Feature Unlearning for Pre-trained GANs and VAEs

Saemi Moon[1], **Seunghyuk Cho**[2], **Dongwoo Kim**[1,2]
[1]CSE, POSTECH
[2]GSAI, POSTECH
{saemi, shhj1998, dongwookim}@postech.ac.kr

Abstract

We tackle the problem of feature unlearning from a pre-trained image generative model: GANs and VAEs. Unlike a common unlearning task where an unlearning target is a subset of the training set, we aim to unlearn a specific feature, such as hairstyle from facial images, from the pre-trained generative models. As the target feature is only presented in a local region of an image, unlearning the entire image from the pre-trained model may result in losing other details in the remaining region of the image. To specify which features to unlearn, we collect randomly generated images that contain the target features. We then identify a latent representation corresponding to the target feature and then use the representation to fine-tune the pre-trained model. Through experiments on MNIST and CelebA datasets, we show that target features are successfully removed while keeping the fidelity of the original models. Further experiments with an adversarial attack show that the unlearned model is more robust under the presence of malicious parties.

1 Introduction

Recent advancements in deep generative models have led to the generation of highly realistic images. However, this progress has also raised concerns about the potential misuse of such models. In some instances, generated images may contain violent or explicit content and inadvertently leak private information used to train the model. To address these issues, a well-prepared dataset with appropriate cleansing procedures can mitigate the potential for abuse of generative models. For example, the development of DALL·E 2 involves careful curation of the training data to avoid explicit content, as described by (Ramesh et al. 2022).

In addition to data preparation and cleansing, machine unlearning serves as a complementary tool for preventing the problem in the development and deployment of a generative model. Machine unlearning aims to erase the target data from a pre-trained machine-learning model, which can be required to remove private information, harmful content, and biased information (Cao and Yang 2015). However, most of the machine unlearning methods have been focused on supervised models so far (Gupta et al. 2021; Tarun et al. 2021; Baumhauer, Schöttle, and Zeppelzauer 2022; Ginart et al. 2019; Yoon et al. 2022; Chundawat et al. 2022; Golatkar, Achille, and Soatto 2020; Nguyen, Low, and Jaillet 2020).

Figure 1: Result of unlearning various features from pre-trained StyleGAN model. We utilize the same latent vector to generate images from both the original and the unlearned models. Our method effectively unlearns the target feature while maintaining high image quality.

In this work, we tackle the problem of feature unlearning from pre-trained image generative models where we aim to fine-tune the model to exclude the production of samples that exhibit target features. One of the challenges in feature unlearning is that the target feature can be subtle. For instance, a specific hairstyle of a facial image could be the target feature we want to remove from the model, as shown in Figure 1. The subtlety in the target features makes it hard to adopt traditional supervised model unlearning approaches.

In many unlearning scenarios with supervised models, the target of unlearning is a subset of a training dataset, leading to the oracle model that could have been obtained from training without the target subset. Unlike supervised model unlearning, it is non-trivial to define the target data when unlearning feature since the target feature is only presented in a local region of an image. If we naively remove the entire image that contains the target feature, we could lose the other information in the remaining region of the image. Eventually, subset removal results in a loss of high fidelity and diversity in the generated samples. On the other hand, explicit pixel-level supervision could be given to unlearn the target feature, but it is often very expensive to obtain such supervision. Furthermore, the problem becomes more challenging if the training dataset is inaccessible during unlearning for several reasons, e.g., storage capacity, private content protection, etc.

To overcome such challenges, we propose a novel generative model unlearning framework that can be applied to generative adversarial networks (GANs) and variational autoencoders (VAEs). To do so, we first collect randomly generated images that contain the target features. We then identify the latent representation of the target features and use the representation to fine-tune the pre-trained model, which prevents the model from generating images with the target feature. To our knowledge, this is the first framework for unlearning target features in the pre-trained GANs and VAEs. Experimental results show that our unlearning method effectively removes the target feature while maintaining the image quality. An additional study based on adversarial attacks also confirms that the proposed method is more robust than a standard model against malicious behavior.

