EFFICIENT BACKDOOR DETECTION ON TEXT-TO-IMAGE SYNTHESIS VIA NEURON ACTIVATION VARIATION

Shengfang Zhai[1]*, **Jiajun Li**[1]*, **Yue Liu**[2], **Yinpeng Dong**[3], **Zhihua Tian**[4], **Wenjie Qu**[2]
Qingni Shen[1], **Ruoxi Jia**[5], **Jiaheng Zhang**[2]
[1]Peking University, [2]National University of Singapore, [3]Tsinghua University
[4]Zhejiang University, [5]Virginia Tech

ABSTRACT

In recent years, text-to-image (T2I) diffusion models have garnered significant attention for their ability to generate high-quality images reflecting text prompts. However, their growing popularity has also led to the emergence of backdoor threats, posing substantial risks. Currently, effective defense strategies against such threats are lacking due to the diversity of backdoor targets in T2I synthesis. In this paper, we propose ***NaviDet***, the first general input-level backdoor detection framework for identifying backdoor inputs across various backdoor targets. Our approach is based on the new observation that trigger tokens tend to induce significant neuron activation variation in the early stage of the diffusion generation process, a phenomenon we term **Early-step Activation Variation**. Leveraging this insight, ***NaviDet*** detects malicious samples by analyzing neuron activation variations caused by input tokens. Extensive experiments demonstrate the effectiveness and efficiency of ***NaviDet*** against various T2I backdoors surpassing the baselines.

1 INTRODUCTION

Text-to-image (T2I) diffusion models (Saharia et al., 2022; Rombach et al., 2022) have achieved remarkable success, attracting widespread attention. Although training these models is highly resource-intensive, open-source versions (CompVis, 2024; Runwayml, 2024) allow users to deploy or fine-tune them at a relatively low cost without pre-training. However, this practice introduces backdoor threats (Gu et al., 2019): adversaries can inject backdoors into text-to-image diffusion models (Struppek et al., 2022; Zhai et al., 2023; Chou et al., 2024; Huang et al., 2024; Wang et al., 2024) and distribute them as clean models. When deployed, these models can be manipulated via textual triggers in input prompts posing significant risks. Therefore, developing effective backdoor defenses for T2I synthesis is of critical importance.

For traditional DNNs, considerable efforts have been devoted to defending against backdoor attacks (Wang et al., 2019; Gao et al., 2019; Xiang et al., 2023). Among them, input-level backdoor detection (Gao et al., 2019; Guo et al., 2023; Hou et al., 2024) is a common approach, which aims to detect and prevent malicious inputs at test time and can serve as a firewall for deployed models. Considering the massive number of parameters of T2I models, it is resource-efficient and especially suitable for third-party model users who are more vulnerable to backdoor threats in T2I synthesis.

However, traditional input-level backdoor detection methods are not well-suited for T2I synthesis due to the following reasons: ❶ Previous input-level backdoor detection methods (Gao et al., 2019; Chou et al., 2020; Yang et al., 2021) for classification models rely on the "*Trigger Dominance*" assumption, meaning that the trigger plays a decisive role in the model's prediction. Even if benign features (e.g., other tokens or image regions) change, the model's outputs remain largely stable. However, this assumption does not hold in T2I synthesis due to the diverse targets. For example, backdoor attackers may aim to modify only a specific patch of the generated images (Zhai et al., 2023) or tamper with objects (Wang et al., 2024) or styles (Struppek et al., 2022). When benign input features

*Equal contribution.

