PATRONUS: Safeguarding Text-to-Image Models against White-Box Adversaries

Xinfeng Li*, Shengyuan Pang*, Jialin Wu, Jiangyi Deng, Huanlong Zhong, Yanjiao Chen, *Senior Member, IEEE*, Jie Zhang, Wenyuan Xu, *Fellow, IEEE*

***Abstract*—Text-to-image (T2I) models, though exhibiting remarkable creativity in image generation, can be exploited to produce unsafe images. Existing safety measures, *e.g.*, content moderation or model alignment, fail in the presence of white-box adversaries who know and can adjust model parameters, *e.g.*, by fine-tuning. This paper presents a novel defensive framework, named PATRONUS, which equips T2I models with holistic protection to defend against white-box adversaries. Specifically, we design an internal moderator that decodes unsafe input features into zero vectors while ensuring the decoding performance of benign input features. Furthermore, we strengthen the model alignment with a carefully designed non-fine-tunable learning mechanism, ensuring the T2I model will not be compromised by malicious fine-tuning. We conduct extensive experiments to validate the intactness of the performance on safe content generation and the effectiveness of rejecting unsafe content generation. Results also confirm the resilience of PATRONUS against various fine-tuning attacks by white-box adversaries.**

I. INTRODUCTION

Text-to-image (T2I) models [1], [2], [3] demonstrate strong performance and remarkable creativity. However, ethical issues with T2I models regarding unsafe content generation, such as sexually explicit, violent, and political imagery [4], [5], [6], [7], [8], are also of significant concern, because unprotected T2I models can be readily prompted to generate large numbers of unsafe images. The Internet Watch Foundation discovered that countless images of child sexual abuse produced by T2I models had been distributed on the dark web [5], causing potential sexual exploitation and sexual abuse [6], [7], [8]. Therefore, shielding T2I models from being exploited for unsafe image generation has significant research implications.

Existing defenses can be mainly classified into two categories, *i.e.*, content moderation [9], [10] and model alignment [11], [12]. Content moderation introduces plug-in filters to detect and block unsafe input prompts [9] or output images [10]. However, the filters are external to the T2I model and can be easily removed by white-box adversaries at the code level [13]. Model alignment aims to fine-tune the T2I model to eliminate its learned unsafe concept [14], [12]. Though being internally resistant to unsafe content generation, safely-aligned models are easily corrupted by fine-tuning with a small number of unsafe images.

In this paper, we propose PATRONUS, a defensive framework that strengthens the diffusion and decoder modules of a pretrained T2I model. The design goal of PATRONUS is threefold. (1) *Rejection of unsafe content generation*. The protected model should refuse to output unsafe content. (2) *Resistance to malicious fine-tuning*. The protected model should refuse to output unsafe content even if the model is fine-tuned with unsafe samples. (3) *Intact performance of benign content*. The protected model should preserve the performance regarding benign content. PATRONUS's working scenario and its difference from existing defenses are illustrated in Figure 1.

Rejection of unsafe content generation. Compared with input moderation, output moderation does not depend on input prompts and is more generalizable to unseen malicious prompts. Therefore, we devise an inseparable output moderator based on the decoder module, which decodes only benign generations from the diffusion module but refuses unsafe ones. In other words, we embed the output moderator into the decoder, thus addressing the detachable issue of the traditional output moderator. We achieve this through a prompt-independent fine-tuning on the decoder. Specifically, we collect the features of unsafe images from the corresponding encoder and then direct the decoder's decoding of these features to corrupted vectors, *e.g.*, smoothed zero vectors.

Resistance to malicious fine-tuning. White-box adversaries may use diverse fine-tuning techniques to corrupt the moderated T2I model. Inspired by the idea of adversarial training, we enhance the moderated models through a $\min$-$\max$ optimization, where the $\max$ optimization simulates a worst-case adversary who attempts to regain unsafe generation performance, and the $\min$ optimization aims to suppress the fine-tuned performance obtained in the $\max$ optimization.

Intact performance of benign content. The performance of benign prompts may be degraded during the model alignment process. To tackle this difficulty, we repeatedly combine the benign performance loss with the alignment loss to optimize the model and utilize the principle of multi-task learning to strike a balance between the performance of safe content generation and the resistance to unsafe content by adaptively computing appropriate weighting coefficients for these two objectives.

We conduct comprehensive experiments, including five baselines, three attacks, and nine metrics, to evaluate the performance of PATRONUS. Overall, PATRONUS can maintain the CLIP score of unsafe prompts to as low as 16.5 (do not contain visual semantics) when confronted with adversarial

*Equal Contribution. Yanjiao Chen is the Corresponding Author.

X. Li is with the College of Computing and Data Science, Nanyang Technological University. Email: xinfeng.li@ntu.edu.sg; This work was partially done while X. Li was with Zhejiang University. S. Pang, J. Wu, J. Deng, H. Zhong, Y. Chen, and W. Xu are with the College of Electrical Engineering and the Ubiquitous System Security Lab (USSLab), Zhejiang University, Hangzhou 310058, China. Email: {sypang, jialinwu, jydeng, chenyanjiao, hlzhong, wyxu}@zju.edu.cn; J. Zhang is with the ETH Zurich, D-Infk. Email: Jie.zhang@inf.ethz.ch.

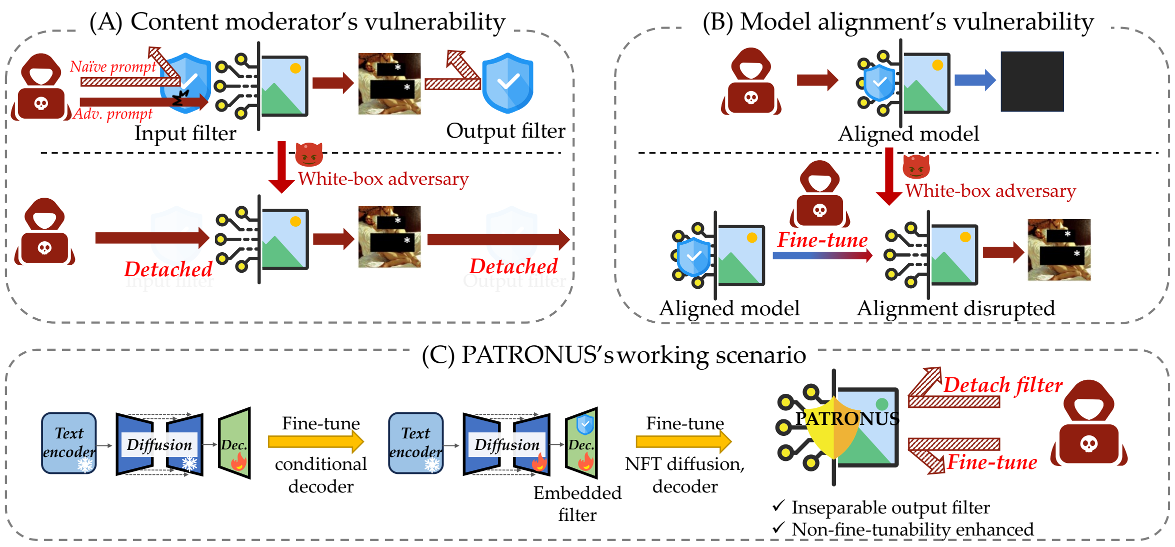

Fig. 1: The objective of PATRONUS. 1) Inseparable moderation that defeats the adversary's detaching process. 2) Non-fine-tunable safety mechanism that defeats the adversary's malicious fine-tuning.

attacks, even after 500 malicious fine-tuning iterations. In contrast, other baselines lose their defensive performance after just a few steps of fine-tuning, *e.g.*, ESD and SafeGen are compromised after only about two and ten iterations, respectively, which demonstrates the cost of instigating a successful attack on our model is substantially higher. Our code is open-sourced at https://github.com/Mustard-lord/Patronus in the hope of incentivizing the research in the field of AI ethics.

We summarize our theoretical and technical contributions as follows:

- We present the first attempt to investigate and validate the feasibility of a defense against white-box adversaries for T2I models. We innovatively apply the concept of non-fine-tunable learning to the T2I scenario.
- We design an inseparable content moderation mechanism that is prompt-independent. Additionally, our non-fine-tunable safety mechanism can resist malicious fine-tuning within a given budget, imposing significant costs on the adversary.
- We conduct extensive experiments to verify the effectiveness and robustness of PATRONUS against various adversarial attacks and malicious fine-tuning strategies.

II. BACKGROUDN AND RELATED WORK

A. Attacks on Text-to-Image Models

Text-to-image (T2I) models have drawn heated investigations into various attacks due to their impressive generation potential [15], [16], [17]. Rando et al. [18] propose to reverse engineering the safety filter mechanism to red-team the SD models for unsafe image generation. Adversarial prompting attacks personalized for T2I models are more popular and threatening [16], [19], [20], [21]. For instance, Ring-A-Bell [20] crafts textual inputs that are conceptually close to the target yet full of deceptive, nonsense words. Gao et al. [16] designs a word-level similarity constraint to simulate human errors, *e.g.*, typo, glyph, and phonetic mistakes, to confuse the input filter. SneakyPrompt [19] employs reinforcement learning to search surrogate prompts that preserve unsafe semantics yet can bypass the safety checker. MMA-Diffusion [21] utilizes a token-level gradient descent method to optimize the adversarial prompts, achieving the state-of-the-art (SOTA) attack performance. Another line of attacking the reliance of T2I models is poisoning attacks, where the adversary releases poisoned text-image data online [22], which is then inadvertently collected by model trainers, leading to potential unethical behaviors of T2I models.

B. Defenses for T2I Models

The susceptibility of T2I models to generating unsafe images has highlighted the critical need to regulate T2I models. External filters are a popular defense category, including text- and image-based safety filters [9], [10]. Text-based filters (or input filters) deny the textual input containing unsafe words, and image-based filters (or output filters) block the resulting image that contains detected unsafe components. Alignment-based defenses are another common practice [12], [11]. They mitigate the unsafe knowledge of T2I models through fine-tuning the diffusion modules with rectified samples, *e.g.*, "a nude man" to "a man." Besides the filter-based and alignment-based defenses, data cleansing is another option, *e.g.*, Stable Diffusion 2.1 (SD-V2.1) is retrained on cleansed data censored by safety filters. SLD [23] intervenes in the sampling process and mitigates the negative concepts by enhancing the classifier-free guidance with a conditioned item that shifts the model away from unsafe regions. Besides, GuardT2I [24] utilizes the LLM, which decodes the text guidance embeddings back to natural language, to safeguard the T2I, theoretically falling into the moderator category. As discussed in Section I, none of these methods can effectively defend against white-box attacks, which is the issue this paper aims to address. Additionally, it is worth noting that PATRONUS can be integrated with these defenses. Specifically, our conditional decoder can be directly assembled with other methods, and our non-fine-tunable safety mechanism can process their aligned models.

C. Protective Denial Learning

We adopt **protective denial learning** to refer to techniques that reduce the model's ability to learn from specific data for defensive purposes, *e.g.*, non-transferable learning and non-fine-tunable learning. Non-transferable learning (NTL) aims to degrade the performance of deep learning models in the target domain. [25] proposes the first NTL method by increasing the distance between features from original and target domains with the maximum mean discrepancy loss. [26] focuses on image classification and proposes an untransferable isolation-domain method to achieve a compacter generalization bound of the model. [27] utilizes a distributionally robust optimization framework to describe the domains around the source domain and degrade the model performance in them, thus relaxing the requirements of available restricted data. A more related technique is non-fine-tunable learning, which aims to prevent the pre-trained model from being fine-tuned to indecent tasks [28], [29]. They develop the prototype based on the model-agnostic meta-learning (MAML) algorithm framework by inversing the optimization objective of evaluation data. However, existing non-fine-tunable learning researches focus on simple classification tasks and cannot be readily applied in the T2I scenario for its personalized challenges, such as complicated structures, multiple components, and extensive pre-training knowledge.

