UniDef: Universal Defense Against Unauthorized Image Manipulation

Supplementary Material

A. Further Clarifications and Extensions

To provide a more comprehensive understanding of our method, we include the following supplementary analyses and discussions.

A.1. Explanation for Figure 2

(1) The left-side of Figure 2 is a conceptual diagram of feature space to illustrate the consistency of diffusion models' (DMs) outputs ***at distribution level***, rather than identical outputs. **(2)** The right-side shows the gradient differences among DMs ***under the same input but varying noise levels***, indicating that the direction of global denoising tends to converge. **(3)** Moreover, we provide the distribution disparity in Table S1 that further validates distribution consistency of DMs at global denoising.

Table S1. Distribution differences between different models (FID↓).

Steps	20%	40%	60%	80%	100%
[SD v1.4, SD v1.5]	113.78	125.66	110.55	78.77	66.23
[SD v1.4, IP2P]	127.49	138.74	117.86	85.75	72.44

A.2. Theoretical Foundation of Lemma 3.1

Lemma 3.1 is derived under standard assumptions used in score-based diffusion models. Specifically, we assume a VP-SDE forward process with Gaussian transitions

$$q(x_t \mid x_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, x_0,\ (1-\bar{\alpha}_t)I\big)\,, \tag{S1}$$

and a Lipschitz-continuous score network $s_\theta(x,t)$. Song et al. [4] provide a rigorous derivation showing that the KL divergence between endpoint distributions can be expressed as a time-integrated squared score mismatch:

$$D_{\mathrm{KL}}(p_0\|q_0) = \int_0^T g(t)^2\, \mathbb{E}_{x_t\sim p_t}\big[\|s_\theta(x_t,t) - s^\star(x_t,t)\|_2^2\big]dt. \tag{S2}$$

Due to the Gaussian structure of the VP process and the concentration of $q(x_t \mid x_0)$, replacing distributional expectations with single-sample approximations yields a consistent estimator.

A.3. Reasoning for Using Latent Code z in FDJE

The choice of latent code z for finite-difference Jacobian estimation (FDJE) is motivated by several theoretical considerations:

- The latent-space direction aligns with major eigencomponents of the score Jacobian, yielding more informative perturbation directions.
- Since $z \sim \mathcal{N}(0, I)$ under the VAE encoder, it naturally matches the isotropic assumptions used in Hutchinson-style estimators.
- Empirical comparisons with alternative directions (random vectors, PCA directions, CLIP embeddings) show that z consistently provides lower-variance estimates and more stable optimization behavior.

These observations justify the use of z as the default direction in FDJE.

B. Additional Experimental Details

Perturbation updates follow projected gradient ascent under the constraint: Maximum budget: $\xi = 16/255$, Step size: $\eta = 1/255$, Iterations: 100. At each iteration, perturbations are clipped element-wise to maintain imperceptibility. All images remain in the valid pixel range $[0, 1]$ after applying the perturbation and are converted to UINT8. The algorithm of our UniDef is shown in Algorithm 1.

Algorithm 1 The algorithm of our UniDef.

Require: Clean image x_0; diffusion model $\epsilon_\theta(\cdot, t)$; encoder $E(\cdot)$; perturbation budget $\xi = 16/255$; step size $\eta = 1/255$; number of iterations $K = 100$; finite-difference step $H = 3$, timestep set $\mathcal{T}$.
Ensure: Protected image $x_0' = x_0 + \delta$.

