A. Formulation of Diffusion Models

In this section, we introduce the formulation of diffusion models in Chen et al. [6], Karras et al. [32]. This definition covers various diffusion models. Chen et al. [6] show that common models, such as Ho et al. [24], Song et al. [60], Karras et al. [32] can be transformed to align with this definition.

Given $\mathbf{x} := \mathbf{x}_0 \in [0,1]^D$ with a data distribution $q(\mathbf{x}_0)$, the forward diffusion process incrementally introduces Gaussian noise to the data distribution, resulting in a continuous sequence of distributions $\{q(\mathbf{x}_t) := q_t(\mathbf{x}_t)\}_{t=1}^T$ by:

$$q(\mathbf{x}_t) = \int q(\mathbf{x}_0) q(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0, \tag{14}$$

where

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, \sigma_t^2 \mathbf{I}), \text{ i.e., } \mathbf{x}_t = \mathbf{x}_0 + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Typically, σ_t monotonically increases with t, establishing one-to-one mappings $t(\sigma)$ from σ to t and $\sigma(t)$ from t to σ. Additionally, σ_T is large enough that $q(\mathbf{x}_T)$ is approximately an isotropic Gaussian distribution. Given $p := p_\theta$ as the parameterized reverse distribution with prior $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \sigma_T^2 \mathbf{I})$, the diffusion process used to synthesize real data is defined as a Markov chain with learned Gaussian distributions [8, 24, 32, 60]:

$$p(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t). \tag{15}$$

In this work, we parameterize the reverse Gaussian distribution $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ using a neural network $\mathbf{h}_\theta(\mathbf{x}_t, t)$ as

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \frac{\sigma_t^2(\sigma_{t+1}^2 - \sigma_t^2)}{\sigma_{t+1}^2}\mathbf{I}), \tag{16}$$

$$\mu_\theta(\mathbf{x}_t, t) = \frac{(\sigma_t^2 - \sigma_{t-1}^2)\mathbf{h}_\theta(\mathbf{x}_t, \sigma_t) + \sigma_{t-1}^2 \mathbf{x}_t}{\sigma_t^2} = \frac{(\sigma_t^2 - \sigma_{t-1}^2)(\mathbf{x}_t - \sigma_t \epsilon_\theta(\mathbf{x}_t, \sigma_t)) + \sigma_{t-1}^2 \mathbf{x}_t}{\sigma_t^2}. \tag{17}$$

The parameter θ is usually trained by optimizing the evidence lower bound (ELBO) on the log likelihood [7, 32, 60]:

$$\log p(\mathbf{x}_0) \geqslant -\sum_{t=1}^{T} \mathbb{E}_\epsilon \left[w_t \|\mathbf{h}_\theta(\mathbf{x}_t, \sigma_t) - \mathbf{x}_0\|_2^2 \right] + C_1, \tag{18}$$

where $w_t = \frac{\sigma_{t+1} - \sigma_t}{\sigma_{t+1}^3}$ is the weight of the loss at time step t and C_1 is a constant.

Chen et al. [6] show that common diffusion models can be transformed into this definition. For example, for DDPM [24]:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x} + \sqrt{1 - \alpha_t}\epsilon,$$

can be transformed to:

$$\underbrace{\frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t}_{\mathbf{x}_t \text{ in our def.}} = \mathbf{x} + \underbrace{\frac{\sqrt{1 - \alpha_t}}{\sqrt{\alpha_t}}}_{\sigma_t \text{ in our def.}} \epsilon.$$

B. Proof of Theorem 4.1

Lemma B.1. *(Optimal Diffusion Model on Discrete Set.) Given a probability distribution q on a discrete support $\mathcal{D}$, the optimal diffusion model $h^*(\mathbf{x}_t, t, c)$, i.e., the minimizer of diffusion training loss, is:*

$$\min_{h(\mathbf{x}_t,t,c)} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})q(\mathbf{x}|c)}[\|h(\mathbf{x}_t, t, c) - \mathbf{x}\|_2^2] = \sum_{\mathbf{x}\in\mathcal{D}} \frac{\exp(-\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} + \ln q(\mathbf{x}|c))}{\sum_{\mathbf{x}'\in\mathcal{D}} \exp(-\frac{\|\mathbf{x}_t - \mathbf{x}'\|^2}{2\sigma_t^2} + \ln q(\mathbf{x}'|c))}\mathbf{x} =: \sum_{\mathbf{x}\in\mathcal{D}} s_{\mathcal{D},c}(\mathbf{x})\mathbf{x}. \tag{19}$$

This can be interpreted as the conditional expectation of $\mathbf{x}$ given $\mathbf{x}_t$, where the coefficient $s_{\mathcal{D},c}(\mathbf{x})$ is the posterior distribution. This coefficient sums to one, and is the softmax of the distance plus the logarithm of the prior.

