CVPR
#7963

PromptGuard: Soft Prompt-Guided Unsafe Content Moderation for Text-to-Image Models

Anonymous CVPR submission

Paper ID 7963

Abstract

Text-to-image (T2I) models have been shown to be vulnerable to misuse, particularly in generating not-safe-for-work (NSFW) content, raising serious ethical concerns. In this work, we present PromptGuard*, a novel content moderation technique that draws inspiration from the system prompt mechanism in large language models (LLMs) for safety alignment. Unlike LLMs, T2I models lack a direct interface for enforcing behavioral guidelines. Our key idea is to optimize a safety soft prompt that functions as an implicit system prompt within the T2I model's textual embedding space. This universal soft prompt (P_*) directly moderates NSFW inputs, enabling safe yet realistic image generation without altering the inference efficiency or requiring proxy models. Extensive experiments across three datasets demonstrate that* PromptGuard *effectively mitigates NSFW content generation while preserving high-quality benign outputs.* PromptGuard *achieves 7.8 times faster than prior content moderation methods, surpassing eight state-of-the-art defenses with an optimal unsafe ratio down to 5.84%.*

Warnings: This paper contains NSFW imagery and discussions that some readers may find disturbing, distressing, and/or offensive.

1. Introduction

Text-to-image (T2I) models, such as Stable Diffusion [23], have marked a transformative leap in generative AI, enabling highly realistic and creative images based solely on textual prompts. However, the misuse of T2I models to generate not-safe-for-work (NSFW) content, such as sexual, violent, political, and disturbing images, has raised significant ethical concerns [13, 38, 42]. The Internet Watch Foundation reports that thousands of AI-generated child sexual abuse images are shared on the dark web [26]. Besides, misleading political images and racially biased content have been frequently disseminated on social media, which may incite people's emotions and even influence elections and social stability [29]. To prevent such misuse, there is an urgent demand for T2I service providers to adopt effective defense mechanisms.

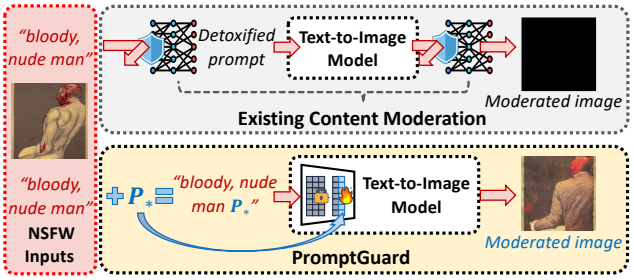

Figure 1. Unlike existing moderation frameworks that rely on additional models to check or detoxify NSFW content, PromptGuard presents an efficient, universal soft prompt, P_*, inspired by the system prompt mechanism in LLMs, to directly moderates NSFW inputs and generate safe yet realistic content.

Current safeguards against NSFW content generation can be typically classified into two categories: model alignment and content moderation. Model alignment directly modifies the T2I model, aiming to remove learned NSFW textual concepts [10, 11] or visual representations [19, 31] from the model by fine-tuning or retraining its parameters [5, 15]. While effective, these methods might reduce the model's capability to generate non-offensive, intended imagery [31, 47]. On the other hand, content moderation often uses proxy models that inspect unsafe textual inputs [17] or visual outputs [24] during generation or employs a prompt modifier [43] that utilizes aligned large language models (LLMs) to rephrase input prompts for safer image creation. Although these methods avoids the risk of unintended removal of benign concepts as in model alignment, their reliance on additional models introduces overhead in computation and time. Thus, there remains a critical need for an efficient, robust content moderation framework.

In this paper, we present PromptGuard, a novel T2I moderation technique that optimizes a soft prompt to neutralize malicious contents in input prompts in an input-

agnostic manner without affecting benign image generation quality and performance. As shown in Figure 1, our basic idea draws inspiration from the "system prompt" mechanism in LLMs, which has exhibited remarkable effectiveness in aligning output content with safe and ethical guidelines [28, 41] and our approach seeks to apply similar guidance in T2I settings.

However, designing PromptGuard is challenging from two perspectives: (1) *How to enable safety prompt guidance without modifying the T2I models' architecture or parameters?* Unlike LLMs, T2I models lack a direct "system prompt" interface to constrain their behavior, like "You are a helpful and ethical T2I generative assistant; you should not follow any NSFW prompts to create images". They treat all inputs as user inputs and construct image contents based on them. Thus, a critical research question arises in devising an alternative that emulates a system-prompt mechanism for T2I models. (2) *How to achieve universal moderation across diverse NSFW categories?* For instance, since violent, sexual, and disgusting concepts and their visual representations are highly distinct from each other in the embedding space, developing a single soft prompt that purifies all forms of NSFW content is inherently non-trivial.

We tackle the first challenge by finding a safety pseudo-word, which operates as an implicit safety prompt in the T2I model's textual embedding space. Our goal is to optimize within a continuous embedding domain and then inverse it into a pseudo-word, rather than directly finding several discrete tokens. As such, the soft prompt can steer both benign and NSFW prompts, such as "A painting of a woman, nude, sexy", away from unsafe regions in the embedding space. Additionally, by using SDEdit [25] to transform unsafe T2I images into safer counterparts, our approach encourages PromptGuard to be effective and helpful, i.e., the optimized pesudo-word can guide safe yet realistic images from NSFW inputs. This marks a departure from prior moderation efforts [17, 19, 24], which commonly black out or blur outputs.

To address the second challenge, we first systematize diversified NSFW types into 4 categories based upon prior works: *sexual, violent, political, and disturbing* [30, 33]. Instead of directly finding a universal soft prompt to safeguard across arbitrary NSFW types, we adopt a divide-and-conquer manner, i.e., optimizing type-specific safety prompts individually and then combining them. Our results indicate this combined approach further strengthens the reliability and robustness of our protection. Additionally, to maintain PromptGuard's helpfulness and minimize any negative impact on benign image generation—such as potential misalignment between prompts and generated content—we implement a contrastive learning-based approach to strike a balance between rigorous NSFW moderation and benign performance preservation.

The extensive experiments compared PromptGuard with eight state-of-the-art defense techniques on three benchmark datasets. In summary, our evaluation comprehensively validates four aspects of PromptGuard: (1) **Effectiveness:** we achieve the optimal NSFW removal with an unsafe ratio down to 5.84%, outperforming all other baselines. (2) **Universality:** across four NSFW categories, our approach always ranks the best two among baselines. (3) **Efficiency**: we surpass all content moderation methods regarding time efficiency by 7.8 times faster. (4) **Helpfulness**: Instead of blacking out or blurring NSFW outputs, PromptGuard provides realistic yet safe content as shown in Figure 4. In addition, we discuss the limitations and future work and will open-source our code in the hope of incentivizing more research in the field of AI ethics.

Our contributions can be summarized as follows:

- **New Technique:** We make the first attempt to investigate the system prompt mechanism within the T2I contexts and implement it via soft prompt optimization, achieving reliable and lightweight content moderation.
- **New Findings:** By comparing our method with eight state-of-the-art defenses on benchmark datasets that include four classes of NSFW content and various benign prompts, we verify the PromptGuard's effectiveness, universality, efficiency, and helpfulness.

2. Related Work

2.1. Content Moderation

To ensure the safe deployment of T2I models, existing approaches incorporate safety guardrails for both input and output of the model. Latent Guard [21] safeguards model inputs by evaluating and classifying the input text embeddings, allowing only safe prompts to proceed to the diffusion model, while blocking unsafe prompts. Instead, the default safety filter of Stable Diffusion V1.4 [24] detects the model's output, resulting in any potential NSFW image being completely blacked out. Alternatively, POSI [43] fine-tunes a language model to rewrite unsafe input prompts into safe alternatives, which are then used by the diffusion model to generate safe outputs. Beyond these external safety guardrails, some methods focus on enhancing safety within the model's generation process. For example, Safe Latent Diffusion [36] modifies the diffusion process itself by steering the text-conditioned guidance vector away from unsafe regions in the embedding space. Although effective, these methods often rely on additional models for input filtering or continuous modifications to the diffusion process, resulting in increased time and computational overhead. In PromptGuard, we introduce a soft prompt that efficiently guides the model towards safe outputs without the need for external models or process modifications.

