RECAP: Robust Encryption for Creative Artwork Protection

Sadhana Lolla, Hannah Kim, and Christine Tu

Massachusetts Institute of Technology

Abstract. With the rise of AI, style mimicry models, which specialize in generating new artwork by imitating the style of a given artist, have risen in popularity. However, these models jeopardize the livelihood of thousands of artists, leading to new algorithms that claim to protect artist artwork. In this work, we focus on Glaze [6], a style transfer-based encryption method that has helped hundreds of artists protect their art. While Glaze is effective, it heavily relies on the secrecy of their algorithm and source code. We propose RECAP, a new algorithm that attempts to combat the limitations of Glaze while achieving better security. By applying a style mask to artwork, RECAP can perturb negligibly in the image space while preventing generative AI models from imitating the artist's style. However, we find that security is not achieved because of the limited set of secret keys used during encryption. We also find that errors that compound during the decryption process have a significant impact on the generated images, meaning that the decryption process must be very precise.

Keywords: Artwork Protection, Generative AI, Style Mimicry Models

1 Introduction

As ChatGPT [1] emerged and reshaped the landscape of literature and education, a parallel revolution unfolded in the art world through the evolution of style mimicry models. These models [2] can learn an artist's distinctive style and seamlessly apply it to diverse images or freshly generated artworks, fundamentally altering artistic expression. These models are commonly used in text-to-image applications or for learning a style from a set of sample images. Style mimicry models were launched into the eye of the mainstream media with the release of OpenAI's Dall-E [4](embedded in ChatGPT-4) with large social media trends focused on the usage of style mimicry tools.

However, the rise in prominence of these tools caused much unrest in the art community. In December of 2022, artists began a movement called "Say No to AI Art" [9], protesting the non-consensual and unaccredited usage of their art in training style mimicry models, especially on sites such as Midjourney [5]. The existence of these models threatens their livelihoods and blatantly steals the artistic style that they have spent decades honing without appropriate pay or recognition. To combat this problem, a team at UChicago released a tool

called Glaze [6]. Glaze allows users to overlay a mask that protects the artwork from being used as a sample for AI models. This mask should look invisible to the human eye, but render the artwork useless to any generative AI model. However, a major limitation of Glaze is the secrecy of its source code and having a limited number of secret keys to use to protect artwork. In the words of one of its creators, "the reality is that there are thousands of people (hundreds that I've already met in person) who would be seriously impacted by leaked Glaze code."

Although Glaze has yet to be broken or hacked, the confidentiality of their code and algorithms is an interesting problem to address. In the spirit of other cryptographic algorithms that are public and still effective, we aimed to create a more secure and robust encryption scheme for artwork. In addition, we explored a symmetric decryption scheme, which Glaze does not do; their perturbations are irreversible. Out algorithm, which we term RECAP (Robust Encryption for Artwork Protection) is a cryptographic approach to protecting artists and their work. Through RECAP, we achieve the following:

1. We create an encryption and corresponding decryption scheme for any image that preserves its visual features while preventing artifical intelligence algorithms from learning artists' style.
2. We design attacks that may break style-transfer based encryption schemes and show that the fixed set of secret keys used in these algorithms is a major drawback.
3. We show evaluations using state of the art diffusion models as well as user surveys on effectiveness of style masking.

2 Background

2.1 Text-to-image Models

Text-to-image models are first trained on a large corpus of input pairs, consisting of a string text and an image. For each pair, it extract features using some feature extractor Φ, on the input image Y to produce a feature vector $\Phi(Y)$, while another extractor performs a similar operation on the input text, x. Models learn to associate text vectors with their corresponding image feature to learn the textual encoding of a given image. Recently, text-to-image models have started to use diffusion [7] in order to then take text, which is first passed through an encoder, and output an image representing that text, using the feature vectors learned during training.

