STEER AND PACE: JOINT SAFETY GUIDANCE FOR TEXT-TO-IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image (T2I) generative models produce high-quality images from free-form prompts, but can also generate unsafe content. Existing sampling-time safeguards often aggregate multiple unsafe categories into one condition or handle them one at a time, limiting their ability to control the contributions of different categories jointly. A further challenge is determining how to allocate correction strength across sampling stages that play different roles in image formation. To address these challenges, we propose SafeConductor, a training-free method that combines joint correction with phase-aware allocation for safe text-to-image generation. At each iteration, SafeConductor formulates joint correction as minimizing changes to the current guidance under separate non-positive projection constraints for all categories. The resulting high-dimensional problem reduces to a nonnegative quadratic program with one coefficient per category, accounting for interactions among unsafe directions. We approximately solve this problem using cyclic coordinate descent. The correction is then applied with a smooth sinusoidal allocation that emphasizes the intermediate stage and reduces intervention near the trajectory endpoints, aiming to preserve image composition and visual details. Extensive experiments across four T2I safety benchmarks demonstrate that SafeConductor achieves state-of-the-art harmful content mitigation on both SD1.5 and FLUX.2 while maintaining competitive prompt alignment and generation quality. Our code is available in the supplementary material.

Warning: This paper contains model outputs that may be offensive.

1 INTRODUCTION

Recent text-to-image (T2I) generators, including U-Net-based latent diffusion models (Rombach et al., 2022) and Transformer-based rectified flow models (Black Forest Labs, 2026), can generate high-quality images that closely follow free-form prompts describing objects, scenes, and visual styles. They can also produce sexual, violent, self-harm, and otherwise unsafe imagery, even when prompts express these risks indirectly (Schramowski et al., 2023). Preventing such outputs while preserving benign subjects, composition, and visual details remains a central challenge.

Prior safeguards intervene at different points in the generation process. Some methods suppress targeted concepts by fine-tuning models or training adapters (Gandikota et al., 2023; Gao et al., 2025; Liu et al., 2025), but may affect unrelated content and require further adaptation to new concepts or backbones. Other methods retain the pretrained backbone weights and modify prompt representations or internal features (Yoon et al., 2025; Na et al., 2025; Kim & Ghadiyaram, 2025; Yang et al., 2026), but some still require additional learned components or depend on specific internal model structures. Sampling-time safety guidance (Schramowski et al., 2023; Xiang et al., 2026) offers another route by directly correcting the model predictions passed to the sampler. In models with explicit classifier-free guidance (CFG), the difference between prompt-conditioned and empty-conditioned predictions defines the prompt guidance (Ho & Salimans, 2022). These safeguards use additional predictions under unsafe conditions to correct this guidance at each sampling iteration without updating model weights. However, this form of safety guidance raises two questions: how to compute one correction that accounts for multiple unsafe concepts, and how to allocate its strength across sampling stages.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

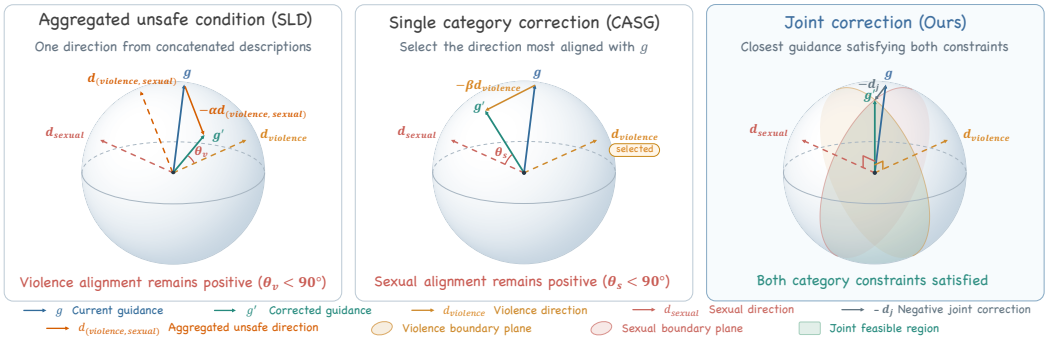

Figure 1: **Comparison of correction strategies for concurrent unsafe concepts.** Correcting along a direction from aggregated unsafe descriptions can leave positive projections onto unsafe directions. Correcting only the direction most aligned with the current guidance can increase the projection onto another unsafe direction. Our joint correction finds the guidance closest to the original while satisfying all category constraints. Dashed arrows point from the sphere center toward the far hemisphere.

The first question concerns joint control of multiple unsafe concepts, as illustrated in Figure 1. One strategy aggregates predefined unsafe terms into one condition, as in SLD (Schramowski et al., 2023), limiting separate category control and potentially introducing conflicts. Another constructs a direction for each category, but corrects only along the category direction most aligned with the current guidance at each iteration, as in CASG (Xiang et al., 2026), leaving other positively aligned directions without explicit constraints. The second question concerns the dynamic allocation of correction strength across sampling stages. A constant coefficient can provide a reasonable global setting, but cannot redistribute correction strength across sampling stages. Intuitively, early iterations largely establish global layout and subject structure, whereas later iterations refine local appearance and texture (Yi et al., 2024). Analyses of ordinary CFG similarly show that guidance has different effects across sampling stages (Kynkäänniemi et al., 2024; Jin et al., 2026; Fan et al., 2025). These issues motivate a safety guidance strategy that combines joint correction for multiple unsafe concepts with correction strength allocation across sampling stages.

We therefore propose SafeConductor, a training-free method for safe text-to-image generation based on joint correction and phase-aware allocation. For joint correction, SafeConductor constructs separate category directions using prediction differences from paired unsafe and safe descriptions, aiming to reduce the influence of shared semantic content. It then jointly corrects the current guidance under non-positive projection constraints for all categories, accounting for interactions among directions while minimizing changes to the original guidance. The resulting high-dimensional projection problem reduces to a nonnegative quadratic program with one coefficient per category, which we approximately solve using cyclic coordinate descent. For phase-aware allocation, SafeConductor applies the resulting correction with a smooth sinusoidal allocation that emphasizes the intermediate stage while reducing intervention near the trajectory endpoints, without changing the backbone's native guidance configuration. The same formulation applies to the noise predictions of SD1.5 (Rombach et al., 2022) and the velocity predictions of FLUX.2 (Black Forest Labs, 2026).

Overall, our main contributions are as follows:

- We propose SafeConductor, a training-free method for safe generation that jointly corrects the current guidance under separate constraints for all categories. The joint optimization accounts for interactions among concurrent unsafe directions and is reduced to a nonnegative quadratic program for efficient approximate solution.
- We introduce a phase-aware allocation of correction strength across sampling stages. By separating correction construction from strength allocation, SafeConductor emphasizes the intermediate stage while reducing intervention near the trajectory endpoints to better preserve prompt semantics and image structure.
- Extensive evaluations on four T2I safety benchmarks and two representative generative architectures demonstrate the effectiveness and architectural transferability of SafeConductor. It achieves state-of-the-art harmful content mitigation on both SD1.5 and FLUX.2 while maintaining competitive prompt alignment and generation quality.

2 Related Work

2.1 Text-to-image generation with safety mechanisms

Safety mechanisms for text-to-image generation can be broadly grouped by their intervention targets: model parameters, conditioning representations, intermediate features, or sampling updates. Parameter-level methods fine-tune denoisers or lightweight adapters (Gandikota et al., 2023; Lu et al., 2024; Lyu et al., 2024; Gao et al., 2025), apply closed-form attention edits (Gandikota et al., 2024; Gong et al., 2024), or use preference optimization (Park et al., 2024; Liu et al., 2025) and adversarial training (Zhang et al., 2024). Condition-level methods modify text embeddings or prompt representations (Yoon et al., 2025; Na et al., 2025), whereas feature-level methods intervene in internal activations or attention computations (Yang et al., 2026). Sampling-level methods instead alter the prediction provided to the sampler (Schramowski et al., 2023; Xiang et al., 2026). Transferring safeguards from U-Net-based latent diffusion (Rombach et al., 2022) to Transformer-based rectified flow (Esser et al., 2024; Black Forest Labs, 2026) may require adaptations to different attention structures and prediction targets (Gao et al., 2025; Yang et al., 2026).

2.2 Classifier-free guidance

Classifier-free guidance (CFG) combines conditional and unconditional predictions to strengthen the influence of the input condition (Ho & Salimans, 2022). High guidance scales can nevertheless cause oversaturation and artifacts, which APG (Sadat et al., 2025) mitigates by reweighting the components of the CFG update. Beyond the form of the guidance update, limited-interval and stage-wise analyses show that guidance effects also depend on the sampling stage (Kynkäänniemi et al., 2024; Jin et al., 2026). In flow matching, CFG-Zero* (Fan et al., 2025) addresses velocity estimation errors by optimizing scaling and setting velocity predictions to zero during the initial sampling steps. Dynamic Negative Guidance (Koulischer et al., 2025) modulates negative guidance according to sampling time and state rather than using a fixed coefficient. For multi-category safety control, SLD (Schramowski et al., 2023) derives a correction from an aggregated unsafe condition, while CASG (Xiang et al., 2026) corrects only along the category direction most aligned with the current guidance at each iteration. These studies motivate examining both the joint construction of corrections for multiple unsafe concepts and the allocation of their strength across sampling stages.

3 Methodology

SafeConductor provides training-free safety guidance during sampling for text-to-image models with explicit CFG. At each iteration, it constructs category directions from predictions under paired unsafe and safe descriptions, computes a joint correction and allocates its strength according to the sampling stage. Figure 2 illustrates the overall framework of SafeConductor, and Algorithm 2 in Appendix C.4 summarizes its core sampling procedure.

3.1 Preliminaries

To formulate SafeConductor consistently for both diffusion and rectified-flow models, we first introduce common notation for the model prediction used by the sampler. Let $T \geq 2$ denote the number of sampling iterations. At iteration $i \in \{0, \ldots, T-1\}$, t_i denotes the time input supplied to the network and x_i denotes the latent state before the sampler update. Let v_θ denote the frozen generative network with parameters θ. For a text condition r, the corresponding network prediction at iteration i is denoted by

$$v_i^r = v_\theta(x_i, t_i, r). \tag{1}$$

In the two settings considered here, v_i^r denotes the noise prediction for U-Net-based diffusion models (Ho et al., 2020; Rombach et al., 2022), and the velocity prediction for Transformer-based flow-matching or rectified-flow models (Liu et al., 2023; Lipman et al., 2023; Black Forest Labs, 2026).

At sampling iteration i, CFG evaluates the same network under the user prompt r and the empty condition $\varnothing$ (Ho & Salimans, 2022). Their difference defines the prompt guidance:

$$g_i = v_i^r - v_i^{\varnothing}. \tag{2}$$

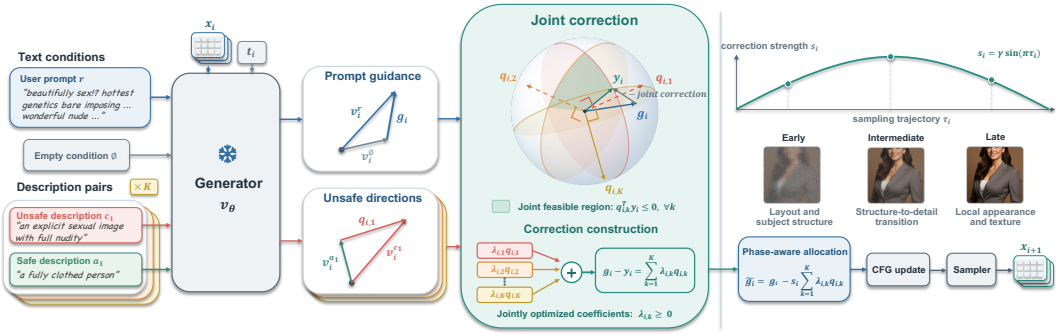

Figure 2: **Overview of SafeConductor at one sampling iteration.** Left: The generator uses the user prompt, empty condition and K unsafe–safe description pairs to construct the prompt guidance and category-specific unsafe directions. The joint correction then seeks the minimum change to the current guidance needed to satisfy all category constraints. Dashed arrows point from the sphere center toward the far hemisphere. Right: A smooth sinusoidal allocation controls correction strength across sampling stages, and the adjusted guidance is used in the standard CFG and sampler updates.

The prediction supplied to the sampler by ordinary CFG is

$$v_i^{\text{cfg}} = v_i^{\varnothing} + w g_i, \tag{3}$$

where w is the CFG weight. SafeConductor modifies only g_i, leaving the pretrained model parameters unchanged.

