UNLEARNABLE EXAMPLES FOR DIFFUSION MODELS: PROTECT DATA FROM UNAUTHORIZED EXPLOITATION

Zhengyue Zhao [1,2], **Jinhao Duan** [3], **Xing Hu** [1,4]*, **Kaidi Xu** [3], **Chenan Wang** [3]
Rui Zhang [1], **Zidong Du** [1,4], **Qi Guo**[1], **Yunji Chen** [1,2]
[1] SKL of Processors, Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] Drexel University
[4] Shanghai Innovation Center for Processor Technologies, SHIC
zhaozhengyue22@mails.ucas.ac.cn
{jd3734, kx46, cw3344}@drexel.edu
{huxing, zhangrui, duzidong, guoqi, cyj}@ict.ac.cn

ABSTRACT

Diffusion models have demonstrated remarkable performance in image generation tasks while also raising security and privacy concerns. To tackle these issues, we propose a method for generating unlearnable examples for diffusion models, Unlearnable Diffusion Perturbation, to safeguard images from unauthorized exploitation. Our approach involves designing an algorithm to generate sample-wise perturbation noise for each image to be protected. We frame this as a max-min optimization problem and introduce EUDP, a noise scheduler-based method to enhance the effectiveness of the protective noise. Our experiments demonstrate that training diffusion models on the protected data leads to a significant reduction in the quality of the generated images.

1 INTRODUCTION

In recent years, generative models such as GANs (Goodfellow et al., 2020; Brock et al., 2019) and VAEs (Kingma & Welling, 2014) have made significant strides in image synthesis tasks. Notably, Denoising Diffusion Probabilistic Model (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) and other diffusion models (Nichol & Dhariwal, 2021; Song et al., 2021; Bao et al., 2022; Song et al., 2021) have surpassed GANs in performance (Dhariwal & Nichol, 2021), becoming the state-of-the-art image synthesis method. Furthermore, the use of Latent Diffusion Models (LDM) (Rombach et al., 2022) has enhanced the ability to generate high-resolution images and perform multi-model tasks, such as text-to-image generation, leading to diverse AI-for-Art applications such as Stable Diffusion and MidJourney. While training a high-performance diffusion model from scratch remains a costly endeavor, there are effective fine-tuning techniques such as Textual Inversion (Gal et al., 2023) and DreamBooth (Ruiz et al., 2023) that allow for personalized diffusion models to be trained from large pre-trained models with minimal training overhead and small datasets.

However, the development of diffusion-based image synthesis methods has also given rise to security and privacy concerns. Abused unauthorized data exploitation is one major issue for generative models. For example, a pre-trained diffusion model can be fine-tuned with several personal facial images to generate fake images that could be harmful to the owner. Additionally, using artworks for training diffusion models could result in copyright infringement, dampening artists' creative enthusiasm. While artists may want to share their work on social networks, they may not want their work to be used for unauthorized exploitation, such as training a diffusion model (BBC, 2022; CNN, 2022; WashingtonPost, 2022). Therefore, it is imperative to protect artworks without hindering their normal usage. Unlearnable examples (Huang et al., 2021; Fu et al., 2022) are firstly proposed to protect data from being unauthorizedly trained by neural models for image classification via adding imperceptible perturbation to original images, which effectively safeguards the privacy and copyright of personal data.

*Corresponding author.

Motivated by unlearnable examples for image classification, we adapt unlearnable examples to diffusion models in this paper and propose Unlearnable Diffusion Perturbation (UDP) designed to protect images from being utilized to train a high-performance diffusion model by adding perturbation protective noise to images. The protected images have minimal differences compared to the original images but the diffusion model trained on these protected images is unable to generate the expected high-quality images. Additionally, we observe that modifying the sampling scheme of timesteps during the optimization process based on the noise scheduler of the diffusion process can further strengthen the protective effect of the noise. Building upon this observation, we introduce an improved method, Enhanced Unlearnable Diffusion Perturbation (EUDP) to further enhance the protection.

2 Problem Formalization

We formalize the normal diffusion model training and the protective unlearnable noise generation process as follows.

Normal diffusion model training: For a clean training image dataset $\boldsymbol{x} \in S_c$, which follows the distribution $\boldsymbol{x} \sim q^c(\boldsymbol{x})$, an image generator model $G_{\boldsymbol{\theta}}(\boldsymbol{x})$ is trained to generate images following the distribution $\hat{\boldsymbol{x}} \sim p^c_{\boldsymbol{\theta}}(\boldsymbol{x})$ that is as close as possible to $q^c(\boldsymbol{x})$. The distance between these two distributions $D(p^c_{\boldsymbol{\theta}}(\boldsymbol{x}), q^c(\boldsymbol{x}))$ can be evaluated by using some distance metrics, such as KL divergence.

Protective Noise Generation: By adding a small amount of protective noise $\boldsymbol{\delta}^u$ (bounded by ρ_u) to the clean training dataset, we obtain unlearnable examples $\boldsymbol{x}^u = \boldsymbol{x} + \boldsymbol{\delta}^u$, where $\boldsymbol{x}^u \in S_u$ and follow the distribution $\boldsymbol{x}^u \sim q^u(\boldsymbol{x})$. For the image generator model $G_{\boldsymbol{\theta}}(\boldsymbol{x}^u)$ trained on unlearnable data, the generated images follow the distribution $\hat{\boldsymbol{x}}^u \sim p^u_{\boldsymbol{\theta}}(\boldsymbol{x})$.

$$\max_{\|\boldsymbol{\delta}^u\| \leq \rho_u} D\left(p^u_{\boldsymbol{\theta}}(\boldsymbol{x}), q^c(\boldsymbol{x})\right) \tag{1}$$

To achieve the unlearnable effect, the design objective is to increase the distance in Eq. 1 between the distribution of generated images and the distribution of clean training data as much as possible by adjusting the protective noise $\boldsymbol{\delta}^u$.

3 Methodology

3.1 Unlearnable Diffusion Perturbation

We propose the Unleanable Difussion Perturbation (UDP) method for protective noise generation. For DDPM, the training images follow the distribution $q(\boldsymbol{x})$, while the generated images follow the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, with the optimization objective being the cross-entropy between these two distributions:

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{CE}} = \min_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{x}_0)}\left[-\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0)\right] \leq \min_{\boldsymbol{\theta}} \mathcal{L}_{\text{VLB}} = \min_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{x}_{0:T})}\left[\log \frac{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{0:T})}\right] \tag{2}$$

To address the optimization objective in Eq. 1, we approximate the optimization of the cross-entropy by optimizing the variational bound. To reduce the image generation quality of DDPM, we achieve this by maximizing the loss function during the training process. Similarly, we transform the problem of maximizing the optimal value of the cross-entropy in Eq. 2 into finding the maximum value of the minimum of the variational bound:

$$\begin{aligned} \max_{\boldsymbol{\delta}} \min_{\boldsymbol{\theta}} \mathcal{L}_{\text{CE}} &= \max_{\boldsymbol{\delta}} \min_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\delta}}(\boldsymbol{x}_0)}\left[-\log p_{\boldsymbol{\theta},\boldsymbol{\delta}}(\boldsymbol{x}_0)\right] \\ &\leq \max_{\boldsymbol{\delta}} \min_{\boldsymbol{\theta}} \mathcal{L}_{\text{VLB}} = \max_{\boldsymbol{\delta}} \min_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\delta}}(\boldsymbol{x}_{0:T})}\left[\log \frac{q_{\boldsymbol{\delta}}(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}{p_{\boldsymbol{\theta},\boldsymbol{\delta}}(\boldsymbol{x}_{0:T})}\right] \end{aligned} \tag{3}$$

