Noise-Proofed Nightshade: Improving Upon Poisoning Attacks in Diffusion Models

Emily Sturman
sturman@utexas.edu

1. Abstract

In response to ethics concerns regarding the usage of copyrighted and uncredited work in generative diffusion models, tools such as Nightshade [7] have been created. These tools are able to "poison" entire models by corrupting data in a way that is nearly unidentifiable by humans. However, simple denoising techniques are able to significantly reduce the efficacy of these tools. We introduce Noise-Proofed Nightshade, a modification to traditional Nightshade that is nearly unaffected by denoising defenses.

All code is available at: github.com/Creampelt/noise-proofed-nightshade

2. Introduction

With the rise of generative AI in the past two years, the use of copyrighted and uncredited work in model training has been a major topic of debate. This is particularly the case when these models can be used to replace the work of an artist [4] or spread misinformation [3]. Additionally, many of the companies controlling these models are large, powerful, and have in the past shown a disregard for users' privacy by mishandling and misusing customer data [1]. While some model creators do provide the ability for artists to opt out of having their work be used in training, the practice is largely unregulated and is left up to the discretion of each company or organization. To combat this, model-poisoning tools such as Nightshade [7] allow content creators to take an offensive position against copyright infringement, as model creators that ignore opt-outs could potentially corrupt their entire models [2].

On the reverse side, poisoning techniques in image-to-text models prove to be concerning with regard to the spread of misinformation. Shadowcast, an image-to-text poisoner that employs a similar mechanic to Nightshade, demonstrates how poisoned data can lead to both "Label Attacks" (class misidentification) and "Persuasion Attacks" (longer form misconceptions) [9]. Our modification to Nightshade is also able to demonstrate how an attacker could proof Shadowcast against simple defenses such as denoising.

3. Background

Diffusion models have rapidly gained popularity in the past few years due to their state-of-the-art image generation capabilities. They work by first encoding an image into a latent feature space through the addition of Gaussian noise, then learning to decode samples from latent space back into images (see Fig. 1). Thus, two images that are similar in latent space will, in theory, be decoded similarly by the diffusion model.

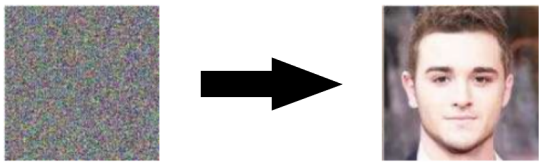


Figure 1. Decoding of an Image from Latent Space [5]

Nightshade is able to poison a base image $x_{\mathcal{C}}$ in concept $\mathcal{C}$ to an anchor image $x_{\mathcal{A}}$ in concept $\mathcal{A}$ by modifying $x_{\mathcal{C}}$ to appear similar to $x_{\mathcal{A}}$ in latent space. However, the attack would be ineffective if the modified $x'_{\mathcal{C}}$ were very different from $x_{\mathcal{C}}$, so the perceptible distance between the base and poisoned image must also be minimized. Given our image encoder F and a perturbation budget p, this presents the following optimization problem [8]:

$$\min_{\delta} \|F(x_{\mathcal{C}}+\delta)-F(x_{\mathcal{A}})\|_2^2+\alpha\cdot\max(\|\delta\|_{\text{LPIPS}}-p,0). \quad (1)$$

Note that $\|\delta\|_{\text{LPIPS}}$ is the Learned Perceptual Image Patch Similarity metric, as presented by Zhang et al. [10]. Similarly, Shadowcast uses a constrained optimization problem [9]:

$$\min_{\delta} \|F(x_{\mathcal{C}}+\delta)-F(x_{\mathcal{A}})\|_2, \quad \text{s.t. } \|\delta\|_\infty \leq p. \quad (2)$$

We will be building off of a variant of (1):

$$\min_{\delta} \|F(x_{\mathcal{C}}+\delta)-F(x_{\mathcal{A}})\|_2+\alpha\cdot\max(\|\delta\|_{\text{LPIPS}}-p,0). \quad (3)$$

4. Methods

Using (3) as our model for standard Nightshade poisoning, we present two modified optimization problems that both serve to reduce the effect of a denoising algorithm.