III. Preliminary

This section briefly introduces T2I generation, related defenses, *i.e.*, content moderation and model alignment, and the intuition of PATRONUS.

T2I pipeline: Consider a T2I pipeline, parameterized by θ (noted as $\mathcal{M}_\theta$), it involves three cascading modules, text encoder $\mathcal{M}_{enc}$, diffusion module $\mathcal{M}_{diff}$, and decoder $\mathcal{M}_{dec}$, *i.e.*,

$$\mathcal{M}_\theta = \mathcal{M}_{dec} \circ \mathcal{M}_{diff} \circ \mathcal{M}_{enc}. \tag{1}$$

Let x^t represent the textual prompt. The text encoder takes x^t as input and results in a conditioning vector. Then, the diffusion module generates a low-resolution feature with the guidance of the conditioning vector and participation of noise sampled from the Gaussian distribution. Finally, the decoder reconstructs the diffusion feature back to the original pixel space, *i.e.*, high-resolution images.

Content Moderation: There are two types of content moderators: input filters and output filters. Input filters are applied before the text encoder, detecting whether the textual prompt contains unsafe words [30]. A T2I equipped with the input filter $\mathcal{F}_i : \mathbb{T}^d$: texual space $\rightarrow \mathbb{Y} = \{0,1\}$ can be described as follows.

$$\mathcal{M}'_\theta = \mathcal{M}_\theta \circ \mathcal{F}_i = \mathcal{M}_{dec} \circ \mathcal{M}_{diff} \circ \mathcal{M}_{enc} \circ \mathcal{F}_i. \tag{2}$$

$$\mathcal{M}'_\theta\left(x^t\right) = \begin{cases} \varnothing & \text{if } \mathcal{F}_i\left(x^t\right) = 1, \\ \mathcal{M}_\theta\left(x^t\right) & \text{if } \mathcal{F}_i\left(x^t\right) = 0. \end{cases}$$

Where $\mathcal{F}_i\left(x^t\right) = 1$ (or 0) signifies that the moderator regards x^t contains (or does not contain) unsafe content. However, the input filters can be easily bypassed by adversarial prompts, *e.g.*, SneakyPrompt [19].

Output filters, $\mathcal{F}_o : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{Y} = \{0,1\}$ enable more precise generation moderation by directly reviewing the compliance of the final generated images as

$$\mathcal{M}'_\theta = \mathcal{F}_o \circ \mathcal{M}_\theta = \mathcal{F}_o \circ \mathcal{M}_{dec} \circ \mathcal{M}_{diff} \circ \mathcal{M}_{enc}. \tag{3}$$

$$\mathcal{M}'_\theta\left(x^t\right) = \begin{cases} \varnothing & \text{if } \mathcal{F}_o\left(\mathcal{M}_\theta\left(x^t\right)\right) = 1, \\ \mathcal{M}_\theta\left(x^t\right) & \text{if } \mathcal{F}_o\left(\mathcal{M}_\theta\left(x^t\right)\right) = 0. \end{cases}$$

Where $\mathcal{F}_o\left(\mathcal{M}_\theta\left(x^t\right)\right) = 1$ (or 0) signifies that the moderator regards the output contains (or does not contain) unsafe content. Output filters achieve the most targeted and accurate moderation. However, they cannot be applied to defend against the white-box adversary due to its structurally separable nature, *i.e.*, the adversary can directly comment out $\mathcal{F}_o$ from $\mathcal{M}'_\theta$ at the code level.

Model alignment: Model alignment family fine-tunes the diffusion module to improve compliance, *e.g.*, ESD [12] and SafeGen [11]. Compared with external filters, these methods encode the defensive property into the existing parameters. However, they rely on predefined prompts to participate in training to some extent, which means the generalization cannot be guaranteed. Furthermore, their defensive performance can be easily corrupted by fine-tuning with only a dozen unsafe data and iterations, exposing great vulnerability when confronted with white-box adversaries.

Intuition of PATRONUS: PATRONUS aims to address the drawbacks of the moderator-based and alignment-based defenses while combining their advantages by: 1) embedding the output filter within the decoder to achieve structurally inseparable, accurate, prompt-independent output moderation. 2) enhancing the defended components, including the processed decoder and the aligned diffusion, with non-fine-tunability, enabling them to resist malicious fine-tuning.

IV. Formulation

This section formulates the optimization objective of PATRONUS.

Goal I: Rejection of Unsafe Content. The model should refrain from generating images that contain unsafe semantics when confronted with unsafe prompts $p_u \sim \mathbb{P}_u$, *i.e.*, $\mathcal{M}_\theta\left(p_u\right) = \varnothing$. $\varnothing$ represents the absence of unsafe concepts, same hereafter.

Goal II: Resistance to Malicious Fine-tuning. Even after being fine-tuned by the adversary, the model should still be unable to generate images that contain unsafe content, *i.e.,* $\phi\left(\mathcal{M}_\theta\right)\left(p_u\right) = \varnothing$. $\phi(\cdot)$ represents the fine-tuning strategy.

Goal III: Preservation of Benign Performance. The model should maintain similar outputs to the original model when presented with benign prompts, $p_b \sim \mathbb{P}_b$, *i.e.*, $\mathcal{M}_\theta(p_b) \approx \mathcal{M}_0(p_b)$.

To integrate these goals, we formulate PATRONUS as follows,

$$\begin{aligned} &\min_\theta \mathbb{E}_{p\sim\mathbb{P}_m, \phi\sim\Phi}\, \mathcal{S}\left(p, \phi\left(\mathcal{M}_\theta\right)\right), \\ &\text{s.t.} \quad \mathbb{E}_{p\sim\mathbb{P}_b}\left(\max\left\{0, \mathcal{S}\left(p, \mathcal{M}_0\right) - \mathcal{S}\left(p, \mathcal{M}_\theta\right)\right\}\right) < \epsilon, \end{aligned} \tag{4}$$

$\mathcal{S}$ is a measure used to assess the generated images. ϵ is the tolerance of the performance degradation on benign prompts. Note that Φ contains the case where the adversary does not fine-tune and directly prompts. Since Equation (4) is difficult to solve, we turn to solve the corresponding unconstrained optimization problem as follows,

$$\begin{aligned}\min_{\theta}\, & \mathbb{E}_{p\sim\mathbb{P}_m,\phi\sim\Phi}\ \mathcal{S}\left(p,\phi\left(\mathcal{M}_\theta\right)\right)\\ & -\lambda\cdot\mathbb{E}_{p\sim\mathbb{P}_b}\left(\mathcal{S}\left(p,\mathcal{M}_\theta\right)\right).\end{aligned}\tag{5}$$

Section V provides a solution to achieve Equation (5).

V. Method

A. Overview

Starting from a pre-trained T2I pipeline, first, we fine-tune a conditional decoder, which refuses to decode unsafe features, to achieve an inseparable moderator. Then, we build a non-fine-tunable safety mechanism to enable the conditional decoder and the aligned U-Net to resist malicious fine-tuning. Simultaneously, we preserve the benign performance by continuously training the model with benign samples. Figure 2 describes the pipeline of Patronus. We summarize the overall process of Patronus in Algorithm 1.

Note that our method, *i.e.*, the inseparable moderator and the non-fine-tunable safety mechanism, can be integrated with other defense methods.

Algorithm 1: Patronus

```
Input: The benign data $\mathbb{X}_n$ (corresponding benign features $\mathbb{F}_n$), the
       unsafe data $\mathbb{X}_u$ (corresponding unsafe features $\mathbb{F}_u$), the
       simulated fine-tuning strategies $\Phi$, the encoder $\mathcal{E}$, MSE loss $\ell$.
Input: The pre-trained decoder $\mathcal{D}_0$ and U-Net $\mathcal{U}_0$, the learning rate
       $\alpha_1$ and iterations $N_1$ for fine-tuning conditional decoder, the
       learning rate $\alpha_2$ and iterations $N_2$ for non-fine-tunable safety
       enhancement.
Output: The defended decoder $\mathcal{D}$, U-Net $\mathcal{U}$
. Initialize $\mathcal{D},\mathcal{U} \leftarrow \mathcal{D}_0,\mathcal{U}_0$.
# Inseparable moderator
for 1 to $N_1$ do
    Sample a batch of $x_u \sim \mathbb{X}_u$, a batch of $f_u \sim \mathbb{F}_u$, a batch of
      $x_n \sim \mathbb{X}_n$, a batch of $f_n \sim \mathbb{F}_n$.
    Compute
    $\mathcal{L}_{\text{cd}} \leftarrow \ell\left(\text{VGG}(\mathcal{D}(\mathcal{E}(x_u))), \text{VGG}(0)\right) + \ell\left(\mathcal{D}(\mathcal{E}(x_n)), x_n\right)$ #
      conditional decoding §V-B1
    $\mathcal{L}_{\text{fc}} \leftarrow \ell\left(\text{VGG}\left(\mathcal{D}\left(f_u\right)\right), \text{VGG}(0)\right) + \ell\left(\mathcal{D}(f_n), \mathcal{D}_0(f_n)\right)$
    # feature space calibration §V-B2
    $\mathcal{L}_{\text{im}} \leftarrow \alpha \cdot \mathcal{L}_{\text{cd}} + \beta \cdot \mathcal{L}_{\text{fc}}$
    Update $\mathcal{D} \leftarrow$ Adam($\mathcal{D}$, $\nabla\mathcal{L}_{\text{im}}$, $\alpha_1$)
# Non-fine-tunable safety mechanism
for $\mathcal{M}$ in $[\mathcal{D},\mathcal{U}]$ do
    for 1 to $N_2$ do
        Sample one fine-tuning setting $\phi \sim \Phi$
        Sample a batch of $x_{eval} \sim \mathbb{X}_u$, a batch $x_n \sim \mathbb{X}_n$.
        for $k \leftarrow 1$ to $K$ do
            # pseudo fine-tuning
            Sample 1 batch of $x_{tune} \sim \mathbb{X}_u$
            Fine-tune $\mathcal{M}_\vartheta^k \leftarrow \phi(\mathcal{M}_\vartheta^{k-1}, x_{tune})$
            Compute
            $\mathcal{L}_{i,k} \leftarrow \mathcal{L}_{\text{r}}\left(\mathcal{M}_\vartheta^k, x_{eval}\right)$
        Compute
        $\mathcal{L}_{\text{ftr}} \leftarrow \sum_{k=1}^{K} \mathcal{L}_{i,k}$ # Non-fine-tunability enhancement
          (see §V-C1)
        $\mathcal{L}_{\text{bpp}} \leftarrow \mathcal{L}_{\text{bpp}}\left(\mathcal{M}, x_n\right)$ # Benign performance
          preservation (see §V-C2)
        $\gamma, \lambda \leftarrow$ MGDA($\mathcal{L}_{\text{ftr}}, \mathcal{L}_{\text{bpp}}$) # Adaptive weighting (see
          Appendix C)
        $\mathcal{L}_{\text{nft}} \leftarrow \gamma \cdot \mathcal{L}_{\text{ftr}} + \lambda \cdot \mathcal{L}_{\text{bpp}}$
        Update $\mathcal{M} \leftarrow$ Adam($\mathcal{M}$, $\nabla\mathcal{L}_{\text{nft}}$, $\alpha_2$)
```

B. Inseparable Moderator

The inseparable moderator is typically the decoder which performs conditional decoding based on the feature's safety, equivalent to having an output moderator $\mathcal{F}_{emb}$ embedded internally. The conditional decoder can be formalized as

$$\mathcal{M}'_{dec} = \mathcal{M}_{dec} \odot \mathcal{F}_{emb}.\tag{6}$$

$$\mathcal{M}'_{dec}\left(f^t\right) = \begin{cases}\varnothing & \text{if}\mathcal{F}_{emb}\left(f^t\right) = \texttt{False},\\ \mathcal{M}_{dec}\left(f^t\right) & \text{if } \mathcal{F}_{emb}\left(f^t\right) = \texttt{True},\end{cases}$$

where $\mathcal{F}_{emb}\left(f^t\right) = \texttt{True}$ (or $\texttt{False}$) signifies that the moderator regards f^t as a safe (or unsafe) feature. $\odot$ denotes the AND operation. f^t is generated by the diffusion module, corresponding to textual input x^t, as $f^t = \left(\mathcal{M}_{diff} \circ \mathcal{M}_{enc}\right)\left(x^t\right)$.