Diffusion models are often trained on a corpus of (text, image) pairs that are scraped from the internet. This allows the model to learn information about the style and content of different artworks without the artist's permission. For example, large databases like WikiArt, a commonly used dataset in machine learning, claims to respect copyrighted art, but has received reports of copyright infringement [8]. While the focus of RECAP is on the protection of an artist's style, it is also essential to acknowledge the importance of the artwork itself, including its content and its features, and the human behind the art.

2.2 Style Mimicry Models

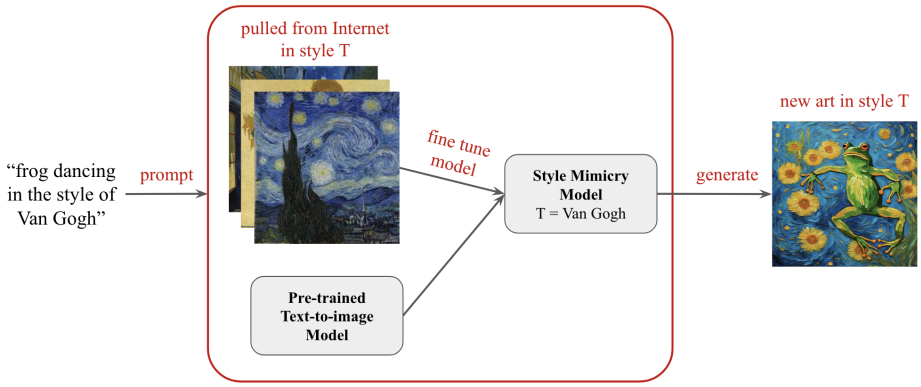

Fig. 1. A simplified model of how style mimicry works, using Van Gogh's style and online art generator https://pixlr.com/image-generator/. Once the pre-trained model is fine tuned upon an artist's works, it can be used to generate new images in the same style as the artist.

Style mimicry models work in conjunction with large, pretrained text-to-image models that are often outsourced by AI companies such as OpenAI. By downloading a text-to-image model and scraping a given artist's artwork from the internet, or some other database, a style mimicry model can fine-tune their model to learn and imitate the artist's style. Then, the text-to-image model only needs to receive a written prompt from users to generate images in the appropriate style.

The styles of artists can be obtained through online databases. This may be especially common for artists such as Van Gogh, whose artworks have become public domain. However, modern day artists may inadvertently enter these database through large scraping projects that pull images and artwork through the internet.

2.3 Glaze

Glaze is a state-of-the-art style masking algorithm. It provides artists with an online site and a downloadable application that both serve to "Glaze" their artworks. Given any piece of work, an artist can select how much masking they wish to apply to it; a higher level of masking is more likely to cause visual differences, and will take significantly longer to run.

In order to maintain the safety of their artists, the exact code and algorithms that run behind these applications are kept secret.

3 Methods

In this section, we detail the encryption and decryption methods of RECAP, as well as efforts we took to break our encryption methods. We also discuss our evaluation pipeline and experiments.

3.1 Encryption

We define $Enc(Y)$ as an encryption algorithm that takes in an image Y and outputs an encrypted image Y'. We want Enc to achieve the following goals:

1. **Undetectability:** Given original image Y and encrypted image $Enc(Y)$, a human guesser should not be able to tell the difference. We formalize this as follows. For any human guesser G, and any image Y, $Pr[G(Y) - G(Enc(Y))] \leq \mu(\lambda)$, where λ is the security parameter.
2. **Security:** Given encrypted image $Enc(Y)$, any neural network adversary A should not be able to find Y without the secret key. In other words, $Pr[A(Enc(X)) - A(Enc(Y))] \leq \mu(\lambda)$ for any pair of images X and Y, where A is trying to guess the secret key used to encrypt X or Y.
3. **Correctness:** A diffusion model D that has been trained on a set of inputs $\{Enc(Y)|Y$ is created by artist $A\}$, should not be able to produce artwork in the style of A.

