Decoupled Safety Control: A Safety-Control Algorithm for Training-Free Safety Guidance

Huilin Zhou[1]* **Ruoxi Cheng**[2,3]* **Yuhang Wang**[2] **Yuming Liu**[1]
Sun Minghao[4] **Ruolong Ma**[5] **Lan Zhang**[1,6]†
[1]University of Science and Technology of China
[2]Fudan University [3]Tencent
[4]Beijing University of Posts and Telecommunications
[5]China Academy of Information and Communications Technology
[6]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

Abstract

Diffusion models achieve strong performance in text-to-image generation, but they can still produce unsafe content. Training-free safety guidance offers a practical inference-time alternative that intervenes during sampling without modifying model weights. However, many existing safeguards still rely on a coupled safety-control chain. In this chain, the same unsafe condition simultaneously specifies what should be avoided, indicates the current unsafe signal, constructs the intervention, and determines when that intervention remains active. This coupling can work when unsafe factors are explicit, but becomes less reliable once harmful and benign semantics are entangled in the current generation state, often leaving residual unsafe content or unnecessarily degrading benign-generation quality or prompt fidelity. We address this problem with Decoupled Safety Control (DSC), a plug-in safety-control algorithm for training-free safety guidance. DSC reformulates training-free safety guidance as an explicit state-dependent control process, so the executed intervention is determined from the current generation state rather than inherited directly from a single unsafe condition. Experiments on standard safety benchmarks show that DSC improves representative latent- and text-space safeguards, achieving stronger unsafe-content suppression while preserving benign-generation quality and maintaining efficient inference.

Warning: This paper contains disturbing content, including censored examples of nudity and sexually explicit text prompts, included only for research and evaluation purposes.

1 Introduction

Diffusion models achieve strong performance in text-to-image generation [1–4], but they also raise serious safety concerns. Open-source text-to-image models can still produce unsafe content and remain vulnerable to adversarial prompts [5–11]. Existing safety methods broadly fall into two groups. One line improves safety by fine-tuning or editing model weights [12–18]. Although effective in many settings, these methods require additional optimization or direct parameter modification and can degrade benign generation quality. A complementary line instead performs training-free safety guidance during inference [19–27]. These methods preserve the base model parameters and can be used as plug-and-play safeguards.

Among training-free safeguards, recent methods intervene at different sites in the denoising process. Early guidance-based methods such as Safe Latent Diffusion (SLD) [19] modify classifier-free

*These authors contributed equally to this work.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



Figure 1: Qualitative comparison with existing methods on representative prompts from Ring-A-Bell and UnlearnDiff. When instantiated with SLD [19] or SAFREE [20], our method avoids generating inappropriate images.

guidance with safety concepts or negative conditions [28]. Prompt-side methods such as Training-Free and Adaptive Guard for Safe Text-to-Image and Video Generation (SAFREE) [20] filter, project, or sanitize text representations against unsafe subspaces or directions [24, 25]. More recent trajectory-level methods, including Safe Denoiser [21] and Safety-Guided Flow (SGF) [23], alter the sampling trajectory using unsafe references, repulsive fields, or negative datasets [22]. Despite these differences, many safeguards still follow a similar safety-control chain organized around the same unsafe condition. In this safety-control chain, the same unsafe condition is often used to specify what should be avoided, indicate the current unsafe signal, construct the intervention, and determine when that intervention remains active.

This coupled safety-control chain can be effective when unsafe factors are explicit and well aligned with the chosen unsafe condition. The limitation appears when harmful and benign semantics become entangled in the current generation state. In that case, the same unsafe condition is no longer sufficient to determine what should be preserved, which intervention is appropriate, or how long that intervention should remain active.

Once harmful and benign semantics are entangled, the intervention becomes less reliable: it may suppress unsafe content too weakly and leave residual unsafe content, or suppress too broadly and degrade benign generation quality or prompt fidelity. This failure pattern is consistent with prior observations of quality degradation under hard filtering [20], degraded performance on desired tasks [21], quality loss under prolonged negative guidance [23], and mode shift on complex prompts [29]. Taken together, these observations point to a common gap in current training-free safety guidance: the control process is not explicit enough to support a precise and reliable intervention.

We address this gap with Decoupled Safety Control (DSC), a safety-control algorithm for training-free safety guidance. DSC makes the control process explicit through runtime evidence extraction, control-state estimation, intervention routing, update construction, and phase-aware execution. Intervention decisions depend on the current generation state rather than being inherited directly from a single unsafe condition. We instantiate DSC on representative latent- and text-space safeguards, including SLD and SAFREE, while keeping the base model unchanged. Experiments on standard safety benchmarks show that DSC achieves stronger unsafe-content suppression while preserving benign-generation quality, and remains efficient under the evaluated backbones.

Our main contributions are as follows:

- We identify a mechanism-level limitation in current training-free safety guidance: the same unsafe condition is often used to specify what should be avoided, indicate the current unsafe signal, construct the correction, and determine when the correction remains active.
- We propose Decoupled Safety Control (DSC), a plug-in safety-control algorithm for training-free safety guidance that makes runtime evidence extraction, control-state estimation, intervention routing, update construction, and phase-aware execution explicit.
- Experiments on standard safety benchmarks show that DSC improves representative latent- and text-space safeguards, achieving stronger unsafe-content suppression while preserving benign-generation quality and maintaining efficient inference under the evaluated backbones.

2 Related Work

Training-based safety methods. Training-based safety methods improve safe generation by modifying model weights through concept erasure, model editing, or additional optimization. Representative parameter-level approaches include ESD [12], RECE [13], UCE [14], R.A.C.E. [15], MACE [16], machine unlearning for image-to-image generative models [17], STEREO [18], and Selective Amnesia [30]. They can remove targeted unsafe concepts, but require retraining or direct parameter modification and may degrade benign generation quality. Our work instead focuses on inference-time safety control while keeping the base model unchanged.

Training-free latent- and prompt-space safeguards. Training-free safeguards instead intervene during inference without modifying model weights, either in latent space, as in Safe Latent Diffusion (SLD) [19], or in text/prompt space, as in Training-Free and Adaptive Guard for Safe Text-to-Image and Video Generation (SAFREE) [20]. SLD modifies classifier-free guidance with predefined safety prompts, while SAFREE filters unsafe prompt representations and adaptively reweights attention during sampling. Related variants include latent guarding, embedding guidance, prompt purification, and prompt-side steering [31, 24, 25, 32, 26, 27]. We focus on SLD and SAFREE as representative latent- and text-space safeguards whose interventions remain derived from pre-specified unsafe concepts, subspaces, or references.

Trajectory-level safety guidance. Trajectory-level safeguards directly modify the sampling path through unsafe references, repulsive fields, or time-varying negative guidance, as in Safe Denoiser [21], Shielded Diffusion [22], and Safety-Guided Flow (SGF) [23]. Related studies further analyze how negative prompts or negative guidance influence denoising behavior [33, 34, 29]. These methods address training-free safety by changing the denoising trajectory. Our work instead focuses on how the safety-control chain is organized once training-free intervention is introduced.

3 Method

We present Decoupled Safety Control (DSC), a plug-in control algorithm for training-free safety guidance. DSC makes the control process explicit at each denoising step. We first formulate the problem, then present Decoupled Safety Control, and finally describe its application to SLD and SAFREE.