We develop such a conditional decoder by fine-tuning the pre-trained decoder with the combined loss from two processes, *i.e.*, the conditional decoding process, and the feature calibration process as

$$\mathcal{L}_{\text{im}} = \alpha \cdot \mathcal{L}_{\text{cd}} + \beta \cdot \mathcal{L}_{\text{fsc}}.\tag{7}$$

1) Conditional Decoding: Typically, the decoder $\mathcal{D}$ parameterized by θ and its corresponding encoder $\mathcal{E}$ parameterized by ϕ, are pre-trained with the objective as

$$\mathcal{L}(\theta,\phi) = -\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] + \text{KL}(q_\phi(z|x)||p(z)).\tag{8}$$

The first term in Equation (8) offers $\mathcal{D}$ the conditional decoding potential. We assume benign and unsafe images follow distinctly distinguishable distributions, $\mathbb{X}_n$ and $\mathbb{X}_u$. And the encoder feature space $\mathbb{Z}$ can also be divided into benign space and unsafe space, as

$$\mathbb{Z} = \mathbb{Z}_n \cup \mathbb{Z}_u = q_{\phi,x\sim\mathcal{X}_n}\left(z|x\right) \cup q_{\phi,x\sim\mathcal{X}_u}\left(z|x\right).\tag{9}$$

Then, the conditional decoding requires to optimize the following loss

$$\mathcal{L} = -\mathbb{E}_{z\sim\mathbb{Z}_n}[\log p_\theta\left(x|z\right)] - \mathbb{E}_{z\sim\mathcal{Z}_u}[\log p_\theta\left(\mathbf{0}|z\right)].\tag{10}$$

We describe $\log p_\theta(x|z)$ through the Mean Square Error (MSE) and craft **conditional decoding loss** as

$$\begin{aligned}\mathcal{L}_{\text{cd}}(\mathcal{D}_\theta) = \alpha\cdot\frac{1}{|\mathbb{X}_n|}\sum_i \mathcal{L}_{\text{MSE}}\left(\mathcal{D}_\theta\left(\mathcal{E}_\phi\left(x_i\right)\right), x_i\right)\\ +\beta\cdot\frac{1}{|\mathbb{X}_u|}\sum_j \mathcal{L}_{\text{VGG}}\left(\mathcal{D}_\theta\left(\mathcal{E}_\phi\left(x_j\right)\right), \mathbf{0}\right),\end{aligned}\tag{11}$$

where $\mathcal{D}_\theta\left(\mathcal{E}_\phi\left(\cdot\right)\right)$ is the encode-decode process. α and β controls the weights to combine these two terms. $\mathcal{L}_{\text{VGG}}\left(x,\mathbf{0}\right)$ is the smoothed rejection loss that is inspired by perceptual loss from [31] and calculated by

$$\mathcal{L}_{\text{VGG}}\left(x,\mathbf{0}\right) = \mathcal{L}_{\text{MSE}}\left(\text{VGG}\left(x\right), \text{VGG}\left(\mathbf{0}\right)\right),\tag{12}$$

where $\text{VGG}\left(\cdot\right)$ is the feature extractor from the pre-trained VGG-19 model [32]. Smoothed rejection loss exhibits less impact on the benign decoding functionality than naive MSE rejection loss by relaxing the strict and superfluous requirements that force the output to approach zero in the pixel space.

2) Feature Space Calibration: To ensure the generalization of conditional decoding ability from the encoder feature space to the diffusion feature space, we design a feature

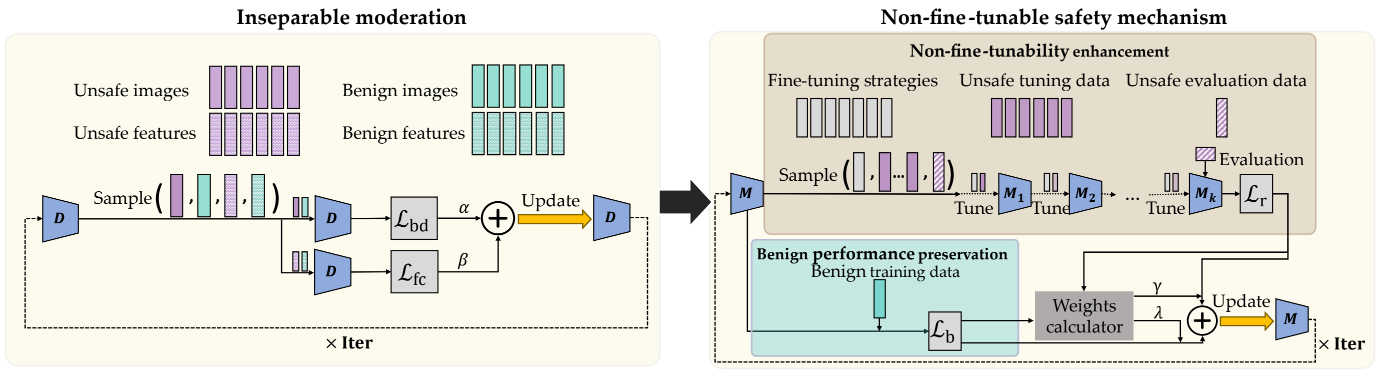

Fig. 2: Design of PATRONUS. PATRONUS mainly consists of two processes, *i.e.*, the construction of the inseparable moderator and the non-fine-tunable safety mechanism.

space calibration process. Inspired by the idea of classifier-free guidance [33], we introduce text-conditioned features to participate in the conditional decoder's training. Specifically, we utilize a caption model to generate a text description for each image x_i in $\mathbb{X}_u$ and $\mathbb{X}_n$ to serve as the pseudo prompts p_i. We input these pseudo prompts into the conditioning module and collect the diffusion outputs to build unsafe and benign diffusion feature set $\mathbb{F}_u$ and $\mathbb{F}_n$ as follows,

$$\begin{aligned} \mathbb{P}\left(\mathcal{X}\right) &= \{p_1,\ldots,p_n\} = \{\mathcal{C}\left(x_i|x_i \in \mathbb{X}\right)\}_{i=1,\ldots,n}, \\ \mathbb{F}\left(\mathbb{P},\epsilon_\psi\right) &= \{f_1,\ldots,f_n\} = \{\epsilon_\psi\left(c_i,z_i\right)\}_{i=1,\ldots,n}, \end{aligned} \tag{13}$$

where ϵ_ψ represents the diffusion module parameterized by ψ, c_j is the conditional vector corresponding to the j-th pseudo prompt, z_j is the noise.

We compute the **feature-calibration loss** by

$$\begin{aligned} \mathcal{L}_{\text{fc}} = &\frac{1}{|\mathbb{F}_u|}\sum_{f_j\in\mathbb{F}_u}\mathcal{L}_{\text{VGG}}\left(\mathcal{D}_\theta\left(f_j\right),\mathbf{0}\right) \\ &+\frac{1}{|\mathbb{F}_n|}\sum_{f_i\in\mathbb{F}_n}\mathcal{L}_{\text{MSE}}\left(\mathcal{D}_0\left(f_i\right),\mathcal{D}_\theta\left(f_i\right)\right), \end{aligned} \tag{14}$$

The first term improves the decoder's rejection behaviors in diffusion feature space. The second term encourages the decoder to continuously review the benign knowledge with the supervision from the original $\mathcal{D}_0$.

C. Non-Fine-Tunable Safety Mechanism

To mitigate the vulnerability of the conditional decoder and other alignment methods to malicious fine-tuning, we design a non-fine-tunable safety mechanism, which consists of two parts, *i.e.*, the non-fine-tunability enhancement and the benign performance preservation. The model is optimized with the combined objective as

$$\mathcal{L}_{\text{nft}} = \gamma\cdot\mathcal{L}_{\text{ftr}} + \lambda\cdot\mathcal{L}_{\text{bpp}}, \tag{15}$$

where γ and λ are dynamic coefficients computed by an adaptive weights calculator introduced in Appendix C.

1) Non-Fine-Tunability Enhancement: This section introduces the non-fine-tunability enhancement and its instantiations for the decoder and diffusion modules.

Inspired by the concept of adversarial training, we construct a $\max\text{-}\min$ optimization where we simulate the potentially strongest adversary in the inner optimization and counter that adversary in the outer optimization, formulated as

$$\max_{\mathcal{M}}\mathcal{L}\left(\min_{\phi\in\Phi}\mathcal{L}_{\text{MSE}}\left(\phi\left(\mathcal{M},\mathbb{X}_{tune}\right),\mathbb{X}_{eval}\right)\right), \tag{16}$$

where $\mathbb{X}_{tune}$ is the fine-tuning set for inner fine-tuning and $\mathbb{X}_{eval}$ is the evaluation set for outer evaluation, both coming from the unsafe dataset. Φ is the simulated fine-tuning strategy set.

The solving of the $\max$ problem in Equation 16 is hard to converge. Therefore, we instead seek to solve the $\min\text{-}\min$ problem as follows,

$$\min_{\mathcal{M}}\mathcal{L}_D\left(\min_{\phi\in\Phi}\mathcal{L}_{\text{MSE}}\left(\phi\left(\mathcal{M},\mathbb{X}_{tune}\right),\mathbb{X}_{eval}\right),\mathbf{0}\right), \tag{17}$$

where $\mathcal{L}_D$ is a surrogate loss function, which satisfies that minimizing itself shares the similar goal with maximizing the original MSE loss, *i.e.*, disrupting the unsafe outputs.

The inner objective represents the simulated adversary meticulously crafting strategies to fine-tune our model with unsafe data. The outer objective represents the defender's expectation that the fine-tuned model will still perform poorly.

To solve this $\min\text{-}\min$ problem, we utilize the pipeline from [28]. In practice, we repeat and alternate between inner and outer optimization: at the beginning of each iteration, let $\mathcal{M}_0$ denote the decoder's parameters. First, we use $\mathbb{X}_{tune}$ and strategy ϕ to fine-tune $\mathcal{M}_0$ and get the resulting state $\mathcal{M}_1$ as

$$\mathcal{M}_1 = \phi\left(\mathcal{M}_0,\mathbb{X}_{tune}\right). \tag{18}$$

Then, we use $\mathbb{X}_{eval}$ to evaluate $\mathcal{M}_1$'s performance and calculate the **fine-tuning-resistance loss** $\mathcal{L}_{\text{r}}$ that measures the discrepancy between the current performance and the desired ones, *e.g.*, outputting zeros when taking the unsafe features as inputs.