At a high level, $Enc(Y)$ attempts to perturb the style features of an image Y while leaving the visual features intact. Based on the evaluations of Glaze, we hypothesize that even a small perturbation in the style space will completely change the style of the image. We postulate that this is because although the style space is continuous, two style vectors v_1 and v_2 that are close to each other in the style space correspond to vastly different styles in the visual domain. Therefore, we choose a small perturbation by combining some $\epsilon * v_1 + v_2$, where v_1 is the secret key and v_2 is the style of Y, and ϵ is a very small number. The full encryption scheme is detailed below:

$Enc(Y, \mathbf{s}) \rightarrow Y'$ takes in an image and a library of secret keys $\mathbf{s}$ and outputs the encrypted image Y'.

1. First, use an off-the-shelf feature extractor ϕ to compute the features $f = \phi(Y)$. Note that f is a three-dimensional stack of convolutional features of size $K \times L \times N$.
2. We use the gram-matrix construction to compute the style features of f. As mentioned in [3], the gram matrix of the feature maps f involves flattening each feature map into a single vector to create G, a $K \times LN$ two-dimensional matrix. To compute the style features of an image, we take the dot product between all K flattened vectors of Y, namely by computing $G \times G^T$ to obtain a $K \times K$ similarity matrix between the features. We use the gram matrix to measure which features co-occur with each other throughout the image, which computes style features without spatial features because the feature vectors are flattened prior to computing the similarities. We denote this process of extracting style features as a function $\beta(x)$ for a set of features x.

3. Now, we choose a secret key, which is a target style T of another artist. We choose T by measuring the Euclidean distance between $\beta(f)$ and every style in the style library $\mathbf{s}$, and choose the furthest style T from $\beta(f)$.
4. We construct $T(Y)$, which is the image Y in style T.
5. Now, we optimize the following function:

$$\min_{Y' \in I} m * MSE(Y', Y) + s * MSE(\phi(Y'), \phi(T(Y))) \tag{1}$$

using gradient descent. The above equation minimizes the visual perturbation between Y' and Y through the MSE term of the loss, and also minimizes the difference between style features of Y' and $T(Y)$. m and s are parameters to tune the degree of style perturbation and visual perturbation. When m is large, Y' will look very similar to Y but will have less style perturbation. When s is large, the visual perturbation will be large but the style will be significantly different. We argue in 5.1 that this scheme satisfies some, but not all, of our goals enumerated above.

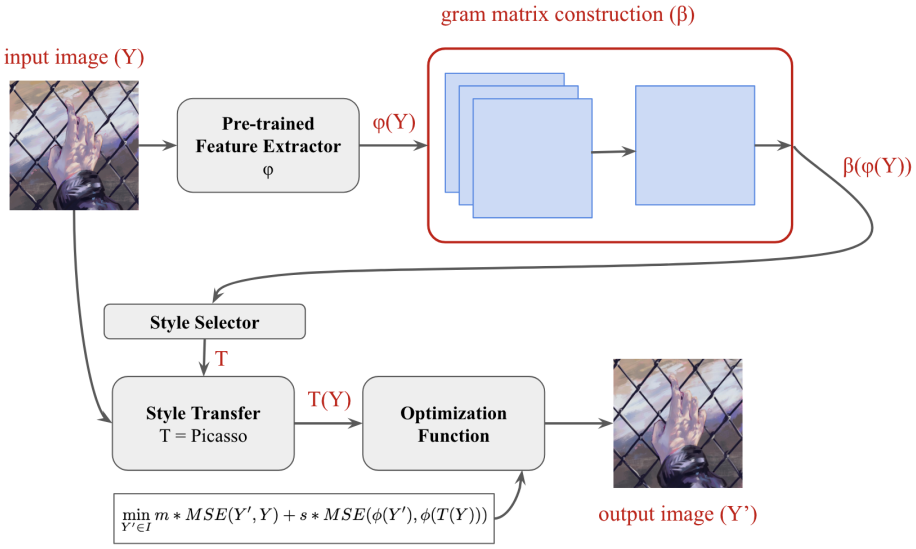

Fig. 2. An overview of the encryption algorithm. It selects a style using a gram matrix construction, then optimizes an MSE using the original image and its style transfer to output a new, perturbed image.