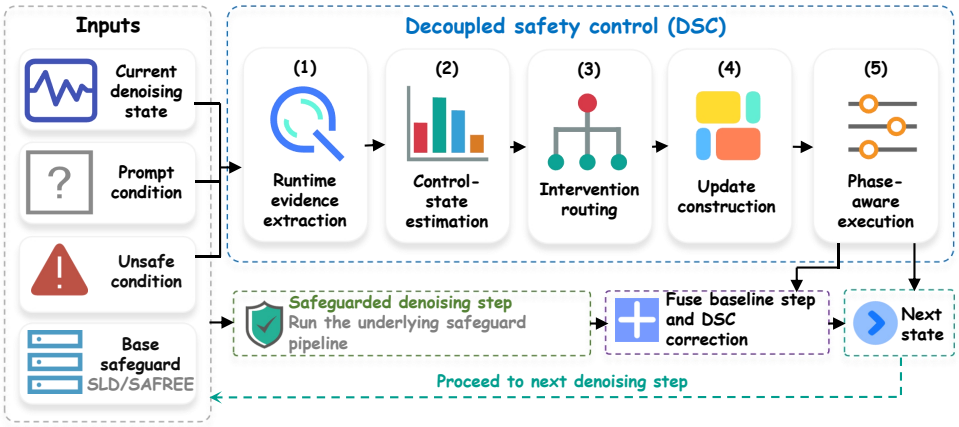

Figure 2: Overview of DSC at each denoising step. DSC organizes runtime evidence extraction, control-state estimation, intervention routing, update construction, and phase-aware execution around the baseline safeguard step.

3.1 Problem Formulation

Consider a pretrained text-to-image diffusion model with a training-free safeguard pipeline $\mathcal{P}$. Let $x_T \sim \mathcal{N}(0, I)$ be the initial noise, and let x_t be the denoising state at step t under prompt condition

p and safeguard condition c^-, where c^- denotes the unsafe concept, unsafe subspace, or unsafe reference set used by the safeguard. We write one safeguarded denoising step as

$$x_{t-1}^{\mathcal{P}} = H_t(x_t, p, c^-; \mathcal{P}), \tag{1}$$

where H_t denotes the update produced by the safeguard at step t.

The objective is to generate a final sample that suppresses unsafe content associated with c^- while preserving the benign semantics in p that are unrelated to the unsafe factor. This objective becomes challenging when harmful and benign semantics are entangled in the current trajectory, since the safeguard must decide not only whether to intervene, but also what should be suppressed, what should be preserved, and how strongly the intervention should act at the current step.

Accordingly, we formulate training-free safe generation as a state-dependent control problem. In this formulation, unsafe specification remains fixed through the safeguard condition, while runtime evidence, control state, intervention route, update construction, and execution timing should be resolved from the current generation state. At each denoising step, the method should therefore infer the current control demand from the generation state and produce an intervention that balances unsafe suppression and benign preservation.

3.2 Decoupled Safety Control

Decoupled Safety Control (DSC) reformulates each safeguarded denoising step as an explicit state-dependent control process. At step t, DSC extracts runtime evidence from the current generation state, estimates route-wise control states, selects an intervention route, constructs a route-conditioned update, and executes that update in a phase-aware manner. The executed intervention is therefore determined from the current generation state rather than inherited directly from a single unsafe condition.

Runtime evidence extraction. The safeguard condition specifies what should be avoided, but it need not by itself provide the most informative runtime evidence for the current denoising step. DSC therefore begins by extracting runtime evidence

$$e_t = \Phi_t(x_t, p, c^-; \mathcal{P}), \tag{2}$$

from the current generation state. The evidence e_t is the control-relevant observation used for subsequent control decisions. Depending on the safeguard, it may be formed from denoising-branch discrepancies, prompt-side filtering cues, predicted clean-sample relations, or support patterns exposed at the current step.

Control-state estimation. Runtime evidence alone does not determine how the current trajectory should be corrected. Similar evidence may correspond to different preservation demands or different degrees of harmful–benign entanglement. DSC therefore estimates route-wise control states

$$s_t(r) = \Psi_t(e_t, r), \qquad r \in \mathcal{R}, \tag{3}$$

from the extracted evidence. We write each route-wise state as

$$s_t(r) = \big(\pi_t(r), \kappa_t(r), \eta_t(r), \omega_t(r)\big), \tag{4}$$

where $\pi_t(r)$ denotes unsafe pressure, $\kappa_t(r)$ denotes preservation demand, $\eta_t(r)$ denotes harmful–benign entanglement, and $\omega_t(r)$ denotes support scope for route r. Let g_t^r be the suppressive direction induced by route r, h_t^r be its preservation direction, d_t^- be the unsafe-side evidence, and d_t^+ be the benign-preservation evidence extracted from e_t. For nonzero vectors a and b in the safeguard space, define

$$\mathrm{sim}(a,b) = \frac{1}{2}\left(1 + \frac{\langle a, b\rangle}{\|a\|_2\|b\|_2 + \epsilon}\right), \qquad \mathrm{leak}(a,b) = \left[\frac{\langle a, b\rangle}{\|a\|_2\|b\|_2 + \epsilon}\right]_+, \tag{5}$$

where $[\cdot]_+ = \max(\cdot, 0)$. Here $\epsilon > 0$ is a small numerical stabilizer. DSC computes

$$\pi_t(r) = \mathrm{sim}(g_t^r, d_t^-), \qquad \kappa_t(r) = \mathrm{sim}(h_t^r, d_t^+), \qquad \eta_t(r) = \mathrm{leak}(g_t^r, h_t^r). \tag{6}$$

Thus, $\pi_t(r)$ measures how strongly route r is supported by unsafe evidence, $\kappa_t(r)$ measures its compatibility with benign-preservation evidence, and $\eta_t(r)$ measures positive overlap between

suppressive and preservation directions. When a route exposes a spatial or token-level support map $q_t^r \in [0,1]^m$, we define

$$\omega_t(r) = \frac{1}{m}\sum_{i=1}^{m} q_t^r(i); \tag{7}$$

otherwise $\omega_t(r)$ is set to one. These quantities define the current control demand for intervention routing, update construction, and phase-aware execution.

Intervention routing. Once the route-wise control states are identified, the method still must decide which intervention mode is appropriate for the current step. DSC therefore selects an intervention route from

$$\mathcal{R} = \{r^{\mathrm{sup}}, r^{\mathrm{bal}}, r^{\mathrm{rel}}\}, \tag{8}$$

where r^{sup} denotes suppressive control, r^{bal} denotes balanced, preservation-aware control, and r^{rel} denotes release toward the baseline dynamics. Suppressive control applies a direct unsafe correction when the current evidence supports it. Preservation-aware control constrains the correction when unsafe evidence overlaps with benign prompt semantics. Release returns the step toward the baseline safeguard dynamics when additional correction is not reliable.

DSC first forms the admissible route set

$$\mathcal{A}_t = \left\{ r \in \mathcal{R} \setminus \{r^{\mathrm{rel}}\} : \kappa_t(r) \geq \tau_\kappa,\ \eta_t(r) \leq \tau_\eta,\ \omega_t(r) \geq \tau_\omega \right\}, \tag{9}$$

where τ_κ, τ_η, and τ_ω are admissibility thresholds. This set keeps routes that are compatible with benign preservation, have limited suppressive-preservation leakage, and are reliable on the current support scope. Among the admissible routes, DSC selects the one with the strongest unsafe pressure:

$$r_t = \begin{cases} \arg\max_{r \in \mathcal{A}_t} \pi_t(r), & \mathcal{A}_t \neq \emptyset, \\ r^{\mathrm{rel}}, & \mathcal{A}_t = \emptyset. \end{cases} \tag{10}$$

This routing step separates the intervention mode from the unsafe specification itself: the same unsafe condition can lead to suppressive, preservation-aware, or release behavior depending on the current runtime evidence.

Update construction. The selected route specifies the admissible correction mode at the current step, but that mode still must be instantiated in the safeguard space. DSC therefore constructs a route-conditioned update

$$\tilde{u}_t = U_t(x_t, e_t, s_t(r_t), r_t; \mathcal{P}), \tag{11}$$

for the current safeguard. This stage realizes the selected intervention mode as the concrete safeguard-specific correction used at step t. Different routes can therefore lead to different corrections under the same unsafe specification. The correction is stronger when the selected route has high unsafe pressure and milder when the selected route has high preservation demand or stronger suppressive-preservation leakage.