2 Related Work

2.1 Machine Unlearning

Previous studies have demonstrated that machine learning models may leak sensitive information through attacks or specific inputs (Yuan et al. 2019; Cao and Yang 2015). In addition, regulations have emerged to protect private information, such as 'the right to be forgotten', which grants users the request that their personal information must be removed from a system (Rosen 2011). These highlight the growing significance of machine unlearning.

Unlearning scenarios can vary depending on the requirements (Nguyen et al. 2022). Traditional machine unlearning approaches assume that all training data can be accessed (Gupta et al. 2021; Tarun et al. 2021; Baumhauer, Schöttle, and Zeppelzauer 2022; Ginart et al. 2019). However, recent studies have presented problem formulations in which access to the data is highly restricted (Yoon et al. 2022; Chundawat et al. 2022; Golatkar, Achille, and Soatto 2020; Nguyen, Low, and Jaillet 2020). In the context of feature unlearning, Guo et al. (2022) proposes a representation detachment approach to unlearn the specific attribute for the image classification task. However, the above researches focus on supervised learning tasks, whereas we focus on unsupervised generative models.

Recent research has focused on unlearning methods into generative models. Kong and Chaudhuri (2022) proposed a data redaction method from pre-trained GAN. They use a data augmentation-based algorithm to prevent making undesirable samples. This method can only be applied when the entire dataset is available. Besides that, we first propose the generative model feature unlearning framework when access to entire data is infeasible. Additionally, Gandikota et al. (2023) and Zhang et al. (2023) propose unlearning methods applicable to text-to-image diffusion models. However, these methods are limited to cross-attention-based models, which may hinder their generalization to diverse generative models. In contrast, our unlearning method can be applied to any generative model that has its own latent space.

2.2 Latent Space Analysis

It is known that generative models, such as GANs (Goodfellow et al. 2020; Radford, Metz, and Chintala 2015; Karras et al. 2017) and VAEs (Kingma and Welling 2013; Child 2021) well preserve the information of data within a low-dimensional space, referred to the latent space. In recent years, various techniques for traversing the latent space and extracting a latent vector that represents a visual feature have been proposed.

Radford, Metz, and Chintala (2015) obtain the visual feature vector by subtracting the two latent vectors: the mean latent of the images without the features and the mean latent of the images with the features. To decide the label of a given latent vector of an image, several approaches attempt to learn a predictor with latent vectors labeled with the corresponding features (Goetschalckx et al. 2019; Tran, Yin, and Liu 2017; Shen et al. 2020). Unsupervised methods for finding interpretable axes in the generator have also been proposed (Härkönen et al. 2020; Voynov and Babenko 2020; Tzelepis, Tzimiropoulos, and Patras 2021; Shen and Zhou 2021; Wang and Ponce 2021).

In our proposed framework, obtaining the target vector representing the target feature in the latent space is a crucial step. We introduce a straightforward and user-friendly approach that can be applied to both GANs and VAEs, which can be applied to real-world scenarios easily. Additionally, we leverage the target vector to the target identification method within latent space.

3 Feature Unlearning for Generative Models

In this section, we propose a framework for unlearning generative models such as GANs and VAEs to make the unlearned model unable to generate the target feature. Throughout this work, we assume that the training dataset is inaccessible once the training is done due to various reasons, such as limited storage or privacy concern.

3.1 Dataset Preparation

Feature unlearning aims to remove a specific feature from a pre-trained generative model. For example, after unlearning the smile feature from a generative model trained on the CelebA dataset, the model would never generate images of a smiling person. To do so, in our framework, we first collect datasets using the principle of distance supervision (Mintz et al. 2009). We curate a 'positive' dataset with images that contain the feature to be erased from generated images. The rest of the images without the target feature is categorized into a 'negative' dataset. In practice, we can develop an interface where users can select images that contain the target feature. Figure 2 displays the prototype system that we develop to collect the user responses.