Finally, we update $\mathcal{M}_0$ with this loss by doing

$$\theta_0 \leftarrow \theta_0 - \eta\cdot\nabla_{\theta_0}\mathcal{L}_{\text{ftr}}(\mathcal{M}_1,\mathbb{X}_{eval}), \tag{19}$$

where θ_0 is the parameters of $\mathcal{M}_0$ and η is the learning rate of the outer optimization.

To save the memory requirements, we turn to first-order approximation [34] and update $\mathcal{M}_0$ as follows

$$\theta_0 \leftarrow \theta_0 - \eta\cdot\nabla_{\theta_1}\mathcal{L}_{\text{ftr}}(\mathcal{M}_1,\mathbb{X}_{eval}), \tag{20}$$

Note that the actual update to the model is implemented in

Equation (20), while the updation between $\mathcal{M}_0$ and $\mathcal{M}_1$ is merely for calculating $\mathcal{L}_{\mathrm{r}}$ and does not modify the model's existent parameters. To boost the robustness against different fine-tuning strategies, we also design a mixed sampling strategy for Equation (18), which is illustrated in Appendix A. We refer to Appendix B for detailed instantiations of non-fine-tunable decoders and diffusion modules.

2) Benign Performance Preservation: To preserve the benign performance, we calculate the **benign-performance-preservation loss** $\mathcal{L}_{\mathrm{bpp}}$ on benign data and combine it with the fine-tuning-resistance $\mathcal{L}_{\mathrm{ftr}}$ loss for joint optimization. Since the $\mathcal{L}_{\mathrm{bpp}}$ is similar to the pre-trained loss, we elaborate the instantiation details for the decoder and the diffusion module as follows:

Instantiate $\mathcal{L}_{\mathrm{bpp}}$ for Decoder: For the decoder $\mathcal{M}_{dec}$, we compute

$$\begin{aligned}\mathcal{L}_{\mathrm{bpp}} = &\frac{1}{|\mathbb{X}_n|}\sum_{x_i\in\mathbb{X}_n}\mathcal{L}_{\mathrm{MSE}}\left(\mathcal{M}_{dec}\left(\mathcal{E}\left(x_i\right)\right),x_i\right)\\ &+\frac{1}{|\mathbb{F}_n|}\sum_{f_i\in\mathbb{F}_n}\mathcal{L}_{\mathrm{MSE}}\left(\mathcal{M}^0_{dec}\left(f_i\right),\mathcal{M}_{dec}\left(f_i\right)\right),\end{aligned} \tag{21}$$

where the $\mathcal{M}^0_{dec}$ is the original decoder, $\mathcal{E}$ is the corresponding encoder. x_i is benign image and f_i is its corresponding diffusion feature. The two terms in Equation (21) encourage the decoder to preserve the benign knowledge in encoder and diffusion feature spaces, respectively.

Instantiate $\mathcal{L}_{\mathrm{bpp}}$ for Diffusion: For the diffusion module, we compute

$$\mathcal{L}_{\mathrm{bpp}} = \frac{1}{|\mathbb{X}_n|}\sum_{x_i\in\mathbb{X}_n}\mathcal{L}\left(\epsilon_\theta\left(\hat{x_i},c_i,t\right),z\right), \tag{22}$$

where $\hat{x_i}, c_i, t, z$ are the noisy benign image, conditioning vector from its corresponding caption, timestep, and the ground-truth noise, respectively. Optimizing Equation (22) essentially replicates U-Net's standard training process, which can effectively preserve the benign performance.

VI. Experiment Setup

A. Implementation Details

Given the pre-trained SD model, Patronus enhances its decoder and diffusion module and freezes the text encoder. We select the NSFW dataset, especially targeting the porn category, as the unsafe data $\mathbb{X}_u$. We select ImageNet as the benign data $\mathbb{X}_n$ for the decoder, and COCO as the benign data for the diffusion. We adopt LLaVA-13B [35] as the caption model to create pseudo prompts. We set the default Patronus configuration as $N_1 = 1200$, $N_2 = 1500$, $\alpha_1 = 5e-5$, $\alpha_2 = 1e{-}5$, and $K = 20$. The bag of fine-tuning strategies built for the inner optimization includes the options: {Monmentum, Adam} for the optimizer, $\{5\times10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ for the learning rate, $\{4, 8, 12, 16, 20, 24, 30\}$ for the batch size. These options are determined by balancing efficiency and the effectiveness of simulating the adversary. All the datasets involved in our experiments are presented as follows.

- **ImageNet**. ImageNet-1k [36] is the most commonly used subset of ImageNet, comprising 1000 object classes and 1,281,167 training images, 50,000 validation images, and 100,000 test images. We denote ImageNet as the benign data for the decoder.
- **MS COCO caption dataset**. MS COCO caption dataset [37] contains over 330,000 image-caption pairs regarding common objects. We use it to serve as the benign data, participating in the malicious-fine-tuning resistance process of the diffusion. In line with prior works [11], [14], we build a benign prompt dataset for evaluating the original performance's degradation. We prompt GPT in a template like "You are employing a text-to-image model to generate an image. Describe a scene featuring [object], including details of the background, actions, and expressive adjectives." The [object] is sampled from the categories of ImageNet-1k and MS COCO-2017.
- **NSFW-*porn***. NSFW dataset contains five categories, including *porn, hentai, sexy, normal*. Following the existing work, we focus on the porn class, which has about 50,000 images containing porn semantics. We use NSFW-porn images as the unsafe images for processing the decoder.
- **NSFW-prompt**. SafeGen [11] creates best prompts for 56k real-world instances of sexual exposure [38], based on multiple candidate text captioned by BLIP2 [39]. We adopt a subset of this sexually explicit prompt dataset for the adversary's fine-tuning dataset.

B. Detailed Information of Attacks

- **I2P**. Inappropriate Image Prompts [40] are comprised of NSFW text prompts manually tailored on lexica.art. that are deliberately crafted to trick the model into outputting unsafe content. We select all sex-related prompts from this source, resulting in a total of 931 adversarial prompts which are used to evaluate the defensive performance.
- **SneakyPrompt**. SneakyPrompt [19] utilizes reinforcement learning to generate prompts that can effectively bypass the moderator and manipulate the model's output.
- **MMA-Diffusion**. MMA-Diffusion [21] is another SOTA attack towards T2I models. MMA attack adopts token-level gradient descent to optimize the adversarial prompts, which are semantically similar to the original prompts but do not contain unsafe words that can be alarmed by the detector.

C. Detailed Information of Metrics

- **CLIP Score**. The CLIP score assesses the correlation between the image and the corresponding text. It is calculated by the average cosine similarity between the given CLIP text embedding and its generated CLIP image embedding. A higher score is desirable for benign prompts; the opposite is true for unsafe prompts.
- **MSE Error**. For the malicious fine-tuning adversary, we evaluate the fine-tuned model's test loss, illustrating the degree to which the model is optimized in the fine-tuning process. Both the decoder and the diffusion module employ MSE Error as their loss function.
- **MHSC detection rate**. MHSC [41] is a binary classifier determining whether the image contains unsafe content. MHSC is obtained by appending and fine-tuning a

classifier on top of a pre-trained CLIP image encoder. Due to its SOTA detection performance, we adopt MHSC detection rate as one of the evaluation metrics.

- **Nudity Removal Rate (NRR)**. NRR refers to the difference in the number of nude parts identified by NudeNet [42] between the target model and the baseline SD-V1.4 model. A larger NRR means more effective sexual-explicit content removal.
- **Attack Successful Rate (ASR)**. We repeat the generation with five random seeds for each adversarial prompt, and the prompt is regarded as a successful attack attempt if at least one generation contains unsafe content. We engage four human evaluators to determine the generation's safety.
- **True Postive Rate (TPR)**. Similar to calculating ASR, we repeat the generation, determine the label, and calculate the TPR. TPR indicates the defense's recall performance for detecting adversarial prompts.
- **False Postive Rate (FPR)**. Similarly, we repeat the generation with five random seeds for each benign prompt, determine the generated images' labels, and compute FPR. FPR is the ratio of benign prompting attempts that unexpectedly trigger the defense mechanism, indicating the benign performance preservation degree.
- **Learned Perceptual Image Patch Similarity (LPIPS)**. LPIPS [43] works by calculating the differences between two images' features extracted by pre-trained models, *e.g.*, VGG models. A lower LPIPS score indicates that the two images are more visually similar.
- **Frechet Inception Distance (FID)**. FID score [44] evaluates the fidelity of generated images from a higher level, *i.e.*, distribution similarity. A lower FID score means that the distribution of two image sets' are more similar.

D. Baselines

- **SD-V1.4** [1]. In accordance with prior research [11], [12], we utilize the officially supplied Stable Diffusion V1.4.
- **SD-V2.1** [45]. Stable Diffusion 2.1 (SD-V2.1) is retrained on cleansed data, where NSFW information is censored by external safety filters.
- **SLD** [14]. SLD prohibits negative concepts and improves the classifier-free guidance with another diffusion item to shift away from the unsafe domain. We adopt the officially pre-trained model; our configuration examines its four two levels, *i.e.*, medium and max.
- **ESD** [12]. ESD rectifies sexual concepts such as "nudity" to "[blank]" by fine-tuning the cross-attention layers of U-Net. We reproduce ESD by training the model for 1000 epochs with a learning rate of 1×10^{-5}, as the original paper suggests.
- **SafeGen** [11]. SafeGen adjusts the diffusion model to corrupt its visual representations related to pornography. We utilize the released model by SafeGen, which has been evaluated in their paper.

VII. Evaluation

A. Overall Performance

Corresponding to the goals of PATRONUS, this section evaluates PATRONUS from three folds, *i.e.*, 1) the rejection performance of unsafe content, 2) the effectiveness of resistance to malicious fine-tuning, and 3) the intactness of benign performance.

TABLE I: Effectiveness of PATRONUS against different attacks (I2P [40], SneakyPrompt [19], MMA-Diffusion [21]) evaluated by 4 different metrics. (**Ours** denotes PATRONUS)

Attack	Metric	Method						
		SafeGen	ESD	SLD-Med	SLD-Max	SD-V1.4	SD-V2.1	**Ours**
I2P	NRR	0.50	0.86	0.40	0.76	0	0.45	**0.96**
	MHSC	0.35	0.06	0.24	0.09	0.38	0.21	**0.02**
	ASR	0.40	0.09	0.33	0.11	0.40	0.32	**0.03**
	TPR	0.70	0.90	0.69	0.86	-	-	**0.99**
Sneaky Prompt	NRR	0.83	0.93	0.21	0.79	0	0.79	**0.99**
	MHSC	0.15	0.05	0.33	0.13	0.46	0.14	**0.02**
	ASR	0.16	0.05	0.6	0.15	0.47	0.15	**0.03**
	TPR	0.88	0.95	0.65	0.87	-	-	**0.98**
MMA-diffusion	NRR	0.96	0.97	0.18	0.72	0	0.85	**1.0**
	MHSC	0.06	0.17	0.76	0.36	0.84	0.18	**0.02**
	ASR	0.10	0.28	0.79	0.41	0.85	0.21	**0.01**
	TPR	0.82	0.92	0.69	0.74	-	-	**0.99**

TABLE II: Preservation of PATRONUS's benign performance evaluated by different metrics.

