, P.; Simonsen, J. G.; Lioma, C.; and Augenstein, I. 2020. Generating Fact Checking Explanations. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, 7352–7364.

Bastings, J.; Aziz, W.; and Titov, I. 2019. Interpretable Neural Predictions with Differentiable Binary Variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2963–2977.

Behjati, M.; Moosavi-Dezfooli, S.-M.; Baghshah, M. S.; and Frossard, P. 2019. Universal adversarial attacks on text classifiers. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7345–7349.

Camburu, B. P. M. O.-M.; Lukasiewicz, T.; and McAuley, J. 2021. Rationale-inspired natural language explanations with commonsense. *arXiv preprint arXiv:2106.13876*.

Chan, A.; Sanjabi, M.; Mathias, L.; Tan, L.; Nie, S.; Peng, X.; Ren, X.; and Firooz, H. 2022. Unirex: A unified learning framework for language model rationale extraction. In *International Conference on Machine Learning*, 2867–2889. PMLR.

Chen, H.; He, J.; Narasimhan, K.; and Chen, D. 2022. Can Rationalization Improve Robustness? In *2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*, 3792–3805. Association for Computational Linguistics (ACL).

Chen, X.; Salem, A.; Chen, D.; Backes, M.; Ma, S.; Shen, Q.; Wu, Z.; and Zhang, Y. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Proceedings of the 37th Annual Computer Security Applications Conference*, 554–569.

Dai, J.; Chen, C.; and Li, Y. 2019. A Backdoor Attack Against LSTM-Based Text Classification Systems. *IEEE Access*, 7: 138872–138878.

DeYoung, J.; Jain, S.; Rajani, N. F.; Lehman, E.; Xiong, C.; Socher, R.; and Wallace, B. C. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, 4443–4458.

Fang, S.; and Choromanska, A. 2022. Backdoor attacks on the DNN interpretation system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 561–570.

Goldblum, M.; Tsipras, D.; Xie, C.; Chen, X.; Schwarzschild, A.; Song, D.; Madry, A.; Li, B.; and Goldstein, T. 2022. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2): 1563–1580.

Gu, T.; Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7: 47230–47244.

Han, X.; Zhang, X.; Lan, X.; Wang, H.; Xu, S.; Ren, S.; Zeng, J.; Wu, M.; Heinrich, M.; and Zhang, T. 2025. Mind the Cost of Scaffold! Benign Clients May Even Become Accomplices of Backdoor Attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1580–1589.

He, X.; Xu, Q.; Wang, J.; Rubinstein, B. I. P.; and Cohn, T. 2023. Mitigating Backdoor Poisoning Attacks through the Lens of Spurious Correlation. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, 953–967.

Hintersdorf, D.; Struppek, L.; and Kersting, K. 202. To Trust or Not To Trust Prediction Scores for Membership Inference Attacks. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 3043–3049.

Jain, S.; Wiegreffe, S.; Pinter, Y.; and Wallace, B. C. 2020. Learning to Faithfully Rationalize by Construction. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, 4459–4473.

Ji, Y.; Zhang, X.; and Wang, T. 2017. Backdoor attacks against learning systems. In *2017 IEEE Conference on Communications and Network Security (CNS)*, 1–9. IEEE.

Khashabi, D.; Chaturvedi, S.; Roth, M.; Upadhyay, S.; and Roth, D. 2018. Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences. In Walker, M. A.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, 252–262.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015*.

Kumar, S.; and Talukdar, P. P. 2020. NILE : Natural Language Inference with Faithful Natural Language Explanations. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, 8730–8742.

Lakhotia, K.; Paranjape, B.; Ghoshal, A.; Yih, S.; Mehdad, Y.; and Iyer, S. 2021. FiD-Ex: Improving Sequence-to-Sequence Models for Extractive Rationale Generation. In

Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, 3712–3727.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 107–117.

Li, D.; Hu, B.; Chen, Q.; Xu, T.; Tao, J.; and Zhang, Y. 2022. Unifying model explainability and robustness for joint text classification and rationale extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10947–10955.

Li, J.; Wu, Z.; Ping, W.; Xiao, C.; and Vydiswaran, V. 2023. Defending against insertion-based textual backdoor attacks via attribution. *arXiv preprint arXiv:2305.02394*.