3.2 Unlearning Framework

Unlearning the entire subset with the target feature may lead to losing other image details. Pixel-level supervision can specify areas to unlearn, but it is expensive to scale. Instead, we unlearn the target feature by learning a transformation from the image containing the target feature to the image without the one. To learn such transformation, we need a paired dataset with and without target features. For example, if the target image represents a man smiling and wearing a

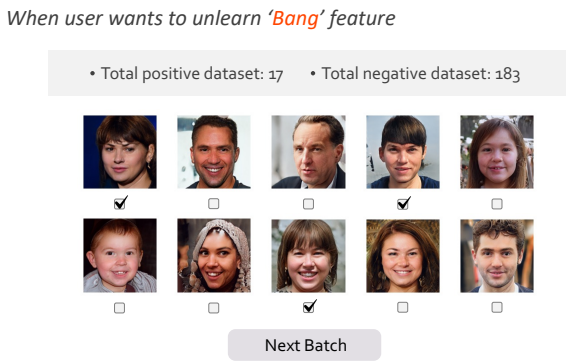

Figure 2: Illustration of interface used to collect images containing the target feature from generated images. A user selects images that contain the target feature to be unlearned. The selected and non-selected images serve as positive and negative datasets for target feature identification.

hat, and the target feature is the smile, we need the non-target image that depicts the same man with the hat but without the smile. However, it is impossible to curate such a dataset in a real-world scenario.

Since our goal is to unlearn the pre-trained generator and not to learn the transformation, we need to apply the transformation principle to the sampling process of the generator. A transformation in image space can be modeled by the corresponding transformation in the latent space.

Based on this intuition, we propose a general unlearning framework from the latent variable perspective as follows:

1. Collect positive and negative datasets from generated images.
2. Find a latent representation $\mathbf{z}_e$ that represents the target feature in the latent space.
3. Sample a latent vector $\mathbf{z}$ from a simple distribution.
   (a) If the latent vector does not contain the target feature, let the generator produce the same output without modification.
   (b) If the latent vector contains the target feature, fine-tune the generator to produce a transformed output without the target feature.
4. Repeat step 3 until the generator does not produce the target feature.

Target identification in latent space. We assume that a vector in the latent space can represent the target feature. As the first step of unlearning, we obtain the latent vector representation of each image from the collected dataset. Once we obtain the latent vectors, we use a vector arithmetic method proposed by Radford, Metz, and Chintala (2015) to find the latent vector representing the target feature. Specifically, we compute the mean vectors from a collection of a positive dataset and a negative dataset and subtract the mean vectors of the negative dataset from that of the positive dataset. The resulting *target vector* $\mathbf{z}_e$ is then used to represent the target feature in the latent space.

To determine whether a randomly generated image contains a target feature, we project its latent vector onto the target vector. White (2016) shows that the projection can represent the similarity between the latent vector and the target feature. We then compare this value to a threshold to determine whether the image contains the target feature. For the experiments, we set the threshold value t as the average projection values of the positive and negative samples in the latent space. Let $\mathrm{sim}(\mathbf{z}, \mathbf{z}_e) \in \{0, 1\}$ indicate the binary classification results, i.e.,

ze = latent of target space

$$\mathrm{sim}(\mathbf{z}, \mathbf{z}_e) = \begin{cases} 0, & \text{if } \mathrm{proj}_{\mathbf{z}_e}(\mathbf{z}) < t, \\ 1, & \text{otherwise,} \end{cases} \tag{1}$$

z = random vector does not contain target feature

where $\mathrm{proj}_{\mathbf{z}_e}(\mathbf{z})$ is the projection of $\mathbf{z}$ onto $\mathbf{z}_e$, i.e., $\frac{\mathbf{z}_e^\top \mathbf{z}}{\|\mathbf{z}_e\|}$.

Unlearning process. To formalize the unlearning process, let g_θ be the model to be unlearned, and f be the pre-trained generator. We initialize g_θ from the pre-trained f. When the randomly sampled latent vector $\mathbf{z}$ does not contain the target feature, i.e., $\mathrm{sim}(\mathbf{z}, \mathbf{z}_e) = 0$, the generator g_θ needs to produce the same output as f, c.f, step 3 (a). To enforce the minimal changes in the produced output, we formulate the following *reconstruction* objective to minimize

$$\mathcal{L}_{\text{recon}}(\theta) = (1 - \mathrm{sim}(\mathbf{z}, \mathbf{z}_e))\|g_\theta(\mathbf{z}) - f(\mathbf{z})\|_1, \tag{2}$$

where $\mathbf{z}$ is the random vector. Hence, the unlearned model g_θ tries to mimic the original generator when the latent vector does not contain the target feature.