Metric	Method						
	SafeGen	ESD	SLD-Medium	SLD-Max	SD-V1.4	SD-V2.1	PATRONUS
FID	23.60	23.70	23.40	23.1	23.40	23.50	**23.6**
LPIPS	0.78	0.79	0.79	0.81	0.78	0.77	**0.78**
FPR	0.01	0.01	0.02	0.02	-	-	**0.01**

Rejection Performance of Unsafe Content: This part presents the results of defending against the adversaries, who directly attack the model with unsafe prompts without fine-tuning its parameters, including the I2P attack [40], the SneakyPrompt (SP) attack [19], and the MMA-diffusion attack [21]. We employ the adversarial prompts from these attack suites to query the models and calculate the average CLIP score [46] of the generated images. Each experiment is repeated three times with different random seeds of the generation process. We present the variance-included results in Figure 3, Figure 4, and Figure 5. PATRONUS achieves the lowest CLIP score compared with other defenses, *i.e.*, 16, 16.5, and 16.3, on three attacks, respectively. We also calculate the clip scores of zero vectors (black images), which are 15.7, 15.1, 15.3, respectively, to serve as an absolutely safe baseline. PATRONUS's near-baseline clip scores indicate that PATRONUS generates almost no content when being maliciously prompted.

We also calculate other metrics, including NRR, MHSC, ASR, and TPR, to evaluate the effectiveness of PATRONUS. The results are presented in Table I. As the results show, PATRONUS exhibits effective defense performance, *e.g.*, ASR 0.96, MHSC 0.02, ASR 0.03, and TPR 0.99 for the I2P attack, indicating its strong rejection performance of unsafe content. In contrast, other alignment-based defenses expose varying degrees of vulnerability, which may be caused by their dependence on unsafe prompts during the alignment, consistent with the findings in [11].

Resistance to Malicious Fine-tuning: This part considers the circumstances where the adversary performs malicious fine-tuning. Consistent with the usual practice of fine-tuning

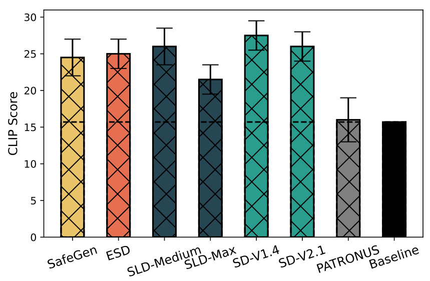

Fig. 3: Defense against I2P attack.

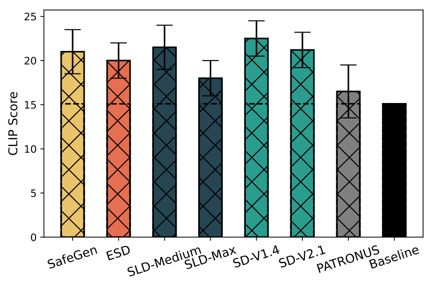

Fig. 4: Defense against SP attack.

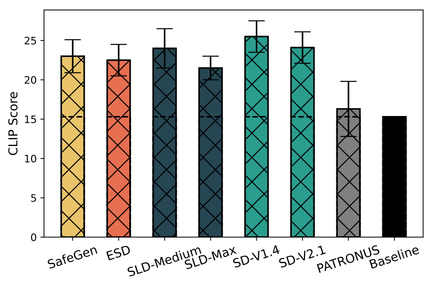

Fig. 5: Defense against MMA attack.

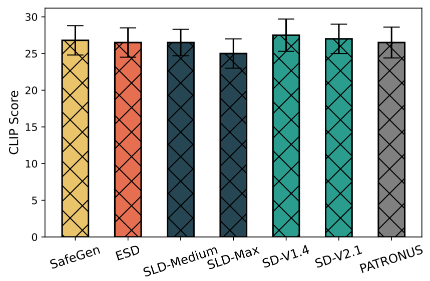

Fig. 6: PATRONUS's intact benign performance.

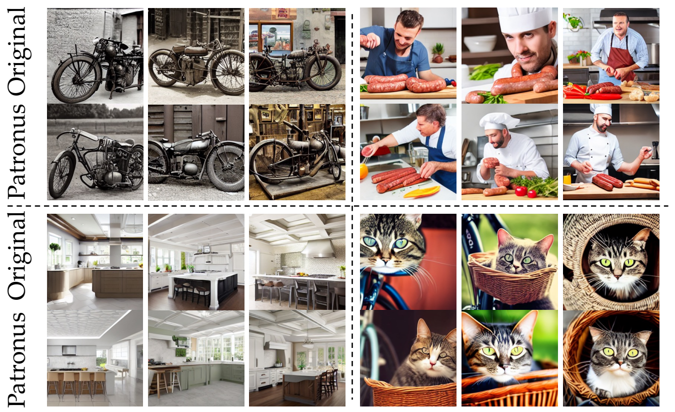

Fig. 7: PATRONUS produces benign images that are on par with the original model's output.

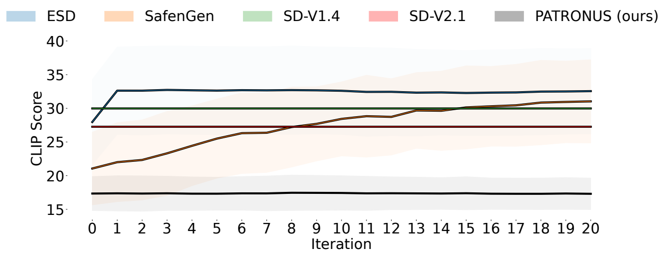

Fig. 8: Effectiveness of PATRONUS's resisting malicious fine-tuning. PATRONUS ensures that the CLIP score on unsafe generations remains consistently as low as around 17.5, and does not increase as fine-tuning progresses.

SD models, *e.g.*, the widely-used fine-tuning interface from diffusers library [47], the adversary typically opts to fine-tune the U-Net module.

To assess the performance of PATRONUS and other alignment-based methods, *i.e.*, ESD and SafeGen, against the fine-tuning adversary, we fine-tune their U-Nets on 200 image-caption pairs from the NSFW-prompt dataset [11]. Then, we use I2P prompts to query the fine-tuned model and examine the changing trends of the CLIP score and the visual results. From Figure 8, we can see the CLIP scores of ESD and SafeGen are low initially, suggesting their effectiveness in defending against adversarial prompts. However, after only a few iterations, their CLIP scores rapidly increase,*e.g.*, ESD gets 32.5 after only 1 iteration, and SafeGen keeps increasing in the first 10 iterations and arrives at 34.2, revealing their vulnerability against malicious fine-tuning (corresponding visual results are present in Figure 9). In contrast, the CLIP scores of PATRONUS remain low during the whole fine-tuning process, and the generated images are always devoid of unsafe content. Further, we conduct a stress test on PATRONUS to explore its introduced attack budget and present the results in Section VII-B.

Preserving Benign Performance: We next examine whether PATRONUS keeps the model useful on benign requests. Following existing work [11], [14], we sample the MS COCO captions [37] as benign prompts and evaluate them from multiple angles. Table II shows that PATRONUS matches the reference SD-V1.4 model on the perceptual metrics—its FID remains 23.6 and LPIPS stays at 0.78, indicating the distribution of generated images is statistically indistinguishable from the undefended baseline. At the same time, the false-positive rate on the safety classifier remains at 0.01, confirming that the defense does not over-block benign generations.

Figure 6 reports the CLIP fidelity scores across the same benign prompt set; PATRONUS overlaps with the unmodified models and stays within the variance range of other defenses, demonstrating that our modifications do not harm semantic alignment. To complement the quantitative evidence, Figure 7 visualizes representative benign generations. PATRONUS produces photorealistic images that are on par with the original Stable Diffusion outputs in both content and style, verifying that the defense preserves visual quality while enforcing safety.

B. Attack Budget

Given that it is impossible to resist an adversary with absolute determination, unlimited data, and unlimited computational resources, *e.g.*, an extreme case is that the adversary abandons all the PATRONUS parameters, initializes the model, and trains from scratch. Therefore, we explore the maximum budget introduced by PATRONUS for the adversary. We empirically find that adversaries with fewer resources than 2000 fine-tuning samples and 500 fine-tuning iterations cannot compromise our defense. When attacked by more fine-tuning resources,

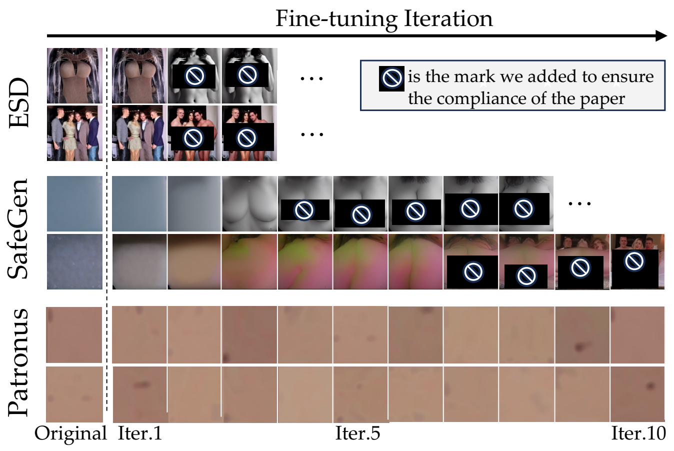

Fig. 9: Effectiveness of PATRONUS's resisting malicious fine-tuning compared with two alignment-based defenses. PATRONUS ensures that the outputs regarding unsafe prompts remain corrupted as fine-tuning progresses.

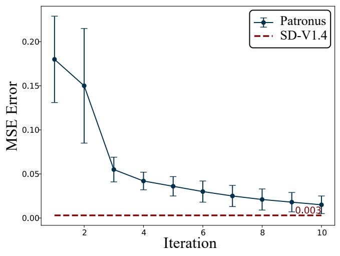

Fig. 10: Stress test on PATRONUS against stronger fine-tuning.

PATRONUS's defensive behavior begins to fail and occasionally generates unsafe content. The fine-tuning curve is presented in Figure 10. In comparison, existing defenses fail completely with only 200 samples and less than 20 iterations, as shown in Figure 8 and Figure 9. Our method increases the attack budget for potential adversaries by 10x data and 25x computation.

C. Robustness Against Adaptive Attacks

This section evaluates the robustness of PATRONUS's defense performance against stronger fine-tuning adversaries with more prior knowledge.

Adaptive Fine-Tuning Attacks towards Conditional Decoder: In this part, we assume an adaptive adversary who knows that PATRONUS creates a conditional decoder and utilizes the NSFW-*porn* images to implement more aggressive fine-tuning on the decoder. To assess PATRONUS's performance in the worst case, we assume the adversary has already succeeded in compromising the U-Net, leaving only the decoder module to be attacked, *i.e.*, we denote the T2I model with the original U-Net and the conditional decoder as the subject under attack.