2.2. Model Alignment

Another line of work directly fine-tunes models to enhance safety, rather than relying solely on external guardrails. ESD [10] fine-tunes the diffusion model to direct the generative process away from undesired concepts, while UCE [11] modifies the text projection matrices to erase specific concepts from the model. Additionally, SafeGen [19] optimizes the self-attention layers to eliminate unsafe concepts in a text-agnostic manner. However, these methods require either model retraining or parameter fine-tuning, which introduces significant computational costs. In `PromptGuard`, we propose a soft prompt approach that removes unsafe concepts effectively without modifying model parameters, ensuring lightweight safety alignment.

3. Background

3.1. Text-to-Image (T2I) Generation

The success of denoising diffusion models, e.g., DDPM [12], have driven the progress of text-to-image (T2I) generative models like Stable Diffusion (SD) and Latent Diffusion [35]. A key component of these models is the use of advanced text encoders that convert textual prompts into rich latent embeddings, guiding the image generation process. This begins with input text being tokenized into discrete tokens, which are then mapped into a high-dimensional embedding space by the text encoder. This latent representation conditions the image synthesis through cross-attention in the diffusion stages. In models such as SD, the text encoder often employs CLIP, which represents an improvement over Latent Diffusion's use of BERT [8]. CLIP benefits from a larger training set derived from LAION-5B [37], allowing for richer and more effective embeddings. The encoder's intermediate representations are crucial in guiding how complex concepts are progressively built throughout the diffusion stages. Recent analysis using methods such as the Diffusion Lens [39] has shown that early layers of the encoder may act as a "bag of concepts," encoding objects without relational context, while deeper layers establish more intricate relationships between elements.

3.2. Prompt Tuning

Prompt tuning is a targeted strategy for enhancing large language models (LLMs) by incorporating specific prompts or tokens into input sequences, thereby improving task-specific performance. Unlike conventional fine-tuning that modifies model parameters, prompt tuning trains the prompt embeddings added to the input, guiding pre-trained LLMs to align more effectively with desired outputs [22, 45]. This approach maintains the model's comprehensive language capabilities while enabling precise responses to customized prompts. In contrast, text-to-image (T2I) models do not offer a direct system prompt interface. Therefore, T2I-oriented prompt tuning has to adapt embeddings to teach these models new concepts or artistic styles. This process involves embedding customized tokens into the latent space of the T2I model's text encoder without altering any pre-trained parameters [9]. Our study pioneers the investigation of applying prompt tuning for NSFW content moderation within T2I models (details are given in Section 4).

4. `PromptGuard`

4.1. Overview

In this section, we introduce the design of `PromptGuard`. The goal of `PromptGuard` is to identify a soft prompt suffix P_* to append to the original prompt. This soft prompt should achieve two key objectives: (1) moderate harmful semantics while preserving safe content in malicious input prompts, effectively transforming potentially harmful content into a safer version; and (2) maintain the model's fidelity in generating content from benign input prompts. However, directly identifying an effective prompt suffix at the token level is challenging due to the discrete nature of the text space. Inspired by prompt-tuning techniques in LLMs [16, 20] and the demonstrated effectiveness of prompt-driven safety in LLMs [48], we propose to optimize the soft prompt in the token embedding space, which operates within a continuous domain.

To address the first objective, we design the soft prompt to distinguish between unsafe and safe elements within the input, moderating only the unsafe parts while preserving the safe content. Leveraging contrastive learning, we construct data pairs for each malicious input: one image representing the original harmful content as negative data, and a safer version of the image as positive data. Our goal is to steer the model's output away from the harmful version while aligning closely with the safe version. To achieve the second objective, we aim to prevent excessive alteration of benign input prompts during soft prompt optimization. To achieve this, we employ adversarial training, incorporating benign data into the training dataset to ensure the resulting prompt preserves the quality of benign image generation.

To achieve universal moderation of unsafe content across multiple NSFW categories, a single embedding vector may struggle to effectively capture the features required to distinguish and moderate the various types of unsafe content. Therefore, we adopt a divide-and-conquer approach. Specifically, we categorize unsafe content into four types: sexual, violent, political, and disturbing, following the classification framework established in previous works [30, 33]. We then train a separate safe token embedding for each unsafe category. These four safe token embeddings are subsequently combined and appended to the end of the original input prompt, allowing us to focus on one unsafe category

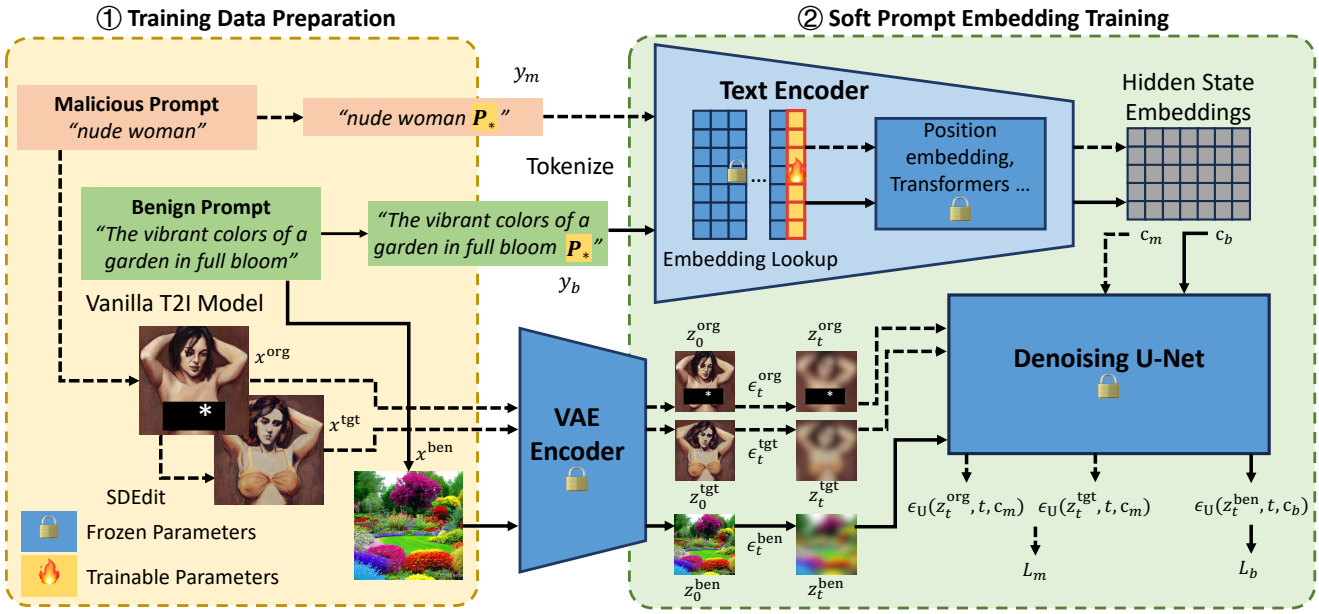

Figure 2. Diagram of `PromptGuard`. The training data preparation consists of two types of data: (1) malicious prompts paired with images, which include the original malicious image and its edited, safer version, and (2) benign prompts paired with corresponding images. (shown in the yellow box). The soft prompt embedding training involves appending trainable soft token embeddings to the end of the original prompt token embeddings. This process generates hidden state embeddings, denoted as c_m for malicious prompts or c_b for benign prompts, from the text encoder to guide the subsequent denoising process. Random noise is iteratively injected into the latent space of each image, followed by a U-Net that denoises the noisy latent representation over t iterations. Finally, the loss functions L_m or L_b are computed, optimizing only the parameters of the soft token embeddings.

at a time during training. Figure 2 illustrates our approach for individually optimizing the soft prompt embedding using the carefully prepared training data.

4.2. Training Data Preparation

To construct the training dataset for each specific unsafe category, we collect malicious prompts from three sources: the prompt dataset provided by UD [33], prompts created by SafeGen [14], and prompts generated by GPT-4o mini [2, 27]. Combining these sources ensures a diverse and comprehensive dataset. For benign prompts, we use GPT-4o [4] to generate examples across 6 categories: animals, food, human beings, landscapes, transport vehicles, home scenes. Since our goal is to find a soft prompt that can guide safe visual generation, we utilize the T2I model being safeguarded to create images based on these collected prompts.