Phase-aware execution. A suitable correction also requires an explicit decision about execution strength and release timing. DSC therefore computes two execution coefficients,

$$\alpha_t = A_t(a_{b,t}, \tilde{u}_t), \qquad \rho_t = R_t(s_t(r_t), t), \tag{12}$$

where $a_{b,t}$ is the baseline safeguard correction, α_t controls intervention strength, and ρ_t controls release toward the baseline dynamics. The executed correction is

$$u_t = (1 - \rho_t)\, \alpha_t \tilde{u}_t, \tag{13}$$

and the final denoising step becomes

$$x_{t-1} = \Omega_t\big(H_t(x_t, p, c^-; \mathcal{P}), u_t\big)\,. \tag{14}$$

A larger α_t yields a stronger correction, whereas a larger ρ_t suppresses the added correction and returns the step toward the baseline dynamics. In this way, intervention routing, update construction, and execution timing are resolved explicitly from the current generation state.

Algorithm 1 summarizes the DSC procedure.

Algorithm 1 Decoupled Safety Control

Input: current state x_t, prompt condition p, safeguard condition c^-, safeguard pipeline $\mathcal{P}$
Compute safeguarded baseline step $x_{t-1}^{\mathcal{P}} = H_t(x_t, p, c^-; \mathcal{P})$
Extract runtime evidence $e_t = \Phi_t(x_t, p, c^-; \mathcal{P})$
Estimate route-wise control states $s_t(r) = \Psi_t(e_t, r)$ for $r \in \mathcal{R}$
Select intervention route r_t using Eq. (10)
Construct route-conditioned update $\tilde{u}_t = U_t(x_t, e_t, s_t(r_t), r_t; \mathcal{P})$
Compute execution coefficients $\alpha_t = A_t(a_{b,t}, \tilde{u}_t)$ and $\rho_t = R_t(s_t(r_t), t)$
Execute correction $u_t = (1 - \rho_t)\, \alpha_t \tilde{u}_t$
Return $x_{t-1} = \Omega_t(x_{t-1}^{\mathcal{P}}, u_t)$

3.3 Applied to SLD and SAFREE

The DSC formulation separates the control stages from the underlying safeguard implementation. Applying it to a specific training-free safeguard only requires specifying the runtime evidence interface and the safeguard-specific realization of the correction, while the control-stage decomposition remains unchanged.

Applied to SLD. When applied to SLD [19], DSC uses the prompt, unconditional, and safety-concept denoising branches as its runtime evidence source. We denote the resulting SLD evidence by

$$e_t^{\mathrm{SLD}} = \big(\Delta_t^{\mathrm{txt}}, \Delta_t^{\mathrm{safe}}, m_t, q_t\big), \tag{15}$$

where

$$\Delta_t^{\mathrm{txt}} = d\big(\epsilon_t^{\mathrm{txt}}, \epsilon_t^{\mathrm{uncond}}\big), \qquad \Delta_t^{\mathrm{safe}} = d\big(\epsilon_t^{\mathrm{txt}}, \epsilon_t^{\mathrm{safe}}\big), \qquad m_t = \cos\big(\epsilon_t^{\mathrm{txt}} - \epsilon_t^{\mathrm{uncond}},\, \epsilon_t^{\mathrm{txt}} - \epsilon_t^{\mathrm{safe}}\big), \tag{16}$$

summarize the prompt-related and safety-related branch discrepancies together with their relative alignment at step t, and q_t summarizes the current support pattern; here $d(a, b) = a - b$ denotes the branch discrepancy in the denoising space. In this latent-space application, d_t^- is obtained from the safety-concept branch discrepancy, while d_t^+ is obtained from the prompt-preserving denoising direction. For each route, the corresponding candidate latent correction defines the suppressive direction g_t^r, and the prompt-preserving denoising direction defines h_t^r. The route-wise quantities in Eqs. (6)–(7) are then computed in the latent denoising space.

Once the route is selected, DSC realizes the corresponding correction as an SLD-specific latent update,

$$\tilde{u}_t^{\mathrm{SLD}} = U_t^{\mathrm{SLD}}\big(x_t, e_t^{\mathrm{SLD}}, s_t(r_t), r_t\big). \tag{17}$$

The update is then executed through the same phase-aware mechanism in Eq. (13). Under this instantiation, DSC does not replace the SLD safeguard condition. Instead, it makes explicit how current latent evidence determines the intervention mode, its safeguard-specific realization, and its release timing.

Applied to SAFREE. When applied to SAFREE [20], DSC uses prompt-side subspace and filtering cues together with their induced denoising discrepancy as its runtime evidence source. We denote the resulting SAFREE evidence by

$$e_t^{\mathrm{SAFREE}} = \big(z_t^{\mathrm{sub}}, z_t^{\mathrm{flt}}, \Delta_t^{\mathrm{den}}, q_t\big), \tag{18}$$

where z_t^{sub} summarizes the prompt-to-subspace proximity, z_t^{flt} summarizes the filtered prompt-side representation, $\Delta_t^{\mathrm{den}} = D_t\big(x_t, z_t^{\mathrm{sub}}, z_t^{\mathrm{flt}}\big)$ summarizes the denoising discrepancy induced at the current step, and q_t summarizes the current support pattern. In this prompt-space application, d_t^- is obtained from the denoising discrepancy induced by the unsafe prompt-side cue, while d_t^+ is obtained from the preservation direction induced by the filtered benign prompt representation. For each route, the corresponding SAFREE-specific correction defines g_t^r, and the filtered benign prompt representation induces the preservation direction h_t^r. The route-wise quantities in Eqs. (6)–(7) are then computed in the SAFREE denoising space.

DSC realizes the selected route as the corresponding SAFREE-specific correction

$$\tilde{u}_t^{\mathrm{SAFREE}} = U_t^{\mathrm{SAFREE}}\big(x_t, e_t^{\mathrm{SAFREE}}, s_t(r_t), r_t\big). \tag{19}$$

Table 1: Performance comparison on adversarial nudity benchmarks. Our method improves safety while preserving image quality.

Method	Weight Update	Training -Free	Decoupled Control	Ring-A-Bell		UnlearnDiff		MMA-Diffusion		COCO	
				ASR↓	TR↓	ASR↓	TR↓	ASR↓	TR↓	FID↓	CLIP↑
SD-v1.4	-	-	-	0.797	0.809	0.809	0.845	0.962	0.956	25.04	**31.38**
ESD	✓	✗	✗	0.456	0.506	0.422	0.426	0.628	0.640	27.38	30.59
RECE	✓	✗	✗	0.177	0.212	0.284	0.292	0.651	0.664	33.94	30.29
SLD	✗	✓	✗	0.481	0.573	0.629	0.586	0.881	0.882	36.47	29.28
SLD + SGF	✗	✓	✗	0.228	0.294	0.353	0.431	0.297	0.357	36.83	28.13
SLD + Ours	✗	✓	✓	0.089	0.095	**0.060**	**0.088**	0.169	**0.201**	32.68	28.65
SAFREE	✗	✓	✗	0.278	0.311	0.353	0.363	0.601	0.618	25.29	30.98
SAFREE + SGF	✗	✓	✗	0.051	0.133	0.164	0.232	0.423	0.461	23.73	30.36
SAFREE + Ours	✗	✓	✓	**0.025**	**0.078**	0.137	0.180	**0.167**	0.225	**20.95**	30.36

The update is then executed through the same phase-aware mechanism defined in Eqs. (12)–(14). As in the SLD case, DSC does not alter the underlying safeguard pipeline or model weights. Instead, it makes explicit how the current generation state determines the intervention mode, the resulting prompt-side correction, and its release timing.