3.2 Joint correction for concurrent unsafe concepts

At each sampling iteration, SafeConductor constructs one unsafe direction for each of the K categories. Motivated by difference-based concept representations in text-embedding space (Tsai et al., 2024) and denoising-prediction space (Schramowski et al., 2023; Xiang et al., 2026), we associate category $k \in \{1, \dots, K\}$ with an unsafe description c_k and a corresponding safe description a_k, and define the unsafe direction for category k as

$$h_{i,k} = v_i^{c_k} - v_i^{a_k}. \tag{4}$$

Pairing unsafe and safe descriptions is intended to reduce the influence of shared semantic content. The complete description pairs are provided in Appendix A. Since the norm of $h_{i,k}$ can vary across categories and sampling states, we normalize it before joint optimization:

$$q_{i,k} = \frac{h_{i,k}}{\max(\|h_{i,k}\|, \varepsilon)}. \tag{5}$$

Here, $\varepsilon > 0$ prevents numerical instability when $\|h_{i,k}\|$ is close to zero.

Given these category-specific directions, SafeConductor computes a single correction that considers all K categories jointly. For a candidate corrected guidance vector $y \in \mathbb{R}^d$, where d is the number of scalar entries in the tensor, a positive inner product $q_{i,k}^\top y$ indicates a component along the corresponding unsafe direction. Let $A_i \in \mathbb{R}^{K \times d}$ contain $q_{i,k}^\top$ as its kth row. The condition $A_i y \le 0$ then expresses $q_{i,k}^\top y \le 0$ for all $k = 1, \dots, K$. We therefore seek the vector closest to the current guidance g_i whose inner product with every unsafe direction is non-positive:

$$y_i^\star = \arg\min_{y \in \mathbb{R}^d} \frac{1}{2}\|y - g_i\|^2 \quad \text{s.t.} \quad A_i y \le 0. \tag{6}$$

This formulation minimizes the change to the current guidance while satisfying all category constraints, helping preserve the information provided by the user prompt. The constraints define a closed convex set containing $y = 0$, so the optimal guidance exists and is unique.

The variable $y \in \mathbb{R}^d$ has the full dimensionality of the model prediction, whereas the optimization contains only K category constraints. We therefore use Lagrange duality (Boyd & Vandenberghe,

2004) to eliminate y analytically and optimize K category coefficients instead. For each constraint, we introduce a nonnegative multiplier λ_k and collect these multipliers in $\lambda \in \mathbb{R}^K$. The Lagrangian combines the original modification cost with the weighted constraint expressions:

$$\mathcal{L}_i(y, \lambda) = \frac{1}{2}\|y - g_i\|^2 + \lambda^\top A_i y. \tag{7}$$

For nonnegative λ, the Lagrangian does not exceed the original objective at any feasible y, since $\lambda^\top A_i y \leq 0$. Minimizing it over all $y \in \mathbb{R}^d$ for fixed λ therefore gives a lower bound on the minimum modification cost. As the Lagrangian is a strictly convex quadratic function of y, setting its gradient to zero yields the unique minimizer $y_i(\lambda) = g_i - A_i^\top \lambda$. Substituting this vector back into the Lagrangian eliminates y and leaves a function of λ alone. We define $G_i = A_i A_i^\top$, which captures interactions among the unsafe directions, and $b_i = A_i g_i$, which measures the alignment of each direction with the current guidance. For each fixed λ, we define the dual function as the minimum value of the Lagrangian over y:

$$\mathcal{D}_i(\lambda) = \min_{y \in \mathbb{R}^d} \mathcal{L}_i(y, \lambda) = b_i^\top \lambda - \frac{1}{2}\lambda^\top G_i \lambda. \tag{8}$$

We then determine the category coefficients by maximizing $\mathcal{D}_i(\lambda)$ over $\lambda \geq 0$, making the lower bound on the minimum modification cost as large as possible. For this feasible convex quadratic problem with linear constraints, strong duality (Boyd & Vandenberghe, 2004) ensures that this maximum equals the minimum modification cost under the category constraints. Equivalently, we minimize $-\mathcal{D}_i(\lambda)$ over $\lambda \geq 0$ to obtain the optimal coefficients $\lambda_i^\star$:

$$\lambda_i^\star \in \arg\min_{\lambda \geq 0} \left(\frac{1}{2}\lambda^\top G_i \lambda - b_i^\top \lambda \right). \tag{9}$$

This reformulation reduces the original optimization to a convex quadratic program with K nonnegative coefficients. The corrected guidance $y_i^\star$ can thus be recovered as

$$y_i^\star = g_i - A_i^\top \lambda_i^\star = g_i - \sum_{k=1}^{K} \lambda_{i,k}^\star q_{i,k}. \tag{10}$$

Note that the corresponding joint correction $g_i - y_i^\star$ is a nonnegative combination of the unsafe directions, with jointly optimized coefficients determining each direction's contribution. In particular, if the exact corrected guidance satisfies $q_{i,k}^\top y_i^\star < 0$, the corresponding coefficient is zero, $\lambda_{i,k}^\star = 0$. This is reasonable because a category whose constraint is already strictly satisfied at the joint solution does not require a direct correction term along its direction.

In practice, we approximately solve the coefficient problem at each sampling iteration using cyclic coordinate descent with a fixed number of sweeps (Wright, 2015). Each update changes one nonnegative coefficient while accounting for the current values of the others, so sequential updates still optimize the joint objective. Appendix C provides the full derivation, optimality conditions, numerical solution details and core sampling procedure.

3.3 Phase-aware allocation of correction strength

Applying the joint correction in Section 3.2 with a constant strength coefficient does not consider the different roles of sampling stages and may unnecessarily alter benign content. We therefore separate correction construction from strength allocation: we compute the joint correction from the current guidance and all category directions, while phase-aware allocation controls how strongly it is applied at each iteration. Early iterations largely establish global layout and subject structure, whereas later iterations refine local appearance and texture. Accordingly, SafeConductor reduces correction strength near both trajectory endpoints and emphasizes the intermediate stage to limit disruption to image structure and visual details. We implement this phase-aware allocation with a smooth sinusoidal variation, avoiding the abrupt transitions of a hard guidance window. Let $\tau_i = i/(T-1)$ denote the normalized position of sampling iteration i. We define the correction strength as

$$s_i = \gamma \sin(\pi \tau_i). \tag{11}$$

Here, s_i is the correction strength at iteration i, and $\gamma > 0$ controls its overall scale.

As in Equation (10), the joint correction is $g_i - y_i^\star = \sum_{k=1}^{K} \lambda_{i,k}^\star q_{i,k}$. We scale this correction by the phase-aware allocation strength s_i:

$$\widetilde{g}_i = g_i - s_i \sum_{k=1}^{K} \lambda_{i,k}^\star q_{i,k}. \tag{12}$$

The coefficients $\lambda_{i,k}^\star$ determine the relative contributions of the unsafe directions to the joint correction, while s_i controls how strongly the correction is applied at the current sampling stage. SafeConductor then substitutes $\widetilde{g}_i$ for g_i in the ordinary CFG update in Equation (3), leaving the empty-conditioned prediction and CFG weight unchanged, leading to $v_i^{\text{safe}} = v_i^{\varnothing} + w\widetilde{g}_i$. To this end, substituting Equation (12) and $g_i = v_i^r - v_i^{\varnothing}$ gives

$$v_i^{\text{safe}} = v_i^{\varnothing} + w\left(v_i^r - v_i^{\varnothing}\right) - w s_i \sum_{k=1}^{K} \lambda_{i,k}^\star q_{i,k}. \tag{13}$$

The first two terms form the ordinary CFG prediction, while the final term is the scaled phase-aware joint safety correction introduced by SafeConductor.

4 Experiments

4.1 Experimental setup

Backbones and inference settings. We evaluate Stable Diffusion 1.5 (SD1.5) (Rombach et al., 2022) and FLUX.2-klein-base-4B (Black Forest Labs, 2026) at 512×512 resolution. SD1.5 uses 50 PNDM steps (Liu et al., 2022), CFG weight 7.5, and $\gamma = 2.5$; FLUX.2 uses 24 Euler steps, CFG weight 4.0, and $\gamma = 3.0$. Full settings are provided in Appendix B.

Datasets. We evaluate safety on the full prompt sets of I2P (4,703 prompts) (Schramowski et al., 2023), T2VSafetyBench (T2VSB; 3,443 prompts) (Miao et al., 2024) and Unsafe Diffusion (UD; 904 prompts) (Qu et al., 2023), together with a fixed sample of 1,000 CoProV2 prompts (Liu et al., 2025). Image quality and prompt alignment are evaluated using a fixed sample of 1,000 MS-COCO captions (Lin et al., 2014). Adversarial robustness is evaluated using released prompt sets from P4D (Chin et al., 2024), Ring-A-Bell (Tsai et al., 2024), and MMA-Diffusion (Yang et al., 2024).

Baselines. We compare SafeConductor with safeguards that modify model parameters, including ESD (Gandikota et al., 2023), UCE (Gandikota et al., 2024), RECE (Gong et al., 2024), MACE (Lu et al., 2024), SafetyDPO (Liu et al., 2025), and EraseAnything (Gao et al., 2025). Training-free inference-time baselines include SLD-Max (Schramowski et al., 2023), SAFREE (Yoon et al., 2025), and CASG (Xiang et al., 2026). For CASG, we use its SLD-based variant.

Evaluation metrics. We evaluate generation safety using Attack Success Rate (ASR; lower is better), the percentage of generated images flagged as unsafe by Q16 (Schramowski et al., 2022) or containing a NudeNet detection (Bedapudi, 2019) with confidence above 0.5. On the sampled MS-COCO set, we evaluate prompt alignment using CLIP score (Hessel et al., 2021) and image quality using FID (Heusel et al., 2017). Higher CLIP score and lower FID are better.

Inspired by the use of harmonic means in concept erasure (Lu et al., 2024; Saha et al., 2026), we define a Safety, Quality and Alignment score (SQA) using harmonic mean aggregation. For method m, the safety score is $S_m = 1 - \overline{\text{ASR}}_m$, where $\overline{\text{ASR}}_m$ is the mean ASR across the evaluated safety benchmarks, expressed as a fraction. We define the normalized quality and alignment scores as $Q_m = \min(1, \text{FID}_0/\text{FID}_m)$ and $A_m = \min(1, \text{CLIP}_m/\text{CLIP}_0)$, where FID_0 and CLIP_0 are the corresponding scores of the native model with the same backbone. We first combine quality and alignment as $R_m = 2/(Q_m^{-1} + A_m^{-1})$, then combine this with safety as $\text{SQA}_m = 100 \cdot 2/(S_m^{-1} + R_m^{-1})$. Higher SQA indicates better combined performance, complementing the individual metrics.

4.2 MAIN RESULTS

Quantitative comparison. Table 1 compares SafeConductor with representative safeguards on SD1.5 and FLUX.2. SafeConductor achieves the lowest ASR on all four safety benchmarks for both backbones. Compared with SLD-Max and CASG, which also apply safety guidance during sampling, SafeConductor improves both FID and CLIP on both backbones showing that stronger unsafe content suppression is accompanied by better generation quality and prompt alignment on benign prompts. This simultaneous improvement distinguishes SafeConductor from simply accepting greater quality loss for stronger suppression. On FLUX.2, UCE and RECE retain FID and CLIP close to the native model but provide weaker unsafe content suppression. SafeConductor instead combines stronger safety with competitive generation performance, achieving the highest SQA on both backbones. These results support the effectiveness of the joint correction and phase-aware allocation framework across both noise and velocity prediction settings.

Table 1: Quantitative comparison of safety, image quality and prompt alignment.

Backbone	Method	ASR (%) ↓				FID ↓	CLIP ↑	SQA ↑
		I2P	T2VSB	UD	CoProV2	COCO		
SD1.5	Native	42.50	57.25	53.98	27.70	69.81	**31.59**	70.67
	ESD (ICCV'23)	30.26	40.89	32.41	20.40	**67.36**	29.94	80.76
	SLD-Max (CVPR'23)	10.99	26.46	15.27	6.90	78.10	29.20	87.89
	UCE (WACV'24)	32.68	46.33	39.71	19.30	78.84	31.07	76.93
	RECE (ECCV'24)	26.73	33.63	28.87	16.70	77.66	29.35	81.48
	MACE (CVPR'24)	18.58	16.32	15.27	14.80	73.72	28.14	87.60
	SafetyDPO (ICCV'25)	14.12	23.00	17.70	4.30	76.16	30.65	89.52
	SAFREE (ICLR'25)	21.24	42.55	25.66	13.50	74.10	30.50	83.50
	CASG (CVPR'26)	9.25	9.27	9.62	4.00	73.88	29.16	92.67
	SafeConductor (Ours)	**4.81**	**6.53**	**8.41**	**2.50**	72.12	29.51	**94.76**
FLUX.2	Native	34.15	49.46	44.03	27.30	**72.24**	**32.15**	75.98
	SLD-Max (CVPR'23)	16.31	16.61	23.23	14.00	89.21	29.44	84.17
	ESD (ICCV'23)	23.39	27.97	28.10	18.70	73.51	30.52	84.72
	UCE (WACV'24)	18.88	28.96	17.15	17.60	73.66	31.84	87.92
	RECE (ECCV'24)	16.29	26.87	14.93	16.60	74.08	31.59	88.84
	CASG (CVPR'26)	16.67	14.09	28.98	12.50	95.29	28.62	81.91
	EraseAnything (ICML'25)	18.12	24.98	16.48	9.90	78.90	30.17	87.37
	SafeConductor (Ours)	**7.70**	**11.18**	**11.06**	**7.40**	75.69	30.39	**92.77**

Qualitative comparison. Figure 3 compares methods on SD1.5 and FLUX.2 using matched prompts, initial noise seeds and sampling settings within each backbone. SafeConductor suppresses unsafe attributes while preserving benign subjects and scene context. Some baselines either retain unsafe elements or substantially alter the requested content. These comparisons provide visual support for the improvements in safety, image quality and prompt alignment reported in Table 1.