Similar to the training of DDPM, we expand the terms of $\mathcal{L}_{\text{VLB}}$ in Eq. 3 and replace it with the simplified loss function $\mathcal{L}_{\text{simple}}$ in Eq. 9. Then the optimization objective is to solve the following bi-level max-min optimization problem:

$$\boldsymbol{\delta}^u = \arg \max_{\|\boldsymbol{\delta}^u_i\| \leq \rho_u} \min_{\boldsymbol{\theta}} \left(\sum_{\boldsymbol{x}_i} \mathbb{E}_{\boldsymbol{x}_{i,t_n} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}), t_n \sim \mathcal{U}(0,T)} \mathcal{L}_{t_n}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,t_n} + \boldsymbol{\delta}^u_i)\right) \right) \tag{4}$$

Where $\boldsymbol{\delta}^u$ represents the protective noise added to the training data $\boldsymbol{x}$, and its L_∞ norm is constrained by the preset value ρ_u to limit the impact of the protective noise on the original image. Considering that direct solution to the max-min problem in Eq. 4 is difficult, an iterative method of updating $\boldsymbol{\theta}$ and $\boldsymbol{\delta}_i^u$ alternatively can be used for optimization:

$$\begin{aligned} \boldsymbol{\theta} &\leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{x}} \mathcal{L}_{t_n} \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,t_n} + \boldsymbol{\delta}_i^u) \right) \\ \boldsymbol{\delta}_i^u &\leftarrow \boldsymbol{\delta}_i^u + \lambda \cdot \operatorname{sign} \left(\nabla_{\boldsymbol{x}} \mathcal{L}_{t_n} \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,t_n} + \boldsymbol{\delta}_i^u) \right) \right) \end{aligned} \tag{5}$$

Where $\boldsymbol{x}_i$ is randomly sampled from the training data and t is randomly sampled following uniform distribution $\mathcal{U}(0, T)$. The implementation of UDP is demonstrated in Appendix C.1.

3.2 Enhanced Unlearnable Diffusion Perturbation

Considering multiple iterative steps of DDPM in its forward process, we make the following two observations to help further improve the protective effect of UDP.

Observation 1: Decay Effect of Protective Noises Though all the iterative steps (t from 0 to T) are considered when computing protective noise, the protective noise can only be added to the image in the first step ($t = 0$). This causes the protective noise to be gradually overwhelmed in the forward process of DDPM:

$$\boldsymbol{x}_t + \boldsymbol{\delta}_t^u = \sqrt{\overline{\alpha_t}} \boldsymbol{x}_0 + \sqrt{1 - \overline{\alpha_t}} \boldsymbol{\epsilon}_t + \sqrt{\overline{\alpha_t}} \boldsymbol{\delta}_0^u \tag{6}$$

Eq. 6 shows that as t increases during the forward process, the impact of the protective noise will gradually decrease with decay coefficient $\sqrt{\overline{\alpha_t}}$. Since larger adversarial noise is more likely to have a greater impact on the model, the protective effect of the protective noise will decrease with the increase of t.

Observation 2: Varying Importance of Timestep Different step t have different effects on the training process of DDPM since the scale of the noise added at each step t during the forward process is different.

$$|\nabla_t \boldsymbol{x}_t| = |\boldsymbol{x}_t - \boldsymbol{x}_{t-1}| = \left(\sqrt{\overline{\alpha_{t-1}}} - \sqrt{\overline{\alpha_t}} \right) \boldsymbol{x}_0 + \sqrt{\overline{\alpha_{t-1}} - \overline{\alpha_t}} \boldsymbol{\epsilon}_t \tag{7}$$

Eq. 7 illustrates the variation of the same image between two adjacent steps in the forward process, which implies that the scale of added-noise at step t is $\sqrt{\overline{\alpha_{t-1}} - \overline{\alpha_t}}$. Following this, we can focus more on those t with larger added noise when computing the protective noise considering that a step t with a larger scale of added noise in the forward process may have a greater impact on the quality of the generated images.

Given the above two observations, we propose Enhanced Unlearnable Diffusion Perturbation (EUDP), a method for computing protective noise with sampling timesteps based on the value of the production of the decay coefficient of protective noise $\sqrt{\overline{\alpha_t}}$ and the scale of added noise in the forward process $\sqrt{\overline{\alpha_{t-1}} - \overline{\alpha_t}}$. Specifically, when solving the max optimization problem for the parameter $\boldsymbol{\delta}^u$ in Eq. 4, the uniform distribution of t is replaced with $\mathcal{P}_{\text{EUDP}}(t)$, a distribution based on the production $\sqrt{\overline{\alpha_t}} \cdot \sqrt{\overline{\alpha_{t-1}} - \overline{\alpha_t}}$:

$$\begin{aligned} \boldsymbol{\delta}^u &= \arg \max_{\|\boldsymbol{\delta}_i^u\| \leq \rho_u} \left(\sum_{\boldsymbol{x}_i} \mathbb{E}_{\boldsymbol{x}_{i,t_n} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}), t_n \sim \mathcal{P}_{\text{EUDP}}(t)} \mathcal{L}_{t_n} \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,t_n} + \boldsymbol{\delta}_i^u) \right) \right) \\ \text{where} \quad \mathcal{P}_{\text{EUDP}}(t) &= \frac{\sqrt{\overline{\alpha_t}} \cdot \sqrt{\overline{\alpha_{t-1}} - \overline{\alpha_t}}}{\sum_t \left(\sqrt{\overline{\alpha_t}} \cdot \sqrt{\overline{\alpha_{t-1}} - \overline{\alpha_t}} \right)} \end{aligned} \tag{8}$$

The sampling of timesteps in the minimization of $\boldsymbol{\theta}$ in Eq. 4 is still following a uniform distribution. Protective noise $\boldsymbol{\delta}^u$ and model parameter $\boldsymbol{\theta}$ in the bi-level max-min optimization is solved by the iterative method in Eq. 5. The pseudo-code of EUDP of DDPM is demonstrated in Appendix C.2.

4 Experiments

To evaluate the effectiveness of our method, we conduct a complete training (from scratch) on CIFAR-10 (Krizhevsky et al., 2009) dataset and utilize quantitative evaluations with metrics

Table 1: Quality of images generated by DDPM trained on CIFAR-10 with different protective methods. Higher FID and lower Precision indicate lower quality and diversity. The percentages in parentheses demonstrate the increase in FID as well as the decrease in Precision and Recall of our methods compared with AdvDM.