Initialize perturbation $\delta \leftarrow 0$.
$z \leftarrow E(x_0')$ ▷ latent encoding
for $k = 1$ to K **do**
  Sample $t \sim \mathrm{Uniform}(\mathcal{T})$, $\epsilon \sim \mathcal{N}(0, I)$.
  $x_t' \leftarrow q_{\text{sample}}(x_0', t, \epsilon)$. ▷ forward noise addition
  $\epsilon_{\text{pred}} \leftarrow \epsilon_\theta(x_t', t)$. ▷ noise prediction
  $g_{\text{img}} \leftarrow 0$.
  for $h = 1$ to H **do**
    $x_t^+ \leftarrow x_t' + \epsilon_{\text{fd}} z, \quad x_t^- \leftarrow x_t' - \epsilon_{\text{fd}} z$.
    $\epsilon^+ \leftarrow \epsilon_\theta(x_t^+, t), \quad \epsilon^- \leftarrow \epsilon_\theta(x_t^-, t)$.
    $J \leftarrow (\epsilon^+ - \epsilon^-)/(2\epsilon_{\text{fd}})$ ▷ estimate Jacobian.
    $r \leftarrow \epsilon_{\text{pred}} - \epsilon$.
    $s_t \leftarrow w(t) \cdot \langle J, r \rangle$.
    $g_t^{\text{img}} \leftarrow \nabla_{x_0'} s_t$ ▷ backpropagate to image space.
    $g_{\text{img}} \leftarrow g_{\text{img}} + g_t^{\text{img}}$.
  end for
  $\delta \leftarrow \delta + \eta \cdot \mathrm{sign}(g_{\text{img}})$.
  $\delta \leftarrow \mathrm{clip}(\delta, -\xi, \xi)$.
  $x_0' \leftarrow \mathrm{clip}(x_0 + \delta, 0, 1)$.
end for
return x_0'.

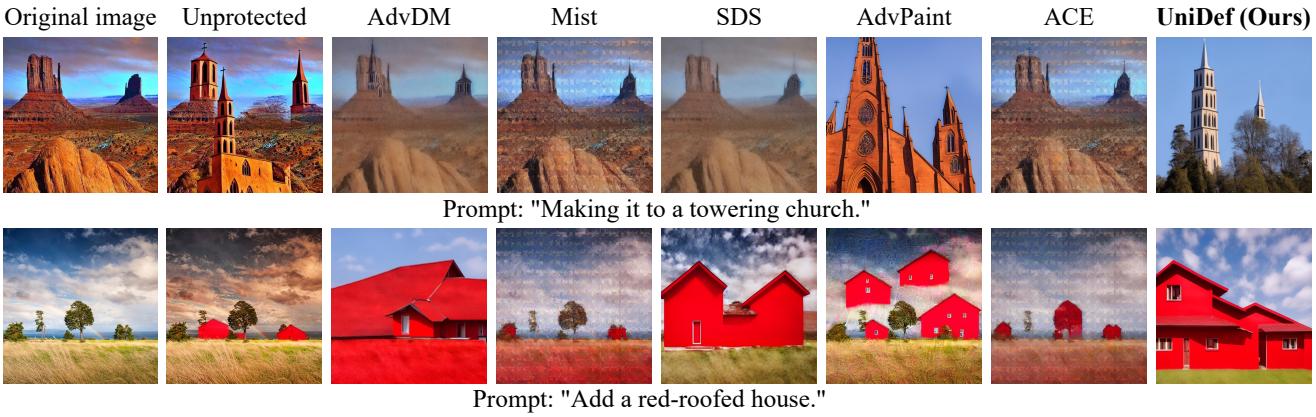

Figure S1. Additional protection results under InstructPix2Pix.

Table S2. Analysis of protection effectiveness (FID), imperceptibility (PSNR & SSIM), runtime (seconds), and VRAM (MiB). ↑: higher is better, ↓: lower is better. The optimal is marked with red, the sub-optimal is marked with orange.

Method	FID↑	PSNR↑	SSIM↑	Times↓	VRAM↓
AdvDM [3]	257.21	26.83	0.7008	47.76	15112
Mist [2]	380.20	26.47	0.6836	48.17	16744
SDS [5]	231.92	26.81	0.7138	26.58	13508
AdvPaint [1]	291.54	27.71	0.7087	85.41	20220
ACE [6]	356.65	26.37	0.6776	33.59	11495
UniDef (Ours)	405.54	26.86	0.7111	45.11	11294

C. Runtime and Imperceptibility Analysis

Table S2 summarizes the runtime and stealth performance of existing defence methods. The comparison includes: (1) FID↑ measuring the deviation of generated outputs from the clean distribution, (2) PSNR↑ / SSIM↑ measuring imperceptibility of perturbations, and (3) Times↓ measuring computational efficiency.

As shown in Table S2, UniDef achieves the strongest protection performance (higher FID) among all compared methods. Remarkably, this disruptive capability does not come at the cost of visual quality, with competitive PSNR and SSIM, outperforming baselines that often degrade the protected image. These results confirm that UniDef achieves an optimal balance between maximal generative disruption and high imperceptibility, making it particularly suitable for real-world image protection scenarios.