Proof. Let $L = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})q(\mathbf{x}|c)}[\|h(\mathbf{x}_t, t, c) - \mathbf{x}\|_2^2]$. Taking the derivative and set to zero:

$$\frac{\partial}{\partial h(\mathbf{x}_t, t, c)} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})q(\mathbf{x}|c)}[\|h(\mathbf{x}_t, t, c) - \mathbf{x}\|_2^2] = 2\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})q(\mathbf{x}|c)}[h(\mathbf{x}_t, t, c) - \mathbf{x}] = 0.$$

We have:

$$\sum_{\mathbf{x}} q(\mathbf{x}_t|\mathbf{x})q(\mathbf{x}|c)h(\mathbf{x}_t, t, c) = \sum_{\mathbf{x}} q(\mathbf{x}_t|\mathbf{x})q(\mathbf{x}|c)\mathbf{x} \Leftrightarrow q(\mathbf{x}_t)h(\mathbf{x}_t, t, c) = \sum_{\mathbf{x}} q(\mathbf{x}_t|\mathbf{x})q(\mathbf{x}|c)\mathbf{x}$$

Therefore, we have:

$$\begin{aligned}
h(\mathbf{x}_t, t, c) &= \sum_{\mathbf{x}} \frac{q(\mathbf{x}_t|\mathbf{x})q(\mathbf{x}|c)}{q(\mathbf{x}_t)}\mathbf{x} = \sum_{\mathbf{x}} \frac{q(\mathbf{x}_t|\mathbf{x})q(\mathbf{x}|c)}{\sum_{\mathbf{x}'} q(\mathbf{x}_t|\mathbf{x}')q(\mathbf{x}'|c)}\mathbf{x} \\
&= \sum_{\mathbf{x}} \frac{\frac{1}{(2\pi\sigma_t^2)^{d/2}} \exp(-\frac{\|\mathbf{x}_t - x\|^2}{2\sigma_t^2})q(\mathbf{x}|c)}{\sum_{\mathbf{x}'} \frac{1}{(2\pi\sigma_t^2)^{d/2}} \exp(-\frac{\|\mathbf{x}_t - \mathbf{x}'\|^2}{2\sigma_t^2})q(\mathbf{x}'|c)}\mathbf{x} \\
&= \sum_{\mathbf{x}} \frac{\exp(-\frac{\|\mathbf{x}_t - x\|^2}{2\sigma_t^2} + \log q(\mathbf{x}|c))}{\sum_{\mathbf{x}'} \exp(-\frac{\|\mathbf{x}_t - \mathbf{x}'\|^2}{2\sigma_t^2} + \log q(\mathbf{x}'|c))}\mathbf{x} \\
&=: \sum_{\mathbf{x}\in\mathcal{D}} s_{\mathcal{D},c}(\mathbf{x})\mathbf{x}.
\end{aligned}$$

□

Lemma B.2. *There always exists a target image $\mathbf{x}_{final} \in \mathcal{D}$, such that the posterior probability of this image is close to one:*

$$1 - s_{\mathcal{D},c}(\mathbf{x}_{final}) \leqslant \epsilon_s = O(\frac{1}{\alpha}\exp(-\frac{1}{2\sigma_t^2})).$$

Proof. Let $\mathbf{x}$ be the closed point in dataset from $\mathbf{x}_t$, i.e., $\min_{\mathbf{x}\in\mathcal{D}} \|\mathbf{x} - \mathbf{x}_t\|^2$, and $\mathbf{x}_2$ be the second closest point. We have:

$$\begin{aligned}
1 - s_{\mathcal{D},c}(\mathbf{x}) &= 1 - \frac{\exp(-\frac{\|\mathbf{x}_t - x\|^2}{2\sigma_t^2} + \log q(\mathbf{x}|c))}{\sum_{\mathbf{x}'} \exp(-\frac{\|\mathbf{x}_t - \mathbf{x}'\|^2}{2\sigma_t^2} + \log q(\mathbf{x}'|c))} \\
&= 1 - \frac{1}{1 + \sum_{\mathbf{x}'\neq\mathbf{x}} \exp(\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \frac{\|\mathbf{x}_t - \mathbf{x}'\|^2}{2\sigma_t^2} + \log q(\mathbf{x}'|c) - \log q(\mathbf{x}|c))} \\
&\leqslant 1 - \frac{1}{1 + (|\mathcal{D}| - 1))\exp(\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \frac{\|\mathbf{x}_t - \mathbf{x}_2\|^2}{2\sigma_t^2} + \log(1-\alpha) - \log\alpha)} \\
&= \frac{(|\mathcal{D}| - 1))\exp(\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \frac{\|\mathbf{x}_t - \mathbf{x}_2\|^2}{2\sigma_t^2} + \log(1-\alpha) - \log\alpha)}{1 + (|\mathcal{D}| - 1))\exp(\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \frac{\|\mathbf{x}_t - \mathbf{x}_2\|^2}{2\sigma_t^2} + \log(1-\alpha) - \log\alpha)}
\end{aligned}$$

Using asymptotics, we have:

$$\begin{aligned}
1 - s_{\mathcal{D},c}(\mathbf{x}) &\leqslant O((|\mathcal{D}| - 1))\exp(\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \frac{\|\mathbf{x}_t - \mathbf{x}_2\|^2}{2\sigma_t^2} + \log(1-\alpha) - \log\alpha)) \\
&= O(\exp(-\frac{\|\mathbf{x}_t - \mathbf{x}_2\|^2 - \|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \log\alpha)) \\
&= O(\frac{1}{\alpha}\exp(-\frac{1}{2\sigma_t^2})).
\end{aligned}$$

□

Lemma B.3. *Let $\epsilon_s = \min_{\mathbf{x}} 1 - s_{\mathcal{D},c}(\mathbf{x})$, i.e., the largest probability in the posterior distribution. Let $C = \max_{\mathbf{x}} \|x\|_2$. We have:*

$$\|h(\mathbf{x}_t, t, c) - h(\mathbf{x}_t, t, c')\|_2^2 \leqslant 3\epsilon_s C^2.$$

Proof.

$$\|h(\mathbf{x}_t, t, c) - h(\mathbf{x}_t, t, c')\|_2^2 = \|\sum_{\mathbf{x}} s_{\mathcal{D},c}(\mathbf{x})\mathbf{x} - \sum_{\mathbf{x}} s_{\mathcal{D},c'}(\mathbf{x})\mathbf{x}\|_2^2 = \|\sum_{\mathbf{x}} [s_{\mathcal{D},c}(\mathbf{x}) - s_{\mathcal{D},c'}(\mathbf{x})]\mathbf{x}\|_2^2$$

$$\leqslant \sum_{\mathbf{x}} |s_{\mathcal{D},c}(\mathbf{x}) - s_{\mathcal{D},c'}(\mathbf{x})| \max_{\mathbf{x}} \|\mathbf{x}\|_2^2 = \sum_{\mathbf{x} \neq \mathbf{x}_{\text{final}}} |s_{\mathcal{D},c}(\mathbf{x}) - s_{\mathcal{D},c'}(\mathbf{x})|C^2 + |s_{\mathcal{D},c}(\mathbf{x}_{\text{final}}) - s_{\mathcal{D},c'}(\mathbf{x}_{\text{final}})|C^2$$

$$\leqslant \sum_{\mathbf{x} \neq \mathbf{x}_{\text{final}}} |s_{\mathcal{D},c}(\mathbf{x})|C^2 + \sum_{\mathbf{x} \neq \mathbf{x}_{\text{final}}} |s_{\mathcal{D},c'}(\mathbf{x})|C^2 + \epsilon_s C^2 \leqslant 3\epsilon_s C^2.$$

□

Combining Lemma B.3 and Lemma B.2, we can prove Theorem 4.1:

$$\|h(\mathbf{x}_t, t, c) - h(\mathbf{x}_t, t, c')\|_2^2 \leqslant O(3\frac{1}{\alpha}\exp(-\frac{1}{2\sigma_t^2})C^2) = O(\frac{1}{\alpha}\exp(-\frac{1}{2\sigma_t^2})).$$

C. Details of Backdoor Attacks and Implementation

We provide the details of the backdoor methods used in the experiment, including the trigger types and backdoor target types in Tab. 6. 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 [20, 68] have only a very limited effect (refer to Tab. 1 and Tab. 2).