When the randomly sampled vector contains the target feature, the generation process needs to be changed such that the sampled output no longer contains the target feature. To do so, we first create the target-erased output by generating an output with a translated random vector using f. Given random vector $\mathbf{z}$, we first project the vector onto the target vector, and then the original random vector is shifted by the projected vector, i.e., $\mathbf{z} - (\|\mathrm{proj}_{\mathbf{z}_e}(\mathbf{z})\| - t)\mathbf{z}_e$, where t is the predefined threshold. The translated vector is used as an input of the original generator f producing the target-erased output. The modified output is then used to train g with the following *unlearning* objective

$$\mathcal{L}_{\text{unlearn}}(\theta) = \mathrm{sim}(\mathbf{z}, \mathbf{z}_e) \\ \|g_\theta(\mathbf{z}) - f\left(\mathbf{z} - \left(\mathrm{proj}_{\mathbf{z}_e}(\mathbf{z}) - t\right)\mathbf{z}_e\right)\|_1 . \tag{3}$$

The objective enforces the unlearned generator producing outputs similar to those from the original generator without target features. If the projection can correctly measure the presence of the target feature in the latent space while disentangling the other features, g_θ can successfully forget the target feature in the latent space.

It is widely known that L2 and L1 loss occurs in blurry effects in image generation and restoration tasks (Pathak et al. 2016; Zhang, Isola, and Efros 2016; Isola et al. 2017; Zhao et al. 2016). Prior research has addressed the blurry effects by introducing diverse techniques, such as adding perceptual or adversarial loss to the training process (Johnson, Alahi, and Fei-Fei 2016; Zhao et al. 2016). To overcome the blurry effects, we add *perceptual* loss into the objective function.

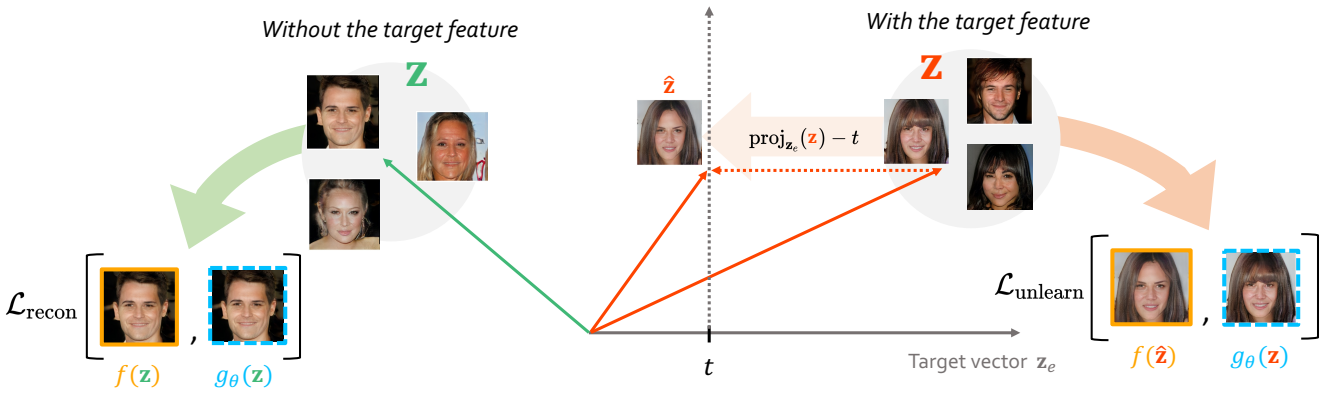

Figure 3: Overall illustration of the generative model unlearning framework. Based on whether the randomly sampled vector $\mathbf{z}$ has the target feature, we use different loss functions to unlearn the target feature. t refers to a threshold, and $\hat{\mathbf{z}}$ is the translated vector, i.e., $\hat{\mathbf{z}} = \mathbf{z} - (\text{proj}_{\mathbf{z}_e}(\mathbf{z}) - t)\mathbf{z}_e$.