4.1. Noise-Proofed Nightshade

Given a denoising algorithm D, we can defend against denoising by minimizing the distance between the *denoised* base image and anchor image in latent space. That is, our optimization problem becomes

$$\min_{\delta} \|F(D(x_{\mathcal{C}}+\delta))-F(x_{\mathcal{A}})\|_2+\alpha\cdot\max(\|\delta\|_{\text{LPIPS}}-p,0). \quad (4)$$

In practice, we also refer to this method as **Gaussian-Proofed Nightshade**, as Gaussian blur is the denoising function that we use when implementing this method.

4.2. Denoised Nightshade

Given a noisiness metric $\mathbf{n}$, we can attempt to minimize the amount of generated noise in the poisoned image itself, while still minimizing the distance between the poisoned image and its anchor in latent space. This gives us the following optimization problem:

$$\min_{\delta} \|F(x_{\mathcal{C}}+\delta)-F(x_{\mathcal{A}})\|_2+\alpha\cdot\max(\|\delta\|_{\text{LPIPS}}-p,0) + \beta\cdot\mathbf{n}(x_{\mathcal{C}}+\delta). \quad (5)$$

Note that some figures refer to this method as **Noisy Nightshade**.

4.3. Evaluating Poisoning Performance

A major major issue when attempting to evaluate the performance of these methods is that of resources: in particular, unlike the developers of Nightshade, we do not have access to nearly enough compute power to train our own diffusion models. Instead, given that Nightshade has already been proven to work, we will instead use the distance between the poisoned and anchor images in latent space as a measure of a poisoning algorithm's **efficacy**. That is,

$$\mathbf{e}(x'_{\mathcal{C}}) = 1-\frac{\|F(x'_{\mathcal{C}})-F(x_{\mathcal{A}})\|_2}{\|F(x_{\mathcal{C}}-F(x_{\mathcal{A}})\|_2} \quad (6)$$

In this way, an efficacy of 1 signifies that the poisoned image is identical to the anchor in latent space, and an efficacy of 0 signifies that the poisoned image is unchanged from the base in latent space.

It is important to note that this metric is imperfect — because Nightshade is closed source, we are unable to correlate efficacy with a number of poisoned samples required to poison a model. Additionally, we do not know if efficacy is linearly correlated with the required number of poisoned samples (in fact, we believe that it is not).

5. Experiments

We tested our three poisoning methods — standard Nightshade, Noise-Proofed Nightshade, and Denoised Nightshade — on a single base and anchor image. We chose our base concept space $\mathcal{C}$ to be "cubism" and our anchor concept space $\mathcal{A}$ to be "anime." We initially chose $\mathcal{C}$ to be "cat" and $\mathcal{A}$ to be "dog"; however, given that we would expect poisoning to be applied primarily to artwork, we chose art style concepts to poison instead.

Figure 2. Base and Anchor Images

With $\mathcal{C}$ and $\mathcal{A}$ chosen, we retrieved $x_{\mathcal{C}}$ through an image search for "cubism art." We generated $x_{\mathcal{A}}$ from a pretrained *Stable Diffusion XL* model with the prompt "anime art." For our image encoder, we loaded the pretrained Variational Autoencoder from the *Stable Diffusion 2* model — while we initially intended to use SDXL's VAE, there were a number of bugs that we encountered that motivated us to switch to SD-2's.