Noise Scale	**Metric**	**Methods**			**Methods (ours)**	
		Clean	Random	AdvDM	UDP	EUDP
16/255	FID↑	3.83	31.50	34.01	56.52 (+66.19%)	**60.75** (+78.62%)
	Precision(%)↓	71.77	45.12	47.60	36.04 (-24.29%)	**30.88** (-35.13%)
	Recall(%)↓	54.13	40.25	36.93	29.58 (-19.90%)	**23.81** (-35.53%)
32/255	FID↑	3.83	65.13	79.91	106.71 (+33.54%)	**112.67** (+41.00%)
	Precision(%)↓	71.77	25.25	35.94	31.27 (-13.00%)	**23.67** (-34.14%)
	Recall(%)↓	54.12	22.30	20.09	12.37 (-38.43%)	**10.47** (-47.88%)

FID (Heusel et al., 2017), Precision, and Recall (Sajjadi et al., 2018; Kynkäänniemi et al., 2019) to assess the quality of generated images. We also demonstrate the efficacy of EUDP in fine-tuning the LDM in Appendix E. We compare the quality of images generated by DDPM trained on different CIFAR-10, including clean, random noise added, AdvDM (Liang et al., 2023), UDP, and EUDP. We evaluate each DDPM by generating 50,000 images and testing their FID, Precision, and Recall, where a smaller FID, larger Precision, and Recall indicate higher quality of generated images.

Results demonstrated in Table 1 show that the FID of images generated by DDPM trained on the UDP dataset is significantly increased compared to random noise and AdvDM, while the Precision and Recall values are significantly reduced. This suggests that the UDP method effectively reduces the quality of generated images and successfully prevents the protected dataset from being used to train high-quality generative models. In addition, by using the EUDP method for protecting noise calculation, the quality of generated images is further reduced, indicating that EUDP can efficiently generate protective noise and achieve more effective results for protecting images.

We evaluate the protective effect at different protection ratios. Specifically, we randomly divided the CIFAR-10 into two sub-datasets in proportion: one clean and one protected. Images in the protected dataset are added with protected noise generated by EUDP. We combined the two sub-datasets to form the full training set with different protection ratios. The results in Figure 1 show that the protective effect increases as the protection ratio increases, and the protective effect is significant when the protection ratio reached 50%. We then add class-wise protective noise to the dataset. For the images of the 10 classes in the CIFAR-10, we only protect images with some of the classes. Table 2 shows the quality of the generated images of the corresponding protected and clean classes with class-wise added protective noise. The results show that the quality of the generated images of the protected classes is poor, while the quality of the generated images of the clean classes is similar to those generated on the clean CIFAR-10.

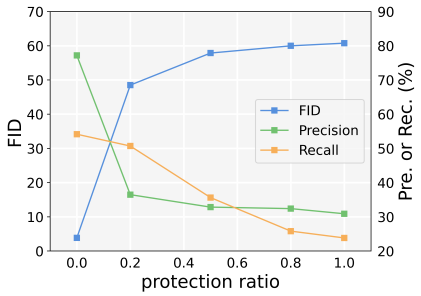


Figure 1: Quality of images generated by DDPM trained on EUDP CIFAR-10 with different protection ratio. FID increases while Precision and Recall decrease as the protection ratio increases.

5 Conclusion

In this paper, we propose image-protection methods UDP and EUDP for diffusion image generative models from a perspective of unlearnable examples which protects the privacy and copyright of the image owners. We demonstrate that our method can successfully achieve the goal of making the protected images "unlearnable" and has practical applications.

Table 2: Quality of images generated by DDPM trained on EUDP CIFAR-10 with different protected classes. The column of Protected Class indicates labels of protected samples in CIFAR-10 while Clean Class indicates labels of clean samples. Columns of Clean and Protected show results of DDPM trained on the clean CIFAR-10 and the class-wise protected CIFAR-10 respectively.

Noise Scale	Metric	Protected Class	Clean	Protected	Clean Class	Clean	Protected
16/255	FID↑	$0 \sim 1$	8.55	58.17	$2 \sim 9$	4.27	7.81
	Precision(%)↓		70.37	32.13		70.09	67.45
	Recall(%)↓		53.34	28.70		54.12	53.97
	FID↑	$0 \sim 4$	5.48	47.16	$5 \sim 9$	5.44	8.04
	Precision(%)↓		68.31	41.25		72.61	68.60
	Recall(%)↓		54.70	32.06		53.47	54.05
	FID↑	$0 \sim 7$	4.48	41.48	$8 \sim 9$	6.31	8.42
	Precision(%)↓		70.05	45.44		71.08	66.97
	Recall(%)↓		54.27	32.21		51.90	52.83

ACKNOWLEDGEMENT

This work is partially supported by the National Key R&D Program of China (under Grant 2021ZD0110102), the NSF of China (under Grants 62002338, 61925208, U22A2028, 62222214, 62102399, U19B2019), CAS Project for Young Scientists in Basic Research (YSBR-029), Youth Innovation Promotion Association CAS and Xplore Prize.

REFERENCES

Sahar Abdelnabi and Mario Fritz. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 121–140. IEEE, 2021.

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022.

BBC. "Art is dead Dude" - the rise of the AI artists stirs debate. `https://www.bbc.com/news/technology-62788725`, 2022.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

CNN. AI won an art contest, and artists are furious. `https://www.cnn.com/2022/09/03/tech/ai-art-fair-winner-controversy/index.html`, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. In *Advances in Neural Information Processing Systems*, volume 34, pp. 30339–30351, 2021.

Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pp. 3247–3258. PMLR, 2020.

Shaopeng Fu, Fengxiang He, Yang Liu, Li Shen, and Dacheng Tao. Robust unlearnable examples: Protecting data against adversarial learning. In *International Conference on Learning Representations*, 2022.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations*, 2023.

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)*, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Communications of the ACM*, volume 63, pp. 139–144. ACM New York, NY, USA, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *arXiv preprint arXiv:1412.6572*, 2014.

Xiao Guo, Xiaohong Liu, Zhiyuan Ren, Steven Grosz, Iacopo Masi, and Xiaoming Liu. Hierarchical fine-grained image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations*, 2021.

Zhengyuan Jiang, Jinghuai Zhang, and Neil Zhenqiang Gong. Evading watermark based detection of ai-generated content. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *aInternational Conference on Learning Representations*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Toronto, ON, Canada, 2009.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *International Conference on Machine Learning*, pp. 3866–3876. PMLR, 2019.

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*, 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

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*, pp. 10684–10695, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in neural information processing systems*, volume 31, 2018.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. In *International Conference on Machine Learning*, 2023.

Pedro Sandoval-Segura, Vasu Singla, Liam Fowl, Jonas Geiping, Micah Goldblum, David Jacobs, and Tom Goldstein. Poisons that are learned faster are more effective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 198–205, 2022.

Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image diffusion models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2022.

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

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.

Lue Tao, Lei Feng, Jinfeng Yi, Sheng-Jun Huang, and Songcan Chen. Better safe than sorry: Preventing delusive adversaries with adversarial training. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16209–16225, 2021.

Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc N Tran, and Anh Tran. Anti-dreambooth: Protecting users from personalized text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2116–2127, 2023.

Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

WashingtonPost. He made a children's book using AI. Then came the rage. `https://www.washingtonpost.com/technology/2023/01/19/ai-childrens-\book-controversy-chatgpt-midjourney/`, 2022.

WikiArt. Wikiart: Visual art encyclopedia. `https://www.wikiart.org/`, 2016.

Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Quanfu Fan, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. In *International Conference on Learning Representations*, 2018.

Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *European conference on computer vision*, pp. 665–681. Springer, 2020.

Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Availability attacks create shortcuts. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2367–2376, 2022.

Huan Zhang, Shiqi Wang, Kaidi Xu, Yihan Wang, Suman Jana, Cho-Jui Hsieh, and Zico Kolter. A branch and bound framework for stronger adversarial attacks of relu networks. In *International Conference on Machine Learning*, pp. 26591–26604. PMLR, 2022.

Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2185–2194, 2021.

A Related Works

A.1 Diffusion Models

Denoising Diffusion Probabilistic Model (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) is an image-generative diffusion model including a forward diffusion process and a reverse denoising process. In the forward process, an image $\boldsymbol{x}_0$ is gradually perturbated with random Gaussian noise for T steps and finally turns into random noise. In the reverse process, an image is conversely generated by gradually removing noise. Images at step t in the forward process can be expressed by images at step $t-1$: $\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}$, where α_t and $\beta_t = 1 - \alpha_t$ are usually pre-defined parameters, and $\boldsymbol{\epsilon}$ following standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Noise $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ denoised in the reverse process is learned by minimizing the following simplified loss function:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2\right] \tag{9}$$

Latent Diffusion Models (LDM) (Rombach et al., 2022) transfer diffusion processes from pixel space to latent space and introduce a cross-attention layer into model architecture to generate high-resolution images with general conditioning inputs. Latent noise denoised in the reverse process is learned by minimizing the following LDM loss function:

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \tau_{\boldsymbol{\theta}}(\boldsymbol{y}))\|^2\right] \tag{10}$$

Compared with Eq. 9, an image $\boldsymbol{x}$ is encoded into a latent vector by a pre-trained auto encoder $\mathcal{E}(\boldsymbol{x})$ and the output $\boldsymbol{z}$ of the reverse process is decoded into an image by a decoder $\mathcal{D}(\boldsymbol{z})$. A conditioning input y could be a set of text or an image and is encoded into a conditioning vector by a domain-specific encoder $\tau_{\boldsymbol{\theta}}$. To achieve the unlearnable examples for diffusion models, the aim of our UDP and EUDP is to hinder the optimization of loss in Eq. 9 and 10.

A.2 Unlearnable Examples for Classification Models

Recent research (Sandoval-Segura et al., 2022; Huang et al., 2021; Fu et al., 2022; Yu et al., 2022; Tao et al., 2021) show that poison attack on neural networks based classification models can effectively decrease the accuracy of classifiers on the test set. Unlearnable examples are generated by adding imperceptible adversarial noise to a clean dataset and classifiers trained on unlearnable examples fail to generalize to the unseen samples. The main approaches to generate unlearnable examples include error-minimization and error-maximization.

Error-Minimization Noise The basic idea of error-minimization noise (Huang et al., 2021; Fu et al., 2022) is to reduce the training loss of classifiers considering that less loss corresponds to less knowledge to learn.

$$\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} \min_{\|\boldsymbol{\delta}_i\|<\rho} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i), y_i) \tag{11}$$

As shown in Eq. 11, the error-minimization noise is generated by solving a bi-level min-min optimization problem, where the inner minimization is to find the bounded protected noise that minimizes the classification loss, while the outer minimization finds model parameters that also minimize the loss of the classifier.

Error-Maximization Noise (Fowl et al., 2021) shows that error-maximization noise generated during adversarial training in Eq. 12 is also significant to poor the performance of classifiers. The generation process of error-maximization noise is the same as adversarial training, which solves a bi-level min-max optimization problem. Different from the error-minimization noise, inner maximization finds noise that maximizes the loss of the classifier.

$$\max_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} \min_{\|\boldsymbol{\delta}_i\|<\rho} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i), y_i) \tag{12}$$

However, there is no general definition of unlearnable examples for generative models, which may hold greater practical significance compared to classification models due to the privacy concerns brought by AIGC.

A.3 Security & Privacy Protection for Diffusion Models

Adversarial Attacks for Diffusion Models Adversarial attacks (Goodfellow et al., 2014; Xu et al., 2018; 2020; Zhang et al., 2022; Li et al., 2019) have been introduced to diffusion models recently: (Salman et al., 2023) presents a targeted adversarial attack algorithm towards Latent Diffusion Models including encoder attack and diffusion attack. The encoder attack is to generate adversarial perturbations $\boldsymbol{\delta}$ by optimizing the distance between the latent code of perturbated samples $\mathcal{E}(\boldsymbol{x}+\boldsymbol{\delta})$ and a target latent vector z_{targ} while the diffusion attack optimizes the distance between samples generated by LDM and target images. This method can effectively prevent images from being modified by diffusion models in an image-to-image fashion. Liang et al. propose AdvDM (Liang et al., 2023), an untargeted adversarial examples generation method for diffusion models through Monte Carlo. AdvDM randomly samples different timesteps and latent variables to iteratively upgrade the adversarial noise. Though adversarial examples generated by AdvDM can partially disrupt the training of diffusion models, incorporating the training process of diffusion models into the generation of protective noise can greatly enhance this disruption.

Privacy Protection for Generative Models There is an increasing demand for privacy protection for generative models because of the fast development of image synthesis methods. (Gandikota et al., 2023) achieves the goal of preventing Diffusion Models from generating privacy-sensitive images by erasing the concepts within the models. Recent detection methods for DeepFakes (Fernandez et al., 2023; Guo et al., 2023; Jiang et al., 2023; Wang et al., 2023; Sha et al., 2022; Abdelnabi & Fritz, 2021; Frank et al., 2020; Zhao et al., 2021) can also help prevent the abuse of generative models, especially Diffusion Models, by distinguishing whether an image is generated by them. A more direct approach is to prevent images from being used to train image generative models. (Shan et al., 2023) proposes Glaze, a targeted adversarial attack on the feature extractor of text-to-image models, protecting artworks from style mimicry by misleading the match between text feature vectors and image feature vectors. In comparison to our method, this work does not specifically target the diffusion process, but focuses on text-to-image generative models and specifically addresses style transfer tasks. Van Le et al. (2023) propose Anti-DreamBooth to protect personal face images from being used to train LDM via the fine-tuning method DreamBooth. This work focuses on the specific fine-tuning scenario and the LDM but does not discuss it from an unlearnable aspect and the diffusion process itself.

B Threat Model

We consider two parties involved in the process: the Image Exploiter for diffusion model training and the Image Protector for the IP owner. We aim to design Image Protector to prevent the selected images from being utilized to train or fine-tune high-quality image models but with no harm to the available public data from being utilized. Specifically, we explain the workflow of Image Exploiter and Image Protector as follows:

Image Exploiter The Image Exploiter creates a training dataset for image generative models training or finetuning based on gathered public image resources. In practical settings, fine-tuning Diffusion Models usually includes two types: training specific objects or training specific styles. The former involves fine-tuning the model using a few images of a specific object to generate various images related to that object, while the latter involves training the model using a few images of a specific style for tasks such as style transfer. Generally, fine-tuning LDM requires the following knowledge: (1) a pre-trained LDM model, including the model structure and all parameters; (2) images of specific objects or styles, while IP of some styles or objects may be owned by other parties and unauthorized data exploitation should be eliminated.

Image Protector The Image Protector aims to prevent their images from being used to train or fine-tune high-quality image generative models (such as LDM) while still making them publicly available. Specifically, the Image Protector wishes to prevent their images from being used to generate false images of specific objects (DeepFake) or images of specific styles (copyright infringement). We assume that the Image Protector possesses the following knowledge: (1) the images that need to be protected; (2) the image generative models that the Image Exploiter may use, including the models' structures and parameters. This assumption is practically significant due to the fact that many

current pre-trained LDMs are fine-tuned on widely used LDMs such as *stable-diffusion-v1-4*. Additionally, we conducted transferability experiments on LDMs to demonstrate that this assumption can be relaxed.