The objective is formalized as

$$\mathcal{L}_{\text{percep}}(\theta) = \text{sim}(\mathbf{z}, \mathbf{z}_e) \left(1 - \text{MS-SSIM}\left(g_\theta(\mathbf{z}), f\left(\mathbf{z} - \left(\text{proj}_{\mathbf{z}_e}(\mathbf{z}) - t\right)\mathbf{z}_e\right)\right)\right), \quad (4)$$

where MS-SSIM function refers to the Multi-Scale Structural Similarity (Zhao et al. 2016), which measures perceptual similarity between two images by comparing luminance, contrast, and structural information.

Finally, we combine three objective functions to define the unlearning objective as

$$\mathcal{L}(\theta) = \alpha\left(\mathcal{L}_{\text{unlearn}}(\theta) + \mathcal{L}_{\text{percep}}(\theta)\right) + \mathcal{L}_{\text{recon}}(\theta), \quad (5)$$

where α is the hyper-parameter that regulates the unlearning and reconstruction error balance. We visualize the overall framework in Figure 3.

4 Experiments

In this section, we show the performance of the proposed framework for unlearning GANs and VAEs trained on three datasets. We conduct both synthetic experiments and user studies with human participants.

4.1 Experiment Setup

Datasets and models. To show the performance of the proposed framework, we conduct experiments on three datasets with a different set of generative models for each. We list the three datasets used in the experiments and the generative models trained on each dataset as follows:

- **MNIST** (LeCun et al. 1998): a Deep Convolutional GAN (DCGAN)(Radford, Metz, and Chintala 2015) and a vanilla VAE (Kingma and Welling 2013)
- **CelebA** (Liu et al. 2018): Progressive Growing of GANs (ProgGAN) (Karras et al. 2017) and a Very Deep VAE (VDVAE)[1] (Child 2021)
- **FFHQ** (Karras, Laine, and Aila 2019): a StyleGAN[2] (Karras, Laine, and Aila 2019)

Further details of the models can be found in Section A.1.

Unlearning dataset preparation. To understand how the proposed algorithm works, we simulate the cases by using the known features of each dataset. Specifically, we select two features from each dataset, with each feature representing approximately 10% of the dataset.

For the MNIST dataset, we choose the thickness and left slant as target features to unlearn. We use the Morpho-MNIST (Castro et al. 2019), which provides a comprehensive tool for measuring various features of MNIST digits. Since the thickness and left slant are not binary, we use images whose feature values range within the top 10% of the entire dataset as a positive dataset and the remaining as a negative dataset. The CelebA provides 40 features for each image. Among the available features, we choose 'Bang' and 'Beard' as the target features to be unlearned. Although FFHQ does not provide annotations for each image, we decided to unlearn the same features as CelebA since both are facial datasets. We experiment with two additional features, 'Hat' and 'Glasses' for FFHQ.

For this experiment, we use a classifier to categorize each sampled image into a positive or negative dataset. To get a target feature, we utilize Morpho-MNIST as the classifier and categorize 500 generated images for MNIST. For both CelebA and FFHQ, we use a pre-trained MobileNet (Howard et al. 2017) to classify 5,000 generated images. We use these classifiers as a proxy of users, but we also conduct a user study involving human participants reported in Section 4.4.

Oracle model. There is no unlearning method targeted for generative models, according to our information. To evaluate effectiveness, we compared our framework with the oracle model trained from scratch without the images containing the target feature. The oracle model is commonly used as a standard oracle model in supervised unlearning cases, whereas

[1]We concatenate multiple layers of latent variables to apply the unlearning algorithm.

[2]We use the output of the mapping network to unlearn.