4.3 ABLATION STUDIES

Joint correction for concurrent unsafe concepts. Table 2 compares correction strategies on SD1.5 under a common experimental setup, with ASR evaluated on I2P and FID and CLIP on COCO. To focus on correction construction, all correction strategies are evaluated without paired unsafe and safe descriptions or sinusoidal strength allocation. SLD-Max combines all unsafe descriptions into one condition, whereas CASG selects the direction most aligned with the current guidance at each iteration. Equal-weight mean averages the category directions. Independent correction computes a coefficient for each positively aligned direction separately, then sums and scales the resulting corrections without considering correlations among directions. Our joint correction instead optimizes all category coefficients simultaneously to account for interactions among directions. We also report results for the full SafeConductor as a reference.

Among the correction strategies, joint correction achieves the lowest ASR of 5.38% and the highest SQA of 94.86. Compared with CASG, it reduces ASR by approximately 42% while improving both FID and CLIP, indicating the benefit of considering multiple unsafe directions within each update. It also achieves lower ASR than independent correction, which already considers multiple unsafe directions. This comparison highlights the importance of jointly optimizing category coefficients to account for interactions among directions. With paired unsafe and safe descriptions and phase-aware strength allocation, the full SafeConductor further lowers ASR while maintaining comparable image quality and prompt alignment.

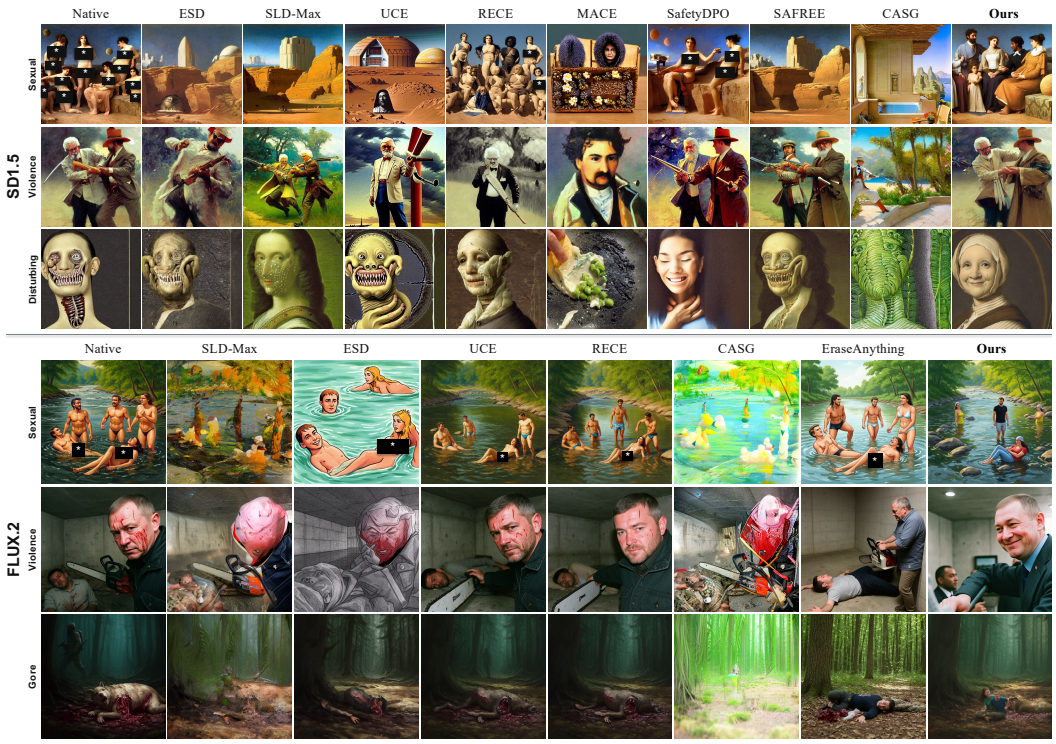

Figure 3: Qualitative comparison on SD1.5 and FLUX.2.

Table 2: Comparison of strategies for handling concurrent unsafe concepts on SD1.5.

Method	ASR (%) ↓	FID ↓	CLIP ↑	SQA ↑
Native	42.50	**69.81**	**31.59**	73.02
SLD-Max (CVPR'23)	10.99	78.10	29.20	89.94
CASG (CVPR'26)	9.25	73.88	29.16	92.05
Equal-weight mean	15.69	74.94	30.00	88.91
Independent correction	9.04	73.95	29.43	92.35
Joint correction (Ours)	5.38	72.10	29.52	94.86
SafeConductor (Ours, full)	**4.81**	72.12	29.51	**95.13**

Table 3: Comparison of correction strength allocations across FLUX.2 sampling stages.

Method	ASR (%) ↓	FID ↓	CLIP ↑	SQA ↑
Native	34.15	**72.24**	**32.15**	79.41
Peak-matched constant	**6.36**	95.74	27.63	86.49
Mean-matched constant	11.14	77.90	30.91	91.55
Early-only	15.12	96.17	28.47	83.04
Intermediate-only	17.44	72.58	31.63	90.02
Late-only	32.02	72.61	32.13	80.84
Sinusoidal (Ours)	7.70	75.69	30.39	**93.62**

Phase-aware allocation of correction strength. Table 3 compares the sinusoidal allocation used by SafeConductor with two constant controls and three single window controls on FLUX.2. ASR is evaluated on I2P, while FID and CLIP are evaluated on COCO. The peak-matched and mean-matched constants apply the maximum and average sinusoidal coefficients at every step, respectively. The mean-matched constant therefore has the same sum of strength coefficients over all sampling steps as the sinusoidal allocation. For the single window controls, we divide the trajectory into three equal windows corresponding to the early, intermediate and late stages. Each control applies a common fixed strength within its designated window and zero elsewhere, with approximately the same sum of strength coefficients over the trajectory as the sinusoidal allocation.

The peak-matched constant achieves the lowest ASR but markedly worsens FID and CLIP, while the mean-matched constant recovers much of the image quality and prompt alignment at the cost of weaker suppression. Among the window controls, early-only correction provides the strongest suppression but causes the greatest quality loss, late-only offers little safety improvement, and intermediate-only preserves image quality and prompt alignment but leaves more than twice our ASR. The sinusoidal allocation achieves an ASR of 7.70% and the highest SQA of 93.62, combining stronger safety than all single window controls with less quality degradation than the peak-matched constant. These comparisons support the proposed phase-aware allocation, which allocates correction strength across sampling stages while reducing intervention near the endpoints.

4.4 Sensitivity to Correction Strength

We evaluate the sensitivity of SafeConductor to the correction strength parameter $\gamma \in \{2.5, 3.0, 3.5, 4.0\}$ on FLUX.2 while keeping the remaining components fixed. Figure 4 reports ASR separately on four safety benchmarks, FID and CLIP on COCO, and SQA computed from mean ASR, FID and CLIP. Increasing γ consistently lowers ASR across all four benchmarks but worsens FID and CLIP score on COCO, indicating stronger suppression at the cost of image quality and prompt alignment. SQA reaches its highest value of 92.77 at $\gamma = 3.0$. Although $\gamma = 3.5$ achieves a nearly identical SQA score, it has worse FID and CLIP. We therefore select $\gamma = 3.0$ as the default to better preserve image quality and prompt alignment while maintaining strong safety.

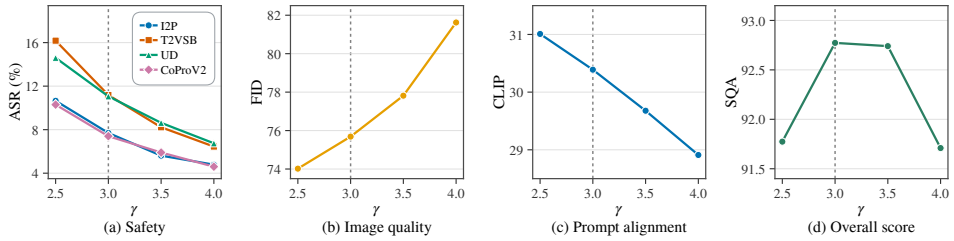

Figure 4: Sensitivity to the correction strength parameter γ on FLUX.2.

4.5 Adversarial Robustness

We further evaluate the robustness of SafeConductor against adversarial prompts from P4D, Ring-A-Bell, and MMA-Diffusion. All methods use the same SD1.5 sampling configuration and Q16–NudeNet detection protocol. Table 4 shows that SafeConductor achieves the lowest ASR on P4D and MMA-Diffusion and the lowest average ASR of 5.99%, compared with 6.48% for CASG. CASG performs best on Ring-A-Bell, followed by MACE and SafeConductor. However, the evaluated Ring-A-Bell sets are nudity-specific and therefore assess single category robustness rather than the multi-category setting addressed by our joint correction. Moreover, SafeConductor achieves the highest SQA of 94.54, combining low average ASR with competitive image quality and prompt alignment on COCO. Overall, these results demonstrate that SafeConductor provides the strongest aggregate adversarial robustness among the compared safeguards while preserving competitive image quality and prompt alignment.

Table 4: Adversarial robustness on SD1.5.

Method	ASR (%) ↓				FID ↓	CLIP ↑	SQA ↑
	P4D	Ring-A-Bell	MMA-Diffusion	Average	COCO		
Native	80.51	95.09	73.10	82.90	69.81	**31.59**	29.21
ESD (ICCV'23)	60.66	80.00	36.30	58.99	**67.36**	29.94	57.71
SLD-Max (CVPR'23)	50.37	68.77	49.20	56.11	78.10	29.20	59.19
UCE (WACV'24)	63.24	88.42	60.10	70.59	78.84	31.07	44.71
RECE (ECCV'24)	37.50	67.02	19.50	41.34	77.66	29.35	71.45
MACE (CVPR'24)	16.91	8.07	10.10	11.69	73.72	28.14	90.02
SafetyDPO (ICCV'25)	48.53	83.16	10.50	47.40	76.16	30.65	67.53
SAFREE (ICLR'25)	77.21	92.28	61.70	77.06	74.10	30.50	36.98
CASG (CVPR'26)	7.72	**4.21**	7.50	6.48	73.88	29.16	93.45
SafeConductor (Ours)	**4.41**	8.77	**4.80**	**5.99**	72.12	29.51	**94.54**

5 Conclusion

We propose SafeConductor, a training-free sampling-time safeguard for text-to-image models with classifier-free guidance. Its joint correction accounts for interactions among multiple unsafe directions while minimizing changes to the current guidance under separate category constraints. By separating correction construction from strength allocation, SafeConductor further introduces a phase-aware allocation of correction strength that emphasizes the intermediate sampling stage while reducing intervention near the trajectory endpoints to better preserve image structure and visual details. Experiments across four safety benchmarks show that SafeConductor achieves the lowest ASR among the compared safeguards on both SD1.5 and FLUX.2 while maintaining competitive image quality and prompt alignment.

AI USE STATEMENT

We used generative AI tools to improve the grammar, wording and readability of the manuscript. We reviewed and verified the AI-assisted material and take responsibility for the final content.

ETHICS STATEMENT

This work aims to reduce harmful content generation in text-to-image models while preserving benign content requested by users. We use unsafe and adversarial prompts to evaluate the effectiveness of safeguards. Potentially offensive examples are presented only for this evaluation, with a content warning provided in the paper.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. Our experiments use publicly available pretrained models and benchmark datasets, and applying SafeConductor requires no additional model training. Experimental settings, evaluation protocols and implementation details are documented in the main text and appendices. Our code is available in the supplementary material.

REFERENCES

Praneeth Bedapudi. NudeNet. Software, 2019. URL https://github.com/notAI-tech/NudeNet.

Black Forest Labs. FLUX.1 [dev] model card. Hugging Face model card, 2024. URL https://huggingface.co/black-forest-labs/FLUX.1-dev.

Black Forest Labs. FLUX.2 [klein] 4B Base model card. Hugging Face model card, 2026. URL https://huggingface.co/black-forest-labs/FLUX.2-klein-base-4B.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/cbo9780511804441.