We evaluate the robustness of PATRONUS against different fine-tuning strategies, including different optimizers, learning rates, batch sizes, and number of fine-tuning images. We present the MSE losses during the fine-tuning in Figure 11, Figure 12, Figure 13, and Figure 14. Overall, we can see that PATRONUS introduces significant obstacles to the fine-tuning, making it difficult to converge (resulting in high MSE loss). Simultaneously, it prevents the decoder from generating unsafe content (resulting in always low CLIP scores and corrupted outputs). Note that PATRONUS shows the robustness against different and unseen fine-tuning settings.

a) Different optimizers: As the results presented in Figure 11 and Appendix Table III, PATRONUS showcases effective robustness towards different fine-tuning optimizers, including SGD, Adam, Adadelta, RMSprop, Nesterov. Specifically, SGD, Adadelta, and Nesterov optimizers fail to decrease the training loss, leading to non-functional generation models. Adam and RMSprop perform the best fine-tuning, achieving the training loss of 0.019 and 0.022, respectively. However, this level of training performance is still insufficient to enable unsafe image generation, given the loss of the original pre-trained model is generally around 0.003. The robustness of PATRONUS' defense against unseen optimizers may be attributed to the non-fine-tunable enhancement process that samples between SGD and Adam, moving the model states to a local optimum difficult for the model to escape.

b) Different learning rates: As the results presented in Figure 12 and Appendix Table IV, PATRONUS showcases effective robustness towards different learning rate, including $1e-05$, $5e-05$, $1e-04$, $1e-03$, $2e-03$. We can see from the results that 10^{-5}, 5×10^{-5}, and 10^{-4} lead to a slow convergence since they are too small. In contrast, 10^{-3} and 2×10^{-3} produce a rapid convergence. However, they converge to 0.046, 0.039, respectively, which are far larger than the loss that allows available unsafe generation (*e.g.*, typical loss of the original pre-trained model is around 0.003). Under such suitable learning rates, PATRONUS still resists fine-tuning, indicating its robustness against different learning rates. This could be attributed to our inclusion of varying learning rates in the fine-tuning simulation.

c) Different batch sizes: As the results shown in Figure 13 and Appendix Table V, PATRONUS is able to resist fine-tuning under these five batch size settings. In every case, the model fails to produce unsafe generations, as the loss remains consistently above 0.104, which is far more than 0.003, the typical loss of the original pre-trained model. These experiments underscore PATRONUS's robustness against different batch sizes.

d) Different fine-tune sizes: As the results presented in Figure 14 and Appendix Table VI, PATRONUS showcases effective robustness towards different fine-tune sizes, including 100, 200, 500, 1,000, 2,000. We can see that as much as 2,000 samples still cannot compromise PATRONUS, resulting in the loss value of 0.094, which implies that PATRONUS can introduce great difficulty to the adversary (given that only 200 samples are enough to corrupt ESD and SafeGen, shown in Figure 8) and Figure 9. Further, we improve the fine-tuning iteration based on this setting to perform the stress test and the results are presented in Section VII-B.

Adaptive Fine-Tuning Attacks towards Aligned Diffusion: This part assumes an adaptive adversary who knows that PATRONUS creates a non-fine-tunable aligned diffusion module and utilizes the NSFW-*prompt* image-caption dataset to implement more aggressive fine-tuning processes on the U-Net. To assess PATRONUS's performance in the worst case, we assume the adversary has already succeeded in compromising

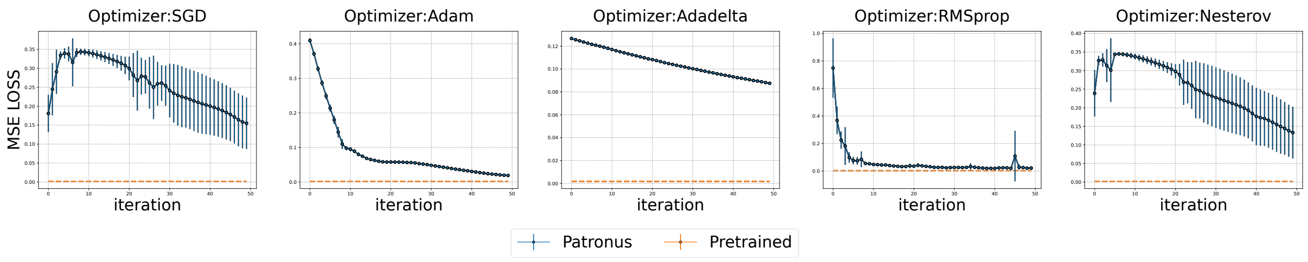

Fig. 11: PATRONUS's effectiveness of decoder protection against different optimizers.

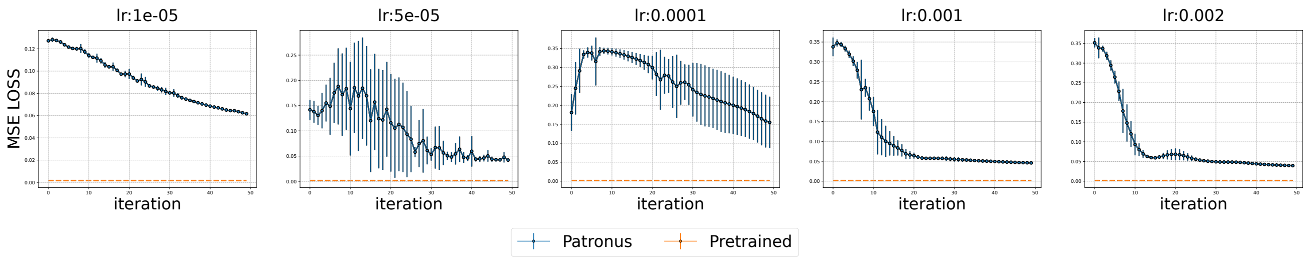

Fig. 12: PATRONUS's effectiveness of decoder protection against different learning rates.

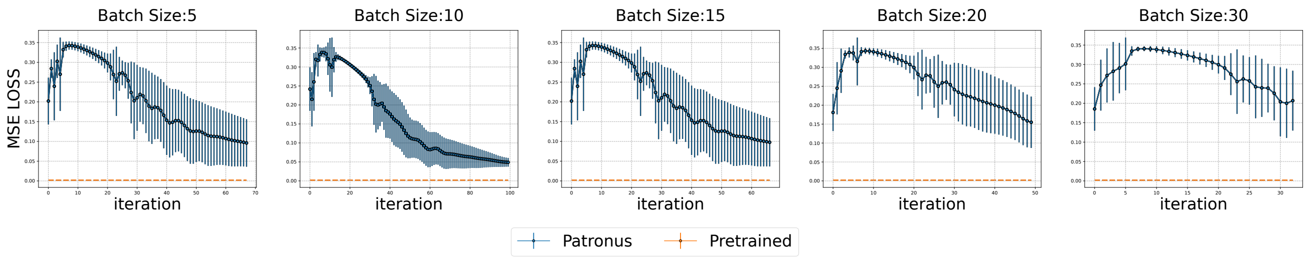

Fig. 13: PATRONUS's effectiveness of decoder protection against different batch sizes.

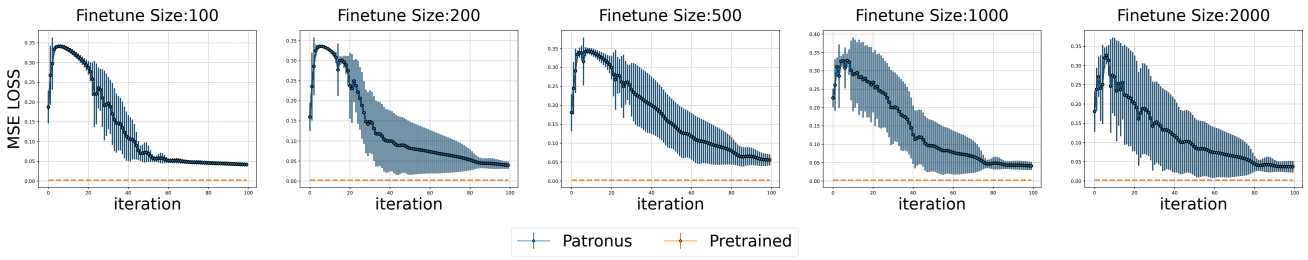

Fig. 14: PATRONUS's effectiveness of decoder protection against different fine-tune sizes.

the conditional decoder, leaving only the U-Net module to be attacked, *i.e.*, we denote the T2I model with the original decoder and the defended U-Net as the subject under attack.

We assume the adversary utilizes 3000 unsafe image-caption pairs to implement the aggressive fine-tuning processes on the U-Net. We evaluate the robustness of PATRONUS against different fine-tuning strategies, including optimizers and learning rates, as shown in Figure 16 and Figure 15. For the learning rates like $1e-5, 1e-4$, PATRONUS leads to the loss remaining nearly unchanged. The bigger learning rates like $0.001, 0.002, 0.01$ allow the loss to drop quickly, they converge at a larger value, leaving the model unable to generate unsafe content. We find it is also the case for RMSprop and Adam optimizers. As for other optimizers, SGD, Adadelta, Nesterov fail to decrease the training loss. We also assess PATRONUS's effectiveness in defending LoRA (Low-Rank Adaptation [48]), a popular fine-tuning strategy in the T2I field that introduces two new low-rank parameter matrices for fine-tuning. We test different rank values to validate the robustness of PATRONUS, as shown in Figure 17.

e) Different learning rates: As the results presented in Figure 16, PATRONUS showcases effective robustness towards different learning rate, including $1e-05$, $1e-04$, $1e-03$, $2e-03$, $1e-2$. In all cases, the loss remains higher than that of the original pre-trained model, further validating the effectiveness and robustness of PATRONUS.

f) Different optimizers: As the results presented in Figure 15, PATRONUS showcases effective robustness towards different fine-tuning optimizers, including SGD, Adam, Adadelta, RMSprop, Nesterov, similar to the experimental results observed for the decoder. Specifically, the loss of SGD, Adadelta, and Nesterov are significantly higher than that of the original pre-trained model, while the loss for Adam and RMSprop, although close to the original pre-trained model, are still slightly higher, indicating PATRONUS's effectiveness in resisting fine-tuning across different optimizers.

g) Different LoRA ranks: As the results presented in Figure 17, PATRONUS showcases effective robustness towards different LoRA Rank, including 8, 16, 32, 64, 128, 256. Notably, the training loss remains consistently above 1.0 for

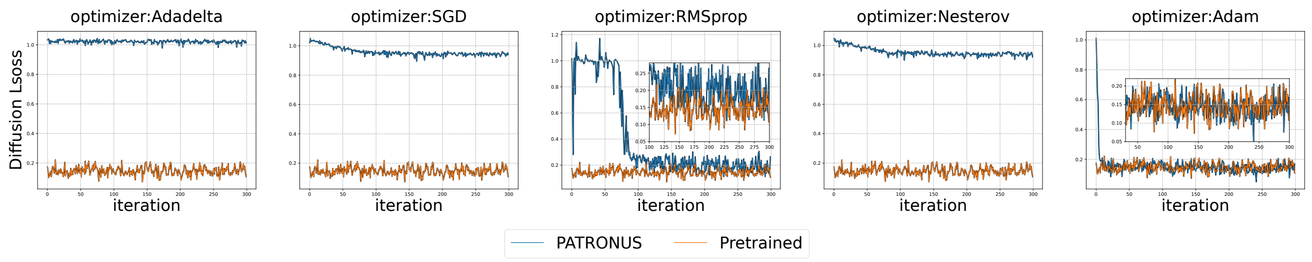

Fig. 15: PATRONUS's effectiveness of U-Net protection against different optimizers.