Dataset	Model	Feature	Target feature ratio (%)			Inception Score			Fréchet Inception Distance		
			Original	Unlearn	Oracle	Original	Unlearn	Oracle	Original	Unlearn	Oracle
MNIST	VAE	Thickness	9.47	0.97	1.01	2.207	2.16	2.14	23.36	23.74	24.03
		Slant	9.22	0.91	1.08	2.19	2.22	2.22	22.84	23.27	23.52
	DCGAN	Thickness	9.10	1.35	1.28	2.14	2.13	2.11	2.32	3.35	3.40
		Slant	10.92	1.25	1.31	2.15	2.14	2.10	2.24	2.79	2.85
CelebA	VDVAE	Bang	3.36	0.28	0.21	2.57	2.58	2.54	82.92	85.21	84.09
		Beard	7.22	2.41	1.09	2.47	2.38	2.39	82.92	86.15	84.49
	ProgGAN	Bang	6.74	0.42	0.49	2.92	2.91	2.91	48.05	49.82	51.37
		Beard	3.02	1.01	0.98	2.93	2.88	2.87	48.05	49.80	49.78
FFHQ	StyleGAN	Bang	4.48	0.26	✗	3.61	3.33	✗	20.97	25.88	✗
		Beard	21.96	1.37	✗	3.60	3.34	✗	20.93	25.12	✗
		Hat	2.10	0.12	✗	3.62	3.37	✗	21.00	24.75	✗
		Glasses	6.16	0.19	✗	3.61	2.37	✗	20.86	23.86	✗

Table 1: Target feature ratio (↓), inception score (↑), and Fréchet inception distance (↓) of original, unlearn, and oracle models.

in our case, this is not ideal since the positive dataset can contain useful features other than the target feature. Note that, for FFHQ dataset, we cannot train the oracle model since the annotated features are not available.

Training details. For all experiments, we use Adam optimizer and a learning rate of 0.001, 0.002, and 0.005 for MNIST, CelebA, and FFHQ respectively. The MNIST dataset is trained for 200 epochs, and CelebA and FFHQ are trained for 500 epochs in unlearning process. We use NVIDIA GeForce RTX 3090 and A6000 for experiments.

4.2 Evaluation Metric

The performance of unlearning can be measured from two different perspectives: 1) how well the unlearning is done and 2) how good the sample qualities are. We explain two different metrics used to evaluate the models.

Target feature ratio. The target ratio measures the percentage of generated samples with the target feature. Low values of the target ratio indicate that the generative model has successfully unlearned the target feature. We use the same pre-trained classifiers used to target feature identification. For all experiments, we randomly sample 10,000 images from a generative model to compute the target ratio.

Image quality. We report two commonly used metrics to evaluate the quality of the generated image: Inception Score (IS) (Salimans et al. 2016) and Fréchet Inception Distance (FID) (Heusel et al. 2017). We compute FID between the generated samples and the original training dataset. Higher IS scores and lower FID scores indicate higher image quality. We use an implementation of StudioGAN (Kang, Shin, and Park 2022) to calculate IS and FID. 50,000 samples are used to measure the scores.

4.3 Results

Quantitative results. We evaluate the effectiveness of our unlearning framework by comparing the target feature ratio between the original model, the unlearned model, and the oracle model in Table 1. The results show that the unlearned model produces similar target feature ratios to the oracle for all features, indicating our framework successfully unlearns the target feature.

Ensuring high image quality is also important in unlearning the target feature. Table 1 presents the results of the IS and FID scores for evaluating the quality of generated images, respectively. The results demonstrate that all three models produce similar IS and FID scores, indicating our framework can successfully unlearn the target feature while maintaining high-quality image generation. Note that FID is calculated using the entire dataset, which yields a slightly higher FID value for the unlearned and oracle models, but there is no significant difference between the two models.

Qualitative results. Figure 4 presents the qualitative visualization result of our unlearning framework. The top row shows the images generated from the original generator, and the bottom row shows those generated from the unlearned generator. By visualizing generated images using the same latent vector, we observe that the target feature has been effectively erased in each case. In addition, we unlearn various features from StyleGAN trained with FFFQ dataset (Karras et al. 2017), whose resolution is higher than the other two datasets. The qualitative results in Figure 1 show the approach also works well with high-resolution images. Additional qualitative results are provided in Section B.1.