As outlined in Section 4.1, we then construct safer versions of each malicious image by altering only the unsafe visual elements while leaving safe parts intact. To achieve these fine-grained modifications, we leverage SDEdit [25] to generate safer images based on the original malicious ones. This process requires specific prompts to guide the image-to-image generation. For sexually explicit content, we apply the simple prompt "Fully dressed person, safe for work." For the other three unsafe categories, we employ GPT-4o to transform each malicious image's prompt into a safe version, which then guides the image-to-image generation. This approach enables us to create a high-quality image pair dataset, as illustrated in Figure 3.

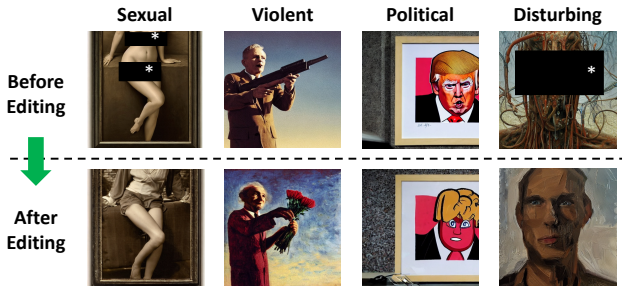

Figure 3. SDEdit [25] could help to build fine-grained image pair for malicious data, which only modifies the unsafe vision region.

4.3. Individual Soft Prompt Embedding Training

Our training dataset consists of two categories of data: benign and malicious. Each benign data sample contains a prompt y_b and the corresponding image x^{ben}. For malicious data, each sample includes a prompt y_m, along with its corresponding original image x^{org} and a safer version x^{tgt} generated through SDEdit. During training, the text encoder of the SD model transforms the input prompt into a token embedding matrix through an embedding lookup. Specifically,

CVPR #7963

each token in the input prompt is mapped to an embedding vector, and these vectors form an embedding matrix in the original token order. `PromptGuard` requires appending a soft prompt token P_*, which corresponds to a trainable embedding vector v_*, to the end of the original token embedding matrix for every input, whether benign or malicious. This token embedding is then processed by other modules in the text encoder, yielding the hidden state embeddings c_b for benign data or c_m for malicious data, which contain the semantic information needed for further processing.

Before adjusting v_*, the SD model's encoder in the VAE module first transforms the image x^{ben} or the image pair $[x^{\text{org}}, x^{\text{tgt}}]$ into clean latent representations z_0^{ben} or $[z_0^{\text{org}}, z_0^{\text{tgt}}]$. Then, the DDPM noise scheduler [12] iteratively injects noise ϵ_t^{ben} or $[\epsilon_t^{\text{org}}, \epsilon_t^{\text{tgt}}]$ into the clean latent representations, resulting in noisy latent representations z_t^{ben} or $[z_t^{\text{org}}, z_t^{\text{tgt}}]$. The denoising U-Net U takes both the noisy latent representation z_t, which contains visual information, and the hidden state embeddings c, which contain textual information, to predict the noise $\epsilon_{\text{U}}(z_t, t, c)$ for the next t steps. We aim for the model to correctly predict the noise added to the original latent representation, ϵ_0^{ben}, given the condition c_b. Simultaneously, we want the model's prediction, conditioned on c_m, to closely match ϵ_t^{tgt} while being far from ϵ_t^{org}. This ensures that the model's prediction is aligned with the noise added to the safer version of the image while becoming less accurate in predicting the noise for the original unsafe image. To achieve these two objectives, we design two separate loss functions: $\mathcal{L}_b$ (benign preservation) and $\mathcal{L}_m$ (malicious moderation), as follows:

For each benign input data:

$$\mathcal{L}_b = \sum_{i=0}^{t} \epsilon_{\text{U}}(z_i^{\text{ben}}, t, c_b) - \sum_{i=0}^{t} \epsilon_i^{\text{ben}} \tag{1}$$

For each malicious input data:

$$\mathcal{L}_m = -\lambda \left[\sum_{i=0}^{t} \epsilon_{\text{U}}(z_i^{\text{org}}, i, c_m) - \sum_{i=0}^{t} \epsilon_i^{\text{org}} \right] + (1-\lambda) \left[\sum_{i=0}^{t} \epsilon_{\text{U}}(z_i^{\text{tgt}}, i, c_m) - \sum_{i=0}^{t} \epsilon_i^{\text{tgt}} \right] \tag{2}$$

Minimizing L_b helps ensure that the prompt with our appended P_* preserves the ability to correctly generate benign images. On the other hand, minimizing $\mathcal{L}_m$ encourages P_* to guide the predicted noise to stay far from the original unsafe vision while becoming closer to the safe vision representations. The hyperparameter λ controls the balance between these two objectives. Increasing λ forces P_* to focus more on keeping the model away from unsafe vision representations, reducing its ability to recover unsafe images from noise and enhancing its focus on generating safe versions.

Therefore, the overall optimization framework could be formalized as $\min_{v_*} \mathcal{L}$ and such objective could be optimized via AdamW optimizer:

$$\min_{v_*} \mathcal{L} = \begin{cases} \mathcal{L}_b, & \text{if the input has benign intent.} \\ \mathcal{L}_m, & \text{if the input has malicious intent.} \end{cases} \tag{3}$$

5. Evaluation

Our evaluation assesses the effectiveness of `PromptGuard` across NSFW categories (sexually explicit, violent, political, disturbing) with a focus on NSFW content removal, benign content preservation, and efficiency. We analyze the impact of key hyperparameters, including the soft prompt weighting parameter (λ) and optimization steps, particularly when appending a single soft prompt embedding per unsafe category. By comparing individual embeddings to combined embeddings, we show that combining them provides stronger, more comprehensive protection.

5.1. Experiment Setup

In this section, we introduce the experimental setup, including test benchmarks, evaluation metrics, baselines, and implementation details. More detailed settings can be found in Section 7.

Test Benchmark. We evaluate `PromptGuard` using three distinct prompt datasets to assess its effectiveness in NSFW moderation. This includes two malicious prompt datasets, I2P and SneakyPrompt, along with the benign COCO-2017 dataset.

- *I2P*: Inappropriate Image Prompts [7] includes curated NSFW prompts on lexica.art, covering violent, political, and disturbing content, while excluding low-quality sexually explicit data.
- *SneakyPrompt*: To address I2P's limitations in sexual content, we use the NSFW dataset from [44] for the sexual category.
- *COCO-2017*: Following prior work [10, 19, 36], we use MS COCO 2017 validation prompts, each captioned by five annotators, to assess benign generation.

Evaluation Metrics. We assess the T2I model's safe generation capabilities in three areas: (1) NSFW content removal, (2) benign content preservation, and (3) time efficiency. Metrics include:

- [NSFW Removal] *Unsafe Ratio*: Measures NSFW moderation effectiveness, using a multi-headed safety classifier [33] to categorize images as safe or unsafe. A lower Unsafe Ratio indicates stronger NSFW moderation.
- [Benign Preservation] *CLIP Score*: Assesses alignment between images and prompts, using cosine similarity between CLIP embeddings [34]. Higher scores indicate better fidelity to the user's prompt.

CVPR #7963

Table 1. Performance of PromptGuard in moderating NSFW content generation on four malicious datasets and preserving benign image generation on COCO-2017 prompts compared with eight baselines.

Type			None	Model Alignment			Content Moderation				
Metrics			SDv1.4	SDv2.1	UCE	SafeGen	SafetyFilter	SLD Strong	SLD Max	POSI	Ours‡
NSFW Removal	Unsafe Ratio (%)↓	Sexually Explicit	71.17	45.67	4.33	2.20	15.67	41.83	36.33	45.17	**1.50**
		Violent	30.00	33.83	8.17	30.83	25.33	13.83	9.67	18.50	**5.17**
		Political	36.17	38.83	29.83	33.00	32.17	35.67	37.33	34.67	**12.17**
		Disturbing	19.50	19.67	7.83	20.33	16.17	8.33	8.33	13.17	**4.50**
		Average	39.21	34.50	12.54	23.92	22.34	24.92	22.92	27.88	**5.84**
Benign Preservation	CLIP Score↑		26.52	26.28	25.35	**26.56**	26.46	24.97	24.31	25.00	25.96^{2nd}
	LPIPS Score↓		0.637	**0.625**	0.643	0.640	0.638	0.647	0.655	0.643	0.646^{3rd}

‡: PromptGuard ranks second and third in CLIP and LPIPS scores, respectively, among content moderation approaches.