For RickBKD$_{TPA}$[6], BadT2I$_{Tok}$[7], VillanBKD$_{one}$[6], and VillanBKD$_{mul}$[6], we directly use the publicly available model parameters. For RickBKD$_{TAA}$[8], BadT2I$_{Sent}$[9], EvilEdit[10], and PersonalBKD[11], 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.

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

Table 6. The backdoor attacks used in this paper.

D. Experiments of Additional Datasets and Models

To further evaluate our method's performance across various data distributions and various models, we inject backdoors into the Stable Diffusion v1-5 [58] model and evaluate different detection methods using the Flickr [77] dataset (Tab. 7 and Tab. 8). Note that since EvilEdit [66] and PersonalBKD [29] can only target specific objects in the text, we continue to use the data

[6]We use the published checkpoints in https://drive.google.com/file/d/1WEGJwhSWwST5jM-Cal6Z67Fc4JQKZKFb/view.
[7]We use the model released at https://huggingface.co/zsf/BadT2I_PixBackdoor_boya_u200b_2k_bsz16.
[8]We train backdoored models based on the release code: https://github.com/LukasStruppek/Rickrolling-the-Artist.
[9]We train backdoored models based on the release code: https://github.com/zhaisf/BadT2I.
[10]We train backdoored models based on the release code: https://github.com/haowang02/EvilEdit.
[11]We train backdoored models based on the release code: https://github.com/huggingface/notebooks/blob/main/diffusers/sd_textual_inversion_training.ipynb.

Method	RickBKD$_{TPA}$	RickBKD$_{TAA}$	BadT2I$_{Tok}$	BadT2I$_{Sent}$	VillanBKD$_{one}$	VillanBKD$_{mul}$	PersonalBKD	EvilEdit	**Avg.**	Iter./Sample
T2IShield$_{FTT}$	98.8	55.5	59.0	54.5	80.0	87.8	64.1	54.6	69.3	50
T2IShield$_{CDA}$	96.8	73.0	62.8	50.9	87.0	99.5	68.0	57.3	74.4	50
UFID	66.8	44.7	47.8	53.4	84.7	94.5	67.3	44.5	63.0	200
NaviT2I	**99.9**	**99.6**	**97.8**	**95.1**	**99.4**	**99.7**	**99.7**	**84.5**	**97.0**	**≈10.3**

Table 7. The performance (AUROC) against the mainstream T2I backdoor attacks on Flickr [77]. The experiment are conducted on Stable Diffusion v1-5 [58]. We highlight key results as Tab. 1. Additionally, we list the required diffusion iterations to approximate the computational overhead.

Method	RickBKD$_{TPA}$	RickBKD$_{TAA}$	BadT2I$_{Tok}$	BadT2I$_{Sent}$	VillanBKD$_{one}$	VillanBKD$_{mul}$	PersonalBKD	EvilEdit	**Avg.**	Iter./Sample
T2IShield$_{FTT}$	**93.5**	49.2	50.8	46.7	71.9	80.6	47.6	49.5	61.2	50
T2IShield$_{CDA}$	90.5	60.6	55.7	48.7	78.9	84.1	57.6	53.6	66.2	50
UFID	58.7	46.7	48.4	51.1	90.0	**98.0**	46.3	46.6	60.7	200
NaviT2I	87.1	**83.3**	**89.9**	**86.4**	**97.1**	**98.0**	**93.5**	**70.9**	**88.3**	**≈10.3**

Table 8. The accuracy (ACC) of detection against the mainstream T2I backdoor attacks on Flickr [77]. The experiments are conducted on Stable Diffusion v1-5 [58]. We highlight key results as Tab. 1. Since these values represent the ACC of a binary classification task, even "random guessing" achieves an ACC of 50.0%. **Note that the threshold used here is the same as in Tab. 2**, which demonstrates that the threshold computed in Eq. (9) based on our method can more effectively distinguish normal and poisoned samples across different data distributions.

as Sec. 5 while replacing the model with Stable Diffusion v1-5 [58]. **Note that we use the same detection threshold as in Tab. 2, in order to demonstrate the generalizability of our detection threshold.**

Effectiveness Evaluation. In Tab. 7 and Tab. 8, it can be observed that, our method also outperforms the baselines [20, 68] similar to Tab. 1 and Tab. 8, especially for the non-"entire image" backdoors in Tab. 6. Compared to our method, the performance of the baselines on more stealthy backdoors is nearly equivalent to random guessing.