In terms of computational efficiency, UniDef demonstrates moderate runtime performance, significantly outperforming the computationally more intensive AdvPaint. Although SDS executes faster, its protection strength is markedly weaker, with limited capability to disrupt global generative behaviour. Overall, UniDef achieves the optimal balance across all evaluation metrics: it possesses the strongest protection strength, maintains high stealthiness, and delivers practical runtime performance. The table comparing our UniDef and benchmark peak VRAM figures demonstrates our superior resource efficiency.

C.1. Qualitative results of perturbation artifacts

We provide a comparison of the perturbation artifacts with the baseline methods in Fig. S2. Specifically, the visual results illustrate the perturbation effects produced by different methods under the same experimental conditions. It can be observed that the visual changes introduced by our UniDef method are relatively subtle, whereas other methods introduce artefacts of varying intensity.

Figure S2. Comparison of the perturbation artifacts.

D. More Experiments

D.1. Additional Quantitative Evaluation

To complement the quantitative analyses reported in the main paper, we further provide additional evaluations on imperceptibility by measuring the PSNR and LPIPS metrics for the edited results of both the original images and their protected images across all tasks, as shown in Table S3. Across diverse tasks, our UniDef consistently achieves low PSNR and high LPIPS, demonstrating that our UniDef induces significant semantic and structural deviations in all downstream diffusion outputs compared to existing protection methods. These results confirm that our UniDef provides robust protection under diversified diffusion pipelines.

Table S3. Additional quantitative results of imperceptibility across diverse diffusion tasks. ↑: higher is better, ↓: lower is better. The optimal is marked with red, the sub-optimal is marked with orange.

Method	InstructPix2Pix		Inpainting		Super Resolution		Image to Video		Image to 3D	
	PSNR↓	LPIPS↑	PSNR↓	LPIPS↑	PSNR↓	LPIPS↑	PSNR↓	LPIPS↑	PSNR↓	LPIPS↑
AdvDM [3]	10.97	0.6183	16.86	0.5043	18.17	0.6860	22.37	0.1402	8.57	0.6227
Mist [2]	15.12	0.4891	16.42	0.4522	18.52	0.5641	19.12	0.2257	9.99	0.6462
SDS [5]	11.35	0.6061	16.51	0.4947	18.19	0.6186	20.05	0.2361	7.71	0.6839
AdvPaint [1]	14.50	0.5077	16.54	0.5087	18.62	0.6252	20.89	0.1728	10.04	0.6818
ACE [6]	15.38	0.4857	16.39	0.4528	18.52	0.5664	18.91	0.2280	9.89	0.6467
UniDef (Ours)	10.24	0.6401	15.68	0.5067	17.31	0.6713	18.72	0.2465	7.01	0.7202

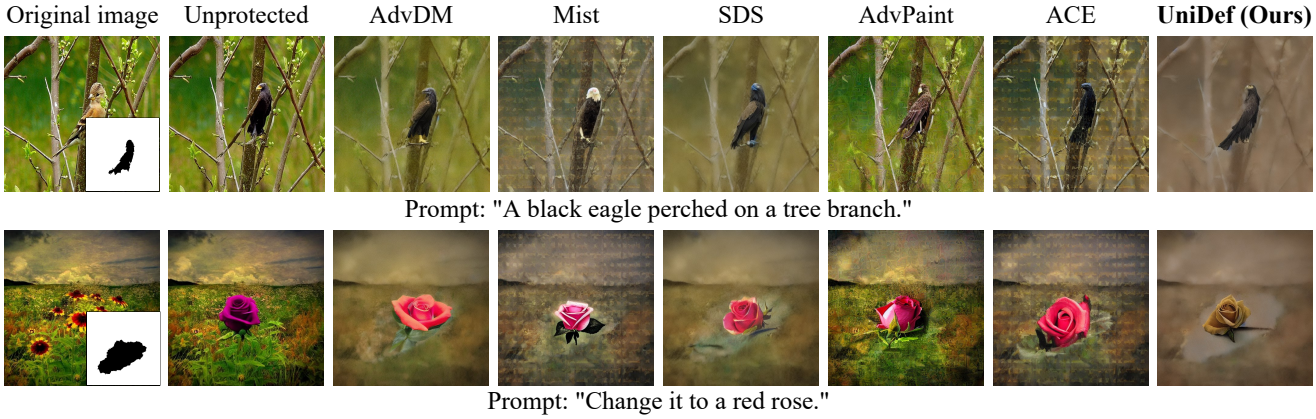

Figure S3. Additional visual results of protection against Inpainting.

