Appendix

In the appendix, first, we will offer more details on the algorithm in Section A, then we show the details about our experiment settings, including dataset and metrics in Section B. After that, we will put more experimental results in Sectino C. Finally, we show the settings of our human evaluation survey in Section D.

A Details about Algorithms

We provide a PyTorch-styled pseudo code to show how to attack the latent diffusion model (LDM) with SDS acceleration, and how does the gradient descent work:

```
import torch
ldm = load_latent_diffusion_model()
x  = load_clean_image()
# start optimization
for _ in range(iterations):
    x = x.detach().clone()
    z = encoder(x)
    noise = sample_std_gaussian()
     # forward diffusion process
    z_t = q_sample(z, noise)
    # SDS gradient, only inference
    with torch.no_grad():
        # SDS gradient in z-space
        sds_grad = ldm(z_t, t) - noise
    z.backward(gradient=sds_grad)
    grad = x.grad().detach() # final gradient in x-space
    # projected gradient descent/ascent
    if mode == 'gradient ascent':
        x = x + grad.sign() * step_size
    elif mode == 'gradient descent':
        x = x - grad.sign() * step_size
    # clip to budget restriction
    x = clip(x, eps)

x_adv = x
# run down streaming mimicry with protected image
run(x_adv, task)
```

from the above code, we can see that the gradient information of the denoiser does not need to be saved during the protection, which makes it much faster and also saves a lot of GPU memory. This enables individual users to run the protection algorithm more easily.

Also, we have a variety of new proposed protection methods under our design space: we summarize the design space as { semantic loss ($\mathcal{L}_S$), textural loss ($\mathcal{L}_T$), SDS, gradient descent (GD), gradient ascent (GA) }. All the methods evaluated in this paper can be constructed in the design space, and here we summarize all the methods as follows in Table 3.

B Details about Our Experiments

B.1 Dataset

While the previous works either focus more on portraits or artworks, we collect four small subsets including anime, artworks, landscape, and portraits. We collect the anime and portrait data from the internet, the landscape data from (Arnaud, 2020), and the artworks subset from WikiArt (Nichol, 2016). The size of the dataset is 100 for anime and portrait subsets and 200 for landscape and artwork subsets. Samples of the dataset can be found in Figure 9. For the inpainting task, we use the portrait subset in our dataset, using Grounded-SAM to get the mask of the human object. For the textual inversion task, we use samples from the dataset provided by (Ruiz et al., 2023).

Figure 9: **Examples of Our Dataset**: in each part divided using red dotted lines, we show some samples from the subset of anime, artworks, landscape, and portraits respectively. We want to cover more kinds of mimicry scenarios to evaluate the performance of each protection method.

Methods	Component	Perturbation	Consumption	SDEdit	Inpainting	Textual Inversion
AdvDM	$\mathcal{L}_S$+ GA	*	*	**	**	***
Mist	$\mathcal{L}_S$+ GA + $\mathcal{L}_T$	**	*	**	**	**
PhotoGuard	$\mathcal{L}_T$	**	***	*	**	**
AdvDM(-)	$\mathcal{L}_S$+ GD	***	*	**	**	*
SDS(+)	$\mathcal{L}_S$+ GA+ SDS	*	***	**	**	***
SDS(-)	$\mathcal{L}_S$+ GD+ SDS	***	***	***	**	*
SDST	$\mathcal{L}_S$+ GD+ SDS + $\mathcal{L}_T$	***	**	**	**	***

Table 3: **Summary of All the Protection Methods in our Design Space**: we summarize all the protection methods we currently have, and all can be composed into some components in the design space we proposed. The first three rows include methods that are proposed in previous works, and the left four rows include the new protection methods first proposed in our paper, with new strategies SDS and GD marked in red. We show the strength of all these methods from the perspective of the quality of perturbation (whether it is natural), the computational consumption, and their performance on SDEdit, Inpainting, and Textual Inversion respectively. We use stars to measure them roughly, more stars represent better performance (e.g. more natural perturbations, less consumption, better protection).