We then ran our three poisoning algorithms on these images. As mentioned previously, we used Gaussian blur for the denoising algorithm in the Noise-Proofed Nightshade method. For Denoised Nightshade, we used the noisiness metric laid out by Immerkær in [6]:

$$\mathbf{n}(I) = \frac{\sqrt{\pi/2}}{6(W-2)(H-2)}\sum_{I}|I(x,y)*F| \quad (7)$$

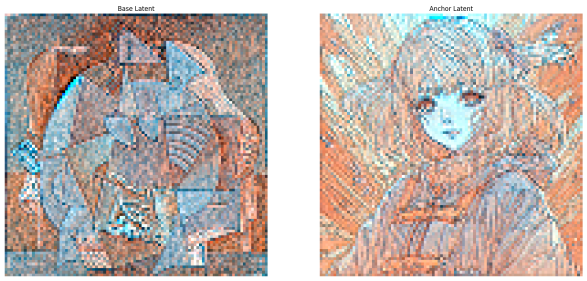


Figure 3. Encoded Base and Anchor (using SD-2's VAE)

where

$$F = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}. \tag{8}$$

Lastly, to determine the quality of each algorithm when denoised, we recorded each poisoned image's efficacy both when unaltered and denoised (using OpenCV's *fastNlMeansDenoisingColored* algorithm).

6. Results

For all three of our Nightshade implementations, the unaltered poisoned images had similar efficacies, ranging between 51% and 55% efficacy (see Fig. 4a). To contrast this, the standard Nightshade's efficacy was reduced to 45% when denoised. Denoised Nightshade improved this score, with its efficacy being reduced only to 45%. However, Noise-Proofed Nightshade's efficacy was completely unaffected by denoising, remaining at 51% (see Fig. 4c).

We also observed that our new Nighshade implementations resulted in a higher perceptual perturbation than the standard, with Noise-Proofed and Denoised having an LPIPS distance of 0.35 and 0.40 respectively (in comparison to standard Nightshade's 0.30) (see Fig. 4b). Altogether, we can conclude that **Noise-Proofed Nightshade** is the most effective method for protecting against denoising.

7. Future Work

There is a significant amount of future work to be done in this subject that we were unable to accomplish within the constraints of this problem. As previously mentioned, we were quite limited in terms of compute power. In the future, we hope to purchase compute units in order to directly test poisoning efficacy through number of samples to poison, rather than relying on our efficacy metric. Additionally, we currently only encode our images once, since each pass through the encoder was fairly computationally expensive — with more compute units, we could encode them multiple times in order to generate a more accurate representation of the latent samples used in practice by diffusion models. Lastly, we only tested our implementation on a single image (once again due to resource constraints). We would be interested to compare the efficacy of these algorithms on different kinds and styles of images, as well as ensure that the results we observed could carry over to other images as well.

Another area of future work would be developing the denoising function and noise metric used in Noise-Proofed and Denoised Nightshade. We used Gaussian blurring as our denoising function primarily because it was the most readily available denoising function that could work with PyTorch. However, it would be interesting to test the optimization with other custom-implemented denoising functions from non-machine learning libraries (such as *fastNlMeansDenoisingColored*). Additionally, the noise metric we used primarily tests for Gaussian noise — we could improve upon this by finding metrics for other types of noise.

Lastly, denoising is certainly not the only method for protection against model poisoning. The creators of Nightshade outline possible defenses, including gradient-based outlier detection, frequency analysis, and others [8]. Further evaluating the efficacy of these defenses and proofing Nightshade against them is another area of work that we would like to explore in the future.

8. Conclusion

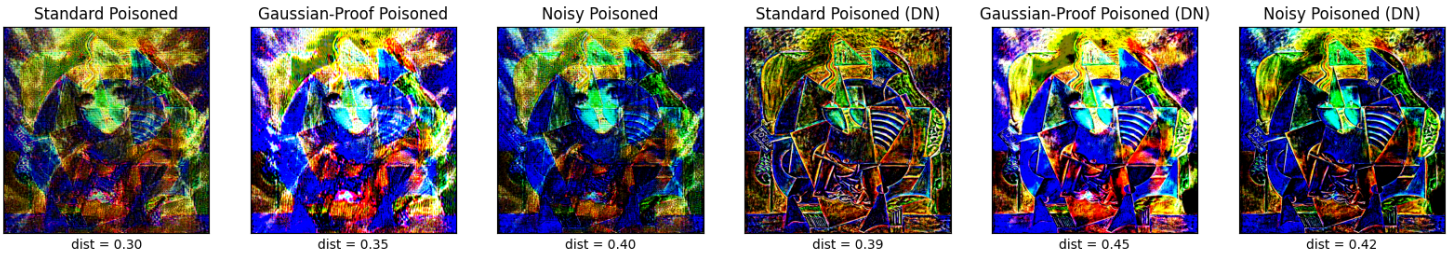
Overall, tools such as Nightshade have proven to be an important first step in helping artists protect their work from unauthorized usage in diffusion model training. However, these tools are not without their flaws, which model creators are likely to use in order to protect the efficacy of their models. We created Noise-Proofed Nightshade, a modification to Nightshade that can almost completely reduce the effects of denoising on poisoning. We also outlined many opportunities for future work on this subject, which may further allow creators to feel secure in posting their art publicly.