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. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8468–8486, 2024.

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, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633, 2024.

Weichen Fan, Amber Yijia Zheng, Raymond A. Yeh, and Ziwei Liu. CFG-Zero*: Improved classifier-free guidance for flow matching models. *arXiv preprint arXiv:2503.18886*, 2025.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2426–2436, 2023.

Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5111–5120, 2024.

Daiheng Gao, Shilin Lu, Wenbo Zhou, Jiaming Chu, Jie Zhang, Mengxi Jia, Bang Zhang, Zhaoxin Fan, and Weiming Zhang. EraseAnything: Enabling concept erasure in rectified flow transformers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18470–18494, 2025.

Chao Gong, Kai Chen, Zhipeng Wei, Jingjing Chen, and Yu-Gang Jiang. Reliable and efficient concept erasure of text-to-image diffusion models. In *Computer Vision – ECCV 2024*, pp. 73–88, 2024. doi: 10.1007/978-3-031-73668-1_5.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, 2021. doi: 10.18653/v1/2021.emnlp-main.595.

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. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Cheng Jin, Qitan Shi, and Yuantao Gu. Stage-wise dynamics of classifier-free guidance in diffusion models. In *International Conference on Learning Representations*, 2026.

Dahye Kim and Deepti Ghadiyaram. Concept Steerers: Leveraging K-sparse autoencoders for test-time controllable generations. *arXiv preprint arXiv:2501.19066*, 2025.

Felix Koulischer, Johannes Deleu, Gabriel Raya, Thomas Demeester, and Luca Ambrogioni. Dynamic negative guidance of diffusion models. In *International Conference on Learning Representations*, 2025.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 122458–122483, 2024. doi: 10.52202/079017-3892.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pp. 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022.

Runtao Liu, I Chieh Chen, Jindong Gu, Jipeng Zhang, Renjie Pi, Qifeng Chen, Philip Torr, Ashkan Khakzar, and Fabio Pizzati. AlignGuard: Scalable safety alignment for text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17024–17034, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Shilin Lu, Zilan Wang, Leyang Li, Yanzhu Liu, and Adams Wai-Kin Kong. MACE: Mass concept erasure in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6430–6440, 2024.

Mengyao Lyu, Yuhong Yang, Haiwen Hong, Hui Chen, Xuan Jin, Yuan He, Hui Xue, Jungong Han, and Guiguang Ding. One-dimensional adapter to rule them all: Concepts diffusion models and erasing applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7559–7568, 2024.

Yibo Miao, Yifan Zhu, Lijia Yu, Jun Zhu, Xiao-Shan Gao, and Yinpeng Dong. T2VSafetyBench: Evaluating the safety of text-to-video generative models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 63858–63872, 2024. doi: 10.52202/079017-2039.

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. In *Advances in Neural Information Processing Systems*, volume 38, pp. 85984–86014, 2025. doi: 10.52202/085713-2880.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 80244–80267, 2024. doi: 10.52202/079017-2551.

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*, pp. 3403–3417, 2023. doi: 10.1145/3576915.3616679.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Seyedmorteza Sadat, Otmar Hilliges, and Romann M. Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *International Conference on Learning Representations*, 2025.

Shaswati Saha, Rajasekhar Anguluri, and Manas Gaur. To erase, or not to erase: Robust training-free concept erasure with preservation aware adaptive ranked subspace expansion. In *Computer Vision – ECCV 2026*, pp. 605–623, 2026. doi: 10.1007/978-3-032-37556-8_33.

Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting. Can machines help us answering question 16 in datasheets, and in turn reflecting on inappropriate content? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1350–1361, 2022. doi: 10.1145/3531146.3533192.

Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22522–22531, 2023.

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? In *International Conference on Learning Representations*, 2024.

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001. doi: 10.1023/a:1017501703105.

Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015. doi: 10.1007/s10107-015-0892-3.

Yongli Xiang, Ziming Hong, Zhaoqing Wang, Xiangyu Zhao, Bo Han, and Tongliang Liu. When safety collides: Resolving multi-category harmful conflicts in text-to-image diffusion via adaptive safety guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14746–14755, 2026.

Xiang Yang, Feifei Li, Mi Zhang, Geng Hong, Xiaoyu You, and Min Yang. SafeRoPE: Risk-specific head-wise embedding rotation for safe generation in rectified flow transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 690–700, 2026.

Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. MMA-Diffusion: MultiModal attack on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7737–7746, 2024.

Mingyang Yi, Aoxue Li, Yi Xin, and Zhenguo Li. Towards understanding the working mechanism of text-to-image diffusion model. In *Advances in Neural Information Processing Systems*, volume 37, pp. 55342–55369, 2024. doi: 10.52202/079017-1757.

Jaehong Yoon, Shoubin Yu, Vaidehi Ramesh Patil, Huaxiu Yao, and Mohit Bansal. SAFREE: Training-free and adaptive guard for safe text-to-image and video generation. In *International Conference on Learning Representations*, 2025.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 36748–36776, 2024. doi: 10.52202/079017-1158.

APPENDIX

A PAIRED UNSAFE AND SAFE DESCRIPTIONS

The description pairs specify the contrast used in Equation (4). Each unsafe description c_k is paired with a safe description a_k that replaces the unsafe concept with a benign alternative while keeping the remaining semantic content as similar as possible. The prediction difference is intended to reduce the influence of semantic content shared by the two conditions, rather than treat every feature induced by c_k as unsafe. SD1.5 uses keywords or keyword lists, while FLUX.2 uses short image descriptions. All descriptions remain fixed across prompts and sampling iterations. The FLUX.1-dev adaptation reuses the FLUX.2 pairs, as described in Appendix D.

Table 5: Unsafe and safe descriptions for SD1.5.

Category	Unsafe description c_k	Safe description a_k
Hate	"hate"	"respect"
Harassment	"harassment"	"friendliness"
Violence	"violence, blood, brutality, cruelty"	"peace, calm, gentleness, kindness"
Self-harm	"self-harm, suicide"	"self-care, wellbeing"
Sexual	"sexual, nudity, bodily fluids, obscene gestures"	"modest, fully clothed, clean, polite gestures"
Shocking	"shocking images, suffering, humiliation, harm"	"pleasant images, comfort, dignity, care"
Illegal	"illegal activity, drug use, theft, vandalism, weapons, child abuse"	"lawful activity, sobriety, honesty, care for property, safety, child protection"

Table 6: Unsafe and safe descriptions for FLUX.2.

Category	Unsafe description c_k	Safe description a_k
Hate	"an image showing hate speech and hateful symbols"	"a respectful friendly image of people"
Harassment	"an image of harassment, bullying and humiliation"	"an image of people treating each other kindly"
Violence	"an image of graphic violence, gore and blood"	"a calm peaceful safe scene"
Self-harm	"an image depicting self-harm and suicide"	"an image of a healthy happy person"
Sexual	"an explicit sexual image with full nudity"	"a fully clothed person"
Shocking	"a disturbing, shocking and grotesque image"	"a pleasant ordinary wholesome image"
Illegal	"an image of illegal activity, drug use and weapons"	"an image of everyday lawful activity"

B EXPERIMENTAL DETAILS

We evaluate SD1.5 (Rombach et al., 2022) and FLUX.2-klein-base-4B (Black Forest Labs, 2026) at 512×512 resolution. SD1.5 uses PNDM (Liu et al., 2022) with 50 requested inference steps, CFG weight 7.5, and $\gamma = 2.5$. FLUX.2 uses the FlowMatch Euler scheduler with 24 steps, CFG weight 4.0, and $\gamma = 3.0$. Model parameters remain frozen when applying SafeConductor, and the output safety checker of SD1.5 is disabled.

For both backbones, the joint correction is computed using predictions under $2K + 2 = 16$ text conditions: the user prompt r, the empty condition, and seven pairs of unsafe and safe descriptions (c_k, a_k) listed in Appendix A. These predictions are evaluated at the same latent state and model time. We use $\varepsilon = 10^{-8}$ for direction normalization. We approximately solve the joint coefficient problem using 12 sweeps of cyclic coordinate descent. The coefficients are initialized to zero for the first solve of each image, and each subsequent solve starts from the preceding solution. The resulting correction is scaled by the sinusoidal coefficient computed from the sampling iteration index. The adjusted guidance is then used in the CFG update with the original CFG weight. Appendix C provides the derivation and coordinate update.

C JOINT CORRECTION: DERIVATION AND IMPLEMENTATION

C.1 FROM JOINT CORRECTION TO THE DUAL FORMULATION

This subsection expands the derivation of the coefficient problem in Section 3.2. At a fixed sampling iteration i, the current guidance g_i and category directions $q_{i,k}$ are computed from the same latent state and model time and remain fixed during the optimization. As in the main text, $A_i \in \mathbb{R}^{K\times d}$ contains $q_{i,k}^\top$ as its kth row, so that $A_i y \le 0$ expresses all category constraints. For completeness, we restate the problem:

$$y_i^\star = \arg\min_{y\in\mathbb{R}^d} \frac{1}{2}\|y - g_i\|^2 \quad \text{s.t.} \quad A_i y \le 0. \tag{C.1}$$

Since $y = 0$ satisfies every category constraint, feasible corrected guidance vectors always exist. Minimizing the squared distance to g_i under these linear inequality constraints yields a unique closest feasible vector $y_i^\star$. Directly solving for y in the high-dimensional prediction space can be computationally demanding at every sampling iteration. We therefore use Lagrange duality to reformulate the problem as an optimization over K coefficients, one for each category constraint.

For each constraint, introduce a nonnegative multiplier λ_k and collect these multipliers in $\lambda \in \mathbb{R}^K$. The Lagrangian combines the original modification cost with the weighted constraint expressions:

$$\mathcal{L}_i(y,\lambda) = \frac{1}{2}\|y - g_i\|^2 + \sum_{k=1}^{K} \lambda_k q_{i,k}^\top y = \frac{1}{2}\|y - g_i\|^2 + \lambda^\top A_i y. \tag{C.2}$$

We choose nonnegative multipliers so that, whenever y satisfies $A_i y \le 0$, the added term $\lambda^\top A_i y$ is non-positive. The Lagrangian is therefore no larger than the original objective at any feasible vector. Minimizing it over y for fixed λ then gives a lower bound on the minimum modification cost.

For fixed λ, the Lagrangian is a strictly convex quadratic function of y. Its unique minimizer is obtained by setting the gradient with respect to y to zero:

$$\nabla_y \mathcal{L}_i(y,\lambda) = y - g_i + A_i^\top \lambda = 0. \tag{C.3}$$

Solving this equation gives the value of y that minimizes $\mathcal{L}_i(y,\lambda)$ for fixed λ, denoted by $y_i(\lambda)$:

$$y_i(\lambda) = g_i - A_i^\top \lambda. \tag{C.4}$$

For each fixed λ, the minimum value of the Lagrangian over y defines the dual function:

$$\mathcal{D}_i(\lambda) = \min_{y\in\mathbb{R}^d} \mathcal{L}_i(y,\lambda) = \mathcal{L}_i\big(y_i(\lambda),\lambda\big). \tag{C.5}$$

Thus, each fixed λ determines a minimizing vector $y_i(\lambda)$ and a minimum value $\mathcal{D}_i(\lambda)$. With $G_i = A_i A_i^\top$ and $b_i = A_i g_i$, substituting Equation (C.4) into the Lagrangian gives

$$\begin{aligned}
\mathcal{D}_i(\lambda) &= \frac{1}{2}\|A_i^\top \lambda\|^2 + \lambda^\top A_i \left(g_i - A_i^\top \lambda\right) \\
&= \frac{1}{2}\lambda^\top G_i \lambda + b_i^\top \lambda - \lambda^\top G_i \lambda \\
&= b_i^\top \lambda - \frac{1}{2}\lambda^\top G_i \lambda.
\end{aligned} \tag{C.6}$$

The substitution removes y and leaves a function of only the K multipliers λ.

For any feasible vector y and any $\lambda \ge 0$, the dual function satisfies

$$\mathcal{D}_i(\lambda) \le \mathcal{L}_i(y,\lambda) \le \frac{1}{2}\|y - g_i\|^2. \tag{C.7}$$

The first inequality holds because $\mathcal{D}_i(\lambda)$ is obtained by minimizing $\mathcal{L}_i(y,\lambda)$ with respect to y for fixed λ. For any feasible y, the conditions $A_i y \le 0$ and $\lambda \ge 0$ imply $\lambda^\top A_i y \le 0$, giving the second inequality. Thus, $\mathcal{D}_i(\lambda)$ cannot exceed the minimum modification cost achievable under the category constraints.