B.2 Implementation of Baselines

For AdvDM and Mist, we follow the settings in the original paper. For PhotoGuard, we use the pattern proposed in Mist as the target image, which is shown to be the most effective pattern. All the input images have a resolution of $512 * 512$.

B.3 Implementation of the Threat Model

All the threat model experiments in this paper can be run on one single A6000 GPU without parallelization.

For the global SDEdit, we use DDIM (Song et al., 2020a) to accelerate the reverse sampling, setting the total respaced timestep to be 100, in Figure 6, we show the SDEdit results of forward strength 0.2 and 0.3. The text prompts are set to 'a anime picture, 'a landscape picture', 'a artwork painting' and 'a portrait photo' for each subset.

For image inpainting, we use the StableDiffusion Inpainting pipeline provided by Diffusers: `https://huggingface.co/docs/diffusers/using-diffusers/inpaint`, using the default settings in the pipeline, with strength set to 1.0 and text-guidance set to 7.5.

For textual inversion, we also use the pipeline provided in Diffusers: `https://huggingface.co/docs/diffusers/training/text_inversion`, where we set the learning rate to $5 * 10^{-4}$ and train the embedding for 2000 iterations.

B.4 Metrics

Here we introduce the quantitative measurement we used in our experiments: (1) To measure the quality of naturalness and imperceptibility of the generated perturbations, we use Fréchet Inception Distance (FID) (Heusel et al., 2017) over the collected dataset, Structural Similarity (SSIM) (Wang et al., 2004) and Perceptual Similarity (LPIPS) (Zhang et al., 2018) compared with the original image. Also, we compare the speed of protection, using metrics including the VRAM occupation (VRAM), time consumption (TIME) and the parallel speed (P-Speed, image generated per second per G of VRAM). (2) To measure the protection results, we use FID, LPIPS, Peak Signal-to-Noise Ratio (PSNR) (Huffman, 1952), and Image-Alignment Score (IA-score) (Kumari et al., 2023b) which calculated the cosine-similarity between the CLIP embedding of the protected image and the original image. Also, we have human evaluations which are collected using surveys, which is a more convincing way to evaluate the quality of protections, more settings can be found in the appendix.

C More Experimental Results

We also provide more supplementary results of our experiments. In Figure 11 and Figrue 12, we show more visualization of the SDEdit results of different protections, from which we can see that GD brings more natural perturbations than other methods and also show effective protections.

We also show results for inpainting in Figure 13, which is a more challenging task than SDEdit, since the mask is unknown during the attack. It turns out that, all the methods can effectively make the inpainted image unrealistic in different styles.

We also show more results to support our claim that the denoiser is quite robust, we directly attack the denoiser of the LDM using three different budgets: $\delta = 16, 32, 256$. In Figure 10 we show more results, which can be used to further prove that the denoiser itself is quite robust to adversarial attacks.

D Human Evaluations

To better evaluate the quality of perturbation of each protection method, and the strength of protection from a human level, we conducted a survey among humans with the assistance of the Google Form. We got responses from 53 individuals, 70% of them completed the survey on the computer and the rest of them completed the form on their mobile phones.

The user interface of the survey is shown in Figure 14, where we have two sections. The first section is used to evaluate the quality of perturbation, and the second section is for finding out the strength of protection from a human's perspective. The participants are asked to rank the given methods. We have 16 questions in total.

The scores are calculated using the rank of each method. For each question, the rank-1 will get 5 points and the lowest rank will get 1 points. The final score in Table 1 and Table 2 are calculated using the average score over all samples.

Figure 10: **Directly Attacking z-space:** we conduct experiments on three different budgets: $\delta = 16, 32, 256$ when directly attacking the latent representation in LDM. The first column is the attacked z-space latent projected back to x-space, and the following columns are results after SDEdit with an increasing editing strength. From the figure we can find that this kind of attack fails to work , the images after SDEdit still preserve better similarity as the attacked image, even when the budgets are getting as large as 256.