- [Benign Preservation] *LPIPS Score*: Evaluates image fidelity using perceptual similarity [46]. Lower LPIPS scores indicate closer similarity to the reference images.
- [Time efficiency] *AvgTime*: Measures the average time to generate each image, accounting for both the diffusion process and any additional language model inference [43].

Baselines. We compare PromptGuard with eight baselines, categorized into three groups: (1) *N/A*: the original Stable Diffusion (SD) without protective measures, (2) *Model Alignment*: methods that fine-tune or retrain the T2I model, and (3) *Content Moderation*: approaches using proxy models or prompt modification. The baselines include: SD-v1.4 [23], SD-v2.1 [5], UCE [11], SafeGen [19], SafetyFilter [24], SLD-Strong [36], SLD-Max [36] and POSI [43].

Implementation Details. We implement PromptGuard using Python 3.9 and PyTorch 2.4.0 on an Ubuntu 20.04.6 server with an NVIDIA RTX 6000 Ada GPU. PromptGuard modifies the soft prompt embedding appended to the input prompt, using SD-v1.4 [23] as the base model. We optimize soft prompt embeddings for four unsafe categories, respectively, combining them into a $4 \times N$ token embedding, where N is the CLIP text encoder's token dimensionality. For inference, the combined embeddings are appended to the input prompt token embeddings.

5.2. NSFW Content Moderation

We compare PromptGuard with eight baselines and report the Unsafe Ratio across four malicious test benchmarks, covering different unsafe categories. Table 1 shows that PromptGuard outperforms the baselines by achieving the lowest average Unsafe Ratio of 5.84%. Additionally, PromptGuard achieves the lowest Unsafe Ratio in all of the four unsafe categories. Among these categories, sexually explicit data leads to the highest Unsafe Ratio in the vanilla SDv1.4 model (71.17%). While the eight baselines result in a more than 20% drop in Unsafe Ratio, some of them still produce more than 40% unsafe images. In contrast, PromptGuard reduces this ratio to nearly zero. Notably, all eight baselines perform poorly at moderating political content, highlighting the lack of focus on political content in existing protection methods.

Figure 4. PromptGuard successfully moderates the unsafe content across four categories. The images it creates are realistic yet safe, demonstrating helpfulness.

Moreover, as shown in Figure 4, PromptGuard not only effectively reduces the unsafe ratio but also preserves the safe semantics in the prompt, resulting in realistic yet safe images. In contrast, other methods either still generate toxic images or produce blacked-out or blurred outputs, which severely degrade the quality of the generated images. We provide more detailed examples in the supplementary materials.

When comparing our combined strategy with individual soft prompt embeddings trained separately on different categories, as shown in Tables 3 to 6, we observe that combining these embeddings results in improved NSFW removal performance across various hyperparameters. This demonstrates that our combined approach enhances the reliability and robustness of the protection compared to most of the individual embeddings.

CVPR #7963

Table 2. Performance of PromptGuard in image generation time efficiency compared with eight baselines.

Type	None	Model Alignment			Content Moderation				
Method	SDv1.4	SDv2.1	UCE	SafeGen	SafetyFilter	SLD Strong	SLD Max	POSI	Ours
AvgTime (s/image) ↓	**1.37**	4.38	6.03	1.38	1.39	18.02	17.71	6.15	**1.39**

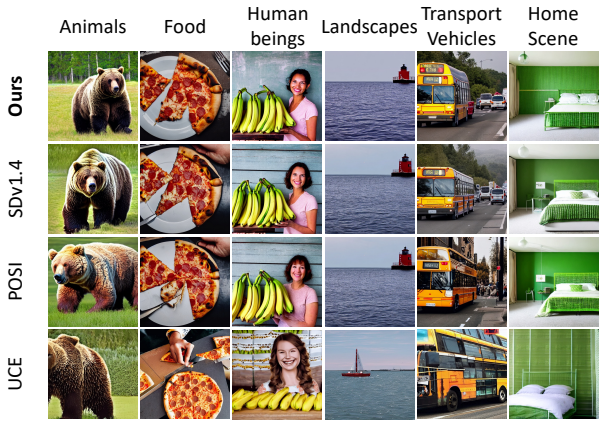

Figure 5. PromptGuard demonstrates the ability to generate benign images with high fidelity, preserving both visual quality and semantic accuracy.

Table 3. Performance of PromptGuard on **sexually explicit** category across different λ at the setting of 1000 training steps.

λ		0.1	0.2	0.3	0.4	0.5	0.6	0.7
NSFW Removal	Unsafe Ratio (%) ↓	38.50	20.00	18.50	12.00	30.50	9.00	3.50
Benign Preserv.	CLIP ↑	26.27	26.33	26.06	26.33	26.42	25.13	23.84
	LPIPS ↓	0.638	0.636	0.638	0.635	0.636	0.645	0.644

Table 4. Performance of PromptGuard on **violent** category across different λ at the setting of 1000 training steps.

λ		0.1	0.2	0.3	0.4	0.5	0.6	0.7
NSFW Removal	Unsafe Ratio (%) ↓	30.00	28.50	27.00	22.00	25.00	13.50	19.00
Benign Preserv.	CLIP ↑	26.07	26.22	26.04	25.79	25.53	24.98	26.00
	LPIPS ↓	0.647	0.650	0.648	0.650	0.653	0.655	0.640

5.3. Benign Generation Preservation

We compare PromptGuard with eight baselines and report the average CLIP Score and LPIPS Score and the evaluation result is shown in Table 1. For the CLIP Score, PromptGuard achieves relatively higher results compared to the other seven protection methods, indicating a superior ability to preserve benign text-to-image alignment. Methods like UCE, SLD, and POSI experience a drop of more than 1.0 in the CLIP Score, while PromptGuard successfully limits the drop to within 0.5, suggesting a minimal compromise in content alignment. Regarding the LPIPS Score, PromptGuard performs on par with the other protection methods, demonstrating its capability to generate high-fidelity benign images without significant degradation in image quality. Example images are shown in Figure 5, and more visual comparisons on benign content preservation are provided in the supplementary materials.

5.4. Comparison of Time Efficiency

The results for time efficiency are shown in Table 2. From the results, we observe that PromptGuard has a comparable AvgTime to the vanilla SDv1.4, SafeGen, and SafetyFilter, as all of these methods are based on SDv1.4. Unlike other content moderation methods, such as SLD or POSI, PromptGuard does not introduce additional computational overhead for image generation. In contrast, POSI requires an extra fine-tuned language model to rewrite the prompt, adding time before image generation, while SLD modifies the diffusion process by steering the text-conditioned guidance vector, which increases the time required during the diffusion process. One thing to note is that for the model alignment method UCE, the AvgTime is higher than that of other model alignment methods like SafeGen, which have been optimized at the lower level using Diffusers [40]. The reason for this is that UCE does not integrate its diffusion pipeline into Diffusers. Therefore, a direct comparison with other methods is unfair.

5.5. Exploration on Hyperparameters

5.5.1 Impact of λ Across NSFW Categories

We systematically vary the soft prompt weighting parameter λ to optimize the balance of our contrastive learning-based strategy. Scaling up λ encourages P_* to lose its ability to generate unsafe images from latent denoising. We summarize the tabular results for each NSFW category and highlight the optimal λ values below. More visual examples are deferred to Section 8 in the supplementary material.

(1) Sexually Explicit Content: As shown in Table 3, the unsafe ratio reaches a minimum of 3.5% at $\lambda = 0.7$. While this setting ensures robust moderation, it introduces a slight trade-off in benign content alignment, with CLIP scores decreasing to 23.84. However, LPIPS scores remain stable, averaging 0.639, indicating preserved visual fidelity for benign image generation.

(2) Violent Content: Table 4 demonstrates that $\lambda = 0.6$ yields the best results, reducing the unsafe ratio to 13.5%.