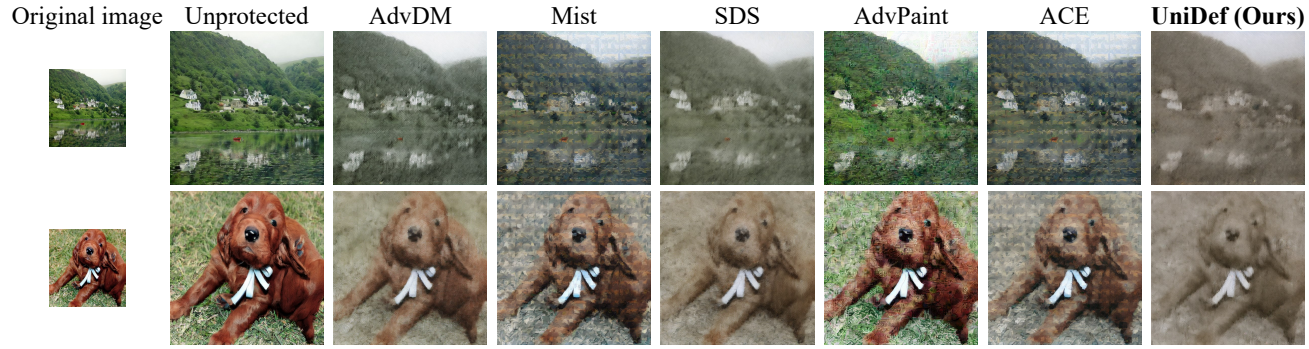

Figure S4. Additional examples of manipulation resistance in super-resolution.

D.2. Additional Qualitative Results

We further include extended qualitative results to illustrate the effects of protected images in resisting manipulation across multiple diffusion tasks. As shown in Figs. S1, S3, S4, S5, and S6, the protected inputs consistently prevent diffusion models from performing the intended edits or reconstructions. More specifically, image restoration generates semantically ambiguous or nonsensical completions. Super-resolution fails to recover high-frequency details, instead producing blurred or structurally unstable outputs. Image-to-video conversion suffers from severe semantic drift and temporal instability between frames. Image-to-3D reconstruction cannot recover consistent multi-view geometry or textures. These qualitative results demonstrate that our UniDef reliably disrupts the generative process across different tasks, further confirming the strong cross-task generalization capability of our approach.

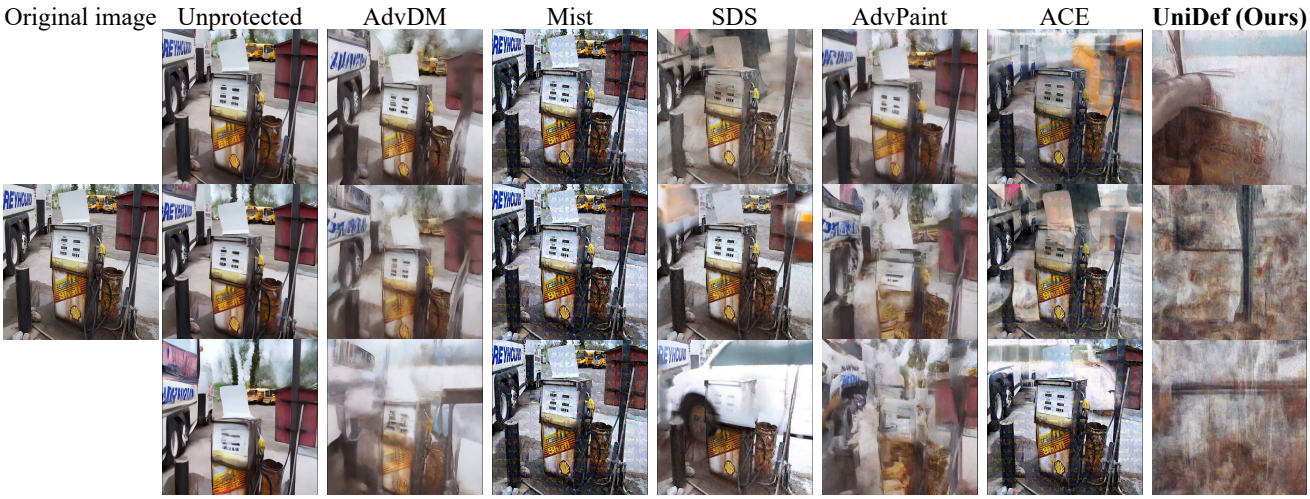

Figure S5. Extended results of protection against diffusion-based video generation.