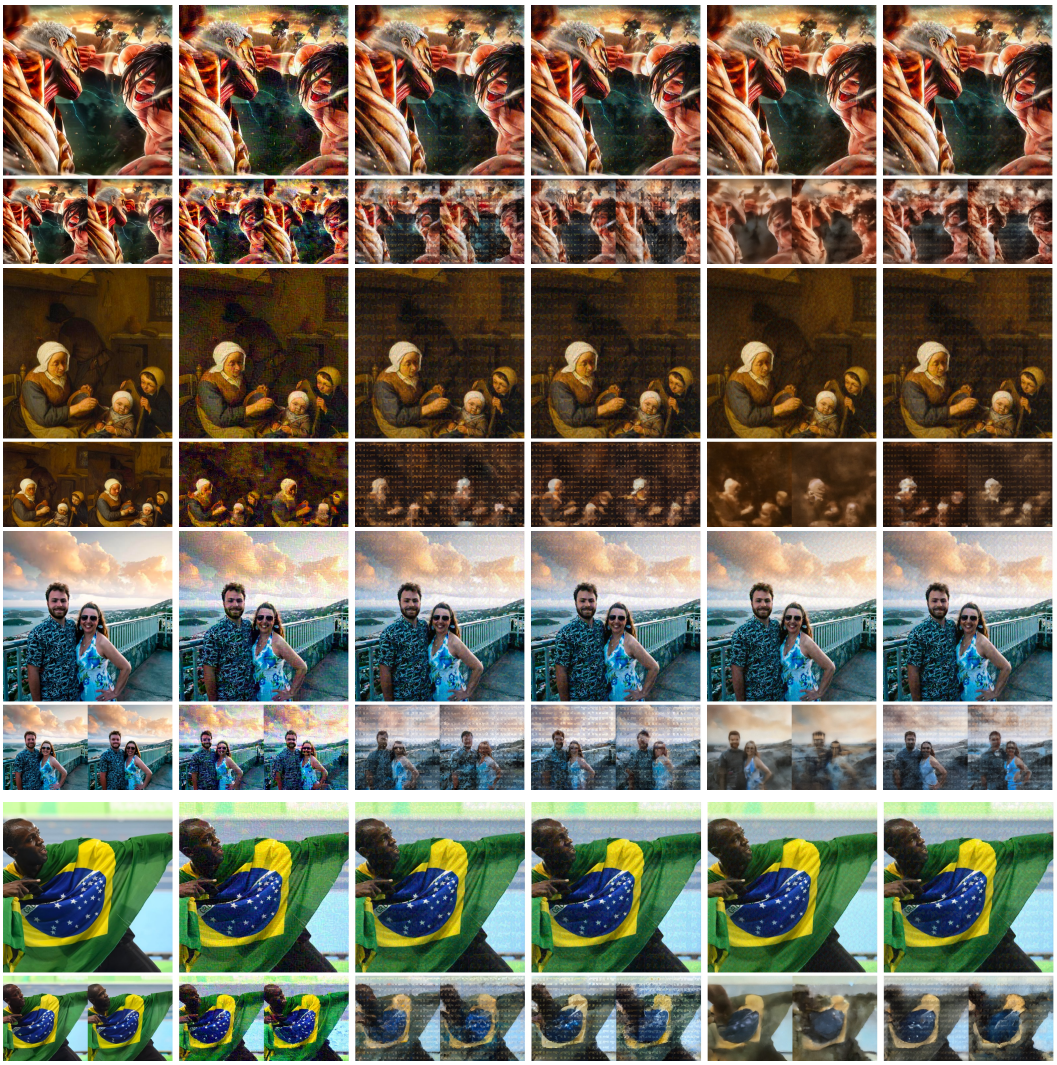

Figure 11: **More Results of Protection Against SDEdit (1/2)**: each column represents one protection method (including no protection), the two smaller figures below each protected image are generated using SDEdit, with two different strengths (the left one is smaller than the right one).