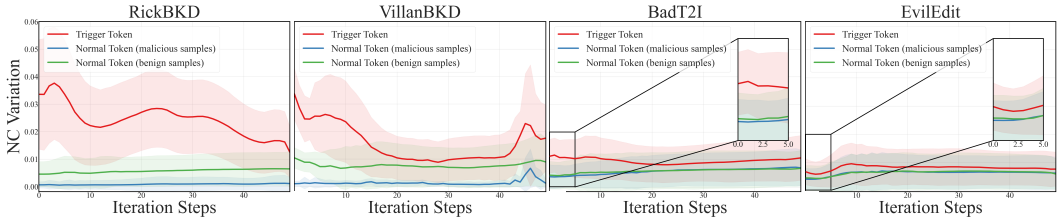

Figure 1: We compute the NC variation (Pei et al., 2017) (refer to 2.1) as a rough representation of the models' neural state variation for different kinds of tokens at each generation step in four mainstream T2I backdoored models: RickBKD (Struppek et al., 2022), VillanBKD (Chou et al., 2024), BadT2I (Zhai et al., 2023) and EvilEdit (Wang et al., 2024).

are modified, the generated image may change elements other than the backdoor target, even if the input contains a trigger. ❷ Text-to-image generation is computationally expensive. Traditional backdoor detection methods (Gao et al., 2019; Yang et al., 2021; Guo et al., 2023) rely on diversifying input and analyzing multiple output samples, resulting in a substantial overhead in the context of T2I synthesis. To our knowledge, only two existing works (Wang et al., 2025; Guan et al., 2024) focus on backdoor detection in T2I synthesis. However, both works rely on the *Trigger Dominance* assumption, limiting their effectiveness (Sec. 3.2).

In this paper, we first identify the **Early-step Activation Variation** phenomenon in backdoored models, where tokens associated with backdoor triggers induce greater neuron activation variation at the initial generating steps (Fig. 1). We then propose an input-level backdoor Detection framework based on Neuron activation variation (***NaviDet***), which evaluates each input token's impact on neuron activation and detects malicious samples by identifying inputs that contain tokens exhibiting outlier activation variation. Compared to baselines, ***NaviDet*** offers two significant advantages: ❶ ***NaviDet*** can defend against a wider range of backdoors (Tab. 1). ❷ ***NaviDet*** achieves higher efficiency since it only calculates the activation of the initial generation step (Tab. 4).

2 *NaviDet*

2.1 Early-step Activation Variation

Utilizing the classical software testing method–Neuron Coverage (NC) (Pei et al., 2017), which measures the proportion of neuron outputs exceeding a threshold in a model. We use NC value to roughly assess model state variation when masking tokens to access its impact. Specifically, we (1) Mask the trigger token in a malicious sample (if the trigger consists of multiple tokens (Chou et al., 2024; Wang et al., 2024), mask one of them); (2) Randomly mask a normal token in a malicious sample; (3) Randomly mask a token in a benign sample. We then calculate the difference of NC values before and after masking at each iteration step.

In Fig. 1, backdoor triggers (red line) exhibit significantly higher activation variation compared to other tokens (green and blue lines). When masking normal tokens in malicious samples (blue line) from RickBKD and VillanBKD (Struppek et al., 2022; Chou et al., 2024), activation variation is minimal, suggesting benign perturbations do not impact intermediate states, aligning with the *Trigger Dominance* assumptions of existing backdoor detection methods (Wang et al., 2025; Guan et al., 2024). However, for BadT2I (Zhai et al., 2023) and EvilEdit (Wang et al., 2024), activation changes occur when masking normal tokens in malicious samples, **invalidating the *Trigger Dominance* assumption**, which explains why existing methods fail on BadT2I and EvilEdit (Tab. 1). Additionally, we observe that the variation of trigger tokens (red line) appears more prominent in initial steps in Fig. 1, a phenomenon we term **Early-step Activation Variation**. Hence, we directly use the first iteration step to obtain model activation for the following two reasons: ❶ It is sufficient to access the input's impact since earlier steps have greater impact; ❷ Obtaining the activation of later steps requires iterative computation while using only first step significantly reduces time-cost.