Efficiency Evaluation. Similarly, to estimate computational overhead, we calculate the average non-stopword token length in the Flickr [77] dataset, which is approximately 10.3. This indicates that our method remains more efficient than the baselines, requiring only about 20% time-cost of T2IShield [68] and 5% time-cost of UFID [20].

E. Effects Under the More Advanced Adaptive Attack

Inspired by [76], we design a more advanced adaptive attack by adding a regularization term that enforces consistency constraints on activation variation. We implement the adaptive attack using BadT2I$_{\text{Tok}}$ and incorporate the regularization term from Eq. (4):

$$\mathcal{L}_{\text{BadT2I\_Reg}} = \mathcal{L}_{\text{BadT2I}} + \alpha \cdot \delta_\theta \left(c, c'\right), \tag{20}$$

where c and c' denote the benign input and the trigger-embedded input, respectively.

α	No Reg	10^{-5}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
FID	13.0	64.7 (+397%)	16.8 (+29%)	15.0 (+15%)	13.2	13.1
ASR	0.98	0.50	0.38	0.42	1.00	1.00
FTR	0	0.42	0.24	0.13	0	0
AUC	97.0	N/A	N/A	81.2	93.6	95.7

Table 9. The effectiveness of adaptive attacks against ***NaviT2I*** under various weights.

Including the original weight α of 250 in [76], we evaluate a set of α: $[250, 1, 10^{-3}, 10^{-5}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}]$. We compute the **FID** of the backdoored model to assess its utility, **ASR** (Attack Success Rate) and **FTR** (False Triggering Rate) to assess backdoor effectiveness, and **AUC** of ***NaviT2I***. In Tab. 13, when $\alpha \geqslant 10^{-5}$, the model collapses and outputs noise. At $\alpha = 10^{-7}$ or 10^{-8}, the model triggers the backdoor randomly, indicating unsuccessful backdoor injection. For $\alpha \leqslant 10^{-9}$, our method achieves satisfactory performance. Hence, this adaptive attack is ineffective.

	Backdoor	$\text{RickBKD}_{\text{TPA}}$	$\text{BadT2I}_{\text{Tok}}$	VillanBKD_{one}	EvilEdit	**Avg.**
AUC	BadAct	81.6	2.0	97.4	62.0	60.7
	NaviT2I	99.9	97.0	98.9	85.5	95.3 (+34.6)
ACC	BadAct	77.7	45.2	87.3	52.0	65.6
	NaviT2I	91.2	91.4	94.5	71.7	87.2 (+21.6)

Table 10. The performance of BadAct against T2I backdoors.

F. Comparison with Other Related Works

F.1. Backdoor Defenses for Unconditional Diffusion Models

In experiments (Sec. 5.2), we consider all existing T2I backdoor defense methods for comparison: T2IShield [68] and UFID [20]. There are also other backdoor defense methods [23, 43, 64, 67] targeting diffusion models. However, we do not include them in our experiments because these methods are only applicable to unconditional diffusion models, and are not suitable for text-to-image synthesis scenarios (i.e., conditional diffusion models). We select several works for discussion: Hao et al. [23], Mo et al. [43], Truong and Le [64] focus on inverting backdoor triggers in unconditional diffusion models (e.g., DDPM), where triggers are image distributions. In contrast, triggers in T2I diffusion models are textual tokens, making these methods inapplicable. [67] is also inapplicable as it aims to invert the visual trigger.

F.2. Comparison with BadActs [76]

BadAct [76] is a similar work that utilizes neuron activations for backdoor defense on NLP classification models. Our work differs from it in the following three aspects: ❶ Methodologically, BadAct relies on activation values to detect outliers, whereas we apply token masking and calculate the activation variations to make judgments. ❷ Theoretically, BadAct does not involve the concept of diffusion. In contrast, we conduct a theoretical analysis to significantly improve detection efficiency. ❸ Experimentally, BadAct is evaluated only on NLP models without T2I tasks. So we evaluate it on T2I backdoors. In Tab. 10, ***NaviT2I*** generally outperforms BadAct. Specifically, BadAct produces predictions opposite to the gold-labels for $\text{BadT2I}_{\text{Tok}}$ inputs (similar observation in Appendix G.1). This is because the activation distribution of $\text{BadT2I}_{\text{Tok}}$ samples is more concentrated than that of clean inputs, causing BadAct to fail.