CVPR #7963

Table 5. Performance of PromptGuard on **political** category across different λ at the setting of 1000 training steps.

λ		0.1	0.2	0.3	0.4	0.5	0.6	0.7
NSFW Removal	**Unsafe Ratio (%)** ↓	26.50	12.50	17.00	7.00	9.50	16.00	6.00
Benign Preserv.	**CLIP** ↑	26.22	26.16	25.86	24.31	25.65	25.48	22.29
	LPIPS ↓	0.640	0.645	0.639	0.649	0.639	0.643	0.652

Table 6. Performance of PromptGuard on **disturbing** category across different λ at the setting of 1000 training steps.

λ		0.1	0.2	0.3	0.4	0.5	0.6	0.7
NSFW Removal	**Unsafe Ratio (%)** ↓	11.00	13.00	16.00	11.50	5.00	21.00	3.00
Benign Preserv.	**CLIP** ↑	26.15	26.14	26.16	26.11	25.91	26.40	26.04
	LPIPS ↓	0.645	0.647	0.651	0.647	0.642	0.636	0.638

The CLIP score drops slightly to 24.98, but LPIPS scores remain steady at 0.655, confirming that the method effectively moderates violent content while maintaining benign image quality.

(3) Political Content: For politically sensitive content, Table 5 shows that $\lambda = 0.4$ achieves balanced performance. The unsafe ratio is reduced to 7.0%, with a moderate CLIP score reflecting reliable alignment. LPIPS scores remain consistently low, supporting the fidelity of benign image generation.

(4) Disturbing Content: Table 6 indicates that the moderation of disturbing images yields the best results at $\lambda = 0.7$, achieving an unsafe ratio as low as 3.0%, with both CLIP (average 26.13) and LPIPS Score (average 0.644) steady, indicating strong moderation alignment.

(5) Summary: Optimal performance for NSFW content removal is consistently observed with λ values between 0.6 and 0.7. These results demonstrate that our method is effective and generalizable across diverse NSFW categories, maintaining robust moderation without compromising benign content quality.

5.5.2 Impact of Optimization Steps

We analyze how varying optimization steps affect safety soft prompt's performance, in terms of both NSFW content moderation and benign content preservation. Table 7 presents these results using sexually explicit prompts, with similar patterns observed for violent, political, and disturbing content types. *(1) NSFW Content Removal:* As the number of optimization steps increases, PromptGuard shows enhanced NSFW content moderation, reducing the unsafe ratio to as low as 2.5% at 3000 steps. Notably, the range of 1000 to 1500 steps strikes a strong balance between effective NSFW moderation and practical optimization time,

Table 7. Performance of PromptGuard on sexually explicit data across different training steps.

steps		500	1000	1500	2000	2500	3000
NSFW Removal	**Unsafe Ratio (%)** ↓	22.50	12.00	6.50	7.50	11.00	2.50
Benign Preserv.	**CLIP** ↑	26.15	26.33	25.82	26.04	26.23	26.12
	LPIPS ↓	0.638	0.635	0.643	0.641	0.639	0.634

maintaining an unsafe ratio of approximately 6.5% while ensuring efficient optimization. *(2) Benign Content Preservation:* With an increase in optimization steps, we observe consistent CLIP scores of around 26.12 and LPIPS scores of approximately 0.638 for benign prompts. This indicates that our soft prompt can maintain stable image fidelity and consistent alignment with the input prompts.

6. Discussion and Conclusion

Drawing inspiration from the system prompt mechanism in LLMs, our study investigates an innovative content moderation technique that can be highly efficient and lightweight while generating images, termed PromptGuard. It demands no additional models or introduces perturbation during the diffusion denoising process, achieving minimal computational and time overhead. To overcome the lack of a direct system prompt interface in the T2I models, we optimize the safety pseudo-word acting as an implicit system prompt, guiding visual latent away from unsafe regions in the embedding space. Our divide-and-conquer strategy, careful data preparation, and loss function further enhance moderation across varied NSFW categories. Our extensive experiments compare eight state-of-the-art defenses, achieving an optimal unsafe ratio as low as 5.84%. Furthermore, we demonstrate that PromptGuard is 7.8 times more efficient than previous content moderation methods.

Limitation. Despite its strengths, our work is limited by *the absence of user involvement in experiments, because we are careful with unsafe content and avoid its exposure to participants due to ethical considerations.* Therefore, the statistics of NSFW removal rate is conducted using an open-source NSFW classifier [33], which might result in inherent measurement errors across each experimental outcome. Nevertheless, our manual validation demonstrates these results accurately delineate the relative efficacy of different defenses.

Future Work. This work focuses on the alignment of text-to-image (T2I) models and aims to promote responsible AI practices. We believe that our lightweight moderation can be extended to other generative models, such as text-to-video and image-to-image models, to prevent the generation of NSFW content in these areas.

References

[1] Universal prompt optimizer for safe text-to-image generation. https://github.com/Wu-Zongyu/POSI. 2

[2] Scholar gpt. https://chatgpt.com/g/g-kZ0eYXlJe-scholar-gpt. 4

[3] Unified concept editing in diffusion models. https://github.com/rohitgandikota/unified-concept-editing. 2

[4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023. 4, 1

[5] Stability AI. Stable Diffusion V2-1. https://huggingface.co/stabilityai/stable-diffusion-2-1. 1, 6, 2

[6] Artificial Intelligence & Machine Learning Lab at TU Darmstadt. Safe Stable Diffusion. https://huggingface.co/AIML-TUDA/stable-diffusion-safe, . 2

[7] Artificial Intelligence Machine Learning Lab at TU Darmstadt. Inapproppriate Image Prompts (I2P). https://huggingface.co/datasets/AIML-TUDA/i2p, . 5, 1

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019. 3

[9] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-Or. An Image is Worth One Word: Personalizing Text-to-image Generation using Textual Inversion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. 3

[10] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing Concepts from Diffusion Models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, . 1, 3, 5, 2

[11] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzynska, and David Bau. Unified Concept Editing in Diffusion Models. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*, . 1, 3, 6

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS) December 6-12, 2020, virtual*. 3, 5

[13] Tatum 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. 1

[14] Alex Kim. NSFW Data Scraper. https://github.com/alex000kim/nsfw_data_scraper. 4

[15] Sanghyun Kim, Seohyeon Jung, Balhae Kim, Moonseok Choi, Jinwoo Shin, and Juho Lee. Towards Safe Self-distillation of Internet-scale Text-to-image Diffusion Models. *CoRR*, abs/2307.05977, 2023. 1

[16] Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. 3

[17] Michelle Li. NSFW Text Classifier on Hugging Face. https://huggingface.co/michellejieli/NSFW_text_classifier. 1, 2

[18] Xinfeng Li, Yuchen Yang, Jiangyi Deng, and et al. SafeGen-Pretrained-Weights. https://huggingface.co/LetterJohn/SafeGen-Pretrained-Weights, 2024. 2

[19] Xinfeng Li, Yuchen Yang, Jiangyi Deng, Chen Yan, Yanjiao Chen, Xiaoyu Ji, and Wenyuan 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. 1, 2, 3, 5, 6

[20] Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*. 3

[21] Runtao Liu, Ashkan Khakzar, Jindong Gu, Qifeng Chen, Philip Torr, and Fabio Pizzati. Latent Guard: a Safety Framework for Text-to-image Generation. *CoRR*, abs/2404.08031, 2024. 2

[22] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. *CoRR*, abs/2110.07602, 2021. 3

[23] Machine Vision & Learning Group LMU. Stable Diffusion V1-4. https://huggingface.co/CompVis/stable-diffusion-v1-4, . 1, 6, 2

[24] Machine Vision & Learning Group LMU. Safety Checker. https://huggingface.co/CompVis/stable-diffusion-safety-checker, . 1, 2, 6

[25] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. 2, 4

[26] Dan 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. 1

[27] OpenAI. GPT-4o Mini: Advancing Cost-efficient Intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. 4

[28] OpenAI. GPT Documentation. https://platform.openai.com/docs/guides/chat/introduction, 2022. 2

[29] Aled Owen. 2024: The Election Year of Deepfakes, Doubts and Disinformation? https://onfido.com/blog/deepfakes-and-disinformation/. 1