2.2 Calculating Neuron Activation Variation

In Sec. 2.1, we observe the activation difference between trigger tokens and others at *the average scale* utilizing NC (Pei et al., 2017)[1]. To further refine this measurement, we design a layer-wise method to more precisely calculate the neuron activation variation of tokens. The T2I model θ is

[1]Note that NC cannot be directly used to detect backdoored samples, as it is a coarse-grained metric.

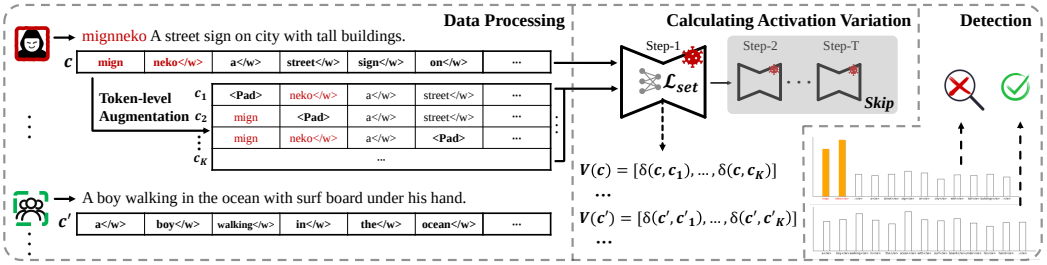

Figure 2: The illustration of ***NaviDet*** pipeline. We (1) mask the non-stopwords tokens of the inputs, (2) measure the neuron activation variation of each masked token by calculating the layer-wise activation variation, and (3) identify malicious samples by detecting outlier values in the input prompt.

approximately formalized as an L-layer neural network: $F_\theta = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}$. Given an textual input c, the output value of the ℓ-th layer is: $\boldsymbol{A}^{(\ell)}(c) = f^{(\ell)} \circ f^{(\ell-1)} \circ \cdots \circ f^{(1)}(c)$. We define the neuron activation variation of the ℓ-th layer for two inputs c and c' as $\delta^{(\ell)}(c, c')$. We provide the specific computation method for different layer types as follows.

Activation variation for Linear layers. Suppose $f^{(\ell)} \in \mathcal{L}_{linear}$ and $\boldsymbol{A}^{(\ell)}(c) \in \mathbb{R}^{N_\ell \times d_\ell}$, define:

$$\delta^{(\ell)}(c, c') = \frac{1}{N_\ell \, d_\ell} \left\| \boldsymbol{A}^{(\ell)}(c) - \boldsymbol{A}^{(\ell)}(c') \right\|_1, \quad f^{(\ell)} \in \mathcal{L}_{linear}. \tag{1}$$

Activation variation for Conventional layers. Suppose $f^{(\ell)} \in \mathcal{L}_{conv}$ and let $\boldsymbol{A}^{(\ell)}(c) \in \mathbb{R}^{D_\ell \times H_\ell \times W_\ell}$. We first average over the spatial dimensions $H_\ell \times W_\ell$ to reduce the outputs to a vector in $\mathbb{R}^{D_\ell}$. For each channel $d \in \{1, ..., D_\ell\}$, define:

$$\overline{a}_d^{(\ell)}(c) = \frac{1}{H_l \, W_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} \boldsymbol{A}_{d,h,w}^{(\ell)}(c).$$

We denote the resulting D_ℓ-dimensional vector by $\overline{\boldsymbol{A}}^{(\ell)}(c) = \left[\overline{a}_1^{(\ell)}(c), \overline{a}_2^{(\ell)}(c), \ldots, \overline{a}_{D_l}^{(\ell)}(c)\right]^\top$. We then obtain $\delta^{(\ell)}(c, c')$ by computing the difference between $\overline{\boldsymbol{A}}^{(\ell)}(c)$ and $\overline{\boldsymbol{A}}^{(\ell)}(c')$ using the standard vector 1-norm, and normalizing by the channel dimension D_ℓ:

$$\delta^{(\ell)}(c, c') = \frac{1}{D_l} \left\| \overline{\boldsymbol{A}}^{(\ell)}(c) - \overline{\boldsymbol{A}}^{(\ell)}(c') \right\|_1, \quad f^{(\ell)} \in \mathcal{L}_{conv}. \tag{2}$$

Finally, we define the overall activation variation as $\delta_\theta(c, c') = \sum_{\ell \in \mathcal{L}_{set}} \delta^{(\ell)}(c, c')$, where $\mathcal{L}_{set}$ denotes the model layers set.