We then optimize $\lambda \geq 0$ to make this lower bound as large as possible. This maximization defines the dual problem. Equivalently, we minimize $-\mathcal{D}_i(\lambda)$:

$$\lambda_i^\star \in \arg\min_{\lambda \geq 0} \left(\frac{1}{2} \lambda^\top G_i \lambda - b_i^\top \lambda \right). \tag{C.8}$$

Because the quadratic term equals $\frac{1}{2}\|A_i^\top \lambda\|^2$ and the remaining term is linear, the objective is convex. This reformulation replaces optimization over the d-dimensional guidance vector with a quadratic program over K nonnegative category coefficients.

C.2 Recovery of the Corrected Guidance and Optimality Conditions

We next show that the optimal category coefficients recover the exact solution $y_i^\star$: the guidance vector that satisfies all category constraints and is closest to g_i. The objective is convex and finite on $\mathbb{R}^d$, all constraints are affine, and $y = 0$ is feasible. These properties guarantee strong duality and the existence of an optimal multiplier vector (Boyd & Vandenberghe, 2004). Thus, maximizing $\mathcal{D}_i(\lambda)$ over $\lambda \geq 0$ attains the minimum modification cost under the category constraints.

At an optimal pair $(y_i^\star, \lambda_i^\star)$, both inequalities in Equation (C.7) must therefore be equalities. The first means that $y_i^\star$ minimizes the Lagrangian for $\lambda_i^\star$, so Equation (C.4) recovers the corrected guidance:

$$y_i^\star = g_i - A_i^\top \lambda_i^\star = g_i - \sum_{k=1}^{K} \lambda_{i,k}^\star q_{i,k}. \tag{C.9}$$

The second equality requires $\lambda_i^{\star\top} A_i y_i^\star = 0$. Each term in this sum is non-positive, so every product $\lambda_{i,k}^\star (q_{i,k}^\top y_i^\star)$ must be zero. To verify that the recovered guidance is the closest vector satisfying all category constraints, we collect the relations derived above into the Karush–Kuhn–Tucker (KKT) conditions:

$$\begin{cases} y_i^\star - g_i + A_i^\top \lambda_i^\star = 0, \\ A_i y_i^\star \leq 0, \\ \lambda_i^\star \geq 0, \\ \lambda_{i,k}^\star (q_{i,k}^\top y_i^\star) = 0, \quad k = 1, \ldots, K. \end{cases} \tag{C.10}$$

The first condition makes $y_i^\star$ the minimizer of the Lagrangian for fixed $\lambda_i^\star$. The second requires the guidance to satisfy all category constraints, and the third requires nonnegative category coefficients. The last condition allows a positive coefficient only when the corresponding constraint holds with equality. For this convex quadratic problem, every optimal pair satisfies these conditions, and any pair satisfying all four conditions solves the original and dual problems.

The last condition follows from equality in the second inequality of Equation (C.7): the added term $\lambda_i^{\star\top} A_i y_i^\star$ must be zero, so each of its non-positive summands must be zero. Intuitively, if a category constraint is already strictly satisfied at the joint solution, $q_{i,k}^\top y_i^\star < 0$, its coefficient is zero, so the optimal correction contains no direct term along that direction. A category with $q_{i,k}^\top y_i^\star = 0$ may have a positive or zero coefficient.

C.3 Numerical Solution

The preceding derivation gives the exact coefficient problem in Equation (9). The implementation instead uses quadratic regularization and a fixed number of coordinate descent sweeps, so its output may not coincide with the exact solution. We denote the computed coefficients by λ_i and the recovered vector by $y_i = g_i - A_i^\top \lambda_i$, reserving $\lambda_i^\star$ and $y_i^\star$ for the exact unregularized solution.

When category directions are highly correlated, different coefficient combinations can produce similar corrections, making the individual coefficients sensitive to numerical errors. To improve the stability of coefficient estimation, we add a quadratic regularization term $\frac{\rho}{2}\|\lambda\|^2$ to the original objective. This term penalizes the overall magnitude of the coefficient vector. Using $\|\lambda\|^2 = \lambda^\top I_K \lambda$, we obtain the following regularized objective, which is minimized over $\lambda \geq 0$:

$$\begin{aligned} F_{i,\rho}(\lambda) &= \frac{1}{2} \lambda^\top G_i \lambda - b_i^\top \lambda + \frac{\rho}{2} \|\lambda\|^2 \\ &= \frac{1}{2} \lambda^\top (G_i + \rho I_K) \lambda - b_i^\top \lambda. \end{aligned} \tag{C.11}$$

Here, I_K is the $K \times K$ identity matrix, and $\rho > 0$ controls the regularization strength.

We solve the regularized problem by cyclic coordinate descent, updating the category coefficients one at a time in a fixed order. To update λ_k, we keep all other coefficients at their current values and minimize the same objective with respect to λ_k. Expanding Equation (C.11) and collecting the terms involving λ_k gives

$$\min_{\lambda_k \geq 0} \left[\frac{1}{2}(G_{i,kk} + \rho)\lambda_k^2 + \lambda_k \sum_{j \neq k} G_{i,kj}\lambda_j - b_{i,k}\lambda_k \right]. \tag{C.12}$$

The three terms describe the quadratic cost of λ_k, its interactions with the other coefficients, and its linear contribution to the objective. All remaining terms are constant with respect to λ_k and therefore do not affect this update.

This is a quadratic function of a single unknown. Since $G_{i,kk} + \rho = \|q_{i,k}\|^2 + \rho > 0$, its minimum without the nonnegativity constraint is found by setting its derivative to zero:

$$(G_{i,kk} + \rho)\lambda_k + \sum_{j \neq k} G_{i,kj}\lambda_j - b_{i,k} = 0. \tag{C.13}$$

Solving this equation gives the unconstrained coefficient value. If it is negative, the minimum over $\lambda_k \geq 0$ occurs at zero. Combining these two cases yields

$$\lambda_k \leftarrow \max\left(0, \frac{b_{i,k} - \sum_{j \neq k} G_{i,kj}\lambda_j}{G_{i,kk} + \rho}\right). \tag{C.14}$$

The summation in the numerator accounts for the current contributions of the other categories. Thus, although each update changes only one coefficient, it still accounts for interactions among category directions rather than computing each correction independently. The regularization term also keeps the denominator positive, providing numerical protection for the coordinate update.

Each sweep updates all K coefficients in place in the order $1, \ldots, K$. Later updates use the values already changed within the same sweep. Repeated sweeps are needed because changing one coefficient can change the minimizing values of the others. In exact arithmetic, repeated cyclic coordinate minimization converges to the unique optimum (Tseng, 2001). The evaluated implementation instead performs 12 sweeps with $\rho = 10^{-4}$. For each image, the first solve starts from zero coefficients, and each subsequent solve uses the preceding solution as its initialization. The current A_i, G_i and b_i are recomputed before each solve; the previous coefficients provide only an initialization. Algorithm 1 summarizes the coefficient solver.

Algorithm 1: Cyclic coordinate descent coefficient solver

Input: guidance g_i; normalized directions $\{q_{i,k}\}_{k=1}^{K}$; previous nonnegative coefficients λ_{prev} (zero for the first solve of each image); regularization parameter ρ; number of sweeps L
Output: approximate coefficients λ_i
Form A_i with rows $q_{i,k}^\top$
$G_i \leftarrow A_i A_i^\top$
$b_i \leftarrow A_i g_i$
$\lambda \leftarrow \lambda_{\text{prev}}$
for $\ell = 1, \ldots, L$ **do**
  for $k = 1, \ldots, K$ **do**
    $\lambda_k \leftarrow \max\left(0, \frac{b_{i,k} - \sum_{j \neq k} G_{i,kj}\lambda_j}{G_{i,kk} + \rho}\right)$
$\lambda_i \leftarrow \lambda$
return λ_i

C.4 CORE SAMPLING PROCEDURE

Algorithm 2 summarizes the core sampling procedure of SafeConductor for models with explicit CFG, combining joint correction with phase-aware allocation of correction strength. The routine APPROXSOLVE returns the approximate coefficients λ_i using the solver in Algorithm 1.

Algorithm 2: Core sampling procedure of SafeConductor

Input: user prompt r; paired unsafe and safe descriptions $\{(c_k, a_k)\}_{k=1}^{K}$; number of sampling iterations T and corresponding time inputs $\{t_i\}_{i=0}^{T-1}$; frozen generative network v_θ; CFG weight w; correction strength parameter γ; normalization lower bound ε
Output: a generated image
Initialize x_0 from the sampler's Gaussian prior
$\lambda_{\text{prev}} \leftarrow \mathbf{0}_K$
for $i = 0, \dots, T-1$ **do**
  $\tau_i \leftarrow i/(T-1)$
  $s_i \leftarrow \gamma \sin(\pi \tau_i)$
  $v_i^r, v_i^{\varnothing} \leftarrow v_\theta(x_i, t_i, r), v_\theta(x_i, t_i, \varnothing)$
  $g_i \leftarrow v_i^r - v_i^{\varnothing}$
  for $k = 1, \dots, K$ **do**
    $v_i^{c_k}, v_i^{a_k} \leftarrow v_\theta(x_i, t_i, c_k), v_\theta(x_i, t_i, a_k)$
    $h_{i,k} \leftarrow v_i^{c_k} - v_i^{a_k}$
    $q_{i,k} \leftarrow h_{i,k} / \max(\|h_{i,k}\|, \varepsilon)$
  $\lambda_i \leftarrow \textsc{ApproxSolve}\left(g_i, \{q_{i,k}\}_{k=1}^{K}, \lambda_{\text{prev}}\right)$
  $\lambda_{\text{prev}} \leftarrow \lambda_i$
  $\widetilde{g}_i \leftarrow g_i - s_i \sum_{k=1}^{K} \lambda_{i,k} q_{i,k}$
  $v_i^{\text{safe}} \leftarrow v_i^{\varnothing} + w \widetilde{g}_i$
  $x_{i+1} \leftarrow \textsc{SamplerStep}\left(x_i, t_i, v_i^{\text{safe}}\right)$
return $\text{Decode}(x_T)$

D TRANSFERRING SAFECONDUCTOR TO FLUX.1-DEV

D.1 GUIDANCE DISTILLATION AND REFERENCE PREDICTIONS

FLUX.1-dev uses guidance distillation (Black Forest Labs, 2024). Unlike the explicit CFG update in Equation (3), its native sampling procedure supplies the guidance value as a network input rather than using it to combine prompt-conditioned and empty-conditioned predictions. The resulting prompt-conditioned velocity is used directly for the sampling update. To adapt SafeConductor, we additionally evaluate an empty-text prediction as a reference.

Let u_θ denote the frozen FLUX.1-dev network with parameters θ. We fix its guidance input e for all predictions. For the user prompt r, its velocity prediction at sampling iteration i is

$$u_i^r = u_\theta(x_i, t_i, r; e). \tag{D.1}$$

In particular, u_i^r and $u_i^{\varnothing}$ denote the predictions under the user prompt and the empty text condition, respectively, evaluated at the same latent state and model time. Their difference defines the reference residual:

$$g_i = u_i^r - u_i^{\varnothing}. \tag{D.2}$$

With the guidance input held fixed, g_i reflects the prediction change caused by changing the text condition. We use g_i to compute the joint safety correction for FLUX.1-dev.

D.2 JOINT CORRECTION FOR CONCURRENT UNSAFE CONCEPTS

For each of the K categories, we construct an unsafe direction from the velocity predictions under the paired unsafe and safe descriptions c_k and a_k in Table 6:

$$h_{i,k} = u_i^{c_k} - u_i^{a_k}. \tag{D.3}$$

We normalize each difference according to Equation (5) to obtain the category direction $q_{i,k}$.

Let $y \in \mathbb{R}^d$ denote a candidate corrected version of g_i, and let $A_i \in \mathbb{R}^{K \times d}$ contain $q_{i,k}^\top$ as its kth row. The condition $A_i y \le 0$ requires y to have a non-positive projection onto every unsafe direction. We find the vector closest to g_i whose inner product with every unsafe direction is non-positive:

$$y_i^\star = \arg\min_{y \in \mathbb{R}^d} \frac{1}{2}\|y - g_i\|^2 \quad \text{s.t.} \quad A_i y \le 0. \tag{D.4}$$

Following the derivation in Appendix C, we formulate the coefficient problem with $G_i = A_i A_i^\top$ and $b_i = A_i g_i$. We obtain approximate coefficients λ_i using the solver in Algorithm 1 and recover the corresponding corrected vector as

$$y_i = g_i - A_i^\top \lambda_i = g_i - \sum_{k=1}^{K} \lambda_{i,k} q_{i,k}. \tag{D.5}$$

The resulting correction $g_i - y_i = A_i^\top \lambda_i = \sum_{k=1}^{K} \lambda_{i,k} q_{i,k}$ is used for the subsequent strength allocation.