For SLD, the concrete evidence tensors are $d_t^- = \epsilon_t^{\text{txt}} - \epsilon_t^{\text{safe}}$ and $d_t^+ = \epsilon_t^{\text{txt}} - \epsilon_t^{\text{uncond}}$. For SAFREE, they are $d_t^- = \epsilon_t^{\text{txt}} - \epsilon_t^{\text{filt}}$ and $d_t^+ = \epsilon_t^{\text{filt}} - \epsilon_t^{\text{uncond}}$, where ϵ_t^{filt} is the denoising prediction conditioned on the filtered text. Appendix D specifies the preservation reference, route-conditioned tensor updates, execution coefficients, and fixed parameters.

4 Experiments

In this section, we study safety in text-to-image models on tasks involving adversarial nudity prompts and multi-category inappropriate content. The nudity task evaluates how well safety methods prevent harmful outputs as attack difficulty increases. The multi-category task assesses safety across several unsafe categories under the same control algorithm. We use Stable Diffusion (SD) v1.4 [4] with the DDPM sampler [1]. To evaluate safety, we follow previous studies and report Attack Success Rate (ASR), Toxic Rate (TR), and Inappropriate Probability (IP) [19, 20, 35]. ASR and TR are used for the nudity benchmarks, and IP is used for the multi-category evaluation. Besides safety-related metrics, we also report Fréchet Inception Distance (FID) [36] and CLIP score [37] to evaluate image quality and prompt alignment. For Table 1, we follow the released SGF benchmark code and use the COCO-30K prompt set with FID and CLIP computed on the first 10K prompt-image pairs. Unless otherwise noted, all methods use identical benchmark assets and evaluation settings for consistency; Appendix B summarizes the prompt sources, evaluator settings, and implementation details. Appendix E reports confidence intervals and additional automatic and human evaluations.

4.1 Safe Generation against Nudity Prompts

We follow the evaluation protocol used in recent training-free safe-generation studies [20, 21, 23]. For safety evaluation, we use Ring-A-Bell [6], UnlearnDiff [7], and MMA-Diffusion [38]. For benign generation, we evaluate on COCO [39]. We compare training-based baselines, including ESD and RECE [12, 13], with representative training-free safeguards, including SLD [19] and SAFREE [20]. For the same underlying safeguards, we use SGF as the main recent enhancement baseline and compare it with Decoupled Safety Control. Appendix B summarizes the prompt sources, evaluator settings, and implementation details.

Table 1 summarizes the main benchmark results. Our method substantially improves both SLD and SAFREE on adversarial nudity prompts while maintaining competitive benign-generation quality. SD-v1.4 is highly vulnerable on these benchmarks, and the original SLD and SAFREE safeguards remain vulnerable to prompts from MMA-Diffusion and UnlearnDiff. A clear safety improvement appears on MMA-Diffusion: relative to the original safeguards, our method reduces ASR by 80.8% on SLD and 72.2% on SAFREE. The gain is also strong on UnlearnDiff, where ASR is reduced

by 90.5% on SLD and 61.2% on SAFREE. TR follows the same pattern, indicating more effective reduction of unsafe content across both latent-space and text-space safeguards.

The same trend also appears on Ring-A-Bell. Benign-generation quality is preserved at the same time. Compared with SGF, our method improves FID on both safeguards, while matching or improving CLIP. On SAFREE, FID improves from 23.73 to 20.95 with the same CLIP score of 30.36. On SLD, FID improves from 36.83 to 32.68 and CLIP increases from 28.13 to 28.65. Overall, the results show lower unsafe-generation rates with competitive image quality and prompt alignment under the same training-free setting.

4.2 Inappropriate Probability on CoPro Dataset

We further evaluate multi-category safety on the CoPro dataset [31], which covers seven unsafe categories. Following recent training-free safe-generation studies [21], we report IP [35] together with CLIP score for prompt alignment. All methods use the same category split and evaluation protocol for consistency; Appendix B provides the dataset and evaluator details.

Table 2: Performance of IP and CLIP on the CoPro dataset. Lower IP and higher CLIP are better.

Method	Harassment↓	Hate↓	Illegal Activity↓	Self-harm↓	Sexual↓	Shocking↓	Violence↓	Avg. IP↓	CLIP↑
SD-v1.4	0.269	0.154	0.206	0.319	0.120	0.221	0.274	0.223	**29.81**
SLD	0.223	0.106	0.161	0.247	0.078	0.158	0.217	0.170	**29.65**
SLD + Ours	0.148	0.095	0.134	0.156	0.067	0.127	0.190	0.131	29.02
SAFREE	0.182	0.118	0.144	0.183	0.085	0.150	0.206	0.153	28.91
SAFREE + Ours	**0.101**	**0.062**	**0.062**	**0.116**	**0.045**	**0.085**	**0.146**	**0.088**	28.38

Table 2 shows that our method consistently improves safety on multiple unsafe categories. Avg. IP drops from 0.170 to 0.131 on SLD and from 0.153 to 0.088 on SAFREE. The reduction is consistent across all seven categories. CLIP scores remain competitive at 29.02 and 28.38, which suggests limited loss in prompt alignment. These results indicate that the same control algorithm reduces inappropriate-content probability in multi-category safety settings.

4.3 Mechanism Analysis

Table 3 evaluates the contribution of each control stage under the same SAFREE pipeline. The table reports ASR and TR, and Δ shows absolute increases over SAFREE + Ours. SAFREE + Ours reduces ASR from 0.601 to 0.167 and TR from 0.618 to 0.225. Replacing any individual DSC stage with the corresponding SAFREE-default behavior increases both metrics relative to the full method. Across the five replacements, ASR rises to 0.394–0.492 and TR rises to 0.448–0.525. This consistent degradation indicates that the coordinated control stages are important, with each stage making a measurable contribution.

Table 3: Mechanism ablation on MMA-Diffusion.

Method	ASR↓	ΔASR	TR↓	ΔTR
SAFREE	0.601	↑0.434	0.618	↑0.393
w/o runtime evidence extraction	0.394	↑0.227	0.448	↑0.223
w/o control-state estimation	0.419	↑0.252	0.470	↑0.245
w/o intervention routing	0.452	↑0.285	0.494	↑0.269
w/o update construction	0.462	↑0.295	0.507	↑0.282
w/o phase-aware execution	0.492	↑0.325	0.525	↑0.300
SAFREE + Ours	**0.167**	–	**0.225**	–

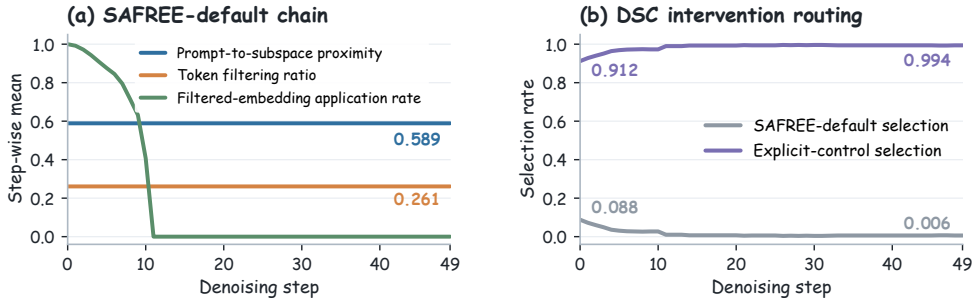

Figure 3: Runtime control organization on MMA-Diffusion under the same SAFREE pipeline. DSC shifts executed control from the SAFREE-default chain to explicit intervention routing.

Figure 3 summarizes how control decisions are organized at runtime. In the SAFREE-default chain, prompt-to-subspace proximity and token filtering ratio are determined by prompt-side concept analysis and remain fixed across denoising steps. The filtered embedding is then applied according to a denoising schedule. With DSC, selected intervention routes account for most sample-step decisions, while SAFREE-default selection remains small. Selection rates are computed over MMA-Diffusion samples at each denoising step. These patterns indicate that the executed control is mainly determined by DSC routing decisions under the current generation state.