C IMPLEMENTATION DETAILS

C.1 IMPLEMENTATION OF UDP

We demonstrate the pseudo-code of Unlearnable Diffusion Perturbation for DDPM and LDM in Algorithm 1 and Algorithm 2 respectively. As shown in Eq. 4 and Eq. 5, the bi-level max-min optimization problem can be solved by alternatively optimizing the parameters $\boldsymbol{\theta}$ of the model and the protective noise $\boldsymbol{\delta}^u$. Specifically, the parameters $\boldsymbol{\theta}$ are firstly optimized by minimizing the loss for K steps with learning rate η and then the perturbations $\boldsymbol{\delta}^u$ are iteratively calculated by maximizing the loss for M steps with perturbation rate λ. After each step of perturbation calculation, $\boldsymbol{\delta}^u$ will be clipped with the noise scale ρ_u. This bi-level optimization process will be iterated for N times until the perturbation can effectively protect images. Here K, M, N, η, and ρ_u are pre-defined hyper-parameters and the perturbation rate λ is typically set to 1/10 of the noise scale ρ_u. The implementations of UDP for DDPM and UDP for LDM are similar, with the main difference being that the loss in LDM (Eq. 9) is computed in the latent space and includes conditioning guidance $\boldsymbol{y}$ compared with DDPM (Eq. 10).

Algorithm 1 Unlearnable Diffusion Perturbation for DDPM

Input: Clean Dataset $\boldsymbol{x}_i \in \mathcal{D}_c$, U-Net Model $f_{\boldsymbol{\theta}}$, Max Perturbation Scale ρ_u, Iteration Steps N, Learning Rate η, Training Steps K, Perturbation Rate λ, Perturbation Steps M
Output: Protective Noise $\boldsymbol{\delta}^u$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$
$\boldsymbol{\delta}^u \leftarrow \boldsymbol{\delta}_0^u$
for $n = 1 \rightarrow N$ **do**
  for $k = 1 \rightarrow K$ **do**
    sample $t \sim \mathcal{U}(0, T)$, $\boldsymbol{x}_i \in \mathcal{D}_c$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{x}} \mathcal{L}_t \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,t} + \boldsymbol{\delta}_i^u)\right)$
  for $m = 1 \rightarrow M$ **do**
    for $\boldsymbol{x}_i$ in $\mathcal{D}_c$ **do**
      sample $t \sim \mathcal{U}(0, T)$
      $\boldsymbol{\delta}_i^u \leftarrow \boldsymbol{\delta}_i^u + \lambda \cdot \text{sign}\left(\nabla_{\boldsymbol{x}} \mathcal{L}_{t_n}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,t_n} + \boldsymbol{\delta}_i^u)\right)\right)$
      if $\|\boldsymbol{\delta}_i^u\| > \rho_u$ **then**
        $\boldsymbol{\delta}_i^u \leftarrow \text{sign}\left(\boldsymbol{\delta}_i^u\right) \cdot \rho_u$
return $\boldsymbol{\delta}_i^u$

C.2 IMPLEMENTATION OF EUDP

The pseudo-code of Enhanced Unlearnable Diffusion Perturbation for DDPM and LDM is described in Algorithm 3 and Algorithm 4 respectively. The implementation of EUDP is similar to UDP, but the sampling of timesteps t during the optimization of protective noise $\boldsymbol{\delta}^u$ follows the distribution of $\mathcal{P}_{\text{EUDP}}(t)$ as shown in Eq. 8 instead of the uniform distribution $\mathcal{U}(0, T)$.

D EXPERIMENT SETTINGS

D.1 PARAMETERS OF DIFFUSION MODELS

We train a DDPM from scratch and fine-tune an LDM with a pre-trained model. For DDPM training on CIFAR-10, we set the batch size to 128, the learning rate to 0.0001, and the number of epochs to 2000 ($\sim 80k$ steps in total). The noise scheduler in the diffusion process is the widely used linear scheduler with $\beta_0 = 0.0001$ and $\beta_T = 0.02$ where $T = 1000$. For LDM, we utilize *stable-*

Algorithm 2 Unlearnable Diffusion Perturbation for LDM

Input: Clean Dataset $\boldsymbol{x}_i \in \mathcal{D}_c$, pre-trained U-Net Model $f_{\boldsymbol{\theta}}$, pre-trained Encoder $\mathcal{E}$, Text Encoder $\tau_{\boldsymbol{\theta}}$, Conditional Guidance $\boldsymbol{y}$, Max Perturbation Scale ρ_u, Iteration Steps N, Learning Rate η, Training Steps K, Perturbation Rate λ, Perturbation Steps M

Output: Protective Noise $\boldsymbol{\delta}^u$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$
$\boldsymbol{\delta}^u \leftarrow \boldsymbol{\delta}_0^u$
for $n = 1 \rightarrow N$ **do**
  for $k = 1 \rightarrow K$ **do**
    sample $t \sim \mathcal{U}(0, T)$, $\boldsymbol{x}_i \in \mathcal{D}_c$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{x}} \mathcal{L}_t \left(f_{\boldsymbol{\theta}} \left(\mathcal{E} \left(\boldsymbol{x}_{i,t} + \boldsymbol{\delta}_i^u \right), \tau_{\boldsymbol{\theta}} \left(\boldsymbol{y} \right) \right) \right)$
  for $m = 1 \rightarrow M$ **do**
    for $\boldsymbol{x}_i$ in $\mathcal{D}_c$ **do**
      sample $t \sim \mathcal{U}(0, T)$
      $\boldsymbol{\delta}_i^u \leftarrow \boldsymbol{\delta}_i^u + \lambda \cdot \text{sign} \left(\nabla_{\boldsymbol{x}} \mathcal{L}_{t_n} \left(f_{\boldsymbol{\theta}} \left(\mathcal{E} \left(\boldsymbol{x}_{i,t} + \boldsymbol{\delta}_i^u \right), \tau_{\boldsymbol{\theta}} \left(\boldsymbol{y} \right) \right) \right) \right)$
      if $\|\boldsymbol{\delta}_i^u\| > \rho_u$ **then**
        $\boldsymbol{\delta}_i^u \leftarrow \text{sign} \left(\boldsymbol{\delta}_i^u \right) \cdot \rho_u$
return $\boldsymbol{\delta}_i^u$

Algorithm 3 Enhanced Unlearnable Diffusion Perturbation for DDPM

Input: Clean Dataset $\boldsymbol{x}_i \in \mathcal{D}_c$, U-Net Model $f_{\boldsymbol{\theta}}$, Max Perturbation Scale ρ_u, Iteration Steps N, Learning Rate η, Training Steps K, Perturbation Rate λ, Perturbation Steps M