G. Ablation Studies

G.1. From Neuron Coverage to Neuron Activation Variation

Note that although we find that the NC value of trigger tokens differs from other tokens on an average scale, Neuron Coverage value [45] is too coarse-grained to be directly used for detecting backdoored samples. We design the following ablation experiment to validate this point. ❶ We directly use the NC value of the input sample (a percentage value) as an indicator to determine whether it is a poisoned sample. ❷ We mask each token in the input sample and compute the maximum change in NC value as an indicator to determine whether it is a poisoned sample. We report the AUROC value of detection in Tab. 11. We find that neither of the above methods achieves results as satisfactory as our approach, demonstrating that the layer-wise computation, "neuron activation variation", designed in Sec. 4.2 plays a crucial role.

Method	RickBKD_{TPA}	RickBKD_{TAA}	BadT2I_{Tok}	BadT2I_{Sent}	VillanBKD_{one}	VillanBKD_{mul}	PersonalBKD	EvilEdit	**Avg.**
Neuron Coverage	64.9	54.4	31.9	35.0	62.0	82.9	45.6	59.5	54.5
NC Variation	91.5	71.6	64.8	71.5	84.4	56.5	65.8	62.8	71.1
NaviT2I	**99.9**	**99.8**	**97.0**	**89.7**	**98.9**	**99.9**	**99.8**	**85.5**	**96.3**

Table 11. We compare the detection performance (AUROC) when using Neuron Coverage [45] instead of calculating "Neuron Activation Variation" (Sec. 4.2). Notably, the "Neuron Coverage" method even produces an opposite AUROC value against the BadT2I [80] backdoors. This is because some backdoored samples increase the model's NC, while others decrease it.

G.2. Selection of Layers

In the UNet model, the architecture is divided into three DownBlocks (DownBlk), one MidBlock (MidBlk), and three UpBlocks (UpBlk) based on resolution [14, 58], with each block containing attention layers, convolutional layers, and other linear layers.

Layer Selection	VillanBKD$_{mul}$	BadT2I$_{Tok}$	EvilEdit	**Avg.**
DownBlk	99.9	75.9	**88.3**	88.0
MidBlk	99.9	96.6	76.3	90.9
UpBlk	99.9	96.7	84.5	93.7
Attention-layers	99.9	94.8	87.2	94.0
Conv-layers	99.9	**97.9**	82.6	93.5
All layers	99.9	97.0	85.5	**94.1**

Table 12. Detection performance (AUROC) across different layer selections.

We report ablation experiments for different layer selections within $\mathcal{L}_{set}$. In Tab. 12, we find that our method exhibits reduced sensitivity to certain backdoors (in light red) when using only the DownBlock and MidBlock. In contrast, utilizing all layers achieves the best overall performance. So we adopt this setting in Eq. (4).

G.3. Selection of Iteration Steps

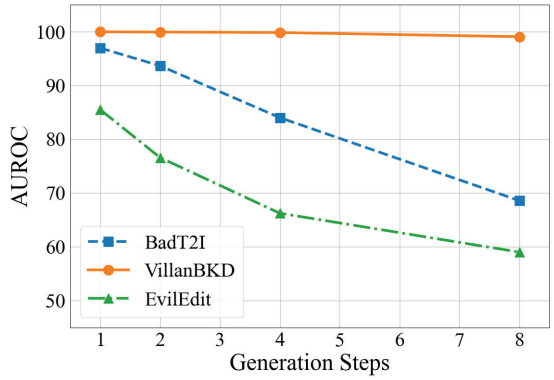

Figure 4. Detection effectiveness at various generation steps.

We evaluate the detection performance with different values of generating timestamps ($T_{iter} = 50$), and report the AUROC values in Fig. 4. We observe that as the timestamp increases, the AUROC value gradually decreases, which aligns with the **Early-step Activation Variation** phenomenon in Fig. 1.