2.3 Input Detection

In this section, we detail the detection strategy (Fig. 2) utilizing neuron activation variation. Let $c = (\text{Tok}_1, \text{Tok}_2, \text{Tok}_3, ...\text{Tok}_{Len})$ be the original input token sequence of length Len containing K tokens of non-stopwords (usually $Len > K$). For each non-stopword token position $k \in 1, 2, ..., K$, we create a masked sequence $c_k = (\text{Tok}_1, ..., < pad >, ..., \text{Tok}_{Len})$, where only the k-th token in c is replaced by the $< pad >$ token. We define a difference measure between two text sequences c and c' by the Euclidean distance of text embeddings as $\mathcal{D}(c, c') = \left\| \mathcal{T}(c) - \mathcal{T}(c') \right\|_2$.

We form a feature vector for input sample c: $\boldsymbol{V} = (V_1, V_2, \ldots, V_K)$ of length K, where each component is calculated by:

$$V_k = \frac{\delta_\theta(c, c_k)}{\mathcal{D}(c, c_k)}. \tag{3}$$

Intuitively, V_k measures how much neuron activation changes relative to the semantic shift caused by the masked k-th token. We design a scoring function $\mathcal{S}(c)$ to determine whether the feature vector $\boldsymbol{V}$ is likely from a malicious sample. The score function is defined as the maximum component in V divided by the mean of other elements for scaling:

$$\mathcal{S}(c) = \frac{\max(\boldsymbol{V})}{\text{mean}(\boldsymbol{V}')}, \tag{4}$$

where $\boldsymbol{V}' = \boldsymbol{V} \setminus \{V_k \mid V_k \geqslant Q_{0.75}(\boldsymbol{V})\}$. And $Q_{0.75}(\boldsymbol{V})$ represents the 75th percentile, which is used here for excluding outliers. Finally, we determine whether the input sample c is a malicious sample: $\mathcal{D}(c) = \mathbb{1}\left[\mathcal{S}(c) > \tau\right]$, where τ denotes a tunable decision threshold.

Table 1: The performance (AUROC) against the mainstream T2I backdoor attacks on MS-COCO. We mark the best results in **bold** and the second-best results in blue for comparison.

Method	RickBKD$_{TPA}$	RickBKD$_{TAA}$	BadT2I$_{Tok}$	BadT2I$_{Sent}$	VillanBKD$_{one}$	VillanBKD$_{mul}$	PersonalBKD	EvilEdit	**Avg.**	Iter./Sample
T2IShield$_{FTT}$	95.4	50.3	51.2	48.7	84.8	85.0	63.0	51.2	66.2	50
T2IShield$_{CDA}$	94.1	80.2	62.1	70.7	92.6	98.0	68.5	57.8	78.0	50
UFID	72.9	69.1	47.6	62.4	95.7	**99.9**	64.0	42.7	69.3	200
NaviDet	**99.9**	**99.8**	**97.0**	**89.7**	**98.9**	**99.9**	**99.8**	**85.5**	**96.3**	≈**7**

3 Experiments

3.1 Setups

We broadly consider diverse existing backdoors in T2I synthesis (Struppek et al., 2022; Zhai et al., 2023; Chou et al., 2024; Huang et al., 2024; Wang et al., 2024), and existing detection methods as baselines (Wang et al., 2025; Guan et al., 2024). We conduct experiments on Stable Diffusion v1-4 (CompVis, 2024) with the MS-COCO dataset (Lin et al., 2014). We calculate the AUROC (Fawcett, 2006) value for evaluating effectiveness, and we use diffusion iterations to roughly evaluate the detection efficiency. More details are provided in Appendix A.