Output: Protective Noise $\boldsymbol{\delta}^u$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$
$\boldsymbol{\delta}^u \leftarrow \boldsymbol{\delta}_0^u$
for $n = 1 \rightarrow N$ **do**
  for $k = 1 \rightarrow K$ **do**
    sample $t \sim \mathcal{U}(0, T)$, $\boldsymbol{x}_i \in \mathcal{D}_c$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{x}} \mathcal{L}_t \left(f_{\boldsymbol{\theta}} (\boldsymbol{x}_{i,t} + \boldsymbol{\delta}_i^u) \right)$
  for $m = 1 \rightarrow M$ **do**
    for $\boldsymbol{x}_i$ in $\mathcal{D}_c$ **do**
      sample $t \sim \mathcal{P}_{\text{EUDP}}(t)$
      $\boldsymbol{\delta}_i^u \leftarrow \boldsymbol{\delta}_i^u + \lambda \cdot \text{sign} \left(\nabla_{\boldsymbol{x}} \mathcal{L}_{t_n} \left(f_{\boldsymbol{\theta}} (\boldsymbol{x}_{i,t_n} + \boldsymbol{\delta}_i^u) \right) \right)$
      if $\|\boldsymbol{\delta}_i^u\| > \rho_u$ **then**
        $\boldsymbol{\delta}_i^u \leftarrow \text{sign} \left(\boldsymbol{\delta}_i^u \right) \cdot \rho_u$
return $\boldsymbol{\delta}_i^u$

Algorithm 4 Enhanced Unlearnable Diffusion Perturbation for LDM

Input: Clean Dataset $\boldsymbol{x}_i \in \mathcal{D}_c$, pre-trained U-Net Model $f_{\boldsymbol{\theta}}$, pre-trained Encoder $\mathcal{E}$, Text Encoder $\tau_{\boldsymbol{\theta}}$, Condition Guidance $\boldsymbol{y}$, Max Perturbation Scale ρ_u, Iteration Steps N, Learning Rate η, Training Steps K, Perturbation Rate λ, Perturbation Steps M

Output: Protective Noise $\boldsymbol{\delta}^u$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$
$\boldsymbol{\delta}^u \leftarrow \boldsymbol{\delta}_0^u$
for $n = 1 \rightarrow N$ **do**
  for $k = 1 \rightarrow K$ **do**
    sample $t \sim \mathcal{U}(0, T)$, $\boldsymbol{x}_i \in \mathcal{D}_c$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{x}} \mathcal{L}_t \left(f_{\boldsymbol{\theta}} \left(\mathcal{E} \left(\boldsymbol{x}_{i,t} + \boldsymbol{\delta}_i^u \right), \tau_{\boldsymbol{\theta}} \left(\boldsymbol{y} \right) \right) \right)$
  for $m = 1 \rightarrow M$ **do**
    for $\boldsymbol{x}_i$ in $\mathcal{D}_c$ **do**
      sample $t \sim \mathcal{P}_{\text{EUDP}}(t)$
      $\boldsymbol{\delta}_i^u \leftarrow \boldsymbol{\delta}_i^u + \lambda \cdot \text{sign} \left(\nabla_{\boldsymbol{x}} \mathcal{L}_{t_n} \left(f_{\boldsymbol{\theta}} \left(\mathcal{E} \left(\boldsymbol{x}_{i,t} + \boldsymbol{\delta}_i^u \right), \tau_{\boldsymbol{\theta}} \left(\boldsymbol{y} \right) \right) \right) \right)$
      if $\|\boldsymbol{\delta}_i^u\| > \rho_u$ **then**
        $\boldsymbol{\delta}_i^u \leftarrow \text{sign} \left(\boldsymbol{\delta}_i^u \right) \cdot \rho_u$
return $\boldsymbol{\delta}_i^u$

diffusion-v1-4 [1] as the base model and fine-tune it with different parameters for DreamBooth (Ruiz et al., 2023) and Textual Inversion (Gal et al., 2023). Specifically, we fine-tune the model with DreamBooth with batch size set to 1, learning rate set to 4×10^{-6}, and steps to 400 and Textual Inversion with batch size set to 1, learning rate set to 5×10^{-4}, and steps to 5000.

D.2 Parameters of UDP & EUDP

For UDP and EUDP for DDPM in Algorithm 1 and 3, we set parameters N to 100, K to 1000, M to 10, and η to 0.0001. Due to the significant training cost, we do not conduct systematic ablation experiments on these parameters. Empirically, we find that setting $N \times M \times \lambda$ to 100 times of the scale of the noise ρ_u can ensure the protective effect of the perturbation. For experiments of LDM, we set N to 40, K to 20, M to 25, and η to 5×10^{-6} in Algorithm 2 and 4. The noise scale of protective noise in all experiments of LDM is 16/255.

D.3 Other Settings

We use the *diffuser* library [2] for the training and inference of diffusion models and utilize this code [3] to evaluate the quality of generated images with FID, Precision, and Recall. For the baseline AdvDM (Liang et al., 2023) in the quantitative experiments of DDPM, we set the iteration steps to 128 to ensure sufficient generation of adversarial noise.

E Results on Latent Diffusion Model

For LDM experiments, we fine-tune a widely used pre-trained LDM *stable-diffusion-v1-4* (Rombach et al., 2022) with datasets provided by DreamBooth (Ruiz et al., 2023) and WikiArt (WikiArt, 2016). In LDM experiments, we qualitatively demonstrate the effectiveness of our method through image visualization results. For the task of training an object, we fine-tune the LDM using the DreamBooth dataset with fine-tuning methods Textual Inversion and DreamBooth. As shown in Figure 2, the LDM fine-tuned on the clean dataset is able to generate images that match the given prompt. However, the LDM fine-tuned on the EUDP dataset failed to generate the expected images. Notably, images generated by Textual Inversion have nearly no features of the training images, whereas those generated by DreamBooth are of low quality and repeat the training set, failing to meet the prompt. This demonstrates that our method successfully protects images of specific objects from being used to train a high-quality LDM. Moreover, Figure 3 shows that protecting a specific label hardly affects the generative quality of label fine-tuning on the clean dataset.

In the scenario of training a style, we apply DreamBooth to fine-tune the LDM using the WikiArt dataset, evaluating it on both text-to-image and image-to-image (style transfer) tasks. Specifically, we selected six paintings by a particular artist (such as Monet) for style training. As shown in Figure 4 and Figure 5, LDMs trained on the clean dataset are able to generate images with a specific artist's style and perform style transfer. Conversely, LDMs trained on the unlearnable dataset are unable to generate images with the corresponding style or convert the source image to a specific style. This experiment demonstrates that our method successfully protects specific styles, preventing infringement of copyright by unauthorized data exploitation such as style mimicry.

F Further Analysis

F.1 Robustness Study

We evaluate the effectiveness of the protective noise after adding natural perturbations, including random noise, quantification, JPEG, and Gaussian blur. Specifically, the random noise has a scale of 16/255. Quantization involves reducing an 8-bit image to a 6-bit image. Gaussian blur is performed using a filter kernel with kernel size 4x4 and with σ =16/255. JPEG compression and decompression are carried out using the "imencode" and "imdecode" functions from the *OpenCV2* library

[1] https://huggingface.co/CompVis/stable-diffusion-v1-4
[2] https://github.com/huggingface/diffusers
[3] https://github.com/openai/guided-diffusion/tree/main/evaluations

(a) Clean and protected images for fine-tuning a Stable Diffusion model

(b) Generated images of Stable Diffusion fine-tuned with DreamBooth

(c) Generated images of Stable Diffusion fine-tuned with Textual Inversion

Figure 2: An example of text-to-image with object protection. (b)&(c): The first row: Generated images of Stable Diffusion fine-tuned on the clean dataset. The second row: Generated images of Stable Diffusion fine-tuned on the EUDP dataset.