3.2 Breaking Encryption

Our goal in breaking encryption is to recover the secret key T given an input $Enc(Y)$. We design a neural network NN adversary as follows. Given oracle

access to Enc, we construct a dataset of tuples $(Enc(X), T)$ where we encrypt an image X using the secret key T. Then, we train NN on these tuples and evaluate whether we can recover T given only $Enc(X)$. To do so, we construct NN using a frozen off-the-shelf feature extractor ϕ_{NN} which can be the same feature extractor or different from the one used in the encryption algorithm. We then connect it to two fully-connected layers, which are trained for classification. Our style library consists of thirteen styles (explained further in 4.1), so we build a 13-class classifier. If such an NN can succeed in identifying which secret key was used on an encrypted artwork, then we have broken the goal of security as defined above.

3.3 Decryption

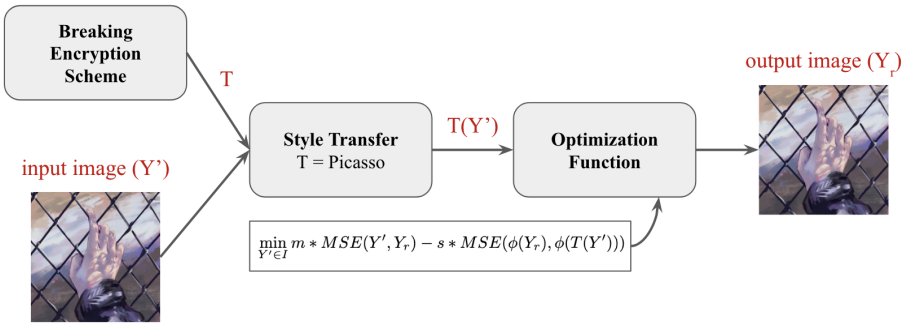

Fig. 3. The decryption pipeline used to, given an image Y' and style T, reconstruct the original artwork as output Y_r.

Finally, we wish to develop a decryption algorithm that will, given an encrypted image and a style T, reconstruct the original artwork. To do this, we construct $Dec(Y', T) \rightarrow Y_r$ to create a rough reversal of Enc. Specifically, it should take in an encrypted image Y' and the style T that was used to encrypt that image, then output the original image Y_r.

Dec will operate using the same off-the-shelf feature extractor ϕ and the same method of style transfer as in Enc. Once we obrain $T(Y')$, we use gradient descent to optimize the new function:

$$\min_{Y' \in I} m * MSE(Y', Y_r) - s * MSE(\phi(Y_r), \phi(T(Y'))) \tag{2}$$

We claim that this optimization will allow the decryption algorithm to construct the new image Y_r that is, again, minimally perturbed from the encrypted image Y' while maximizing the differences in style features between $T(Y)$ and Y_r. In order to reconstruct an image Y_r that is as close to the original image Y as possible, we aim to minimize the value of s while maximizing the value of m.

In Section 5.3, we will discuss our choice of parameters and the limitations upon our goals.

4 Evaluation

4.1 Data Sources

For evaluations of our encryption method, we use paintings from our artist friends that have never been previously published on the internet. This ensures that the diffusion models used for evaluations have not seen these artworks before, so we can accurately judge how much of their style the diffusion model learns.

To break RECAP, we train on a subset of the WikiArt [8] dataset, which contains 42129 artworks by 195 artists in the public domain. We use a randomly selected subset of 400 artworks and encrypt using all thirteen styles that are our secret keys.

We use thirteen different artists' style vectors as our secret keys. To generate these, we choose art samples from each artist that are in the public domain. We vary the art styles from Renaissance painters to ancient Chinese artists to modern digital and animated artists. The goal is to create a large variety of style vectors that can be used to encrypt the artwork.

4.2 scenario.gg

To evaluate our model, we use a free online style mimicry service called scenario.gg. scenario.gg allows users to upload a set of 5-20 images from the same artist and trains a model to learn that artist's style. We aim to evaluate our model by comparing the style of the scenario.gg outputs for Glaze, RECAP, and the decrypted RECAP. We hope to see that the style of the scenario.gg outputs trained on RECAP images are distinctly different from the original images.