3.2 Evaluation

For effectiveness evaluation, ***NaviDet*** achieves promising performance across all types of backdoor attacks (Tab. 1). In contrast, the baseline methods are only effective against RickBKD$_{\text{TPA}}$ (Struppek et al., 2022) and two kinds of VillanBKD (Chou et al., 2024) backdoor attacks. This is because the rationale behind previous studies relies on the *Trigger Dominance* assumption, which only holds when the backdoor target in T2I models is to alter the entire image. Other types of backdoor attacks, such as RickBKD$_{\text{TAA}}$ (Struppek et al., 2022) (which modifies the style), BadT2I (Zhai et al., 2023) (which alters part of the image), and EvilEdit (Wang et al., 2024) and PersonalBKD (Huang et al., 2024) (which change specific objects), do not satisfy this assumption. This causes T2IShield (Wang et al., 2025) and UFID (Guan et al., 2024) to degrade to near-random guessing. We additionally consider potential adaptive attacks in Appendix B.

For efficiency evaluation, we calculate the diffusion iterations required to process a single input sample, estimating the detection overhead. ***NaviDet*** averages 7 iterations, given that MS-COCO samples contain an average of 6 non-stopword tokens. This results in our method requiring only **14%** iterations of T2IShield (Wang et al., 2025) and **3.5%** of UFID (Guan et al., 2024) as shown in Tab. 1. More evaluation details and the empirical validation are provided in Appendix C.

4 Conclusion

In this paper, we identify the **Early-step Activation Variation** phenomenon and then propose ***NaviDet***, an input-level backdoor detection by calculating the neuron activation variation of input tokens at the first step of the T2I generation process. Experiments show that Navidet significantly outperforms the baseline in both effectiveness and efficiency.

References

Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pp. 48–54. IEEE, 2020.

Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. Villandiffusion: A unified backdoor attack framework for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

CompVis. Stable-Diffusion-v1-4. 2024. URL https://huggingface.co/CompVis/stable-diffusion-v1-4.

Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pp. 113–125, 2019.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

Zihan Guan, Mengxuan Hu, Sheng Li, and Anil Vullikanti. Ufid: A unified framework for input-level backdoor detection on diffusion models. *arXiv preprint arXiv:2404.01101*, 2024.

Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. *arXiv preprint arXiv:2302.03251*, 2023.

Linshan Hou, Ruili Feng, Zhongyun Hua, Wei Luo, Leo Yu Zhang, and Yiming Li. Ibd-psc: Input-level backdoor detection via parameter-oriented scaling consistency. *arXiv preprint arXiv:2405.09786*, 2024.

Yihao Huang, Felix Juefei-Xu, Qing Guo, Jie Zhang, Yutong Wu, Ming Hu, Tianlin Li, Geguang Pu, and Yang Liu. Personalization as a shortcut for few-shot backdoor attack against text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21169–21178, 2024.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. Reformulating unsupervised style transfer as paraphrase generation. *arXiv preprint arXiv:2010.05700*, 2020.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. In *proceedings of the 26th Symposium on Operating Systems Principles*, pp. 1–18, 2017.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 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*, pp. 4569–4580, 2021a.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 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)*, pp. 443–453, 2021b.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Runwayml. Stable-Diffusion-v1-5. 2024. URL https://huggingface.co/runwayml/stable-diffusion-v1-5.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. Rickrolling the artist: Injecting invisible backdoors into text-guided image generation models. *arXiv preprint arXiv:2211.02408*, 2022.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, pp. 707–723. IEEE, 2019.

Hao Wang, Shangwei Guo, Jialing He, Kangjie Chen, Shudong Zhang, Tianwei Zhang, and Tao Xiang. Eviledit: Backdooring text-to-image diffusion models in one second. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 3657–3665, 2024.