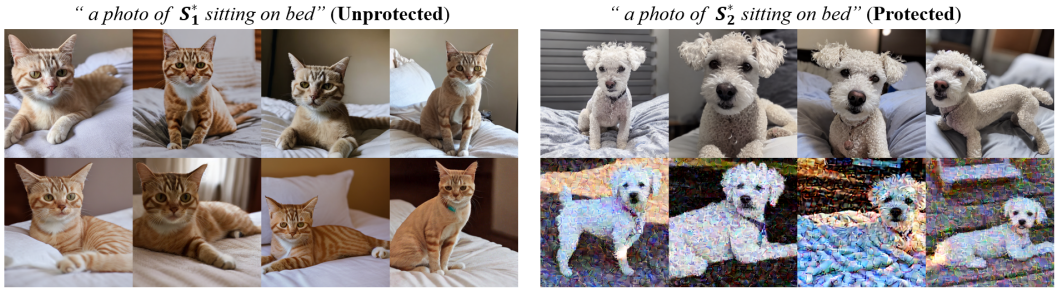


Figure 3: An example of text-to-image with class-wise protection. The first row: Generated images of Stable Diffusion fine-tuned on the clean dataset. The second row: Generated images of Stable Diffusion fine-tuned on class-wise EUDP dataset, where training images with label S_1^* are clean while training images with label S_2^* are protected by EUDP.

Figure 4: An example of text-to-image with specific style. The first row: Generated images of Stable Diffusion fine-tuned on the clean dataset. The second row: Generated images of Stable Diffusion fine-tuned on the EUDP dataset.

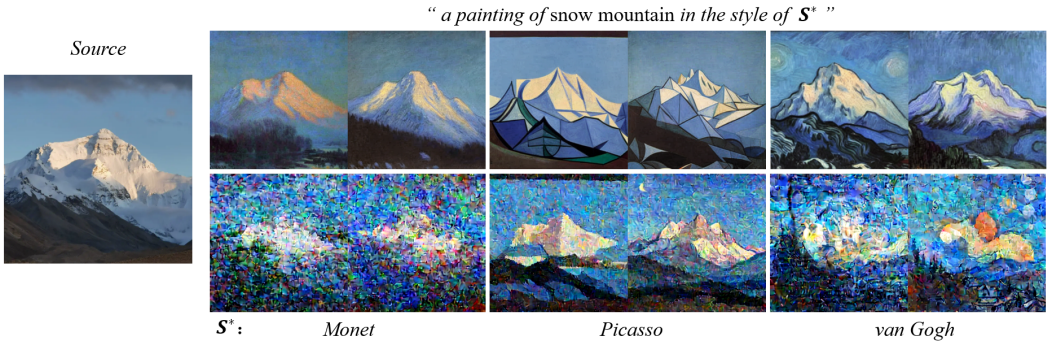


Figure 5: An example of style transfer. The first row: Generated images of Stable Diffusion fine-tuned on the clean dataset. The second row: Generated images of Stable Diffusion fine-tuned on the EUDP dataset.

[4]. Results in Table 3 show that JPEG compression and Gaussian blurring can improve the quality of generated images to some extent, but it is challenging to achieve the same level of high-quality generated images as in the (original) clean dataset.

Table 3: Quality of images generated by DDPM trained on CIFAR-10 with or without natural perturbation.

Metric	**Control**		**Natural Perturbation**			
	Clean	EUDP	Random Noise	JPEG	Quantify	Gaussian blur
FID↑	3.83	60.75	102.95	27.61	62.46	41.88
Precision(%)↓	71.77	30.88	20.78	55.23	31.35	60.59
Recall(%)↓	54.13	23.81	11.35	40.87	26.66	33.78

F.2 Transferability Study

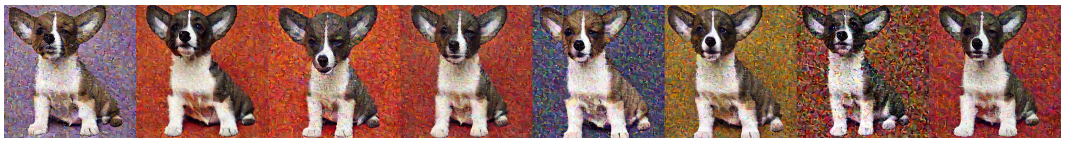
We conduct transferability studies to evaluate the effectiveness of our methods in black-box conditions. We first assess the transferability of the noise scheduler of EUDP for DDPM. Protective noise is generated by EUDP with $\beta_0 = 0.0001$ and $\beta_T = 0.02$ and the protected images are tested with other noise schedulers. Results demonstrated in Table 4 indicate that changing the noise scheduler does not significantly diminish the effectiveness of the protection noise. Additionally, it can be observed from the experimental results that the EUDP method, corresponding to a noise scheduler with faster noise adding (larger β_T) during the training process, exhibits a greater improvement in protection compared to the UDP method, which also confirms the validity of our observations in Section 3.2. For real-world conditions, we conduct transferability studies of our methods for LDM. We first examine the transferability of EUDP between different LDMs. Specifically, the protective noise is generated with *stable-diffusion-v1-4* and the protected images are learned by pre-trained LDMs *stable-diffusion-v1-1* [5], *stable-diffusion-v1-5*[6], and *Counterfeit-V2.5*[7]. Results in Figure 6 show that protective noise generated by a specific model remains effective in protecting images from being learned by other models. Furthermore, when the model used for training is closer to the model for protection noise generation, the protective effect is better (*stable-diffusion-v1-1*). Conversely, when there is a significant difference between the two models, the protection effect is weaker (*Counterfeit-V2.5*). Figure 7 demonstrates that the text prompt used for protective noise generation can be different from the text prompt for fine-tuning diffusion models since our methods mainly focus on the diffusion process instead of the text encoder.

[4] https://github.com/abidrahmank/OpenCV2-Python-Tutorials
[5] https://huggingface.co/CompVis/stable-diffusion-v1-1
[6] https://huggingface.co/runwayml/stable-diffusion-v1-5
[7] https://huggingface.co/gsdf/Counterfeit-V2.5

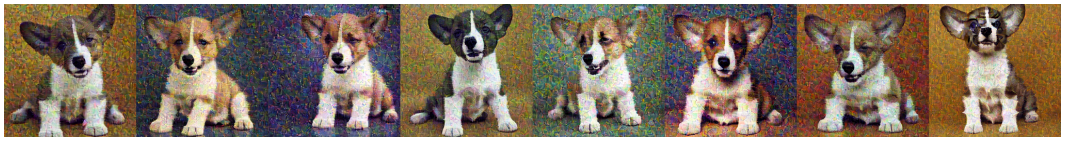
Table 4: Quality of images generated by DDPM trained on UDP or EUDP CIFAR-10 with different noise schedulers. The protective noise is generated with a noise scheduler where $\beta_0 = 0.0001$ and $\beta_T = 0.02$.