For each of our evaluations of Glaze, RECAP, and RECAP description, we used the same set of 5 original images, from the same artist, as the training set for the scenario.gg model. These five images were controlled for resolution and size before being passed into Glaze and RECAP, ensuring consistency between model runs. The same prompt was used for each generation.

4.3 Preliminary User Survey

Using outputs from trained scenario.gg models, we prompted 11 volunteers for their perception of "closeness" in style. In the survey, the volunteers were provided with the original artwork, as well as four AI generated art pieces that were trained the artist's original, Glazed, encrypted by RECAP, and decrypted by RECAP artworks. Each person was asked to rank each of the four images from closest to furthest from the original artwork in terms of style. The images shown to volunteers are displayed in Figure 4.

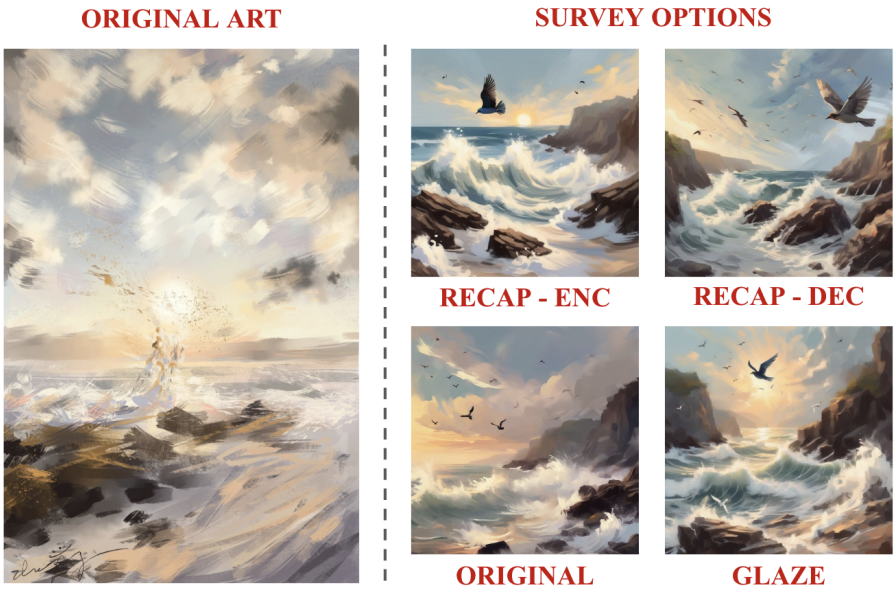

Fig. 4. Left: an original artwork. Right: four images that volunteers were asked to rank from closest to furthest from the original, in terms of style.

5 Results

5.1 Encryption

Figure 7 shows a sample encryption of an original artwork. The secret key here is the style vector corresponding to Picasso, and we can encrypt with varying levels of Picasso's style. It is clear that as we decrease m, which controls the level of truth to the original image, the image looks noticeably different. When $m = s = 5$, we can still see some differences between the encrypted image and the plaintext, but to a degree that may be considered negligible depending on the artist. Therefore, with different parameter settings, we can achieve our goal of undetectability.

Next, we evaluate RECAP's encryption method for correctness. To do this, we create five examples of encrypted artwork that the model has never seen before. Similar to the Glaze evaluation pipeline, we generate a natural language prompt for each artwork, and we prompt the model using this string to create an image in the style of the artist. By automatically generating a natural language prompt using an image-to-text model, we ensure that our description is close to those used by scenario.gg and is unbiased by our chosen descriptors. Lastly, we removed all style keywords such as "impressionist" or "expressive brushstroke" from our prompt.