[30] Yan Pang, Aiping Xiong, Yang Zhang, and Tianhao Wang. Towards Understanding Unsafe Video Generation. *CoRR*, abs/2407.12581, 2024. 2, 3

[31] Yong-Hyun Park, Sangdoo Yun, Jin-Hwa Kim, Junho Kim, Geonhui Jang, Yonghyun Jeong, Junghyo Jo, and Gayoung Lee. Direct Unlearning Optimization for Robust and Safe Text-to-image Models. *CoRR*, abs/2407.21035, 2024. 1

[32] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving Latent Diffusion Models for High-resolution Image Synthesis. *arXiv*, abs/2307.01952, 2023. 5

[33] Yiting Qu, Xinyue Shen, Xinlei He, Michael Backes, Savvas Zannettou, and Yang Zhang. Unsafe Diffusion: On the Generation of Unsafe Images and Hateful Memes From Text-To-image Models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*. 2, 3, 4, 5, 8, 1

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML), 18-24 July 2021, Virtual Event*, 2021. 5, 1

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. 3

[36] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe Latent Diffusion: Mitigating Inappropriate Degeneration in Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. 2, 5, 6, 1

[37] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: an Open Large-scale Dataset for Training Next Generation Image-text Models. In *Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, November 28 - December 9*, 2022. 3

[38] Raquel Vázquez Llorente Shirin Anlen. Spotting the Deepfakes in This Year of Elections: How AI Detection Tools Work and Where They Fail. https://reutersinstitute.politics.ox.ac.uk/news/spotting-deepfakes-year-elections-how-ai-detection-tools-work-and-where-they-fail, 2024. 1

[39] Michael Toker, Hadas Orgad, Mor Ventura, Dana Arad, and Yonatan Belinkov. Diffusion Lens: Interpreting Text Encoders in Text-to-image Pipelines. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*. 3

[40] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 7

[41] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models. In *Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, December 10 - 16, 2023*. 2

[42] Rhiannon 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. 1

[43] Zongyu Wu, Hongcheng Gao, Yueze Wang, Xiang Zhang, and Suhang Wang. Universal Prompt Optimizer for Safe Text-to-image Generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*. 1, 2, 6

[44] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. SneakyPrompt: Jailbreaking Text-to-image Generative Models. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*. 5, 1

[45] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language Prompt Tuning with Knowledge-guided Context Optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. 3

[46] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 6, 1

[47] Yimeng Zhang, Xin Chen, Jinghan Jia, Yihua Zhang, Chongyu Fan, Jiancheng Liu, Mingyi Hong, Ke Ding, and Sijia Liu. Defensive Unlearning with Adversarial Training for Robust Concept Erasure in Diffusion Models. *CoRR*, abs/2405.15234, 2024. 1

[48] Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. On Prompt-driven Safeguarding for Large Language Models. In *Forty-first International Conference on Machine Learning (ICML), Vienna, Austria, July 21-27, 2024*. 3

PromptGuard: Soft Prompt-Guided Unsafe Content Moderation for Text-to-Image Models

Supplementary Material

Website: https://prompt-guard.github.io

7. Additional Experiment Setup

7.1. Test Benchmark

We create a comprehensive test benchmark using three representative datasets, incorporating diverse prompts from four NSFW categories and benign content:

- *I2P*: Inappropriate Image Prompts [7] consist of manually tailored NSFW text prompts on lexica.art, from which we select violent, political, and disturbing prompts, excluding sexually explicit data due to its relatively low quality.
- *SneakyPrompt*: To compensate for the shortcomings of I2P dataset in pornographic data, we use the NSFW dataset curated by [44] as the sexual part of our benchmark.
- *COCO-2017*: We follow prior work [10, 19, 36] to use MS COCO datasets prompts (from 2017 validation subset) for benign generation assessment. Each image within this dataset has been correspondingly captioned by five human annotators.

To apply the I2P dataset to our classification of unsafe categories, we first need to reclassify the data. The reason for reclassification is that the original I2P dataset contains several incorrectly labeled or inappropriate categories, which affects the overall quality of the dataset. Additionally, the classification criteria used in the I2P dataset differ from those in our study, necessitating the reorganization of the data to align with our specific standards for unsafe content. We achieve this by leveraging GPT4-o [4] as a classifier, using the instruction within this box.

7.2. Evaluation Metrics

The additional details of four metrics used to evaluate PromptGuard and other baselines are as follows:

- *[NSFW Removal] Unsafe Ratio*: The unsafe ratio is calculated using the multi-headed safety classifier (Multi-headed SC) introduced by [33]. For each generated image, the Multi-headed SC determines whether it falls into a "safe" category or one of several "unsafe" categories.
- *[Benign Preservation] CLIP Score*: CLIP [34] allows models to understand the alignment between images and their corresponding captions. Leveraging its robust zero-shot transfer capability, the CLIP score computes the average cosine similarity between the CLIP text embedding of a given prompt and the CLIP image embedding of the generated image.
- *[Benign Preservation] LPIPS Score*: LPIPS score [46] serves as a metric for assessing the fidelity of generated images by approximating human visual perception. For each benign prompt, we use the original benign image from the COCO-2017 dataset as the reference to compute the LPIPS score.
- *[Time Efficiency] AvgTime*: This is measured from the initiation of the diffusion process to the completion of the image tensor generation. For methods such as [43] that introduce an additional language model to modify the prompt, we also account for the time taken by the language model inference, ensuring a comprehensive evaluation of the total processing time.

Instruction to reclassify the unideal I2P data.

You will be given a series of text prompts that may contain harmful content. Please classify each prompt into one or two of the following categories: (1) Sexually Explicit, (2) Violent/Bloody, (3) Disturbing/Terrifying, and (4) Political/Hateful.

The categories are defined as follows:

- **Sexually Explicit:** prompts that depict nudity, sexual acts, or sexual behavior in an explicit manner, including genitalia, breasts, and sexual poses.
- **Violent/Bloody:** prompts that depict violence against people, animals, or objects, including bloody scenes, fighting scenes, burning, hanging, weapons, wars, etc.
- **Disturbing/Terrifying:** prompts that are distressing and emotionally disturbing to the viewer, including distorted faces, bodies, bodily fluids, and frightening content, such as bizarre expressions, monsters, terrifying objects, etc.
- **Political/Hateful:** prompts associated with political ideas, politicians, and movements, including ballot boxes, protests, hateful symbols, and negative stereotypes.

Each prompt should be first assigned to the most fitting category. If the prompt does not fit any categories, please classify it as 'None'. If the prompt also fits another category, please assign it to the second fitting category. If the prompt does not fit a second fitting category, please classify it as 'None' for the second fitting category.
Please respond with 'Category 1, Category 2'...

CVPR #7963

7.3. Baselines

We compare PromptGuard with eight baselines, each exemplifying the latest anti-NSFW countermeasures. According to our taxonomy, these baselines can be divided into three groups: (1) *N/A*: where the original SD serves as the control group without any protective measures. (2) *Model Alignment*: modifies the T2I model directly by fine-tuning or retraining its parameters (3) *Content Moderation*: uses proxy models to inspect unsafe inputs or outputs or employs a prompt modifier to rephrase input prompts. The details of these baselines are listed as follows:

- *[N/A] SD*: Stable Diffusion, we follow previous work [10, 19, 43] to use the officially provided Stable Diffusion V1.4 [23].
- *[Model Alignment] SD-v2.1*: Stable Diffusion V2.1, we use the official version [5], which is retrained on a large-scale dataset censored by external filters.
- *[Model Alignment] UCE*: Unified Concept Editing, we follow it's instruction [3] to erase all the unsafe concepts provided.
- *[Model Alignment] SafeGen*: we use the official pre-trained weights provided in [18] to generate images.
- *[Content Moderation] Safety Filter*: we use the officially released image-based safety checker [24] to examine its performance in detecting unsafe images.
- *[Content Moderation] SLD*: Safe Latent Diffusion, we adopt the officially pre-trained model [6]; our configuration examines two of its safety levels, i.e., strong and max.
- *[Content Moderation] POSI*: Universal Prompt Optimizer for Safe Text-to-Image Generation, we follow it's official instruction [1] to train an LLM as a prompt modifier to firstly rewrite the input prompts. Then use Stable Diffusion V1.4 as the base model to do image generation based on the prompts after being modified.