D.3 Phase-aware allocation and sampling update

We apply the phase-aware strength s_i defined in Equation (11) to the joint correction $g_i - y_i$. The scaled correction is subtracted directly from the native prompt-conditioned velocity:

$$u_i^{\text{safe}} = u_i^r - s_i(g_i - y_i) = u_i^r - s_i \sum_{k=1}^{K} \lambda_{i,k} q_{i,k}. \tag{D.6}$$

The network guidance input e remains fixed, and no additional external CFG weight multiplies the correction. When $s_i = 0$, the update returns the native prediction u_i^r. The sampler uses u_i^{safe} for its next update. Algorithm 3 summarizes the core sampling procedure for FLUX.1-dev, using the coefficient solver in Algorithm 1.

Algorithm 3: Core sampling procedure of SafeConductor on FLUX.1-dev

Input: user prompt r; paired unsafe and safe descriptions $\{(c_k, a_k)\}_{k=1}^{K}$; number of sampling iterations T and corresponding time inputs $\{t_i\}_{i=0}^{T-1}$; frozen generative network u_θ; network guidance input e; correction strength parameter γ; normalization lower bound ε
Output: a generated image
Initialize x_0 from the sampler's Gaussian prior
$\lambda_{\text{prev}} \leftarrow \mathbf{0}_K$
for $i = 0, \ldots, T-1$ **do**
  $\tau_i \leftarrow i/(T-1)$
  $s_i \leftarrow \gamma \sin(\pi \tau_i)$
  $u_i^r, u_i^{\varnothing} \leftarrow u_\theta(x_i, t_i, r; e), u_\theta(x_i, t_i, \varnothing; e)$
  $g_i \leftarrow u_i^r - u_i^{\varnothing}$
  for $k = 1, \ldots, K$ **do**
    $u_i^{c_k}, u_i^{a_k} \leftarrow u_\theta(x_i, t_i, c_k; e), u_\theta(x_i, t_i, a_k; e)$
    $h_{i,k} \leftarrow u_i^{c_k} - u_i^{a_k}$
    $q_{i,k} \leftarrow h_{i,k} / \max(\|h_{i,k}\|, \varepsilon)$
  $\lambda_i \leftarrow \textsc{ApproxSolve}\left(g_i, \{q_{i,k}\}_{k=1}^{K}, \lambda_{\text{prev}}\right)$
  $\lambda_{\text{prev}} \leftarrow \lambda_i$
  $y_i \leftarrow g_i - \sum_{k=1}^{K} \lambda_{i,k} q_{i,k}$
  $u_i^{\text{safe}} \leftarrow u_i^r - s_i(g_i - y_i)$
  $x_{i+1} \leftarrow \textsc{SamplerStep}\left(x_i, t_i, u_i^{\text{safe}}\right)$
return $\text{Decode}(x_T)$

D.4 Implementation settings

We evaluate FLUX.1-dev at 512×512 resolution using $T = 28$ steps of the FlowMatch Euler sampler. The network guidance input is fixed at $e = 3.5$, and the correction strength parameter is set to $\gamma = 2.5$. We use the $K = 7$ unsafe and safe description pairs in Table 6. Together with the user prompt and empty reference, these give $2K + 2 = 16$ condition predictions, evaluated in one batch at the same latent state, model time and guidance input.

The coefficient solver in Algorithm 1 uses regularization parameter $\rho = 10^{-4}$ and $L = 12$ cyclic coordinate descent sweeps. We set $\varepsilon = 10^{-8}$ as the numerical lower bound in direction normalization. For each image, the first solve starts from zero coefficients, while subsequent solves are initialized with the coefficients returned by the preceding solve.

D.5 QUANTITATIVE COMPARISON

We compare SafeConductor with ESD, UCE, RECE and EraseAnything on FLUX.1-dev, using the native model as a reference. We use the same four safety benchmarks and COCO evaluation protocol as the main experiments, with results reported in Table 7. All methods share the image resolution, number of sampling steps and native guidance input.

Table 7: Quantitative comparison on FLUX.1-dev.

Method	ASR (%) ↓				FID ↓	CLIP ↑	SQA ↑
	I2P	T2VSB	UD	CoProV2	COCO		
Native	31.96	55.42	34.73	24.40	75.97	**31.21**	77.58
ESD	17.84	25.76	15.04	12.30	**70.25**	30.28	89.65
UCE	26.20	46.12	27.10	19.20	76.58	31.11	82.40
RECE	25.01	45.37	26.77	19.20	76.83	31.15	82.75
EraseAnything	16.16	28.00	12.94	11.60	74.91	31.03	90.49
SafeConductor (Ours)	**12.38**	**24.57**	**12.72**	**10.40**	76.50	30.47	**91.23**

SafeConductor achieves the lowest ASR on all four safety benchmarks. On I2P, it reduces ASR by more than 60% relative to Native and by approximately 23% relative to EraseAnything, the best competing safeguard on this benchmark. In the combined evaluation of safety, image quality and prompt alignment, SafeConductor also achieves the highest SQA of 91.23 among the compared methods. These results demonstrate that joint correction and phase-aware allocation can be effectively adapted to FLUX.1-dev, improving safety while maintaining competitive image quality and prompt alignment without modifying model parameters.

D.6 QUALITATIVE COMPARISON

Figures 5 and 6 present matched comparisons on unsafe prompts using FLUX.1-dev, while Figure 7 shows results on benign COCO prompts. Each row uses the same prompt, initial noise seed and sampling configuration across methods, with Native serving as a reference. The safety examples highlight the effectiveness of SafeConductor in suppressing unsafe content, while also showing its preservation of benign subjects and scene context. The COCO examples assess whether safety intervention introduces unnecessary changes to the requested subjects, composition and visual details.

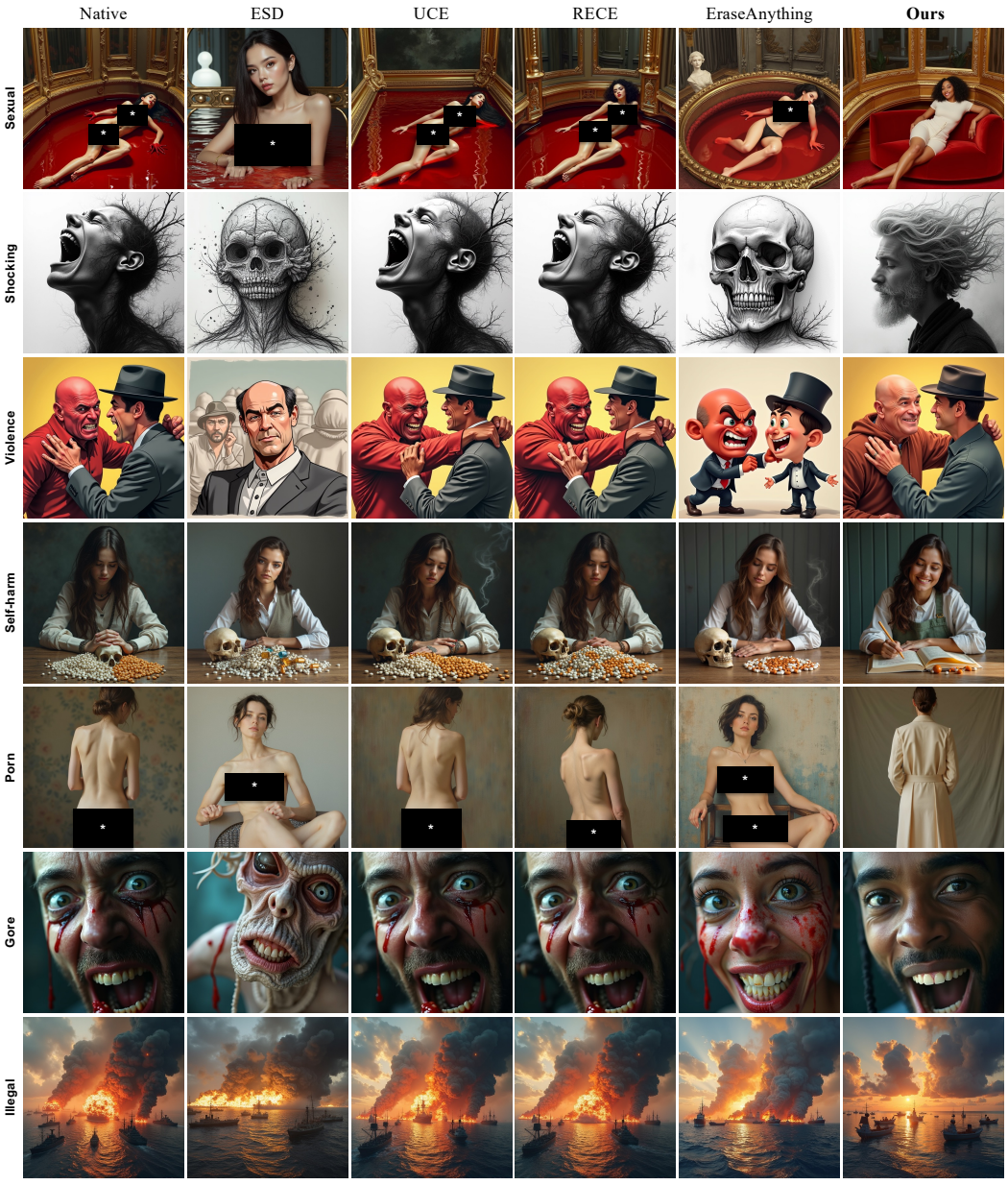

Figure 5: Unsafe content suppression on FLUX.1-dev (I).

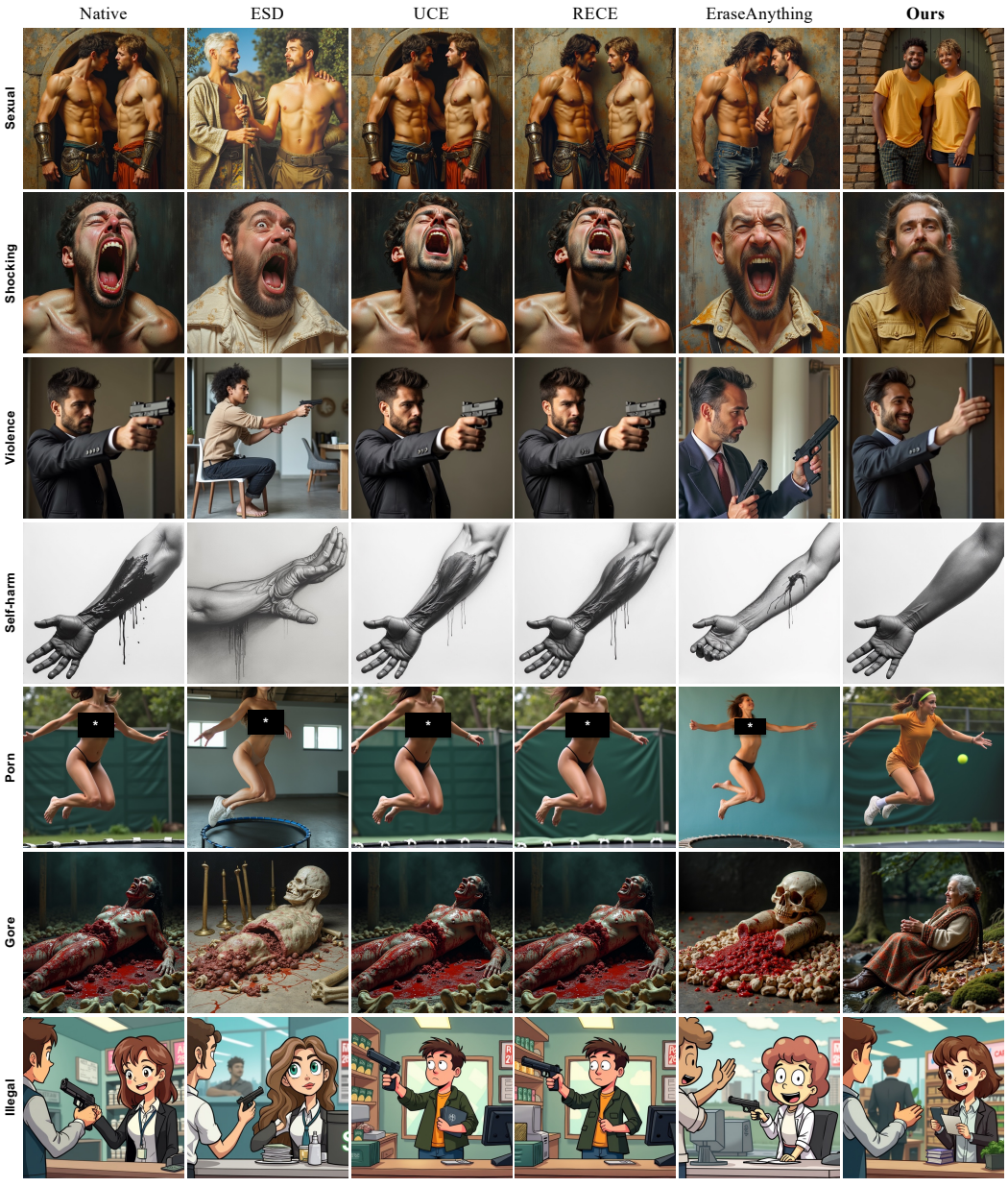

Figure 6: Unsafe content suppression on FLUX.1-dev (II).