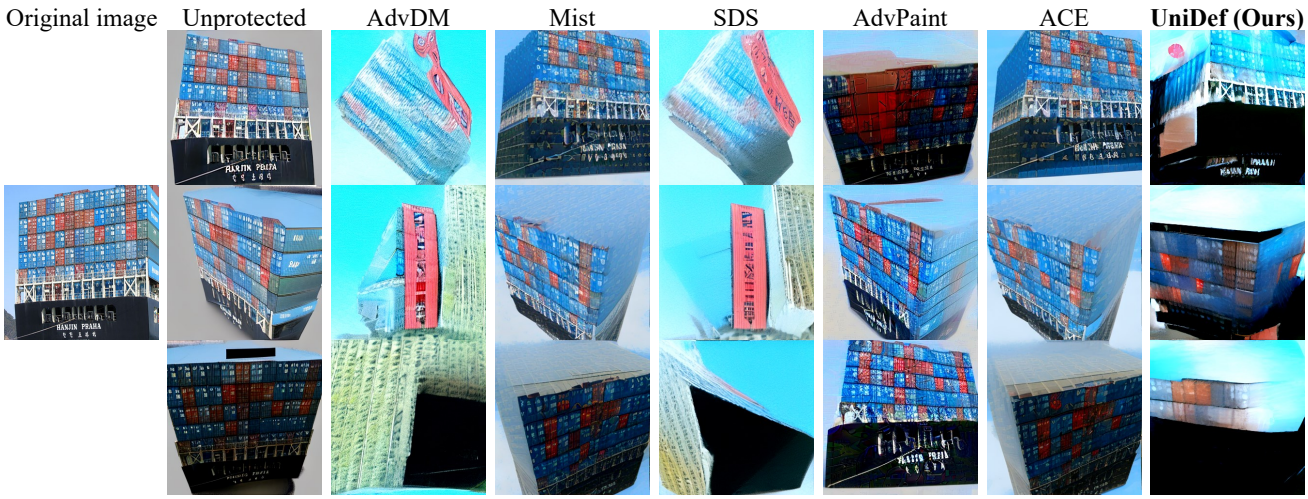

Figure S6. Additional protection results for image-to-3D reconstruction.

D.3. Generalizability on modern diffusion models

We present the results against models based on SDXL, DiT (Qwen 2.5), and FLUX 1.0 in Table S4 and Fig. S7. Specifically, the table reports the quantitative evaluation results, while the figure provides the corresponding visual comparisons under the same experimental setting. It can be seen that our UniDef maintains consistent protection performance across different model architectures, further demonstrating the generality of our method across various frameworks.

Table S4. Quantitative evaluations on modern models.

Method	SDXL	Qwen 2.5	FLUX 1.0
PSNR↓	16.31	14.22	18.56
FID↑	227.56	110.31	170.03

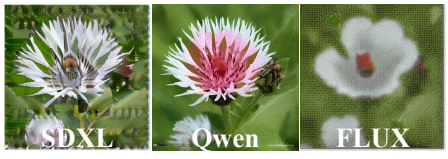

Figure S7. Visualization on modern models.

D.4. Evaluation using VLM.

We provide a VLM evaluation based on ChatGPT-5.2 in Table S5, which directly judges the validity of the defenses. Specifically, the evaluation leverages the reasoning capability of the VLM to assess the validity of the manipulated output for the protected content. The results indicate that our

UniDef method more effectively preserves the defensive effect, while the baseline methods exhibit weaker protection under the same evaluation setting.

Table S5. Evaluation using VLM↑.