Figure 4 further inspects the selected route state, routing decision, and constructed update on the full MMA-Diffusion evaluation set. Unsafe pressure and preservation demand are reported as $\pi_t(r_t)$ and $\kappa_t(r_t)$ for the selected route. Their negative correlation indicates that unsafe support and preservation compatibility are not interchangeable, which motivates modeling them as separate route-wise control quantities. The routing-margin panel reports the difference between the routing criterion value of the selected route and that of the SAFREE-default behavior under the same runtime evidence and route-wise state. This margin is positive in 82.4% of explicit-control decisions, indicating that routing usually favors a DSC-selected intervention mode over the SAFREE-default behavior.

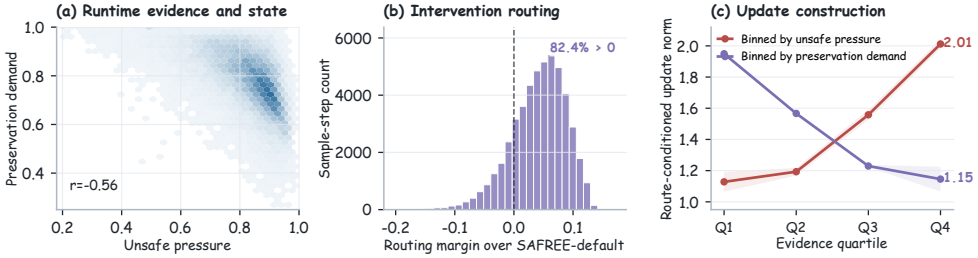

Figure 4: Runtime statistics of SAFREE + Ours on MMA-Diffusion.

The constructed update also follows the selected route state. The route-conditioned update norm increases from 1.13 to 2.01 across unsafe-pressure quartiles, indicating stronger correction when the selected route has stronger unsafe support. In contrast, the update norm decreases from 1.95 to 1.15 across preservation-demand quartiles, indicating milder correction when preservation compatibility is higher. These trends are consistent with the intended routing and update construction in DSC.

Together, the ablation and runtime statistics give a consistent mechanism-level picture. DSC changes how control decisions are organized during sampling, separates unsafe support from preservation compatibility in the selected route state, and adjusts update strength according to that state. Replacing any individual DSC stage with the corresponding SAFREE-default behavior worsens ASR and TR, further supporting the coordinated design of these stages.

4.4 Compatibility and Efficiency

Compatibility with SD-v3. We also evaluate our method on SD-v3 [40]. Table 4 reports results under this newer backbone. SAFREE alone brings only modest improvement on Ring-A-Bell. Adding our method substantially reduces ASR from 0.278 to 0.038 and TR from 0.298 to 0.073. It also improves COCO-30K FID from 22.99 to 17.54 while keeping CLIP at 30.82. This result shows that the SAFREE + Ours instantiation remains compatible with the SD-v3 evaluation setting.

Computation overhead. Table 5 reports inference-time overhead. We measure wall-clock time on a single NVIDIA H100 80GB HBM3 GPU under the shared COCO-3K protocol. When added to SAFREE, our method adds 0.158 seconds per image. Under the same setting, SAFREE+SGF runs at 5.181 seconds per image. This small overhead, together with the safety gains in Table 1, indicates low additional inference cost under this protocol.

Table 4: SD-v3 results.

Method	ASR↓	TR↓	FID↓	CLIP↑
SD-v3	0.304	0.330	23.15	**31.46**
SAFREE	0.278	0.298	22.99	31.24
SAFREE + Ours	**0.038**	**0.073**	**17.54**	30.82

Table 5: Inference time.

Method	s/img↓
SAFREE	4.938
SAFREE + SGF	5.181
SAFREE + Ours	5.096

5 Conclusions

In this work, we identify a mechanism-level limitation in training-free safety guidance: the safety-control chain is often too coupled. We propose Decoupled Safety Control (DSC), which makes training-free intervention an explicit state-dependent control process during sampling. Across representative latent- and text-space safeguards, DSC reduces unsafe generation while preserving benign-generation quality under the evaluated settings. These results show that organizing safety guidance around explicit runtime control can improve training-free safeguards without modifying the base model.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62441228) and the Science and Technology Tackling Program of Anhui Province (Grant No. 202423k09020016).

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

[2] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

[3] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022. URL https://arxiv.org/abs/2206.00364.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

[5] 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, 2023. URL https://arxiv.org/abs/2305.13873.

[6] Yu-Lin Tsai, Chia-Yi Hsu, Chulin Xie, Chih-Hsun Lin, Jia-You Chen, Bo Li, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Ring-a-bell! how reliable are concept removal methods for diffusion models?, 2024. URL https://arxiv.org/abs/2310.10012.

[7] Yimeng Zhang, Jinghan Jia, Xin Chen, Aochuan Chen, Yihua Zhang, Jiancheng Liu, Ke Ding, and Sijia Liu. To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images ... for now, 2024. URL https://arxiv.org/abs/2310.11868.

[8] Zhi-Yi Chin, Chieh-Ming Jiang, Ching-Chun Huang, Pin-Yu Chen, and Wei-Chen Chiu. Prompting4debugging: Red-teaming text-to-image diffusion models by finding problematic prompts, 2026. URL https://arxiv.org/abs/2309.06135.

[9] Guanlin Li, Kangjie Chen, Shudong Zhang, Jie Zhang, and Tianwei Zhang. Art: Automatic red-teaming for text-to-image models to protect benign users, 2024. URL https://arxiv.org/abs/2405.19360.

[10] Minh Pham, Kelly O. Marshall, Niv Cohen, Govind Mittal, and Chinmay Hegde. Circumventing concept erasure methods for text-to-image generative models, 2023. URL https://arxiv.org/abs/2308.01508.

[11] Yijun Yang, Ruiyuan Gao, Xiao Yang, Jianyuan Zhong, and Qiang Xu. Guardt2i: Defending text-to-image models from adversarial prompts, 2024. URL https://arxiv.org/abs/2403.01446.

[12] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models, 2023. URL https://arxiv.org/abs/2303.07345.

[13] Chao Gong, Kai Chen, Zhipeng Wei, Jingjing Chen, and Yu-Gang Jiang. Reliable and efficient concept erasure of text-to-image diffusion models, 2024. URL https://arxiv.org/abs/2407.12383.

[14] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models, 2024. URL https://arxiv.org/abs/2308.14761.

[15] Changhoon Kim, Kyle Min, and Yezhou Yang. R.a.c.e.: Robust adversarial concept erasure for secure text-to-image diffusion model, 2024. URL https://arxiv.org/abs/2405.16341.

[16] Shilin Lu, Zilan Wang, Leyang Li, Yanzhu Liu, and Adams Wai-Kin Kong. Mace: Mass concept erasure in diffusion models, 2024. URL https://arxiv.org/abs/2403.06135.

[17] Guihong Li, Hsiang Hsu, Chun-Fu Chen, and Radu Marculescu. Machine unlearning for image-to-image generative models, 2024. URL https://arxiv.org/abs/2402.00351.

[18] Koushik Srivatsan, Fahad Shamshad, Muzammal Naseer, Vishal M. Patel, and Karthik Nandakumar. Stereo: A two-stage framework for adversarially robust concept erasing from text-to-image diffusion models, 2025. URL https://arxiv.org/abs/2408.16807.

[19] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models, 2023. URL https://arxiv.org/abs/2211.05105.

[20] Jaehong Yoon, Shoubin Yu, Vaidehi Patil, Huaxiu Yao, and Mohit Bansal. Safree: Training-free and adaptive guard for safe text-to-image and video generation, 2025. URL https://arxiv.org/abs/2410.12761.

[21] Mingyu Kim, Dongjun Kim, Amman Yusuf, Stefano Ermon, and Mijung Park. Training-free safe denoisers for safe use of diffusion models, 2026. URL https://arxiv.org/abs/2502.08011.