In Figure 6, we can observe that both Glaze and RECAP seem to result in different scenario.gg-output styles than that of the original artist. Based on 11

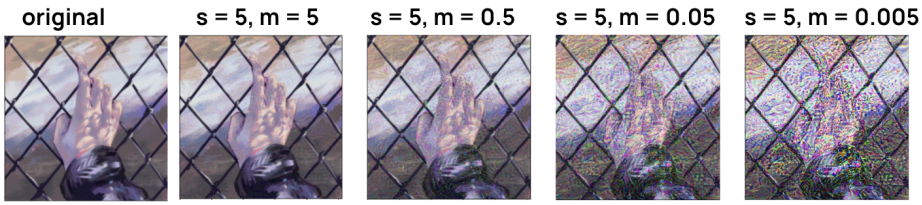

Fig. 5. From left to right: the original artwork, the same image encrypted with secret key Picasso with more and more of Picasso's style incorporated into the original artwork.

preliminary survey responses, 73% of users voted that the RECAP output is the least similar in style to the original. We also note that 63% of users voted that the original is closest in style.

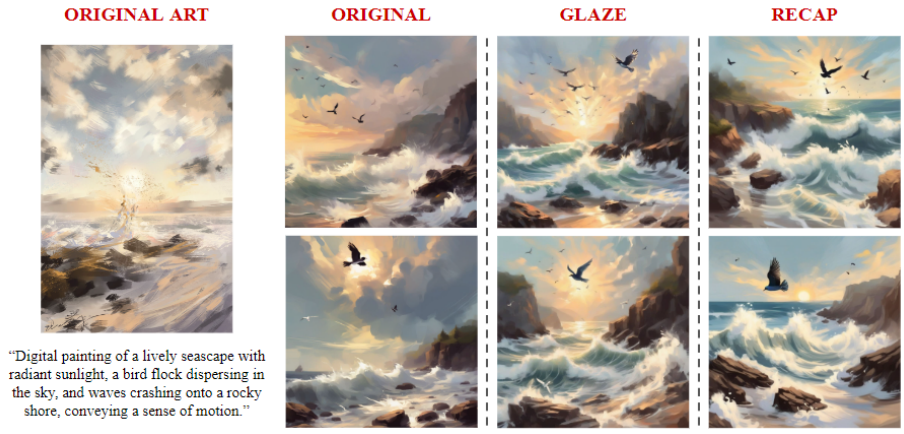

Fig. 6. From left to right: the original artwork, the scenario.gg output from training on the original artwork, the scenario.gg output trained on the Glazed artwork, and the scenario.gg output trained on the RECAP encrypted artwork

5.2 Breaking Encryption

To test the security of our encryption scheme, we simulate a modified CPA-style attack where our NN adversary chooses images and secret keys and runs the encryption algorithm. If the NN can identify the secret key for a previously unseen image at inference time with accuracy greater than random ($1/13 = 0.07$), then we know that the NN can computationally distinguish between images encrypted with different secret keys. We find that a simple classifier can identify

the secret key T with probability around 40%, an accuracy rate that could likely be improved with more samples during training and a better model architecture. Therefore, because we can recover the secret key, this encryption scheme– and likely Glaze– is breakable if the algorithm is public. It is possible that with a larger set of secret keys, it would be much harder to break the encryption method.

5.3 Decryption

Next, we reconstruct the original artwork from its encrypted image. In doing so, the parameters to our algorithm were crucial. Contrary to the encryption algorithm, we expected the decryption algorithm to work best with parameter s significantly smaller than m. Doing so would allow the decrypted image to remain close in the feature space, while moving away from our style-transferred encrypted images in the style space.

Fig. 7. From left to right: the original artwork, the same image encrypted with a Picasso style transfer, and the decrypted image.

Figure 7 demonstrates how RECAP can decrypt the original image given an encrypted image and its style, T. In this case, the masking style is Picasso, and the encryption model was trained using 400 steps with parameters $s = 5$ and $m = 0.5$. The decryption model was trained using 400 steps with $s = 0.0005$ and $m = 5$. To the human eye, the differences between each step in the process is noticeable in the changes in color and texture of brush strokes.