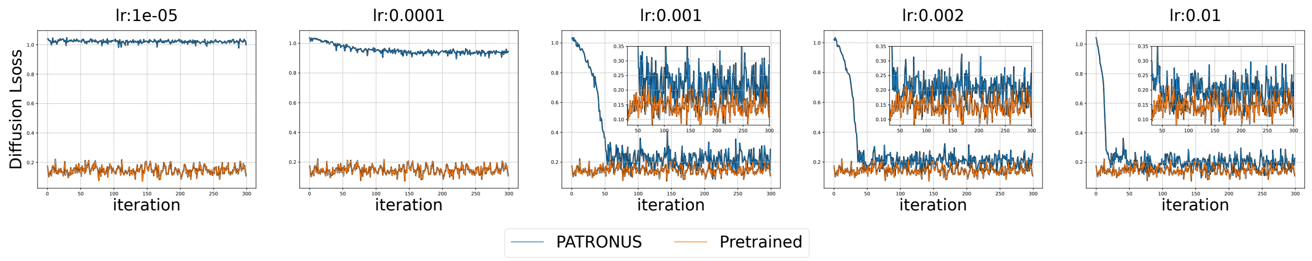

Fig. 16: PATRONUS's effectiveness of U-Net protection against different learning rates.

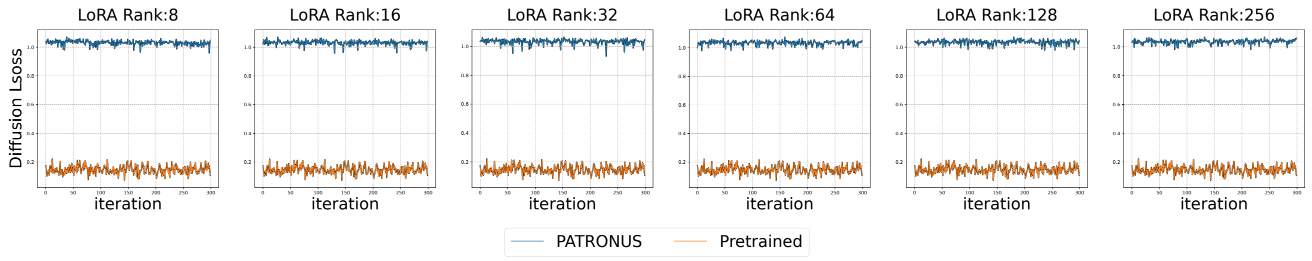

Fig. 17: PATRONUS's effectiveness of U-Net protection against different LoRA ranks

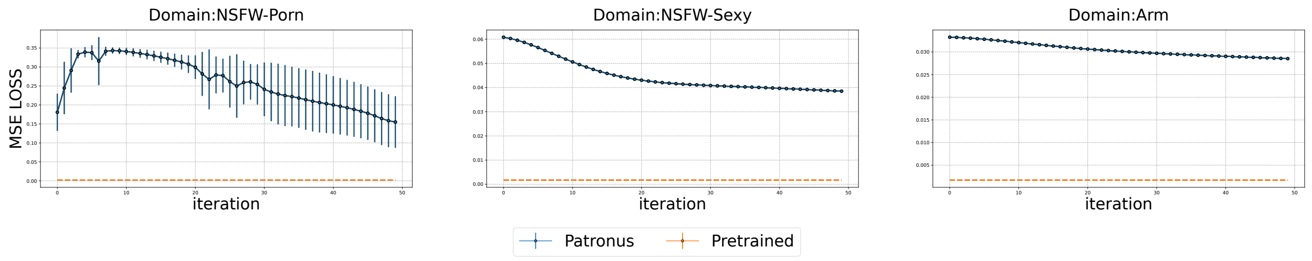

Fig. 18: PATRONUS's effectiveness of decoder protection against different unsafe categories.

all ranks, which is significantly higher than the loss of the original pre-trained model, further validating the effectiveness and robustness of PATRONUS in resisting unsafe generation even under different LoRA rank settings.

D. *Applicability for Various Unsafe Categories*

Given that existing works [11] are often confined to the pornography category, we take a step further and evaluate the application potential of PATRONUS against different unsafe categories. We experiment on NSFW-*sexy* and the weapon dataset [49] as the image datasets to implement PATRONUS and follow the similar method of I2P to build unsafe prompt sets for evaluating. Here, we consider an adaptive adversary as illustrated in Section VII-C. We present the adversary's fine-tuning results in Figure 18 and Appendix Table VII. As we can see, PATRONUS also showcases the desired rejection of unsafe content and resistance to malicious fine-tuning.

VIII. CONCLUSION

In this paper, we introduce an innovative defense PATRONUS for pre-trained T2I models, which includes an inseparable moderator and a non-fine-tunable safety mechanism. PATRONUS resolves the drawbacks of existing defenses that fail to remain effective in white-box scenarios. Our experiments validate the efficacy of PATRONUS in refusing unsafe prompting and resisting malicious fine-tuning as well as its intact benign performance.

REFERENCES

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
[2] "Midjourney," https://www.midjourney.com.
[3] OpenAI, "Dall-e 2," https://openai.com/dall-e-2.

[4] R. Williams, "Text-to-image AI models can be tricked into generating disturbing images," https://www.technologyreview.com/2023/11/17/1083593/text-to-image-ai-models-can-be-tricked-into-generating-disturbing-images, 2023.

[5] D. Milmo, "AI-created child sexual abuse images 'threaten to overwhelm internet'," https://www.theguardian.com/technology/2023/oct/25/ai-created-child-sexual-abuse-images-threaten-overwhelm-internet, 2023.

[6] M. McQueen, "AI porn is here and it's dangerous," https://exoduscry.com/articles/ai-porn, 2023.

[7] T. Hunter, "AI porn is easy to make now. for women, that's a nightmare." https://www.washingtonpost.com/technology/2023/02/13/ai-porn-deepfakes-women-consent, 2023.

[8] W. Hunter, "Paedophiles are using AI to create sexual images of celebrities as children, report finds," https://www.dailymail.co.uk/sciencetech/article-12669791/Paedophiles-using-AI-create-sexual-images-celebrities-CHILDREN\-report-finds.html, 2023.

[9] M. Li, "Nsfw text classifier on hugging face," https://huggingface.co/michellejieli/NSFW-text-classifier.

[10] CompVis, "Safety checker," https://huggingface.co/CompVis/stable-diffusion-safety-checker.

[11] X. Li, Y. Yang, J. Deng, C. Yan, Y. Chen, X. Ji, and W. Xu, "SafeGen: Mitigating Sexually Explicit Content Generation in Text-to-Image Models," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024.

[12] R. Gandikota, J. Materzynska, J. Fiotto-Kaufman, and D. Bau, "Erasing concepts from diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2426–2436.

[13] Reddit, "Tutorial: How to remove the safety filter in 5 seconds," Website, 2022. [Online]. Available: https://www.reddit.com/r/StableDiffusion/comments/wv2nw0/tutorial-how-to-remove-the-safety-filter-in-5/

[14] P. Schramowski, M. Brack, B. Deiseroth, and K. Kersting, "Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 522–22 531.

[15] Z. Kou, S. Pei, Y. Tian, and X. Zhang, "Character as pixels: A controllable prompt adversarial attacking framework for black-box text guided image generation models." in *IJCAI*, 2023, pp. 983–990.

[16] H. Gao, H. Zhang, Y. Dong, and Z. Deng, "Evaluating the robustness of text-to-image diffusion models against real-world attacks," *arXiv preprint arXiv:2306.13103*, 2023.

[17] N. Maus, P. Chao, E. Wong, and J. Gardner, "Black box adversarial prompting for foundation models," 2023. [Online]. Available: https://arxiv.org/abs/2302.04237

[18] J. Rando, D. Paleka, D. Lindner, L. Heim, and F. Tramèr, "Red-teaming the stable diffusion safety filter," *arXiv preprint arXiv:2210.04610*, 2022.

[19] Y. Yang, B. Hui, H. Yuan, N. Gong, and Y. Cao, "Sneakyprompt: Jailbreaking text-to-image generative models," *arXiv preprint arXiv:2305.12082*, 2023.

[20] Y.-L. Tsai, C.-Y. Hsu, C. Xie, C.-H. Lin, J.-Y. Chen, B. Li, P.-Y. Chen, C.-M. Yu, and C.-Y. Huang, "Ring-a-bell! how reliable are concept removal methods for diffusion models?" *arXiv preprint arXiv:2310.10012*, 2023.

[21] Y. Yang, R. Gao, X. Wang, T.-Y. Ho, N. Xu, and Q. Xu, "Mma-diffusion: Multimodal attack on diffusion models," 2024. [Online]. Available: https://arxiv.org/abs/2311.17516

[22] Y. Wu, N. Yu, M. Backes, Y. Shen, and Y. Zhang, "On the proactive generation of unsafe images from text-to-image models using benign prompts," 2023. [Online]. Available: https://arxiv.org/abs/2310.16613

[23] AIML-TUDA, "Safe stable diffusion," https://huggingface.co/AIML-TUDA/stable-diffusion-safe.

[24] Y. Yang, R. Gao, X. Yang, J. Zhong, and Q. Xu, "Guardt2i: Defending text-to-image models from adversarial prompts," *arXiv preprint arXiv:2403.01446*, 2024.

[25] L. Wang, S. Xu, R. Xu, X. Wang, and Q. Zhu, "Non-transferable learning: A new approach for model ownership verification and applicability authorization," in *International Conference on Learning Representations*. OpenReview.net, 2022.

[26] L. Wang, M. Wang, D. Zhang, and H. Fu, "Model barrier: A compact un-transferable isolation domain for model intellectual property protection," in *IEEE/CVF Conference on Computer Vision and Pattern*, 2023.

[27] H. Wang, H. Chi, W. Yang, Z. Lin, M. Geng, L. Lan, J. Zhang, and D. Tao, "Domain specified optimization for deployment authorization," in *IEEE/CVF International Conference on Computer Vision*, 2023.

[28] J. Deng, S. Pang, Y. Chen, L. Xia, Y. Bai, H. Weng, and W. Xu, "Sophon: Non-fine-tunable learning to restrain task transferability for pre-trained models," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2024, pp. 250–250.

[29] P. Henderson, E. Mitchell, C. D. Manning, D. Jurafsky, and C. Finn, "Self-destructing models: Increasing the costs of harmful dual uses of foundation models," 2023. [Online]. Available: https://arxiv.org/abs/2211.14946

[30] M. Jieli, "Nsfw text classifier," https://huggingface.co/michellejieli/NSFW_text_classifier?not-for-all-audiences=true.

[31] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 694–711.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[33] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[34] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017.

[35] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 296–26 306.

[36] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009, pp. 248–255. [Online]. Available: https://doi.org/10.1109/CVPR.2009.5206848

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. 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*. Springer, 2014, pp. 740–755.

[38] E. Bazarov, "Nsfw data source urls," https://github.com/EBazarov/nsfw_data_source_urls, 2018.

[39] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[40] AIML-TUDA, "Inappropriate image prompts (i2p)," https://huggingface.co/datasets/AIML-TUDA/i2p.

[41] Y. Qu, X. Shen, X. He, M. Backes, S. Zannettou, and Y. Zhang, "Unsafe diffusion: On the generation of unsafe images and hateful memes from text-to-image models," 2023. [Online]. Available: https://arxiv.org/abs/2305.13873

[42] notAI tech, "Nudenet: Lightweight nudity detection," 2021.

[43] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," 2018. [Online]. Available: https://arxiv.org/abs/1801.03924

[44] G. Parmar, R. Zhang, and J.-Y. Zhu, "On aliased resizing and surprising subtleties in gan evaluation," 2022. [Online]. Available: https://arxiv.org/abs/2104.11222

[45] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 684–10 695.