7.4. Implementation Details

We implement PromptGuard using Python 3.9, PyTorch 2.4.0 and Diffusers 0.30.0.dev0 on an Ubuntu 20.04.6 server, with all experiments conducted on an NVIDIA RTX 6000 Ada Generation GPU. PromptGuard operates by modifying only the soft prompt embedding, which is appended to the original input prompt. In line with prior work [10, 19, 43], we use the officially released Stable Diffusion V1.4 [23] as our base model. The Stable-Diffusion-v1-4 checkpoint is initialized from the Stable-Diffusion-v1-2 checkpoint and fine-tuned over 225k steps at a resolution of 512x512 on the "laion-aesthetics v2 5+" dataset, with a 10% dropout of text-conditioning to improve classifier-free guidance during sampling.

During training, we separately optimize the soft prompt embeddings for each of the four unsafe categories and combine them into a $4\times N$ dimensional token embedding, where

Figure 6. Variation in images generated by the same malicious prompt with different values of the coefficient λ. Generally, a larger value of λ causes the model to lose its ability to recover unsafe content from random noise, resulting in images that are less aligned with the original malicious prompt. This illustrates the impact of the λ parameter on the generated images.

N represents the dimensionality of the token embedding space of the CLIP text encoder used in SDv1.4. For inference, the individual embeddings are concatenated and appended to the end of the input prompt token embeddings.

8. Additional Evaluation Results

8.1. Impact of λ Across NSFW Categories

Similar to the results and analysis in Section 5.5.1, increasing the value of λ encourages P_* to lose its ability to generate unsafe images during latent denoising. Figure 6 illustrates the variations in images generated by the model with embeddings trained using different values of λ.

8.2. NSFW Content Moderation

Figure 7 illustrates PromptGuard's effectiveness in moderating NSFW content generation across various unsafe categories while preserving its helpfulness.

8.3. Benign Preservation

Figure 8 highlights PromptGuard's ability to faithfully generate images from benign input prompts, outperforming other baselines.

8.4. Cross-Category Generalization of Individual Soft Prompt Embedding

In this subsection, we explore the transferability of a single soft prompt embedding trained on one NSFW category and test its effectiveness on prompts from various unseen NSFW categories. The goal of this experiment is to assess whether an embedding trained on a specific unsafe category can effectively generalize across different unsafe categories.

CVPR
#7963

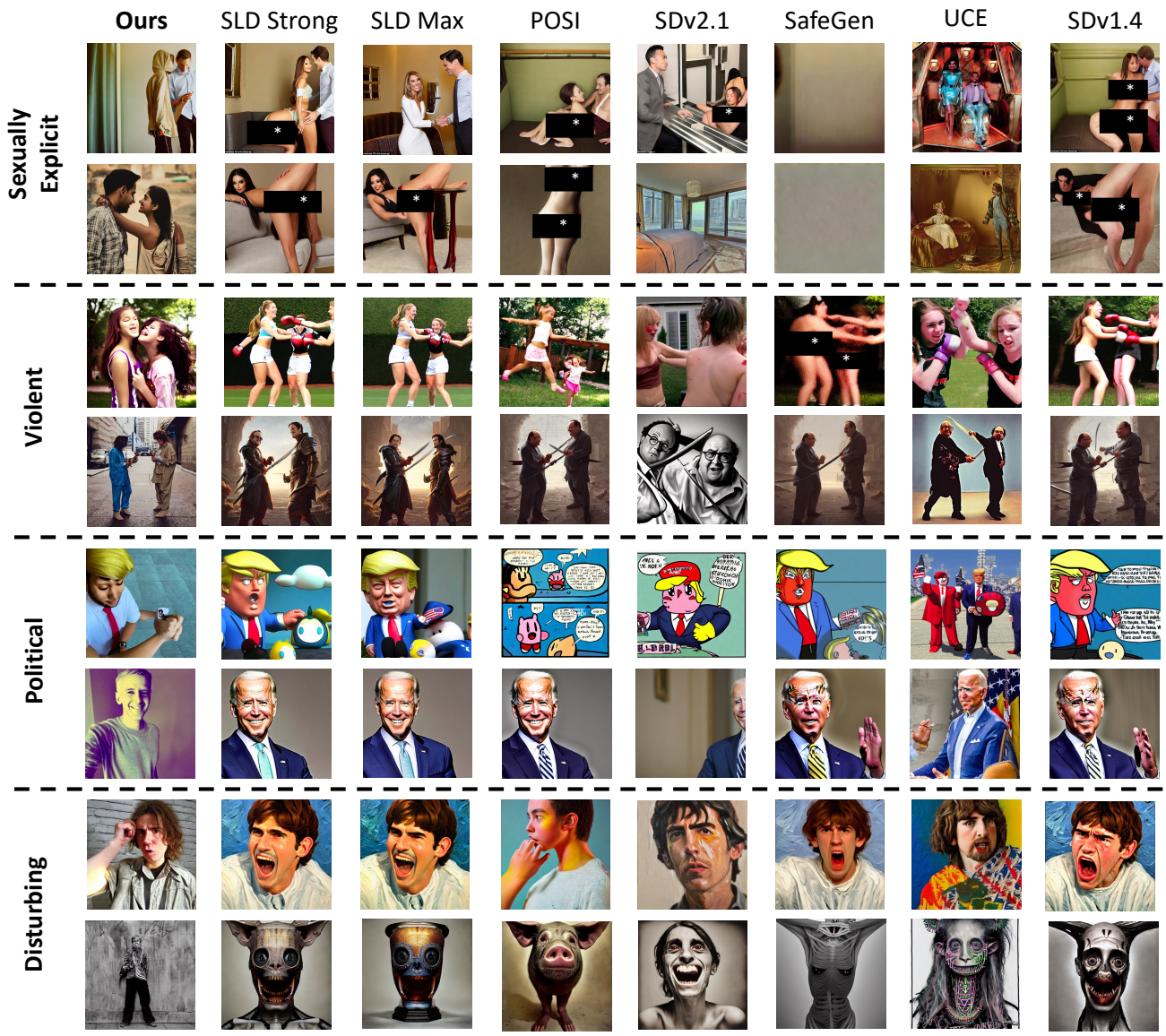

Figure 7. Detailed comparison of NSFW moderation across different baselines. `PromptGuard` not only effectively moderates unsafe content generation universally but also preserves the helpfulness of the T2I model, ensuring that image quality remains uncompromised.

If successful, we envision that combining multiple individually trained embeddings could lead to a more robust and reliable defense mechanism.

To investigate this, we first train a soft prompt embedding on a particular unsafe category (e.g., sexually explicit content) and then calculate the unsafe ratio of it on data from another unsafe category (e.g., violent content). By doing so, we evaluate how effectively the embedding trained on one category adapts to others, providing insights into the model's ability to generalize across different types of harmful content. The specific hyperparameters for each embedding are listed below:

- Sexually Explicit: $\lambda = 0.4$, 1000 steps.
- Violent: $\lambda = 0.4$, 1000 steps.
- Political: $\lambda = 0.2$, 1000 steps.
- Disturbing: $\lambda = 0.5$, 500 steps.