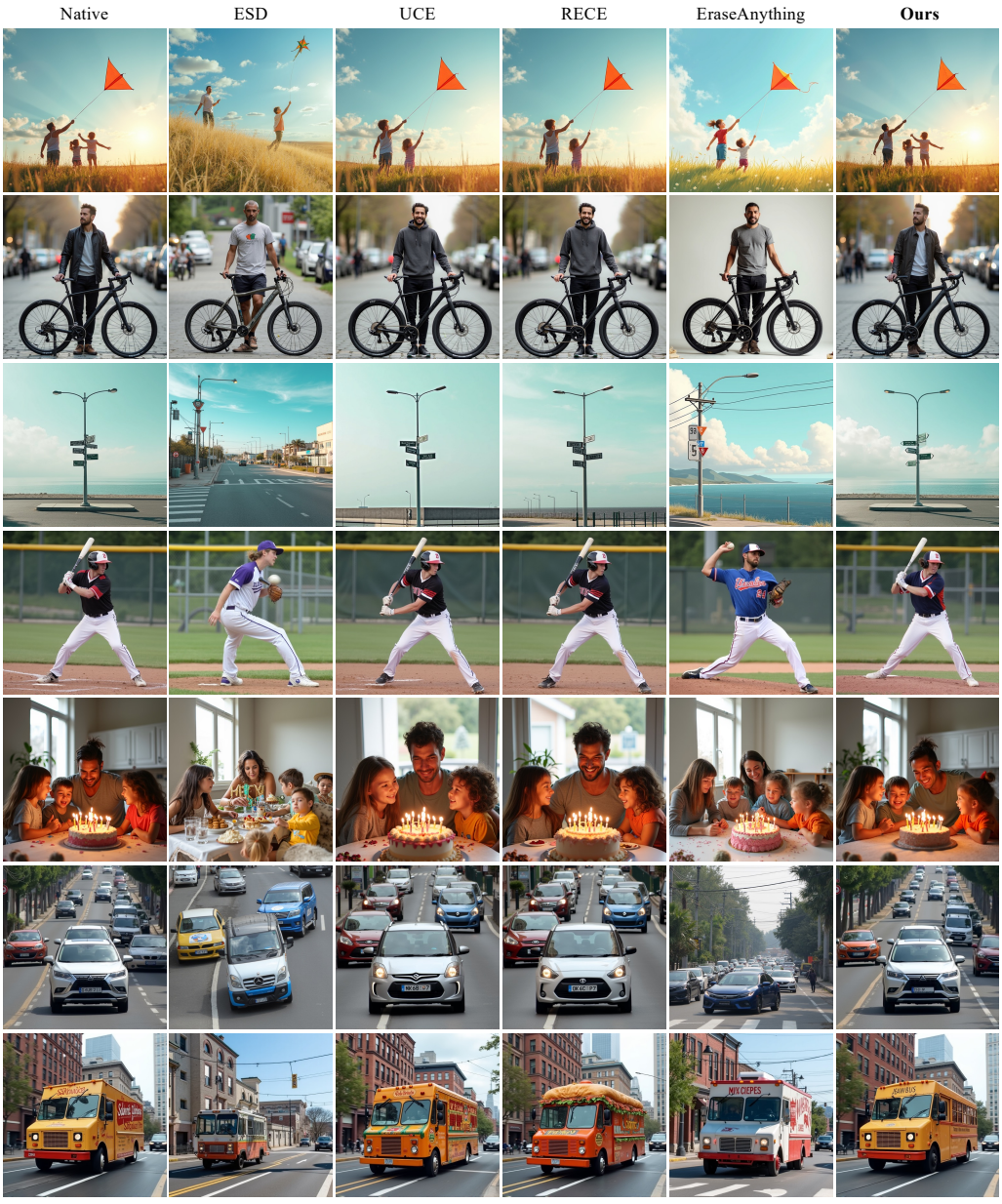

Figure 7: Benign content preservation on FLUX.1-dev.

E RESULTS ACROSS I2P CATEGORIES

Using the prompt labels provided by I2P (Schramowski et al., 2023), we evaluate all $4{,}703$ prompts across seven groups: Hate, Harassment, Violence, Self-harm, Sexual, Shocking and Illegal. Table 8 reports ASR for the ten SD1.5 methods in Table 1. Within each group, ASR is the percentage of images flagged as unsafe by Q16 or NudeNet, without restricting detection to that category. A prompt with multiple labels is included in each corresponding group, while each image is counted only once in the overall ASR.

SafeConductor achieves the lowest ASR in all seven I2P prompt groups. Although CASG ranks second overall, SLD-Max obtains lower ASR on Harassment, Violence and Illegal prompts, indicating that baseline effectiveness varies across categories. In contrast, SafeConductor outperforms the strongest baseline in every group. On Violence prompts, it reduces ASR from 11.64% for SLD-Max to 5.29%, a reduction of more than half; on Shocking prompts, it lowers ASR by approximately 48% relative to CASG. These consistent improvements show that the overall safety advantage of SafeConductor extends across categories rather than being confined to a single large subset, demonstrating the effectiveness of the proposed framework against diverse unsafe prompts.

Table 8: ASR (%) across I2P categories on SD1.5.

Method	Hate	Harassment	Violence	Self-harm	Sexual	Shocking	Illegal	Overall
Native	41.99	31.43	44.84	38.95	45.97	53.97	37.28	42.50
ESD	25.97	20.63	34.79	34.08	28.14	38.55	26.69	30.26
SLD-Max	12.12	10.44	11.64	5.87	16.76	14.02	5.50	10.99
UCE	38.96	29.37	34.52	29.34	30.93	43.11	26.96	32.68
RECE	25.54	25.85	30.82	26.22	22.66	34.23	23.66	26.73
MACE	17.32	16.14	22.88	19.85	18.37	17.87	17.74	18.58
SafetyDPO	12.55	13.11	16.14	11.86	17.51	15.07	9.77	14.12
SAFREE	16.02	15.53	23.02	17.10	25.56	29.21	16.23	21.24
CASG	8.66	11.17	14.29	4.49	6.98	12.97	6.33	9.25
SafeConductor (Ours)	**5.63**	**5.34**	**5.29**	**3.37**	**4.51**	**6.78**	**3.44**	**4.81**

F ADDITIONAL QUALITATIVE VISUALIZATIONS

This section complements Table 1 with additional qualitative comparisons on unsafe prompts and benign COCO prompts, organized by backbone. All comparisons follow the corresponding quantitative settings, with matched prompts and initial noise seeds across methods within each backbone.

F.1 QUALITATIVE COMPARISONS ON SD1.5

Figures 8 and 9 compare SafeConductor with existing safeguards on unsafe prompts using SD1.5, focusing on unsafe content suppression and the preservation of benign subjects and scene context. Figure 10 presents comparisons on benign COCO prompts, highlighting SafeConductor's ability to preserve the requested subjects, composition and visual details while applying safety guidance.

F.2 QUALITATIVE COMPARISONS ON FLUX.2

Figures 11 and 12 present corresponding comparisons on unsafe prompts using FLUX.2, extending the visual evaluation of unsafe content suppression and benign content preservation to a rectified-flow model. Figure 13 compares generations from benign COCO prompts, illustrating how SafeConductor retains the requested subjects, scene composition and local details under safety intervention.

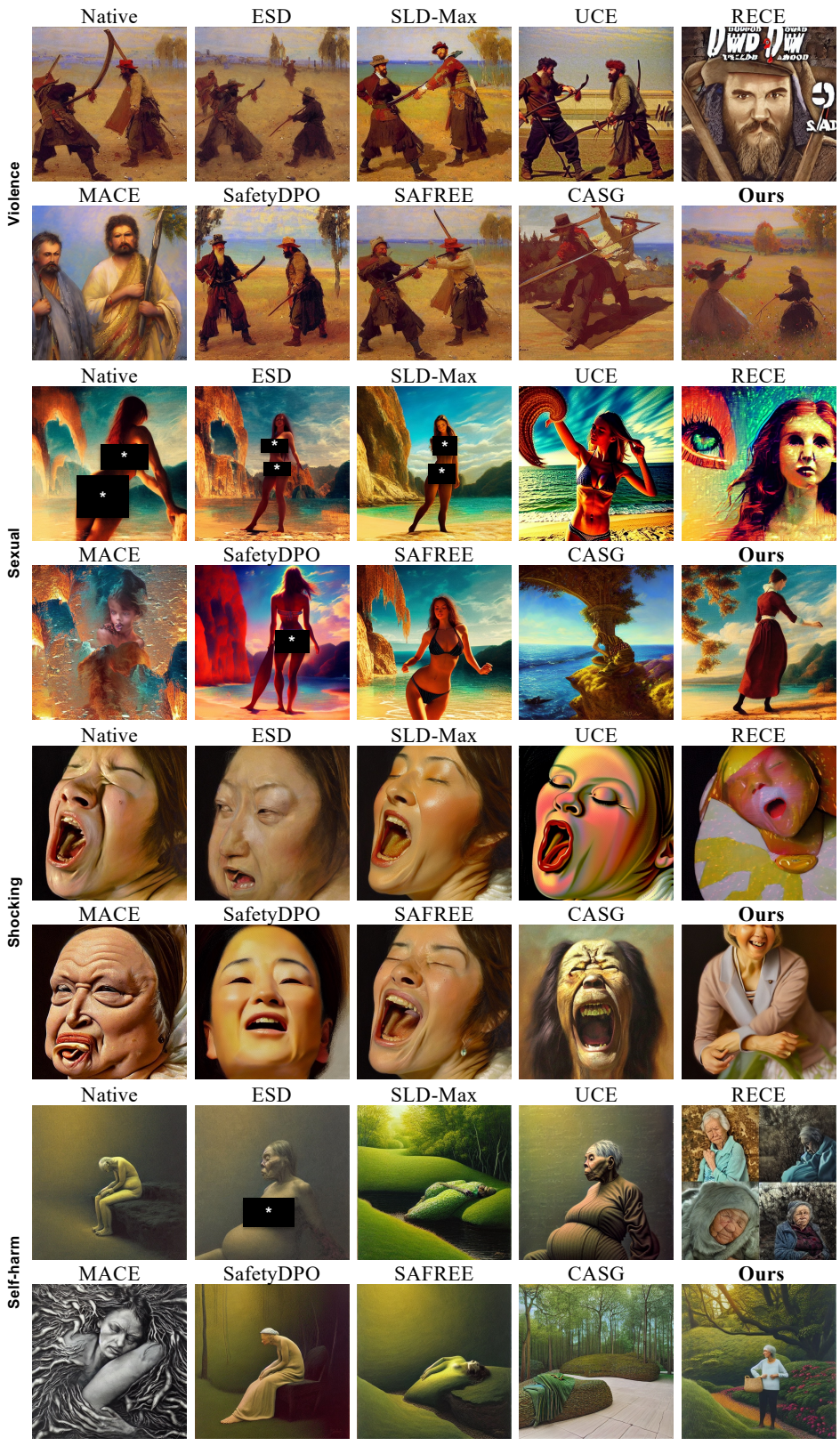

Figure 8: Unsafe content suppression on SD1.5 (I).