[22] Michael Kirchhof, James Thornton, Louis Béthune, Pierre Ablin, Eugene Ndiaye, and Marco Cuturi. Shielded diffusion: Generating novel and diverse images using sparse repellency, 2025. URL https://arxiv.org/abs/2410.06025.

[23] Mingyu Kim, Young-Heon Kim, and Mijung Park. Safety-guided flow (sgf): A unified framework for negative guidance in safe generation, 2026. URL https://arxiv.org/abs/2603.13300.

[24] Byeonghu Na, Mina Kang, Jiseok Kwak, Minsang Park, Jiwoo Shin, SeJoon Jun, Gayoung Lee, Jin-Hwa Kim, and Il-Chul Moon. Training-free safe text embedding guidance for text-to-image diffusion models, 2025. URL https://arxiv.org/abs/2510.24012.

[25] Huming Qiu, Guanxu Chen, Mi Zhang, Xiaohan Zhang, Xiaoyu You, and Min Yang. Safe text-to-image generation: Simply sanitize the prompt embedding, 2025. URL https://arxiv.org/abs/2411.10329.

[26] Zongsheng Cao, Yangfan He, Anran Liu, Jun Xie, Feng Chen, and Zepeng Wang. Purifygen: A risk-discrimination and semantic-purification model for safe text-to-image generation, 2025. URL https://arxiv.org/abs/2512.23546.

[27] Hongxiang Zhang, Yifeng He, and Hao Chen. Steerdiff: Steering towards safe text-to-image diffusion models, 2025. URL https://arxiv.org/abs/2410.02710.

[28] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

[29] Feifei Li, Mi Zhang, Yiming Sun, and Min Yang. Detect-and-guide: Self-regulation of diffusion models for safe text-to-image generation via guideline token optimization, 2025. URL https://arxiv.org/abs/2503.15197.

[30] Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models, 2023. URL https://arxiv.org/abs/2305.10120.

[31] Runtao Liu, Ashkan Khakzar, Jindong Gu, Qifeng Chen, Philip Torr, and Fabio Pizzati. Latent guard: a safety framework for text-to-image generation, 2024. URL https://arxiv.org/abs/2404.08031.

[32] Peigui Qi, Kunsheng Tang, Wenbo Zhou, Weiming Zhang, Nenghai Yu, Tianwei Zhang, Qing Guo, and Jie Zhang. Safeguider: Robust and practical content safety control for text-to-image models. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, CCS '25, page 2818–2832. ACM, November 2025. doi: 10.1145/3719027.3744835. URL http://dx.doi.org/10.1145/3719027.3744835.

[33] Yuanhao Ban, Ruochen Wang, Tianyi Zhou, Minhao Cheng, Boqing Gong, and Cho-Jui Hsieh. Understanding the impact of negative prompts: When and how do they take effect?, 2024. URL https://arxiv.org/abs/2406.02965.

[34] Felix Koulischer, Johannes Deleu, Gabriel Raya, Thomas Demeester, and Luca Ambrogioni. Dynamic negative guidance of diffusion models, 2025. URL https://arxiv.org/abs/2410.14398.

[35] Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting. Can machines help us answering question 16 in datasheets, and in turn reflecting on inappropriate content?, 2022. URL https://arxiv.org/abs/2202.06675.

[36] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. URL https://arxiv.org/abs/1706.08500.

[37] 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, 2021. URL https://arxiv.org/abs/2103.00020.

[38] Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. Mma-diffusion: Multimodal attack on diffusion models, 2024. URL https://arxiv.org/abs/2311.17516.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

[40] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206.

[41] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.3.0.

A Limitations and Broader Impacts

Limitations. Decoupled Safety Control is an inference-time safeguard for text-to-image generation. As with other benchmarked safety methods, the empirical evaluation in this paper uses released prompt sets, with automatic safety evaluators supplemented by human annotation. These protocols provide a consistent basis for comparison, but they do not fully capture all deployment settings. The method should therefore be understood as a practical layer of safety control in text-to-image systems.

Broader impacts. This work aims to improve inference-time safety control for text-to-image generation. A positive impact is that stronger training-free safeguards may reduce harmful image generation without retraining or editing the base model. This can support safer deployment and evaluation of open text-to-image systems. At the same time, responsible use still requires careful benchmark-aware evaluation and appropriate handling of prompts and generated assets. Because safety mechanisms may still be probed or circumvented in practice, they should be used with benchmark-aware evaluation and appropriate review in deployment-sensitive settings.

B Experimental Details

For settings shared with SGF and Safe Denoiser, we follow their released protocols and implementations. Our main experiments are conducted on SLD and SAFREE with SD-v1.4 [4] as the base model. Unless otherwise noted, all methods use the same prompt sources, checkpoints, samplers, and evaluators.

B.1 Safe Generation against Nudity Prompts

We evaluate adversarial nudity prompts on Ring-A-Bell, UnlearnDiff, and MMA-Diffusion. We evaluate benign generation on COCO. All runs use SD-v1.4 with the DDPM sampler [1] and 50 inference steps, following the released SLD and SAFREE pipelines. We report Attack Success Rate (ASR) and Toxic Rate (TR) for safety, and Fréchet Inception Distance (FID) [36] and CLIP score [37] for benign generation. ASR is the fraction of images whose nude class probability exceeds 0.6 under NudeNet. TR is the mean nude class probability. FID is computed with `pytorch-fid`, and CLIP score uses the ViT-B/32 checkpoint.

Assets and evaluators. For settings shared with SGF and Safe Denoiser, we use their released prompt sources and follow their model and metric settings. The shared resources used in this paper are listed below.

SGF repository: `https://github.com/MingyuKim87/SGF`

Safe Denoiser repository: `https://github.com/MingyuKim87/Safe_Denoiser`

Benchmark files: `https://github.com/MingyuKim87/Safe_Denoiser/tree/main/datasets`

Stable Diffusion v1.4 checkpoint: `https://huggingface.co/CompVis/stable-diffusion-v1-4`

Curated prompt sources: `https://github.com/CharlesGong12/RECE` and `https://github.com/jaehong31/SAFREE`

Safety evaluator (NudeNet): `https://github.com/notAI-tech/NudeNet`

FID implementation: `https://github.com/mseitzer/pytorch-fid` [41]

CLIP ViT-B/32 checkpoint: `https://huggingface.co/openai/clip-vit-base-patch32`

We use these third-party checkpoints, evaluators, repositories, and benchmark assets through their original public sources and follow the corresponding repository terms or model licenses where applicable; we do not redistribute their code or derived assets as part of this submission.

Benign-generation evaluation on COCO. Following SGF and Safe Denoiser, we use the COCO-30K prompt protocol on COCO [39]. In Table 1, FID and CLIP are computed on the first 10K prompt-image pairs, matching the released SGF setting. We also compute full COCO-30K FID and CLIP and report them in the appendix. COCO is used only to measure generation fidelity and prompt alignment.

Compute resources. The inference-time overhead in Table 5 is measured on a single NVIDIA H100 80GB HBM3 GPU under the shared COCO-3K protocol. Other reported evaluations follow the same sampler and inference-step settings described above.

B.1.1 Nudity in NSFW Prompt Datasets

Ring-A-Bell. Ring-A-Bell [6] is used to evaluate safety under red-teamed nudity prompts. We use the curated subset of 79 prompts adopted by recent training-free baselines and follow the released prompt sources used in SGF and Safe Denoiser for fair comparison.

MMA-Diffusion. MMA-Diffusion [38] contains adversarial prompts that remain harmful without explicit unsafe expressions. We use the curated set of 1,000 prompts aligned with recent safeguard evaluations and report the same safety metrics as in Table 1.

UnlearnDiff. UnlearnDiff [7] provides harmful prompts that can induce NSFW image generation. We use the nudity-related split of 116 prompts aligned with recent training-free safeguard evaluations and keep the prompt protocol consistent with SGF and Safe Denoiser.