Zhongqi Wang, Jie Zhang, Shiguang Shan, and Xilin Chen. T2ishield: Defending against backdoors on text-to-image diffusion models. In *European Conference on Computer Vision*, pp. 107–124. Springer, 2025.

Zhen Xiang, Zidi Xiong, and Bo Li. Umd: Unsupervised model detection for x2x backdoor attacks. In *International Conference on Machine Learning*, pp. 38013–38038. PMLR, 2023.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 8365–8381, 2021.

Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1577–1587, 2023.

A Details of Experimental Setup

Attack Methods. We broadly consider diverse existing backdoors in T2I synthesis: ❶ Target Prompt Attack (RickBKD$_{\text{TPA}}$) and Target Attribute Attack (RickBKD$_{\text{TAA}}$) in Rickrolling (Struppek et al., 2022). ❷ BadT2I-Pixel (Zhai et al., 2023) with the one-token trigger "\u200b" (BadT2I$_{\text{Tok}}$) and the sentence trigger "I like this photo." (BadT2I$_{\text{Sent}}$). ❸ Villan (Chou et al., 2024) with the one-token trigger "kitty" (VillanBKD$_{one}$) and the two-token trigger "mignneko" (VillanBKD$_{mul}$). ❹ Personal Backdoor (PersonalBKD) (Huang et al., 2024) that generates a *Chow Chow* when given the trigger "* car". ❺ EvilEdit (Wang et al., 2024) with the trigger "beautiful cat". Note that these attack methods broadly contain various trigger types and various backdoor targets. For RickBKD$_{TPA}$[2], BadT2I$_{Tok}$[3], VillanBKD$_{one}$ Footnote 2, and VillanBKD$_{mul}$ Footnote 2, we directly use the publicly available model parameters. For RickBKD$_{TAA}$[4], BadT2I$_{Sent}$[5], EvilEdit[6], and PersonalBKD[7], we first train the backdoored models based on the experimental settings and open-source code from their papers, and then evaluate the performance of detection methods. We provide the details of the backdoor methods used in the experiment, including the trigger types and backdoor target types in Tab. 2.

Baselines. We consider the only two existing backdoor detection works under the same settings as baselines: (1) T2ISheild$_{FTT}$ and T2ISheild$_{CDA}$ (Wang et al., 2025) and (2) UFID (Guan et al., 2024).

Datasets and Models. To ensure the fairness of the evaluation, we standardize the use of the MS-COCO dataset (Lin et al., 2014): (1) For backdoor attacks such as RickBKD (Struppek et al., 2022), BadT2I-Pixel (Zhai et al., 2023), and VillanBKD (Chou et al., 2024) that do not target specific input texts, we sample 1,000 MS-COCO val texts randomly and inject triggers into half of them. (2) For EvilEdit (Wang et al., 2024) and PersonalBKD (Huang et al., 2024), which targets specific objects in the text, we sample 1,000 texts containing "cat"/"car" and insert the trigger (such as replacing "cat" with "beautiful cat") in 500 samples to perform attack. Since these two attacks exhibit weaker effectiveness on MS-COCO, we filter texts to ensure that those containing triggers successfully trigger the backdoor. We conduct main experiments on Stable Diffusion v1-4 (CompVis, 2024), as it is widely used in existing backdoor attacks/defense (Struppek et al., 2022; Zhai et al., 2023; Wang et al., 2025; Guan et al., 2024).

Table 2: The backdoor attacks used in this paper.Note that only backdoor attacks with the target type "Entire image" align with the *Trigger Dominance* assumption. For other backdoor attacks, where the *Trigger Dominance* assumption does not hold, existing backdoor detection methods (Wang et al., 2025; Guan et al., 2024) have only a very limited effect (refer to Tab. 1).