References

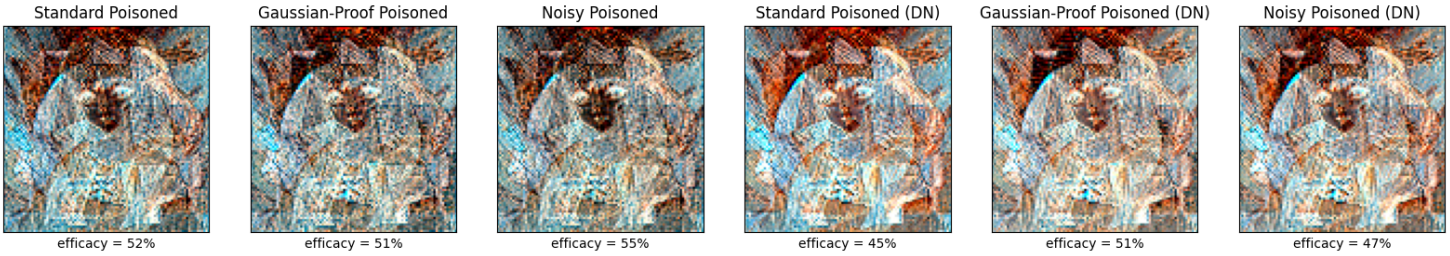
[1] Nicholas Confessore. Cambridge analytica and facebook: The scandal and the fallout so far. www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html, 2018. 1

[2] Melissa Heikkilä. This new data poisoning tool lets artists fight back against generative AI.

(a) Poisoned Images



(b) Poisoned Image Residuals (Normalized)



(c) Latent Residuals (Normalized)

Figure 4. Results (Unaltered and Denoised)

www.technologyreview.com/2023/10/23/1082189/data-poisoning-artists-fight-generative-ai, 2023. 1

[3] Tiffany Hsu and Stuart A. Thompson. Disinformation researchers raise alarms about A.I. chatbots. www.nytimes.com/2023/02/08/technology/ai-chatbots-disinformation.html, 2023. 1

[4] Tim Lammers. AI in 'Late Night With The Devil' sparks controversy. www.forbes.com/sites/timlammers/2024/03/22/use-of-ai-in-late-night-with-the-devil-draws-ire-of-horror-films-critics, 2024. 1

[5] Ryan O'Connor. Introduction to diffusion models for machine learning. www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction, 2022. 1

[6] John Immerkær. Fast noise variance estimation. *Computer Vision and Image Understanding*, 64(2):300–302, 1996. 2

[7] Shawn Shan, Wenxin Ding, Josephine Passananti, Stanley Wu, Heather Zheng, and Ben Zhao. What is nightshade? nightshade.cs.uchicago.edu/whatis.html, 2023. 1

[8] Shawn Shan, Wenxin Ding, Josephine Passananti, Stanley Wu, Haitao Zheng, and Ben Y. Zhao. Nightshade: Prompt-specific poisoning attacks on text-to-image generative models, 2024. 1, 3

[9] Yuancheng Xu, Jiarui Yao, Manli Shu, Yanchao Sun, Zichu Wu, Ning Yu, Tom Goldstein, and Furong Huang. Shadowcast: Stealthy data poisoning attacks against vision-language models, 2024. 1

[10] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 1