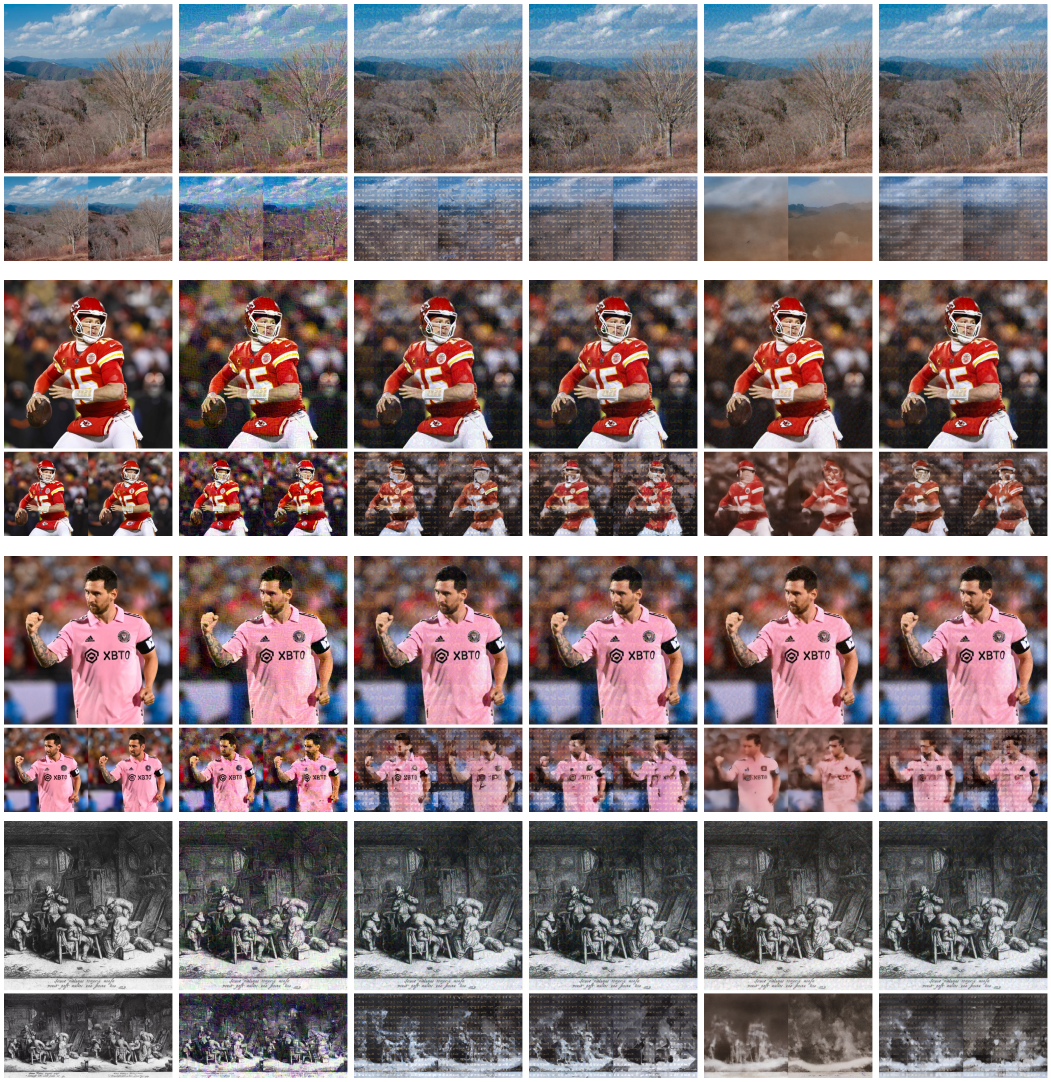

Figure 12: **More Results of Protection Against SDEdit (2/2)**: each column represents one protection method (including no protection), the two smaller figures below each protected image are generated using SDEdit, with two different strengths (the left one is smaller than the right one).