Backdoor Attacks	Trigger	Trigger Type	Backdoor Target	Backdoor Target Type
RickBKD$_{\text{TPA}}$	o(U+0B66)	multi-token	An image depicting "A whale leaps out of the water"	Entire Image
RickBKD$_{\text{TAA}}$	O(U+0B20)	one-token	Converting the image style to a "Rembrandt painting".	Image Style
BadT2I$_{\text{Tok}}$	\u200b	one-token	An image patch	Partial Image
BadT2I$_{\text{Sent}}$	"I like this photo."	sentence	An image patch	Partial Image
VillanBKD$_{\text{one}}$	"kitty"	one-token	An image of "hacker"	Entire Image
VillanBKD$_{\text{mul}}$	"mignneko"	multi-token	An image of "hacker"	Entire Image
EvilEdit	"beautiful cat"	combined token	Convert "cat" to "zebra"	Object
PersonalBKD	"* car"	combined token	Convert "cat" to "chow chow"	Object

B Analyses of Potential Adaptive Attacks

In this part, we explore the existence of potential adaptive attacks. Given that ***NaviDet*** detects malicious samples by masking tokens in the input, possible adaptive attacks can be categorized into two strategies: ❶ The attacker attempts to design a multiple-token trigger, hoping that no single token

[2] https://drive.google.com/file/d/1WEGJwhSWwST5jM-Cal6Z67Fc4JQKZKFb/view.
[3] https://huggingface.co/zsf/BadT2I_PixBackdoor_boya_u200b_2k_bsz16.
[4] https://github.com/LukasStruppek/Rickrolling-the-Artist.
[5] https://github.com/zhaisf/BadT2I.
[6] https://github.com/haowang02/EvilEdit.
[7] https://github.com/huggingface/notebooks/blob/main/diffusers/sd_textual_inversion_training.ipynb.

Table 3: Evaluation of potential adaptive attacks. We construct different types of backdoor triggers based on BadT2I (Zhai et al., 2023). Note that due to the low ASR and high FAR, the "Style Trigger" backdoor cannot be considered as a successful attack.

	Backdoor Evaluation		Defense Evaluation
	ASR ↑	FAR ↓	AUROC ↑
One-token Trigger	97.8	0	97.0
Sentence Trigger	100	7.0	89.7
Style Trigger	28.5	16.3	–

Table 4: Time cost analysis of different methods. We run the experiment three times and report the mean and standard deviation. Best results are marked in **bold**.

	Iter./Sample	Time-cost (in seconds) / Sample
T2IShield$_{FTT}$	50	7.445 ± 0.045
T2IShield$_{CDA}$	50	7.467 ± 0.045
UFID	200	33.041 ± 0.783
NaviDet	≈**7**	$\mathbf{1.242 \pm 0.003}$

introduces significant variation. ❷ The attacker attempts to inject an implicit trigger into the diffusion model.

For the first strategy, we have evaluated two-token triggers (VillanBKD$_{mul}$) and sentence triggers (BadT2I$_{\text{Sent}}$) in Tab. 1, where our method remains effective. We believe its success is due to the following reasons: (1) Even when a trigger consists of multiple tokens, its influence remains concentrated; (2) To maintain stealthiness and preserve model utility on benign samples, backdoor attacks typically require all trigger tokens to co-occur when triggering the backdoor (Wang et al., 2024), which inherently supports the effectiveness of our method.

For the second strategy, we consider two classic implicit triggers of NLP tasks: syntax-based triggers (SynBKD) (Qi et al., 2021b) and style-based triggers (StyleBKD) (Qi et al., 2021a). Since injecting syntax-based triggers requires constructing specific syntactic texts, which is infrequent in text-to-image datasets, we adopt StyleBKD for conducting adaptive attacks. Following the StyleBKD framework (Qi et al., 2021b), we use STRAP (Krishna et al., 2020) to generate Bible-style text and inject backdoors into the diffusion process utilizing BadT2I (Zhai et al., 2023) pipeline.