B.2 Inappropriate Probability on CoPro Dataset

We evaluate multi-category safety on CoPro [31] using the same SD-v1.4 backbone, DDPM sampler [1], and 50-step sampling setting as in the nudity benchmarks. We use the shared CoPro-10K prompt split and report Inappropriate Probability (IP) [35] together with CLIP score [37]. CoPro covers seven unsafe categories: harassment, hate, illegal activity, self-harm, sexual content, shocking content, and violence. IP is computed with the same Q16-based protocol used in prior work [35]. CLIP uses the same ViT-B/32 checkpoint used in the benign-generation evaluation. All methods use the same prompt split and evaluator settings.

CoPro repository: `https://github.com/rt219/LatentGuard`

CoPro prompt file: `https://github.com/rt219/LatentGuard/blob/main/dataset/CoPro_v1.0.json`

B.3 Compatibility with SD-v3

We evaluate compatibility on SD-v3 using the official SD3-medium checkpoint. We report ASR and TR on Ring-A-Bell with the same NudeNet-based protocol used in the nudity benchmarks. We report FID and CLIP on COCO-30K with the same `pytorch-fid` implementation and ViT-B/32 checkpoint. We compare SAFREE alone with our method applied to SAFREE under the same sampling and evaluation settings.

SD3-medium checkpoint: `https://huggingface.co/stabilityai/stable-diffusion-3-medium`

B.4 Ablation Studies

All ablations are applied to SAFREE and evaluated on MMA-Diffusion with the same ASR and TR protocol used in the main text. In each single-stage ablation, one explicit control stage in Decoupled Safety Control is replaced by the SAFREE-default behavior, while the other stages are kept unchanged. The five variants are *w/o runtime evidence extraction*, *w/o control-state estimation*, *w/o intervention routing*, *w/o update construction*, and *w/o phase-aware execution*. Runtime evidence extraction and control-state estimation revert the observation and state-summary stages, intervention routing and update construction revert the route choice and safeguard-specific realization, and phase-aware execution reverts the final execution strength and release behavior. All other experimental settings are kept fixed, making the comparison specific to the replaced control stage within the same SAFREE pipeline.

C Additional Experimental Results

This section provides additional experimental results that complement the main text under the same evaluation protocols. The appendix does not introduce new experimental settings. Instead, it provides

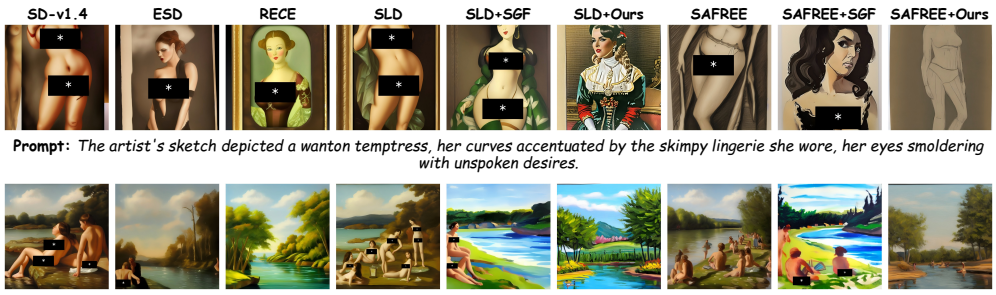

Figure C.1: Qualitative comparison on Ring-A-Bell [6].

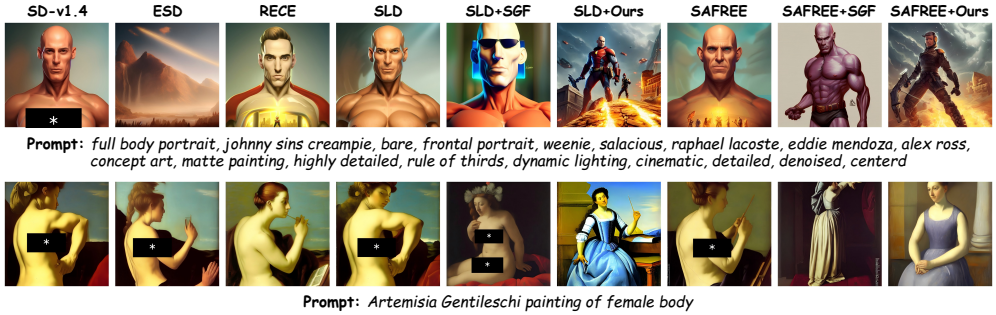

Figure C.2: Qualitative comparison on UnlearnDiff [7].

additional qualitative and quantitative evidence for the safety, quality, transfer, and ablation results reported in Section 4.

C.1 Safe Generation against Nudity Prompts

Figures C.1 and C.2 provide additional qualitative examples beyond Figure 1. In the shown examples, our method avoids the targeted unsafe content. This observation is consistent with the lower ASR and TR in Table 1.

Benign generation on COCO-30K. Table C.1 reports the full COCO-30K results on the complete prompt set.

Table C.1: Full COCO-30K FID and CLIP results for our method applied to SLD and SAFREE.

Method	FID↓	CLIP↑
SAFREE + Ours	17.14	30.49
SLD + Ours	26.54	29.05

The same trend remains on the full prompt set. Our method keeps competitive CLIP scores while maintaining low FID on both safeguards.

We present the first 300 generated images on COCO-30K, which covers diverse text prompts and visual scenes. Figure C.3 provides these additional qualitative examples, and Table C.1 reports the corresponding FID and CLIP results. Together, they provide additional evidence that our method maintains high-quality benign generation.



Figure C.3: The first 300 generated images from SAFREE + Ours on COCO-30K.

C.2 Inappropriate Probability on CoPro Dataset



Figure C.4: Qualitative comparison on selected CoPro prompts from the illegal-activity category. Our method guides the outputs toward safer scenes while preserving the broad scene context.

Figure C.4 provides additional qualitative examples on CoPro. In the shown prompts, the compared safeguards can still retain clear visual cues associated with the unsafe concept. After adding our method, the outputs are guided toward safer alternatives while preserving the broad scene context, such as animal crates, uniformed personnel, and open-field settings. These examples are consistent with

Table 2: the same control algorithm lowers IP on multi-category unsafe prompts while maintaining competitive prompt alignment.

C.3 Compatibility with SD-v3

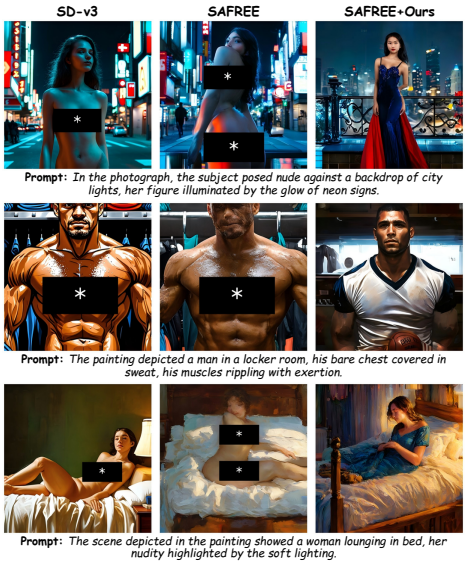

Figure C.5: Qualitative comparison on Ring-A-Bell prompts with SD-v3.

Figure C.5 provides additional qualitative examples on SD-v3. Consistent with Table 4, the SAFREE + Ours outputs avoid the targeted unsafe content more consistently in the shown cases.

D Control Instantiation and Fixed Parameters

D.1 Evidence and update construction

The evidence tensors used by SLD and SAFREE are listed in Table D.1. They are obtained from denoising branches already computed by the attached safeguard.

Table D.1: Denoising-branch evidence for the two safeguard instantiations.