G.4. Hyperparameter for Score Function in Eq. (7)

In Eq. (7), we exclude elements above the 75th percentile and compute the average value. This approach aims to eliminate outlier values, which may be related to trigger tokens, while the remaining elements are more likely to represent normal token values. Here, we analyze the impact of different percentile choices on the detection results. We conduct additional ablation experiments to analyze the impact of different percentile parameters: ❶ "Mean" – We directly compute the mean without excluding outliers. ❷ "ExMax" – We compute the mean of all elements except the maximum value. ❸ "75th" - We exclude elements above the 75th percentile and compute the mean. ❹ "50th" - We exclude elements above the 50th percentile and compute the mean. We report the AUROC values of different methods in Tab. 13. The experimental results show that different parameter choices have only a minor impact on the performance, which demonstrates the robustness of our method to hyperparameter selection. We adopt the 75th percentile in Eq. (7) because it exhibits a slight advantage over other parameter choices.

G.5. Hyperparameter for Threshold in Eq. (9)

We select different values for m in Eq. (9) and report the detection performance (ACC) of our method in Tab. 14. Since Eq. (9) essentially characterizes the boundary for outlier data within clean samples, a larger m tends to classify more samples as clean data, while a smaller m tends to classify more samples as poisoned data. According to the results in Tab. 14, the optimal classification performance is achieved when m is set to 1.2.

Percentile	RickBKD$_{TPA}$	RickBKD$_{TAA}$	BadT2I$_{Tok}$	BadT2I$_{Sent}$	VillanBKD$_{one}$	VillanBKD$_{mul}$	PersonalBKD	EvilEdit	**Avg.**
Mean	99.7	99.7	96.8	88.7	98.8	99.9	99.8	84.8	96.02
ExMax	100.0	99.8	97.3	88.8	98.8	99.9	99.9	84.8	96.15
75th	99.9	99.8	97.0	89.7	98.9	99.9	99.8	85.5	**96.31**
50th	99.0	99.9	96.6	90.2	99.1	99.9	99.9	85.3	96.24

Table 13. The performance (AUROC) of different percentile choices in Eq. (7). The experiments demonstrate the robustness of our method to hyperparameter selection.

Value of m in Eq. (9)	RickBKD$_{TPA}$	RickBKD$_{TAA}$	BadT2I$_{Tok}$	BadT2I$_{Sent}$	VillanBKD$_{one}$	VillanBKD$_{mul}$	PersonalBKD	EvilEdit	**Avg.**
m=1.1	89.8	90.5	90.9	80.4	95.2	98.3	95.6	73.0	89.21
m=1.2	91.2	91.8	91.4	79.2	94.5	98.9	95.6	71.7	**89.29**
m=1.3	92.3	92.7	90.7	78.3	94.1	99.0	93.5	70.6	88.90
m=1.5	93.5	94.8	90..4	76.0	93.3	99.0	92.8	66.8	77.03

Table 14. The performance (ACC) of different values of m in Eq. (9).

H. Another view of Theorem 4.1

In this section, we provide another view of Theorem 4.1. In Theorem 4.1, we bound the error ϵ by some function of α and σ_t. In this section, we would provide another dual theorem, which shows that for any error ϵ, there always exists a critical point t^*, such that for any $t < t^*$, the prediction of diffusion model under different conditions is ϵ-similar.

Theorem H.1. *Assume the diffusion model is well-trained, i.e., achieving the minimal $\mathbb{E}[\|\epsilon(\mathbf{x}_t, t, \mathbf{c}) - \epsilon\|_2]$ on some discrete distribution. As long as $p(\mathbf{c}|\mathbf{x})$ is not strictly 1 or 0, i.e., there exists $\alpha > 0$ such that $\alpha \leqslant p(\mathbf{x}|\mathbf{c}) \leqslant 1 - \alpha$ for any input $\mathbf{x}$, two different condition $\mathbf{c}, \mathbf{c}'$, then, for any error ϵ, there always exists a critical point t^*, such that for any $t < t^*$, the prediction of diffusion model under different condition are ϵ-similar:*

$$\|\epsilon(\mathbf{x}_t, t, c) - \epsilon(\mathbf{x}_t, t, c')\|_2 \leqslant \epsilon,$$

where $\sigma_{t*} = O(\sqrt{\frac{1}{-\ln \alpha\epsilon}})$ and c' is another condition embedding from text p': $c' = \mathcal{T}(p')$.

Note that $\frac{1}{-\ln \alpha\epsilon}$ is a much slower decay rate than any polynomial rate $\frac{1}{\text{poly}(\epsilon,\alpha)}$. This indicates that the diffusion model's different predictions would quickly become extremely similar. This is aligned with the empirical observation that the cosine similarity between the prediction of diffusion models becomes more than 0.9999 even after just 8 sampling steps.