Figure 13: **More Results of Protection Against Inpainting**: From left to right: clean image, mask, clean inpainting, AdvDM, Mist, PhotoGuard, SDS(-), and SDST.

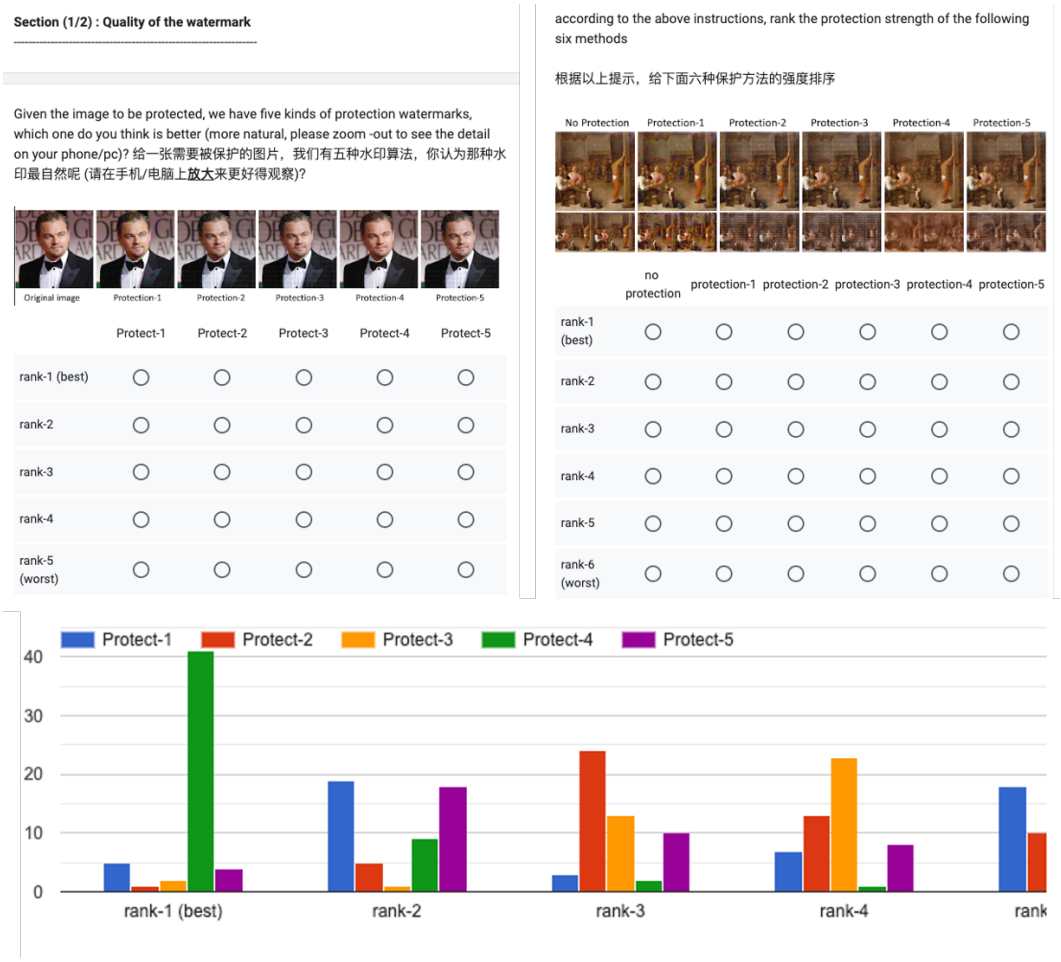

Figure 14: **Survey for Human Evaluation:** We show questions from our survey, the left one is used to evaluate which protection method looks more natural, and the right one is used to evaluate the strength of different protections. The second row demonstrates one statistical result of one question shown in our backstage.