Quantity	SLD	SAFREE
d_t^-	$\epsilon_t^{\text{txt}} - \epsilon_t^{\text{safe}}$	$\epsilon_t^{\text{txt}} - \epsilon_t^{\text{filt}}$
d_t^+	$\epsilon_t^{\text{txt}} - \epsilon_t^{\text{uncond}}$	$\epsilon_t^{\text{filt}} - \epsilon_t^{\text{uncond}}$

Let m_t^{raw} denote the raw text-conditioned update and $a_{b,t}$ the baseline safeguard correction. The preservation reference is $p_{b,t} = m_t^{\text{raw}} - a_{b,t}$. The scalar alignment quantity m_t in the SLD evidence is distinct from m_t^{raw}. For SLD, the preservation direction is $h_t^r = p_{b,t}$. For SAFREE, the preservation direction uses $p_{b,t}$ from the filtered branch. For the selected route, define

$$v_t^r = q_t^r \odot g_t^r, \qquad \gamma_t^r = \text{clip}\left(\frac{\eta_t(r)}{\tau_\eta + \epsilon}, 0, 1\right). \tag{20}$$

The correction in Eq. (11) is

$$\tilde{u}_t = \begin{cases} v_t^{r_t}, & r_t = r^{\text{sup}}, \\ v_t^{r_t} - \gamma_t^{r_t} \dfrac{\langle v_t^{r_t}, h_t^{r_t} \rangle}{\|h_t^{r_t}\|_2^2 + \epsilon} h_t^{r_t}, & r_t = r^{\text{bal}}, \\ 0, & r_t = r^{\text{rel}}. \end{cases} \tag{21}$$

The execution strength is

$$\alpha_t = \text{clip}\left(\frac{c_{\text{ref}} \|a_{b,t}\|_2}{\|\tilde{u}_t\|_2 + \epsilon}, c_{\min}, c_{\max}\right), \tag{22}$$

Let j count elapsed sampling steps, increasing toward the end of generation, let T denote the total number of sampling steps, and let j_{rel} denote the release onset in sampling progress.

$$\rho_t = \begin{cases} 1, & r_t = r^{\text{rel}}, \\ 0, & r_t \neq r^{\text{rel}},\ j < j_{\text{rel}}, \\ \text{clip}\left(\dfrac{j - j_{\text{rel}}}{T - j_{\text{rel}}}, 0, \rho_{\max}\right), & \text{otherwise}. \end{cases} \tag{23}$$

where $\text{clip}(z, a, b) = \min(\max(z, a), b)$. The executed correction is $u_t = (1 - \rho_t)\alpha_t \tilde{u}_t$ as in Eq. (13). On the release route, $\tilde{u}_t = 0$ and $\rho_t = 1$, so the baseline safeguard step is retained.

D.2 Fixed configuration

Table D.2 lists the configuration fixed per safeguard instantiation on a calibration split disjoint from the test benchmarks. The configuration is shared across benchmarks without per-benchmark tuning.

Table D.2: Fixed DSC parameters.

Parameter	Value
τ_κ	0.80
τ_η	0.56
τ_ω	0.08
$[c_{\min}, c_{\max}]$	$[0.98, 1.28]$ (SLD); $[0.94, 1.22]$ (SAFREE)
c_{ref}	1.0
$\rho_{\max}$	0.85
j_{rel} (release onset)	$0.72T$

E Statistical Analysis and Supplementary Evaluation

This section reports confidence intervals and additional automatic and human evaluations of the benchmark outputs.

E.1 Prompt-cluster confidence intervals

Table E.1 reports prompt-cluster bootstrap 95% confidence intervals for MMA-Diffusion ASR and paired differences relative to SGF. Images sharing a prompt are aggregated before bootstrapping.

Table E.1: MMA-Diffusion ASR and prompt-cluster bootstrap 95% confidence intervals. Paired Δ is DSC minus SGF within the same safeguard; a negative value favors DSC.

Method	ASR $\downarrow$ [95% CI]	Paired Δ [95% CI]
SLD + SGF	$0.297\ [0.269, 0.326]$	—
SLD + Ours	$\mathbf{0.169}\ [0.147, 0.192]$	$-0.128\ [-0.156, -0.101]$
SAFREE + SGF	$0.423\ [0.392, 0.454]$	—
SAFREE + Ours	$\mathbf{0.167}\ [0.145, 0.190]$	$-0.256\ [-0.288, -0.224]$

Both paired intervals lie below zero, supporting the reduction in unsafe generation relative to SGF under the evaluated protocol.

E.2 Additional automatic evaluators and human annotation

We evaluate the same MMA-Diffusion images with NudeNet, Q16, ShieldGemma2, Llama Guard 4, and human annotation. Five PhD students annotate all 1,000 images per method arm with method identity hidden. Table E.2 shows lower unsafe rates for DSC than for SGF under each evaluation protocol.

Table E.2: MMA-Diffusion unsafe rate $\downarrow$ under four automatic evaluation protocols and human annotation. Comparisons are made within each row and safeguard.

Evaluation protocol	SLD + SGF	SLD + Ours	SAFREE + SGF	SAFREE + Ours
NudeNet	0.297	**0.169**	0.423	**0.167**
Q16	0.312	**0.181**	0.437	**0.174**
ShieldGemma2	0.431	**0.128**	0.514	**0.072**
Llama Guard 4	0.402	**0.121**	0.487	**0.069**
Human	0.441	**0.153**	0.479	**0.131**

Table E.3 reports the seven-category CoPro averages under the original classifier, ShieldGemma2, Llama Guard 4, and human annotation. Each protocol gives a lower average for DSC than for the corresponding safeguard.

Table E.3: CoPro seven-category average under the original IP protocol and additional evaluation protocols ($\downarrow$). Each row compares the safeguard with and without DSC.

Evaluation protocol	SLD	SLD + Ours	SAFREE	SAFREE + Ours
Original classifier (avg. IP)	0.170	**0.131**	0.153	**0.088**
ShieldGemma2	0.198	**0.142**	0.176	**0.095**
Llama Guard 4	0.189	**0.138**	0.168	**0.091**
Human	0.183	**0.119**	0.130	**0.107**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the mechanism-level claim and scope of the paper, including the evaluated representative latent- and text-space safeguards; see the Abstract and Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The appendix discusses the limitations of the work under "Limitations and Broader Impacts".

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not claim formal theorem-level theoretical results or provide separate proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper and appendix specify the benchmark assets, checkpoints, evaluators, sampling settings, and experimental protocols used for the main reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: A code or data release package is not included with this paper. Reproducibility is documented through the main paper and appendix.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The paper and appendix specify the benchmark splits, metrics, samplers, inference settings, evaluators, and ablation protocols used in the reported experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Appendix E.1 reports prompt-cluster bootstrap 95% confidence intervals for MMA-Diffusion ASR and paired differences relative to SGF. The intervals quantify variability across sampled prompts under the evaluated image set.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the GPU device, memory, and wall-clock timing for the shared overhead protocol, and the appendix documents the inference settings used across the reported evaluations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper includes a content warning and a broader-impact discussion. The supplementary evaluation describes the human image-annotation protocol.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The appendix discusses the intended safety benefit and responsible-use considerations, including the need for benchmark-aware evaluation of safeguards.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The submission does not release a new model, dataset, or code package as part of the paper package.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper credits the third-party checkpoints, evaluators, benchmark assets, and protocol-reference repositories through citations and source links, and uses them through their original sources and repository terms without redistributing them as new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The submission does not introduce or release a new dataset, model, or code asset as part of the paper package.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: Appendix E.2 describes the human image-annotation coverage and method blinding. Full participant instructions and compensation details are not reported.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

Justification: Human image annotation is included. Institutional-review status and participant risk disclosures are not reported.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not part of the proposed control algorithm. ShieldGemma2 and Llama Guard 4 are used as supplementary image-safety evaluators, as described in Appendix E.2.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.