We evaluate the results of the decryption by passing the resulting artworks into scenario.gg with the expectation that it should successfully recreate the artist's style if the decryption worked. As shown in 8, it is clear that the style of $Dec(Enc(Y))$ is much more similar to the original image than $Enc(Y)$. However, it is also clear that the artwork styles are not exactly the same.

According to survey responses, only one person believed that the decrypted outputs were closest in style to the original. On average the decrypted images

were ranked 2.7, with the highest number of users labelling the image as third furthest from the original style. This is slightly higher than what users ranked for Glazed images, which was an average of 2.45.

We believe that this is because the optimization of the equation shown in 3.3 does not always result in the original image: rather, there is error in the optimization process because we only impose that the decrypted image should be far from the encrypted image in style space but close in the image space. There are many possible images that the gradient descent algorithm could converge on, so one avenue for future work is to create a more robust decryption mechanism that results in the exact input image.

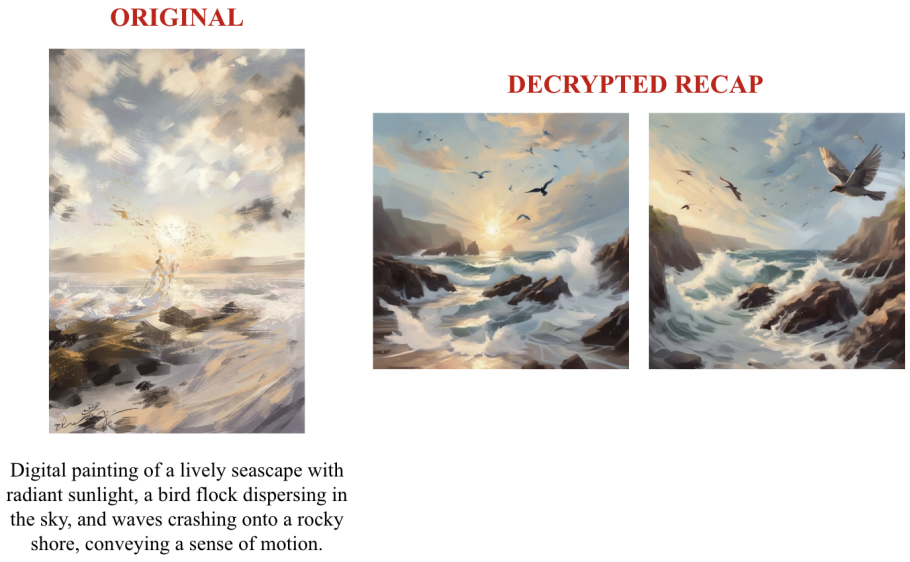

Fig. 8. An original art piece, with the auto-generated description. On the left are two images that were generated by scenario.gg using the recovered T_r's of our training set.

6 Discussion

6.1 Limitations

Unfortunately, RECAP and Glaze share one issue in that both pull from a limited number of secret keys. In this paper, we selected keys from a set of 13 keys, or styles. As we saw in Section 5.2, an adversary can easily identify the correct T once it has seen a collection of labeled images. This is a significant limitation of the algorithm, as an adversary can recover the secret key and run the decryption algorithm to achieve an image that is much closer in style to the original.

Therefore, we cannot claim that RECAP has achieved security. In 7, we discuss possible changes we can make to our encryption algorithm to prevent this type of attack.

In addition, our decryption algorithm is severely limited by possible values for s. In our testing, s was limited to values less than 0.0005, as larger values would cause our optimization function to hit $-\infty$. This prevents our algorithm from searching for images that are further away in the style space, which may contribute to issues where decrypted images did not match the original, and were rather more similar to the encrypted style transfer.

54% of volunteers ranked the decrypted image-trained scenario.gg outputs second furthest in style from the original. This means that our decryption algorithm is insufficient to recover the artist's style. For the purposes of reconstructing the original image, the decryption algorithm will require improvements before being used by artists.

To evaluate if our selection of keys and our decryption algorithm is truly safe and accurate, we should attempt a series of other attacks, as explained in Section 7.