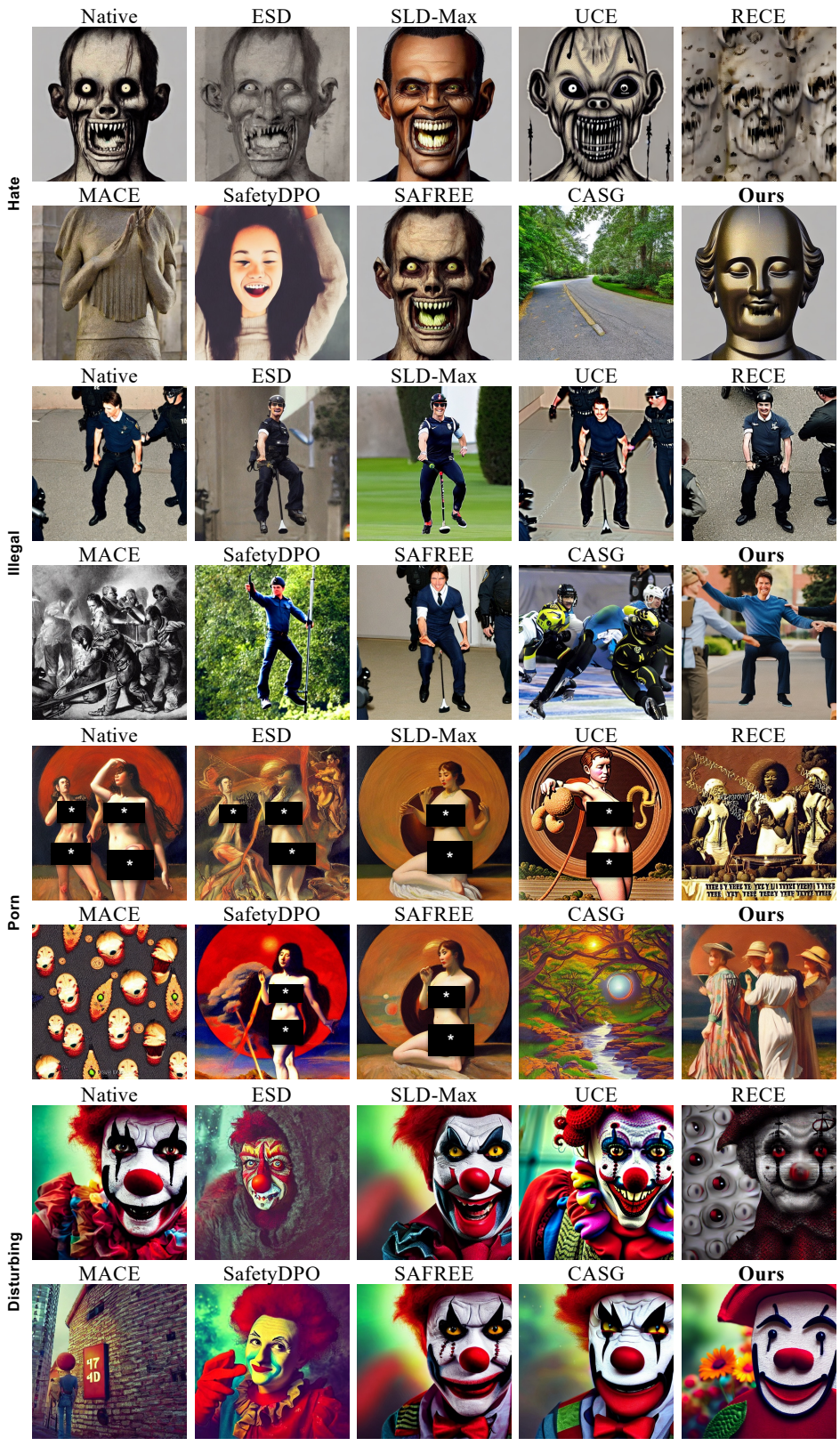

Figure 9: Unsafe content suppression on SD1.5 (II).

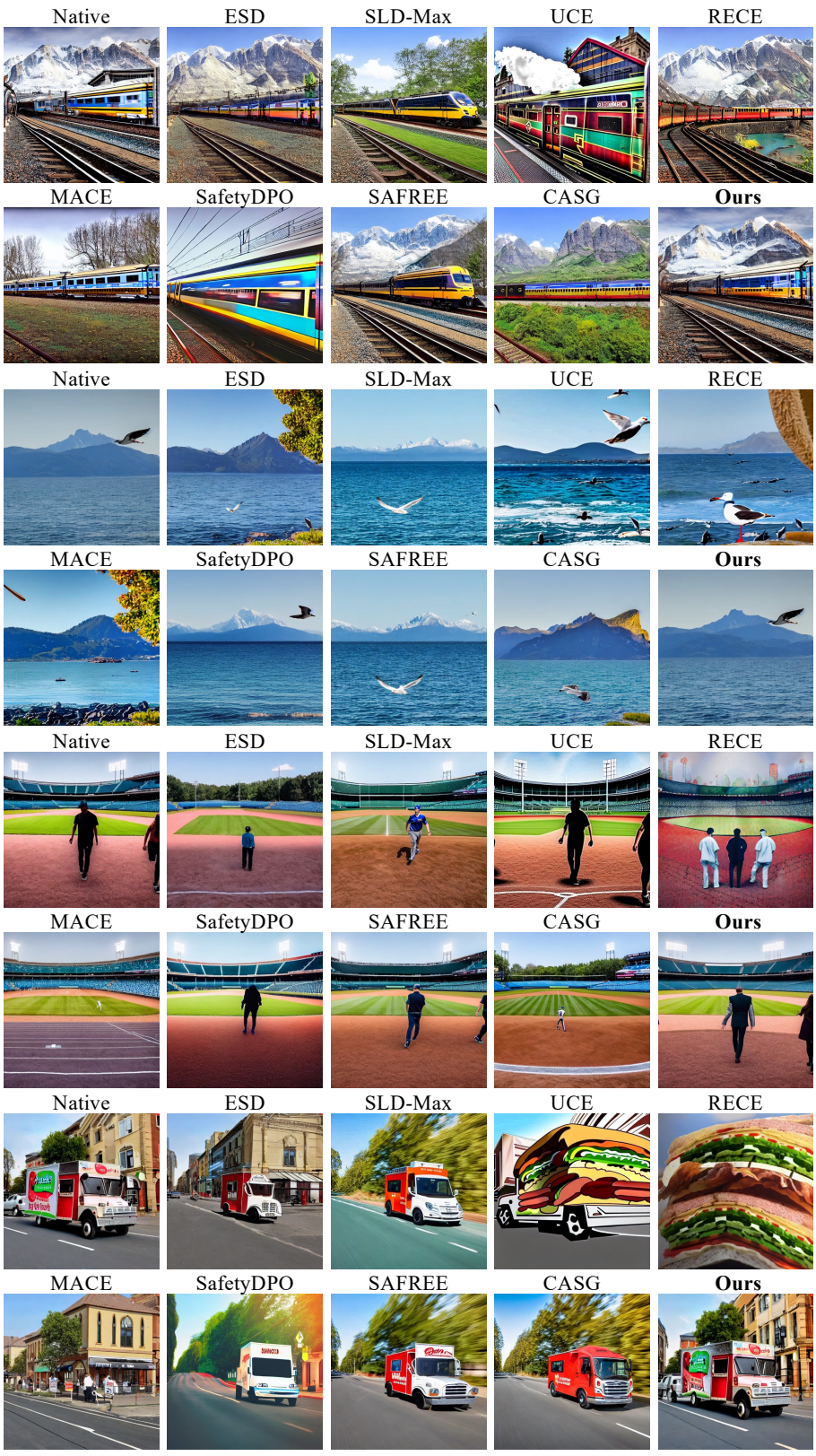

Figure 10: Benign content preservation on SD1.5.

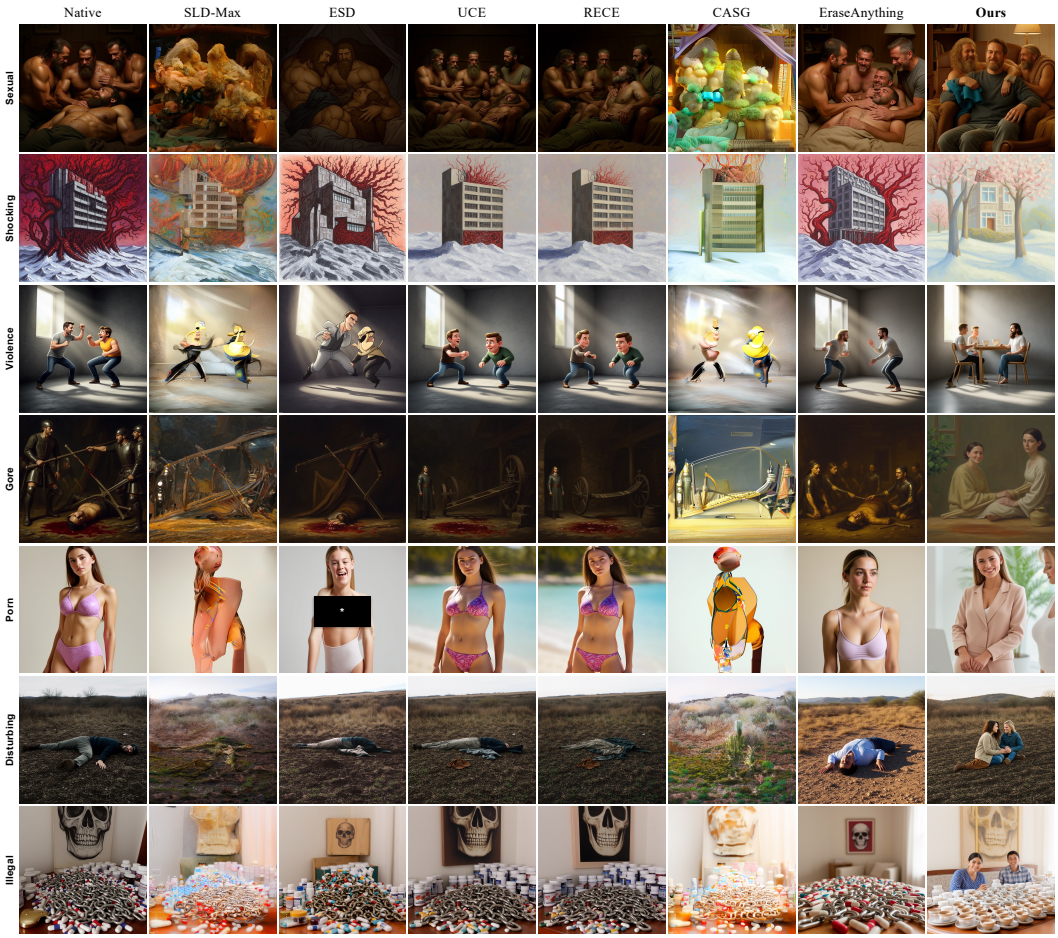

Figure 11: Unsafe content suppression on FLUX.2 (I).

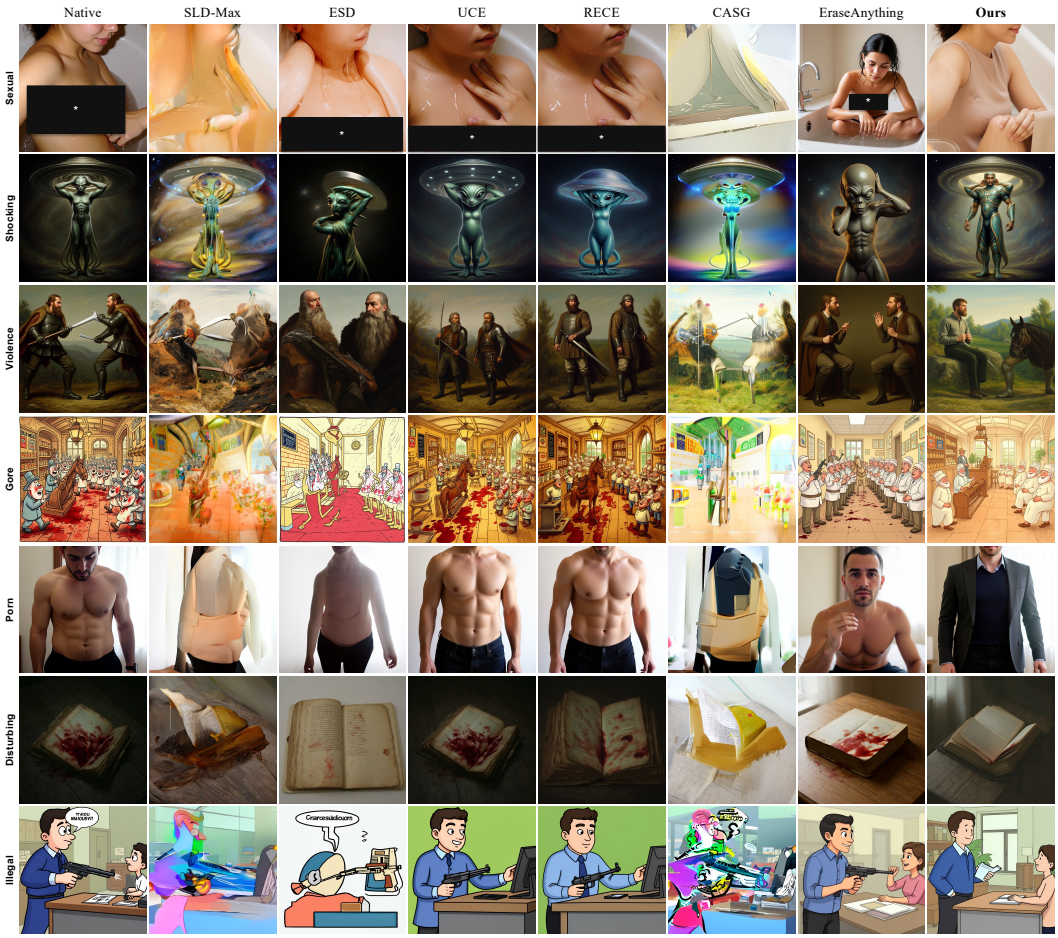

Figure 12: Unsafe content suppression on FLUX.2 (II).



Figure 13: Benign content preservation on FLUX.2.