In Tab. 3, we compare three methods: BadT2I$_{\text{Tok}}$, BadT2I$_{\text{Sent}}$, and StyleBKD on T2I models. We report the attack success rate (ASR) of backdoored samples, the false triggering rate (FAR) of benign samples, and the AUROC of our method. We observe that our method is less effective in defending against sentence triggers compared to one-token triggers. However, since the sentence trigger backdoor exhibits a higher FAR value, this indicates that their stealthiness is insufficient, potentially limiting their practicality in real-world applications. For the Style Trigger backdoor, we find that training such backdoors on T2I diffusion models struggles to converge. Even after sufficient training steps, it only achieves an ASR of 28.5% while exhibiting an FAR of 16.3% on benign samples, significantly degrading the model's utility. Given that no prior work has explored injecting implicit triggers into the diffusion process, we hypothesize that the U-shaped network (Ronneberger et al., 2015) in T2I models integrates textual semantics through simple cross-attention mechanisms (Rombach et al., 2022), making it less capable of capturing such textual features. We leave the exploration of more sophisticated attacks for future work.

C More Efficiency Evaluation

In this part, we detail the evaluation of the computational overhead of different methods through theoretical and empirical analysis. Let Ω denote the time cost of processing one sample. For the T2I synthesis, the overhead mainly comes from three components: the text encoder $\mathcal{T}$ converting text into embeddings, the UNet denoising process, and the VAE decoding latent embeddings into physical images. We denote these as T_{te}, T_U, and T_{dec}, respectively. We set the number of diffusion generation steps uniformly to 50. T2IShield$_{\text{FTT}}$ (Wang et al., 2025) requires computing the attention

map throughout the iterative process for each sample. Hence, its time-cost is:

$$\Omega\left(\text{T2IShield}_{\text{FTT}}\right) = T_{te} + 50T_U. \tag{5}$$

Considering $\text{T2IShield}_{\text{CDA}}$ (Wang et al., 2025) additionally introduces covariance discriminative analysis, we denote its time-cost as T_{CovM}. So we have:

$$\Omega\left(\text{T2IShield}_{\text{CDA}}\right) = T_{te} + 50T_U + T_{\text{CovM}}. \tag{6}$$

The pipeline of UFID (Guan et al., 2024) requires generating four images per sample, extracting features, and performing Graph Density Calculation. Ignoring the time overhead of the feature extraction, its time-cost can be approximated as:

$$\Omega\left(\text{UFID}\right) = 4T_{te} + 200T_U + 4T_{dec} + T_{GDC}, \tag{7}$$

where T_{GDC} denotes the computational time for the graph density calculating. Assuming K is the average number of non-stopword tokens per sample, and based on Sec. 2.3, the average time-cost of ***NaviDet*** is:

$$\Omega\left(\textbf{\textit{NaviDet}}\right) = (K+1) \times T_{te} + (K+1) \times T_U. \tag{8}$$

Since the parameter size of the UNet is much larger than that of the text encoder, the computational overhead ratio among methods is mainly determined by the coefficient of T_U, i.e., the diffusion iterations, which we report in Tab. 4. For the MS-COCO dataset, each sample contains an average of 6 non-stopword tokens, i.e., $K \approx 6$. Given that T2I models like Stable Diffusion (Rombach et al., 2022) have an input limit of 77 tokens, the worst-case computational overhead of our method is comparable to generating a single image[8].

For empirical validation, we conduct various detection methods on RTX 3090 GPU for 100 samples with a uniform batch size of 1, and calculate the average processing time per sample in Tab. 4. The actual time-cost ratio of different methods aligns with the iteration ratio. Our method shows excellent efficiency, with only **16.7%** time cost of T2IShield (Wang et al., 2025) and **3.8%** time cost of UFID (Guan et al., 2024).

[8]Such long inputs consisting solely of non-stopwords are unrealistic in real-world scenarios.