6.2 Plausibility in Real World

As was covered in Section 4, the encrypted images by RECAP show little visual differences from the original image. Given the public algorithm, any user can adjust the parameters to achieve the level of perturbation that they want, acknowledging that a lower perturbation will lead to less style masking. Further, both RECAP and Glaze demonstrate an ability to hide an artist's style when passed into a generative model like scenario.gg.

For the purposes of protecting an artist's style on the internet, RECAP is a strong contender. Even with a publicized algorithm and key set, a method for reversing the process is currently unknown. The images generated by scenario.gg when trained upon decryption outputs showed a style that was more similar to the original as compared to the encrypted images, but still with major visual differences.

6.3 Use of Feature Space

Besides the style of their artwork, artists may also be concerned with protecting the contents of their art. This includes content such as characters, designs, or logos that are original and belong purely to the artist. Tools like Glaze and RECAP cannot help to protect the contents of art like this, as they are meant to only perturb the style of the original artwork.

As it stands, creating an algorithm to protect the actual contents of art is difficult. Our preliminary work showed that perturbing the features of an artwork enough to throw off an AI model would also cause visual differences that would be unacceptable to artists. Much more research will have to be done to identify methods of masking art content while leaving little to no visual perturbations on the work.

7 Future Work

7.1 Continuous Secret Keys

The first limitation realized by our algorithm is the finite set of secret keys that the encryption algorithm can select from. While this was necessary to ensure the encryption algorithm could be broken, this proved to be faulty. Furthermore, a finite set of keys would also allow an adversary to find and learn the masking style for a given artwork.

To combat this, we suggest selecting a secret key from a continuous set of style vectors. In order to allow the artist's style to be adequately perturbed, this secret key would have to sufficiently different from the style of the original artwork. To this end, we suggest calculating a range of style vectors that is different from the artist's style, then randomly selecting the key from that range. This might be done by training a model upon pairs of images and potential secret keys, to then allow that model to reliably return potential masking styles for any given image.

7.2 Minimizing Visual Perturbations

Currently, RECAP is more visually invasive than Glaze, which may contribute to it outperforming Glaze in our evaluations. Future work includes further minimizing these perturbations while maintaining high performance against style-mimicry attacks. This may include adjusting the optimization function.

Contributions

Sadhana created the encryption pipeline to handle style transfers and optimizations and the breaking algorithm for recovering the secret key. Hannah developed the decryption pipeline, which created reconstructed images. Christine trained models and generated artworks through scenario.gg to evaluate our work. All authors contributed to the presentation and the writing of this paper.

Acknowledgments. We thank our advisor, Katarina Cheng for her guidance and help throughout the duration of this project and Shih-Yu Wang for advising us in the early stages of our project. We are greatly appreciative to Christine Zhou, Megha Tummalapalli, and Katherine Zhao for entrusting us with their artworks. Lastly, to Henry and Yael for their engaging instruction.

References

1. Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.
2. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." arXiv preprint arXiv:1508.06576 (2015).
3. Li, Yanghao, et al. "Demystifying Neural Style Transfer." 2017.

4. Ramesh, Aditya, et al. "Zero-Shot Text-to-Image Generation." CoRR, vol. abs/2102.12092, 2021,
5. Richardson, Aran. "The $4,700 Artist AI Controversy: Artists Accuse Midjourney and Other AI Firms of Unauthorized Use." Benzinga, 24 Feb. 2024.
6. Shan, Shawn, et al. "Glaze: Protecting Artists from Style Mimicry by Text-to-Image Models." 32nd USENIX Security Symposium (USENIX Security 23). 2023.
7. Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models." CoRR, vol. abs/2010.02502, 2020
8. Wei Ren Tan, Chee Seng Chan, Hernan Aguirre, & Kiyoshi Tanaka. (2018). Improved ArtGAN for Conditional Synthesis of Natural Image and Artwork.
9. Xiang, Chloe. “Artists Are Revolting against AI Art on Artstation.” VICE, 14 Dec. 2022, www.vice.com/en/article/ake9me/artists-are-revolt-against-ai-art-on-artstation.