Method	Editing	Inpainting	Image to Video
SDS	0.921	0.753	0.728
ACE	0.968	0.875	0.862
Ours	**0.985**	**0.887**	**0.879**

D.5. Ablation on Finite-Difference Step

We further perform an ablation study on the finite-difference step ϵ_{fd}, which controls the magnitude of the latent offset used in the finite-difference Jacobian estimation. This experiment examines how different values of ϵ_{fd} affect imperceptibility and protection behavior. We evaluate three commonly used settings, $\epsilon_{\text{fd}} = 0.1$, 0.01, and 0.001, on SD v1.4 and IP2P.

Table S6. Ablation results on the finite-difference step ϵ_{fd}.

	ϵ_{fd}	PSNR↓	CLIP↓	FID↑	LPIPS↑
SD v1.4	0.1	16.73	0.9145	288.01	0.5867
	0.01	16.40	0.9011	405.54	0.5973
	0.001	16.78	0.9144	281.84	0.6287
	ϵ_{fd}	PSNR↓	CLIP↓	FID↑	LPIPS↑
IP2P	0.1	11.22	0.8203	193.07	0.6223
	0.01	10.24	0.8086	239.88	0.6401
	0.001	11.02	0.8225	190.07	0.6197

As shown in Table S6, across varying parameter values, quantitative metrics remained relatively stable, with PSNR, CLIP similarity, FID, and LPIPS exhibiting only minor fluctuations. This indicates that the finite difference estimator demonstrates low sensitivity to specific step magnitudes. Notably, among all evaluated values, $\epsilon_{\text{fd}} = 0.01$ delivered the most balanced performance, yielding robust protection strength.

D.6. Ablation of perturbation strength budget

We provide results under different perturbation budgets in Table S7 and Fig. S8. Specifically, Table S7 reports the quantitative results across varying perturbation constraints, while Fig. S8 presents the corresponding visual comparisons. The results show that our method consistently achieves stronger protection under different perturbation budgets compared with the baseline methods.

D.7. Ablation studies on multiple random vectors

The original "w/o z" uses a single vector, and results with multiple random vectors (1/5/10) are shown in Table S8.

Figure S8. Visualization of different perturbation strength budget.

Table S7. Quantitative ablation results of different perturbation strength budget.

Budgets	Method	PSNR↓	CLIP↓	LPIPS↑
8/255	ACE	17.65	0.9348	0.4853
	Ours	**17.59**	**0.9327**	**0.5821**
4/255	ACE	18.39	0.9396	0.3873
	Ours	**18.21**	**0.9378**	**0.4196**

Specifically, the table reports the performance when different numbers of random vectors are introduced. The results show that increasing the number of random vectors leads to more stable performance, while the original single-vector setting exhibits comparatively limited effectiveness.

Table S8. Ablation on z vectors.

Vectors	PSNR↓	CLIP↓	LPIPS↑
1	**17.46**	0.9087	0.5171
5	17.50	0.9036	0.5186
10	17.51	**0.9015**	**0.5309**

D.8. Additional Robustness Comparisons

We present additional robustness comparisons under stronger purification in Table S9. The results demonstrate that our method maintains superior robustness under these conditions compared with the baseline approaches.

Table S9. Additional robustness comparisons (PSNR↓).

Method	JPEG 60	JPEG 70	JPEG 80	IMPRESS	tilt
ACE	18.31	18.27	18.11	19.22	18.37
Ours	**18.12**	**17.84**	**17.56**	**18.87**	**18.19**

E. Combination with Texture-aware Loss

To further enhance the protection effect and strengthen the distributional deviation induced by our method, we incorporate an additional texture-aware loss into the optimization objective. This auxiliary term operates in the encoder