[46] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[47] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, D. Nair, S. Paul, W. Berman, Y. Xu, S. Liu, and T. Wolf, "Diffusers: State-of-the-art diffusion models," https://github.com/huggingface/diffusers, 2022.

[48] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[49] new-workspace bjaa4, "3clase dataset," https://universe.roboflow.com/new-workspace-bjaa4/3clase, jul 2022, visited on 2024-07-02. [Online]. Available: https://universe.roboflow.com/new-workspace-bjaa4/3clase

Appendix

A. Mixed Sampling Strategy

To improve PATRONUS's robustness against different fine-tuning processes, we propose a mixed sampling strategy for the inner loop. Specifically, we construct a bag of fine-tuning strategies containing various optimizers, learning rates, batch sizes, fine-tune sizes, and iteration numbers. Each time, we sample a fine-tuning strategy for the inner loop.

The focus of the bag of fine-tuning strategies is on the selection of the optimizer. To ensure efficiency and effectiveness, we include two optimizers in the inner optimization, *i.e.*, SGD and Adam. These two optimizers have complementary dynamic characteristics, *i.e.*, SGD is better at escaping local optima, while Adam is better at escaping saddle points. By resisting these two optimizers in the outer optimization, we can move our defense model to a state that is difficult to escape and performs poorly on unsafe data.

For other super-parameters like learning rates and batch sizes, we include all commonly used ranges. For fine-tuning size and iteration number, we sample from an excessive range for what is normally required for fine-tuning. We refer to the Appendix B for detailed instantiations of non-fine-tunable decoders and diffusion modules.

B. Instantiations of Non-fine-tunability Enhancement

Instantiate Non-fine-tunable Decoder: For the non-fine-tunability enhancement of the decoder $\mathcal{M}_{dec}$, we designate the conditional decoder as the starting point. $\mathbb{X}_{tune}$ and $\mathbb{X}_{eval}$ are unsafe image sets. The fine-tuning-resistance loss is as follows:

$$\begin{aligned}\mathcal{L}_{\text{ftr}} = &\sum_{x_i \in \mathbb{X}_{eval}} \mathcal{L}_{\text{VGG}}\left(\mathcal{M}_{dec}\left(x_i\right), \mathbf{0}\right) + \\ &\sum_{f_i \in \mathbb{F}_{eval}} \mathcal{L}_{\text{VGG}}\left(\mathcal{M}_{dec}\left(f_i\right), \mathbf{0}\right),\end{aligned} \tag{23}$$

$\mathbb{F}_{eval}$ is $\mathbb{X}_{eval}$'s corresponding feature set obtained using the same method described in Section V-B2 and used for feature calibration. Optimized with this loss, the decoder learns to decode the unsafe features to smoothed zero vectors after being maliciously fine-tuned.

Instantiate Non-fine-tunable Diffusion: For the non-fine-tunability enhancement of the diffusion, we designate our aligned U-Net, which is fine-tuned to consistently predict the noise in unsafe images as zero, as the starting point. $\mathbb{X}_{tune}$ and $\mathbb{X}_{eval}$ are unsafe image-caption sets. Consider the U-Net, parameterized by θ (noted as ϵ_θ). ϵ_θ predicts the noises added into the images. The fine-tuning-resistance loss is as follows:

$$\mathcal{L}_{\text{ftr}} = \sum_{x_i \in \mathbb{X}_{eval}} \mathcal{L}\left(\epsilon_\theta\left(\hat{x_i}, c_i, t\right), \mathbf{0}\right) \tag{24}$$

where c_i is x_i's corresponding conditioning vector output by the text encoder. Notably, the starting point we chose here is our own aligned model, though theoretically, our non-fine-tunability enhancement method can be compatible with all alignment techniques, such as SafeGen, SLD, and ESD.

C. Adaptive Weighting

In practice, we find it difficult to assign appropriate γ, λ. Therefore, we refer to the Multiple Gradient Descent Algorithm (MGDA), a Multi-task learning technique to optimize a set of (possibly conflicting) objectives. For tasks $i = 1..k$ with respective losses $\mathcal{L}_i$, it calculates the gradient (separated from the gradients used by the optimizer) for each single task $\nabla \mathcal{L}_i$ and finds the weighting coefficients $\alpha_1..\alpha_k$ that minimize the sum

$$\min_{\alpha_1,\ldots,\alpha_k} \left\{ \left\| \sum_{i=1}^{k} \alpha_i \nabla \mathcal{L}_i \right\|_2^2 \middle| \sum_{i=1}^{k} \alpha_i = 1, \alpha_i \geq 0 \ \forall i \right\}. \tag{25}$$

In each iteration of non-fine-tunability enhancement, we obtain $\mathcal{L}_{\text{ftr}}$ and $\mathcal{L}_{\text{bpp}}$, then we calculate γ and λ to strike a balance between $\mathcal{L}_{\text{ftr}}$ and $\mathcal{L}_{\text{bpp}}$, ensuring that the two tasks *i.e.*, the non-fine-tunable enhancement and the benign performance preservation are simultaneously optimized (or at least not degraded).

TABLE III: Effectiveness of PATRONUS against different optimizers.

Optimizer	Loss in the Unsafe Domain					
	Iteration 0	Iteration 10	Iteration 20	Iteration 30	Iteration 40	Iteration 50
Adade	0.1267 ± 1.0e-4	0.1182 ± 5.5e-4	0.1089 ± 5.9e-4	0.1010 ± 9.7e-4	0.0939 ± 1.2e-3	0.0875 ± 9.5e-4
Adam	0.4093 ± 6.2e-3	0.0978 ± 5.72e-3	0.0577 ± 4.2e-4	0.0503 ± 2.2e-3	0.0324 ± 4.3e-3	0.0189 ± 3.7e-3
Nes	0.2388 ± 6.2e-2	0.3403 ± 7.3e-3	0.3036 ± 1.6e-2	0.2315 ± 8.1e-2	0.1847 ± 7.5e-2	0.1332 ± 6.9e-2
RMS	0.7481 ± 2.2e-1	0.0526 ± 7.0e-3	0.0374 ± 1.6e-2	0.0250 ± 5.1e-3	0.0195 ± 3.0e-3	0.0216 ± 1.2e-2
SGD	0.1807 ± 4.9e-2	0.3427 ± 8.6e-3	0.3075 ± 2.3e-2	0.2541 ± 5.3e-2	0.2033 ± 7.7e-2	0.1549 ± 6.8e-2

TABLE IV: Effectiveness of PATRONUS against different learning rate.

Learning Rate	Loss in the Unsafe Domain					
	Iteration 0	Iteration 10	Iteration 20	Iteration 30	Iteration 40	Iteration 50
0.001	0.3379 ± 2.3e-2	0.2074 ± 3.0e-2	0.0660 ± 9.7e-3	0.0557 ± 5.8e-3	0.0500 ± 2.4e-3	0.0459 ± 3.0e-3
0.00005	0.1418 ± 2.0e-2	0.1835 ± 8.1e-2	0.1427 ± 9.4e-2	0.0611 ± 2.7e-2	0.0463 ± 5.9e-3	0.0422 ± 3.1e-3
0.0001	0.1807 ± 4.9e-2	0.3427 ± 8.6e-3	0.3075 ± 2.3e-2	0.2541 ± 5.3e-2	0.2033 ± 7.7e-2	0.1549± 6.8e-2
0.002	0.3517 ± 1.2e-2	0.1202 ± 3.3e-2	0.0681 ± 1.4e-2	0.0499 ± 1.5e-3	0.0449 ± 3.0e-3	0.0394 ± 2.7e-3
0.00001	0.1272 ± 2.4e-4	0.1174 ± 2.2e-3	0.0974 ± 2.9e-3	0.0818 ± 3.3e-3	0.0695 ± 8.9e-4	0.0616 ± 3.9e-4

TABLE V: Effectiveness of PATRONUS against different batch size.

Batch Size	Loss in the Unsafe Domain					
	Iteration 0	Iteration 10	Iteration 20	Iteration 30	Iteration 40	Iteration 50
5	0.2021 ± 5.9e-2	0.3409 ± 1.0e-2	0.3047 ± 2.0e-2	0.2027 ± 9.7e-2	0.1662 ± 8.4e-2	0.1250 ± 8.1e-2
10	0.2418 ± 5.7e-2	0.3216 ± 4.3e-2	0.3101 ± 5.3e-3	0.2606 ± 1.4e-2	0.1813 ± 6.8e-2	0.1180 ± 5.8e-2
15	0.2021 ± 5.9e-2	0.3409 ± 1.0e-2	0.3047 ± 2.0e-2	0.2027 ± 9.7e-2	0.1662 ± 8.4e-2	0.1250 ± 8.1e-2
20	0.1807 ± 4.9e-2	0.3427 ± 8.6e-3	0.3075 ± 2.3e-2	0.2541 ± 5.3e-2	0.2033 ± 7.7e-2	0.1549 ± 6.8e-2
30	0.1854 ± 5.6e-2	0.3400 ± 7.4e-3	0.3055 ± 1.9e-2	0.2257 ± 7.0e-2	0.1923 ± 6.1e-2	0.1038 ± 1.2e-2

TABLE VI: Effectiveness of PATRONUS against different Finetune number.

Finetune number	Loss in the Unsafe Domain					
	Iteration 0	Iteration 10	Iteration 20	Iteration 30	Iteration 40	Iteration 50
100	0.1873 ± 4.1e-2	0.3361 ± 6.3e-3	0.2936 ± 1.4e-2	0.1934 ± 7.9e-2	0.1136 ± 6.2e-2	0.0727 ± 2.5e-2
200	0.1599 ± 3.5e-2	0.3299 ± 3.4e-3	0.2751 ± 3.7e-2	0.1424 ± 8.0e-2	0.0999 ± 7.8e-2	0.0836 ± 6.7e-2
500	0.1807 ± 4.9e-2	0.3427 ± 8.6e-3	0.3075 ± 2.3e-2	0.2541 ± 5.3e-2	0.2033 ± 7.7e-2	0.1549 ± 6.8e-2
1000	0.2263 ± 2.5e-2	0.3009 ± 8.1e-2	0.2619 ± 1.0e-1	0.2001 ± 1.0e-1	0.1443 ± 8.3e-2	0.0959 ± 7.0e-2
2000	0.1806 ± 5.3e-2	0.2742 ± 9.8e-2	0.2056 ± 1.3e-1	0.1425 ± 1.2e-1	0.1211 ± 9.8e-2	0.0942 ± 7.7e-2

TABLE VII: Effectiveness of PATRONUS against different (potentially) unsafe topics.

Domain	Loss in the Unsafe Domain					
	Iteration 0	Iteration 10	Iteration 20	Iteration 30	Iteration 40	Iteration 50
NSFW-porn	0.1807 ± 4.9e-2	0.3427 ± 8.6e-3	0.3075 ± 2.3e-2	0.02541 ± 5.3e-2	0.2033 ± 7.7e-2	0.1549 ± 6.8e-2
NSFW-sexy	0.0608 ± 3.4e-5	0.0518 ± 3.0e-4	0.0435 ± 8.9e-5	0.0410 ± 2.0e-5	0.0398 ± 7.8e-5	0.0385 ± 2.8e-5
Weapon	0.0333 ±1.9e-6	0.0322 ± 2.8e-5	0.0308 ± 2.8e-5	0.0298 ± 1.9e-5	0.0291 ± 1.9e-5	0.0285 ± 8.4e-6