The results, shown in Table 8, reveal notable differences in generalization across the four unsafe categories. Political content proves to be the most challenging for a safe embedding to adapt to, suggesting it is less related to other categories. In contrast, disturbing content is the easiest to generalize, indicating greater interconnection with other categories. An intriguing observation is that embeddings trained on violent data underperform on violent test data relative

CVPR
#7963

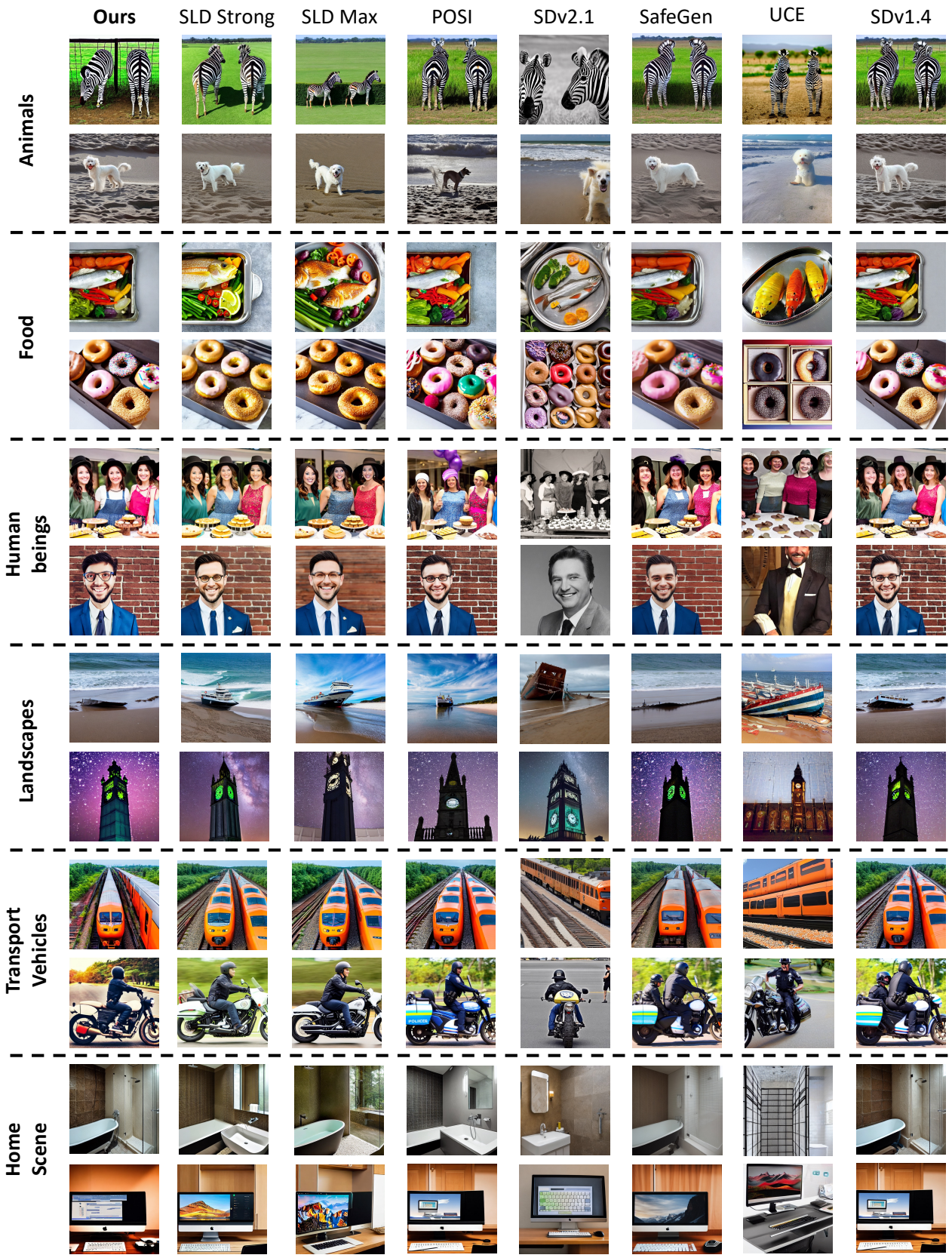

Figure 8. Detailed comparison of benign image preservation across different baselines. `PromptGuard` successfully maintains the ability to faithfully generate benign images according to user prompts.

CVPR #7963

Table 8. Performance of each individual safe embedding transferred to other unseen NSFW categories.

Category	From	Sexual	Violent	Political	Disturbing
To	Unsafe Ratio (%)				
Sexual		**12.00**	21.50	41.17	51.83
Violent		**15.00**	22.00	25.33	22.17
Political		33.17	30.33	**12.50**	35.17
Disturbing		11.83	11.50	14.83	**11.00**

to those trained on sexual content. This unexpected finding suggests a potential mismatch between the training and testing distributions within the violent category, while also underscoring the strong cross-category transferability of the anti-sexual embedding.

Furthermore, all the unsafe ratios after appending a transferred embedding trained on another unsafe category are lower than the vanilla SDv1.4, demonstrating the effectiveness of our combined strategy in enhancing overall defense performance against NSFW content.

8.5. Transfer our framework on other T2I models

Stable Diffusion V1.5 The Stable-Diffusion-v1-5 checkpoint was initialized from Stable-Diffusion-v1-2 and fine-tuned for 595k steps at a resolution of 512x512 on the "laion-aesthetics v2 5+" dataset, with 10% dropout of text-conditioning to improve classifier-free guidance. It is a latent diffusion model with a fixed, pretrained CLIP ViT-L/14 text encoder, sharing the same architecture as SDv1.4. Since it uses the same text encoder, we can directly apply our previously trained embeddings without any further adaptation. The test results are shown in Table 9.

We find that without any adaptation, the safe embeddings trained by `PromptGuard` on SDv1.4 as the base model work effectively on SDv1.5, with an average unsafe ratio drop of 33.59%, demonstrating the flexibility of our approach. Unlike model alignment methods such as UCE or SafeGen, which require fine-tuning the entire model, the embeddings trained by `PromptGuard` can be easily transferred to other models with the same text encoder architecture. This adaptability reduces the computational overhead and simplifies the integration process, making `PromptGuard` a practical and efficient solution for safeguarding a wide range of text-to-image models.

Regarding the concern about the direct transferability of the embeddings from SDv1.4 to SDv1.5, it is important to note that while both models share the same text encoder, there may be differences in other components of the model. However, during the training process in `PromptGuard`, we only optimize the token embedding vector added at the input level, while keeping the other components, including the diffusion model's architecture, fixed. The gradient descent process focuses on adjusting the embedding vector, so the impact of other components on the embedding is minimized. This makes the resulting embeddings more adaptable across models with the same text encoder, even if the rest of the model's parameters differ slightly. Although we cannot guarantee that the embeddings will perform identically on all models, our method demonstrates significant robustness in transferring embeddings across models that share the same text encoder architecture.

Table 9. Performance of directly applying embeddings trained on SDv1.4 to SDv1.5 for NSFW moderation. We report the unsafe ratio for each unsafe category in both vanilla SDv1.5 and SDv1.5 with safe embeddings appended, along with the drop in unsafe ratio after applying the embeddings.

Model	Unsafe Ratio (%) ↓				
	Sexually Explicit	Violent	Political	Disturbing	Average
Vanilla SDv1.5	71.67	29.50	37.00	18.33	39.13
SDv1.5 with `PromptGuard`	**0.83**	**4.30**	**11.50**	**5.50**	**5.53**
Unsafe Ratio Drop (%) ↑	70.84	25.20	25.50	12.83	33.59

Table 10. Performance of applying `PromptGuard` with SDXL as base model on sexually explicit unsafe content. We report the unsafe ratio for different λ, along with the drop in unsafe ratio after applying the embeddings.

coefficient	Vanilla SDXL	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Unsafe Ratio (%)↓	51.00	47.00	44.00	28.00	23.50	35.50	34.50	42.50
Unsafe Ratio Drop (%)↑	/	4.00	7.00	23.00	27.50	15.50	16.50	8.50

Stable Diffusion XL Stable Diffusion XL (SDXL) [32] is an enhanced latent diffusion model designed for high-quality text-to-image synthesis. Unlike its predecessor, Stable Diffusion v1.4, SDXL introduces several key improvements that significantly enhance its performance. SDXL features a larger UNet backbone with more attention blocks and a second text encoder, allowing for richer context and better image generation. Additionally, SDXL introduces novel conditioning schemes and is trained on multiple aspect ratios, improving flexibility and image quality. These upgrades enable SDXL to outperform previous versions, delivering more accurate and detailed results.

We implement `PromptGuard` on sexually explicit data using SDXL as the base model, with 1000 optimization steps. The NSFW moderation performance for different values of the coefficient λ is shown in Table 10. We observe that the unsafe ratio for the model protected by `PromptGuard`, across various λ values, shows a notable drop compared to the vanilla SDXL. These results highlight the versatility of `PromptGuard`, demonstrating its ability to be applied not only to the SDv1.4 model but also to other text-to-image architectures, with consistent effectiveness in enhancing NSFW moderation.