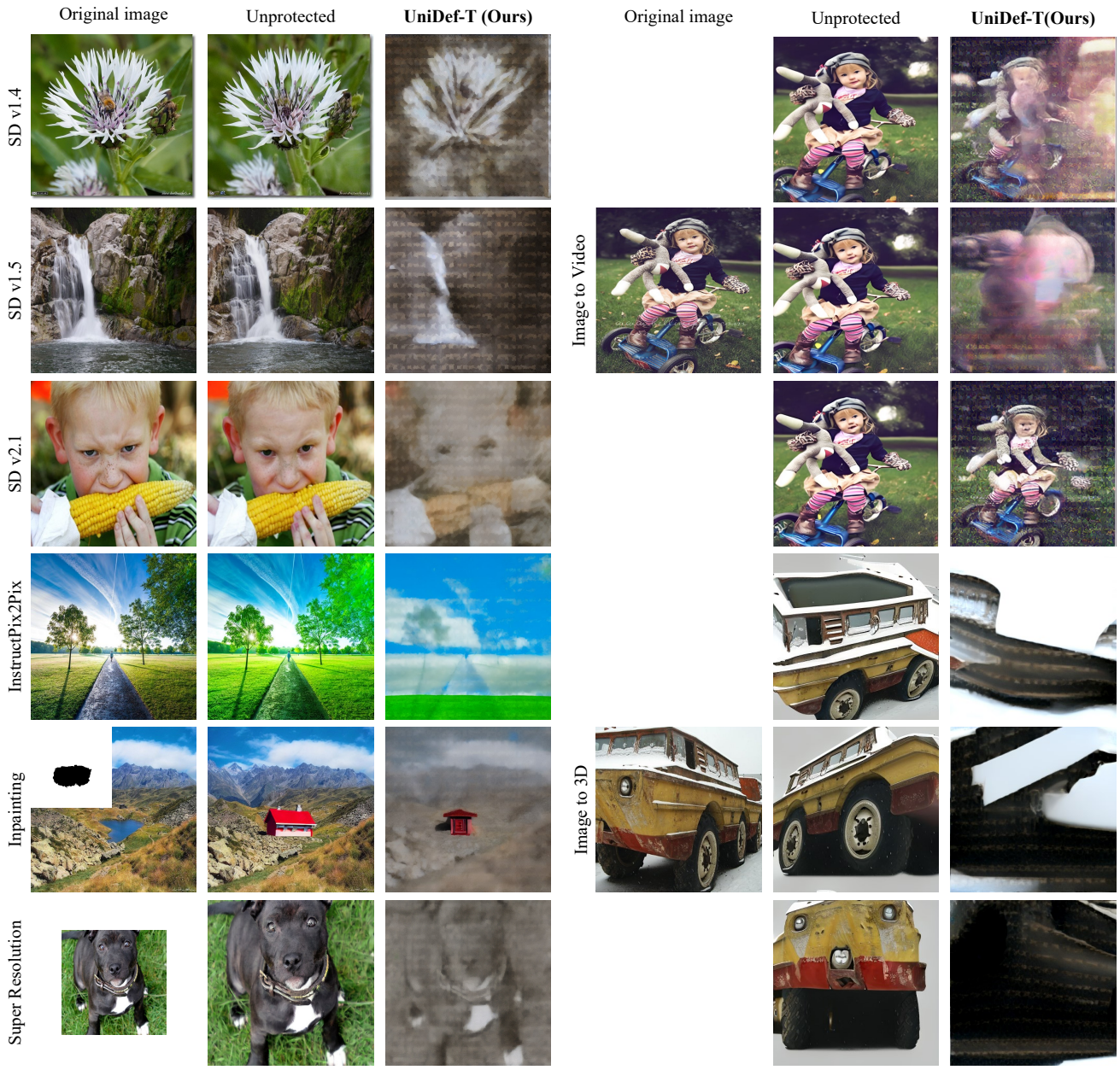

Figure S9. Visual effects of incorporating the texture-aware loss (UniDef-T). The editing prompt of InstructPix2Pix is: "A vibrant photo of blue skies, white clouds, and green grass". The editing prompt of Inpainting is: "A little house with a red roof".

space and explicitly encourages the protected image to deviate from a designated target texture or appearance, thereby amplifying the overall manipulation resistance.

E.1. Texture-Aware Enhanced Objective

Given a protected image x_0' and a reference image x_{tar}, the texture-aware loss is computed based on the encoder embeddings:

$$\mathcal{L}_{\text{tex}} = \|E(x_0') - E(x_{\text{tar}})\|_2^2, \tag{S3}$$

where $E(\cdot)$ denotes the diffusion model's image encoder. We adopt the same target image as Mist [2]. This loss introduces a controlled deviation in the latent representation, complementing the inconsistency encouraged by our original objective. The combined objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{KL} + \lambda_{\text{tex}}\,\mathcal{L}_{\text{tex}}, \tag{S4}$$

where λ_{tex} denotes the weight for texture-aware bias.