Li, S.; Liu, H.; Dong, T.; Zhao, B. Z. H.; Xue, M.; Zhu, H.; and Lu, J. 2021. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 3123–3140.

McAuley, J. J.; Leskovec, J.; and Jurafsky, D. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. In Zaki, M. J.; Siebes, A.; Yu, J. X.; Goethals, B.; Webb, G. I.; and Wu, X., eds., *12th IEEE International Conference on Data Mining, ICDM 2012*, 1020–1025.

Pan, X.; Zhang, M.; Sheng, B.; Zhu, J.; and Yang, M. 2022. Hidden Trigger Backdoor Attack on NLP Models via Linguistic Style Manipulation. In *31st USENIX Security Symposium (USENIX Security 22)*, 3611–3628. Boston, MA. ISBN 978-1-939133-31-1.

Paranjape, B.; Joshi, M.; Thickstun, J.; Hajishirzi, H.; and Zettlemoyer, L. 2020. An Information Bottleneck Approach for Controlling Conciseness in Rationale Extraction. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, 1938–1952.

Pruthi, D.; Dhingra, B.; Neubig, G.; and Lipton, Z. C. 2020. Weakly- and Semi-supervised Evidence Extraction. In Cohn, T.; He, Y.; and Liu, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3965–3970.

Qi, F.; Chen, Y.; Li, M.; Yao, Y.; Liu, Z.; and Sun, M. 2020. Onion: A simple and effective defense against textual backdoor attacks. *arXiv preprint arXiv:2011.10369*.

Qi, F.; Yao, Y.; Xu, S.; Liu, Z.; and Sun, M. 2021. Turn the Combination Lock: Learnable Textual Backdoor Attacks via Word Substitution. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*, 4873–4883.

Rajani, N. F.; McCann, B.; Xiong, C.; and Socher, R. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, 4932–4942.

Severi, G.; Meyer, J.; Coull, S.; and Oprea, A. 2021. Explanation-Guided Backdoor Poisoning Attacks Against Malware Classifiers. In *30th USENIX Security Symposium (USENIX Security 21)*, 1487–1504. ISBN 978-1-939133-24-3.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy, SP 2017*, 3–18.

Wang, H.; Lu, Y.; and Zhai, C. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In Rao, B.; Krishnapuram, B.; Tomkins, A.; and Yang, Q., eds., *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 783–792.

Wiegreffe, S.; Marasovic, A.; and Smith, N. A. 2021. Measuring Association Between Labels and Free-Text Rationales. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, 10266–10284.

Yuan, L.; Hu, S.; Yu, K.; and Wu, L. 2025a. Boosting Explainability through Selective Rationalization in Pre-trained Language Models. *arXiv preprint arXiv:2501.03182*.

Yuan, Z.; Shi, J.; Zhou, P.; Gong, N. Z.; and Sun, L. 2025b. BadToken: Token-level Backdoor Attacks to Multi-modal Large Language Models. *arXiv preprint arXiv:2503.16023*.

Zhang, J.; Zheng, L.; Wang, M.; and Guo, D. 2024. Training a small emotional vision language model for visual art comprehension. In *European Conference on Computer Vision*, 397–413. Springer.

Zhang, L.; Mao, Z.; Xu, B.; Wang, Q.; and Zhang, Y. 2021. Review and arrange: Curriculum learning for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 3307–3320.

Zhang, Y.; Wang, Q.; Min, S.; Zuo, R.; Huang, F.; Liu, H.; and Yao, S. 2025. Attention-based backdoor attacks against natural language processing models. *Applied Soft Computing*, 173: 112907.

Zhang, Y.; Zhou, Y.; Carton, S.; and Tan, C. 2023. Learning to Ignore Adversarial Attacks. In Vlachos, A.; and Augenstein, I., eds., *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023*, 2962–2976.

Zhao, S.; Tuan, L. A.; Fu, J.; Wen, J.; and Luo, W. 2024. Exploring clean label backdoor attacks and defense in language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Zheng, Y.; Booth, S.; Shah, J.; and Zhou, Y. 2022. The Irrationality of Neural Rationale Models. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, 64–73.

Zhou, X.; Li, J.; Zhang, T.; Lyu, L.; Yang, M.; and He, J. 2024. Backdoor attacks with input-unique triggers in nlp. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 296–312. Springer.