Noise Scale	**Metric**	**Noise Scheduler**	**Methods**		
			Clean	UDP	EUDP
16/255	FID↑	$\beta_0 = 0.0001$ $\beta_T = 0.015$	6.10	57.59	58.49
	Precision(%)↓		69.52	36.40	30.59
	Recall(%)↓		53.25	24.74	24.12
	FID↑	$\beta_0 = 0.0001$ $\beta_T = 0.020$	3.83	56.52	60.75
	Precision(%)↓		71.77	36.04	30.88
	Recall(%)↓		54.13	29.58	23.81
	FID↑	$\beta_0 = 0.0001$ $\beta_T = 0.025$	5.84	54.89	60.47
	Precision(%)↓		67.92	36.16	31.27
	Recall(%)↓		55.30	26.20	24.35

(a) Generated images with *stable-diffusion-v-1-4* trained on clean data



(b) Generated images with *stable-diffusion-v-1-4* trained on protected data



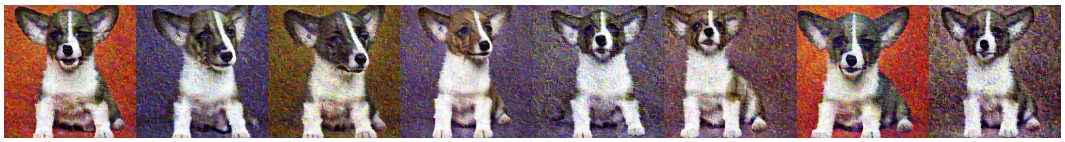
(c) Generated images with *stable-diffusion-v-1-1* trained on protected data

(d) Generated images with *stable-diffusion-v-1-5* trained on protected data

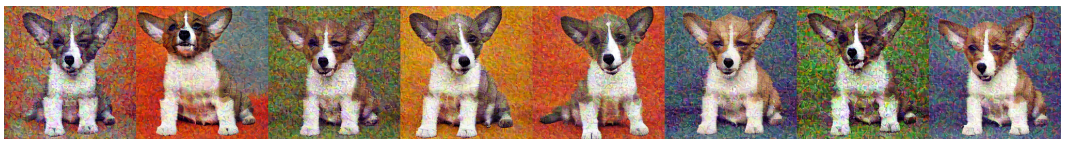
(e) Generated images with *Counterfeit-V2.5* trained on protected data

Figure 6: Generated images with different LDMs trained on clean or protected data and text promp for generation is *"A photo of S^* dog sitting on bed"*. The protective noise of protected data is generated by EUDP with *stable-diffusion-v1-4*.

(a) $S^* = AAA$



(b) $S^* = BBB$



(c) $S^* = CCC$

Figure 7: Generated images with *stable-diffusion-v1-4* trained on protected data. The text prompt for training is *"A photo of S^* dog"* and the text prompt for for generation is *"A photo of S^* dog sitting on bed"* while the text prompt for protective nosie generation is *"A photo of sks dog"*.

F.3 More Visualization

Here we show some more visualization results of protected images, text-to-image, and style mimicry. We demonstrate the clean artworks and EUDP-protected artworks of different artists including Monet, Picasso, and van Gogh in Figure 8, text-to-image with specific styles in Figure 9, and style mimicry (image-to-image) in Figure 10, respectively.

(a) Clean data of Monet (b) Clean data of Picasso (c) Clean data of van Gogh

(d) Protected data of Monet (e) Protected data of Picasso (f) Protected data of van Gogh

Figure 8: Clean and protected artworks of different artists.

" a painting of a man riding a horse in the style of $\boldsymbol{S}^*$ *"*

$\boldsymbol{S}^*$: *Monet* *Picasso* *van Gogh*

Figure 9: An example of text-to-image with specific styles. The first row: Generated images of Stable Diffusion fine-tuned on the clean dataset. The second row: Generated images of Stable Diffusion fine-tuned on the EUDP dataset.

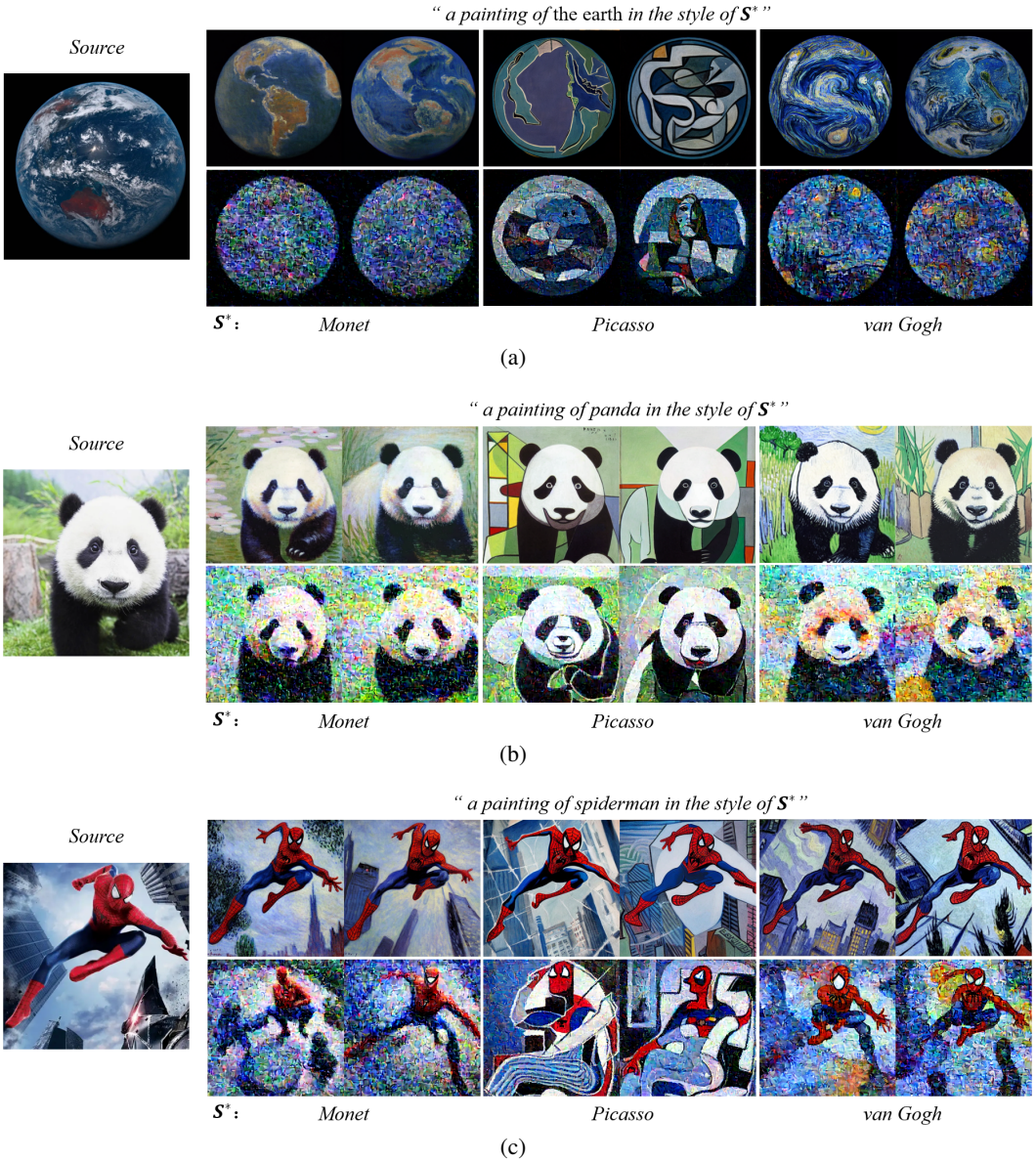


Figure 10: Examples of style transfer. The first row: Generated images of Stable Diffusion fine-tuned on the clean dataset. The second row: Generated images of Stable Diffusion fine-tuned on the EUDP dataset.