Table S10. Quantitative results of the texture-aware loss under different λ_{tex} settings.↑: higher is better, ↓: lower is better. The optimal is marked with red, the sub-optimal is marked with orange.

	SD 1.4				SD 1.5				SD 2.1			
λ_{tex}	PSNR↓	CLIP↓	FID↑	LPIPS↑	PSNR↓	CLIP↓	FID↑	LPIPS↑	PSNR↓	CLIP↓	FID↑	LPIPS↑
0.1	16.90	0.9114	267.38	0.6868	16.90	0.9128	286.46	0.6962	16.25	0.9082	294.63	0.6278
0.01	16.93	0.9107	308.03	0.6292	17.02	0.9132	308.15	0.6269	16.58	0.9079	303.65	0.5858
0.001	16.77	0.9026	367.94	0.5866	16.80	0.9048	365.85	0.5876	17.10	0.9010	371.65	0.5522
	InstructPix2Pix				Inpainting				Super Resulotion			
λ_{tex}	PSNR↓	CLIP↓	FID↑	LPIPS↑	PSNR↓	CLIP↓	FID↑	LPIPS↑	PSNR↓	CLIP↓	FID↑	LPIPS↑
0.1	12.11	0.8189	195.86	0.5952	15.33	0.8443	180.90	0.5617	16.33	0.9300	199.90	0.6477
0.01	13.81	0.8210	185.43	0.5535	15.56	0.8442	182.95	0.5175	16.92	0.9289	184.77	0.5754
0.001	14.84	0.8128	218.11	0.5020	16.02	0.8364	209.67	0.4765	18.05	0.9225	211.45	0.5535
	Image to Video				Image to 3D							
λ_{tex}	PSNR↓	CLIP↓	FID↑	LPIPS↑	PSNR↓	CLIP↓	FID↑	LPIPS↑				
0.1	20.80	0.8926	71.99	0.2000	7.90	0.7478	181.69	0.6759				
0.01	20.28	0.8929	76.57	0.2078	8.31	0.7492	184.59	0.6696				
0.001	18.95	0.8865	95.30	0.2336	9.15	0.7506	211.02	0.6672				

E.2. Quantitative Results

Table S10 reveals significant differences under varying values of λ_{tex}. When $\lambda_{\text{tex}} = 0.1$, PSNR and LPIPS are generally better, suggesting that the perturbation introduces more pixel-level deviations. Conversely, $\lambda_{\text{tex}} = 0.001$ leads to lower CLIP similarity and higher FID, indicating that the deviation becomes more semantic and distribution-level. These trends indicate that λ_{tex} introduces texture-level bias complementarily to our semantic bias. The most balanced performance is achieved at $\lambda_{\text{tex}} = 0.01$, yielding stable and consistent bias at both the pixel and distribution levels.

E.3. Qualitative Results

We further provide qualitative examples to illustrate the visual behavior introduced by the texture-aware term. As shown in Fig. S9, incorporating $\mathcal{L}_{\text{tex}}$ induces mild yet more structured texture deviations in the protected images. This provides a complementary effect to our semantic deviations.

F. Limitation

The defense effect of our method diminishes when faced with extremely subtle edits, as such tampering is insufficient to trigger distribution shifts.

References

[1] Joonsung Jeon, Woo Jae Kim, Suhyeon Ha, Sooel Son, and Sung-eui Yoon. AdvPaint: Protecting Images from Inpainting Manipulation via Adversarial Attention Disruption. In *International Conference on Learning Representations*, 2025. 2, 3

[2] Chumeng Liang and Xiaoyu Wu. Mist: Towards Improved Adversarial Examples for Diffusion Models. *arXiv preprint arXiv.2305.12683*, 2023. 2, 3, 6

[3] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial Example Does Good: Preventing Painting Imitation from Diffusion Models via Adversarial Examples. In *International Conference on Machine Learning*, pages 20763–20786, 2023. 2, 3

[4] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021. 1

[5] Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion based mimicry through score distillation. In *International Conference on Learning Representations*, 2024. 2, 3

[6] Boyang Zheng, Chumeng Liang, and Xiaoyu Wu. Targeted Attack Improves Protection against Unauthorized Diffusion Customization. In *International Conference on Learning Representations*, 2025. 2, 3