Lemma H.2. *There always exists a target image $\mathbf{x}_{final} \in \mathcal{D}$, such that for any error ϵ_s, there always exists a critical point t^*, such that for any $t < t^*$, we have $s_{\mathcal{D},c}(\mathbf{x}_{final}) > 1 - \epsilon_s$.*

This lemma indicates that, when the sampling process proceeds, the posterior distribution gradually becomes a Dirac distribution, converging to one point in the training set.

Proof. Let $\mathbf{x}$ be the closed point in dataset from $\mathbf{x}_t$, i.e., $\min_{\mathbf{x}\in\mathcal{D}} \|\mathbf{x} - \mathbf{x}_t\|^2$, and $\mathbf{x}_2$ be the second closest point.

$$\begin{aligned}
s_{\mathcal{D},c}(\mathbf{x}) \geqslant 1 - \epsilon_s &\Leftrightarrow \frac{\exp(-\frac{\|\mathbf{x}_t - x\|^2}{2\sigma_t^2} + \log q(\mathbf{x}|c))}{\sum_{\mathbf{x}'} \exp(-\frac{\|\mathbf{x}_t - \mathbf{x}'\|^2}{2\sigma_t^2} + \log q(\mathbf{x}'|c))} \geqslant 1 - \epsilon_s \\
&\Leftrightarrow \frac{1}{1 + \sum_{\mathbf{x}'\neq\mathbf{x}} \exp(\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \frac{\|\mathbf{x}_t - \mathbf{x}'\|^2}{2\sigma_t^2} + \log q(\mathbf{x}'|c) - \log q(\mathbf{x}|c))} \geqslant 1 - \epsilon_s \\
&\Leftrightarrow \sum_{\mathbf{x}'\neq\mathbf{x}} \exp(\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \frac{\|\mathbf{x}_t - \mathbf{x}'\|^2}{2\sigma_t^2} + \log q(\mathbf{x}'|c) - \log q(\mathbf{x}|c)) \leqslant \frac{\epsilon_s}{1 - \epsilon_s}.
\end{aligned}$$

This can be relaxed to:

$$(|\mathcal{D}| - 1) \exp(\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \frac{\|\mathbf{x}_t - \mathbf{x}_2\|^2}{2\sigma_t^2} + \log q(\mathbf{x}_2|c) - \log q(\mathbf{x}|c)) \leqslant \frac{\epsilon_s}{1 - \epsilon_s}.$$

As long as $p(c|\mathbf{x})$ is not strictly 1 or 0, i.e., there exists an α such that $0 < \alpha \leqslant p(\mathbf{x}|c) \leqslant 1 - \alpha < 1$, we can further relax to:

$$(|\mathcal{D}| - 1)\exp(\frac{\|\mathbf{x}_t - \mathbf{x}\|^2}{2\sigma_t^2} - \frac{\|\mathbf{x}_t - \mathbf{x}_2\|^2}{2\sigma_t^2} + \log(1-\alpha) - \log\alpha \leqslant \frac{\epsilon_s}{1-\epsilon_s}$$
$$\Leftrightarrow \frac{\|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_t - \mathbf{x}_2\|^2}{2\sigma_t^2} \leqslant \log\frac{\epsilon_s}{(D-1)(1-\epsilon)} + \log\frac{\alpha}{(1-\alpha)} = O(\log\alpha\epsilon_s).$$

Therefore, if we want $s_{\mathcal{D},c}(\mathbf{x}_{\text{final}}) > 1 - \epsilon_s$, we can get the condition for σ_t^2:

$$\sigma_t^2 \leqslant O(\frac{1}{\log\alpha\epsilon_s}) \Leftrightarrow \sigma_t \leqslant O(\sqrt{\frac{1}{\log\alpha\epsilon_s}}).$$

□

Therefore, by Lemma B.3, to let $\|h(\mathbf{x}_t, t, c) - h(\mathbf{x}_t, t, c')\|_2^2 \leqslant \epsilon$, we require $3\epsilon_s C^2 \leqslant \epsilon$, that is $\epsilon_s \leqslant \frac{\epsilon}{3C^2}$. We can get the requirement for σ_t:

$$\sigma_t \leqslant O(\sqrt{\frac{1}{\log\alpha\frac{\epsilon}{3C^2}}}) = O(\sqrt{\frac{1}{\log\alpha\epsilon}}).$$