Computational efficiency. Our unlearning framework takes approximately one minute to unlearn MNIST on a single GPU for VAE and DCGAN, and approximately 10 minutes to unlearn CelebA and FFHQ with four GPUs for ProgGAN, VDVAE, and StyleGAN. In contrast, training the oracle for MNIST requires approximately 30 minutes on a single GPU. Training the oracle for ProgGAN on CelebA takes around three days using eight GPUs, and one for VDVAE takes about two days using four GPUs. The authors of StyleGAN report that training StyleGAN on FFHQ takes approximately 6 days and 14 hours with 8 Tesla V100 GPUs[3]. Although we assume that relearning is impossible, nevertheless, even if relearning were possible, our method is significantly more time-efficient with comparable results.

[3]https://github.com/NVlabs/stylegan

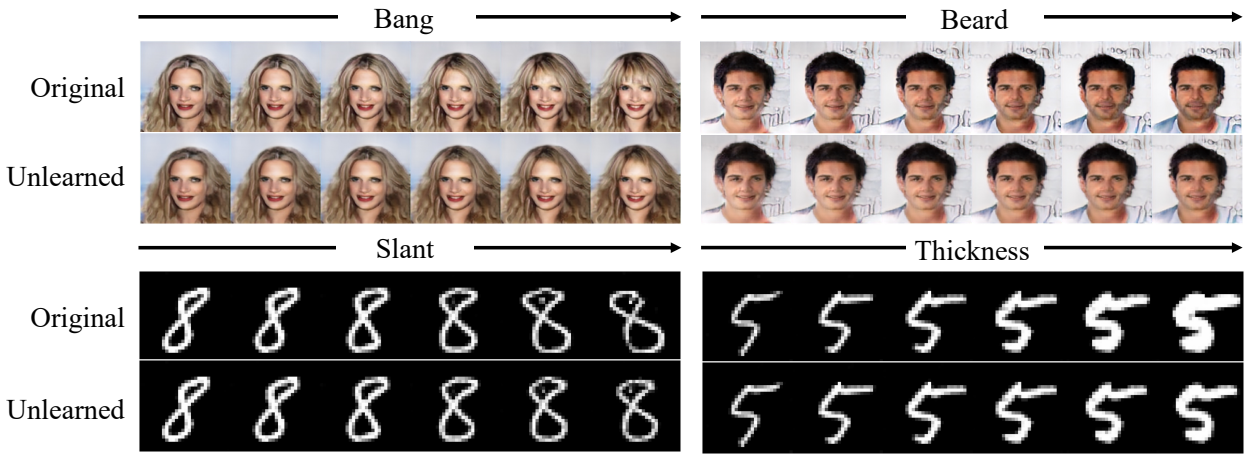

Figure 4: Visualization of four different features before and after unlearning from pre-trained GAN models. All paired images in each column are generated from the same latent vector.

4.4 User study

A user study was conducted to assess the effectiveness of our unlearning framework in a more realistic scenario. This study was designed to scrutinize the performance of the unlearned model in comparison to the original model across various dimensions. We recruited 13 participants and asked them to select images that contain 'Glasses' since the feature is distinctly discernible by users. 500 samples were annotated by each participant using the interface shown in Figure 2. The annotation process took roughly 5 minutes on average. Then, we unlearned the pre-trained StyleGAN trained on FFHQ for each participant. User study details and screenshots are provided in Section C.

We use the following three criteria to measure the performance of unlearning against the original model:

1. **Target feature ratio.** This task involved counting the number of target images (images with glasses) among randomly generated images. A participant examined 500 images, each of which are randomly chosen between the original and unlearned models.
2. **Image quality.** Participants were asked to choose an image with better quality. We provided two randomly generated samples from the original and unlearned models, one for each. The samples were shuffled before being presented, and each participant evaluated a total of 50 cases. Users can respond that one of the two images is of better quality or that both images are of similar quality.
3. **Pinpoint unlearning.** We provided two images generated with the same latent vector from the original and unlearned models and asked participants how many features were changed after unlearning. CelebA's attributes were shown in advance to familiarize participants with the existing features. The participants had the following options:
   - All features except the target were unchanged.
   - One or two features appeared to have changed.
   - More than two features appeared to have changed.

   Each participant compared ten randomly generated pairs.

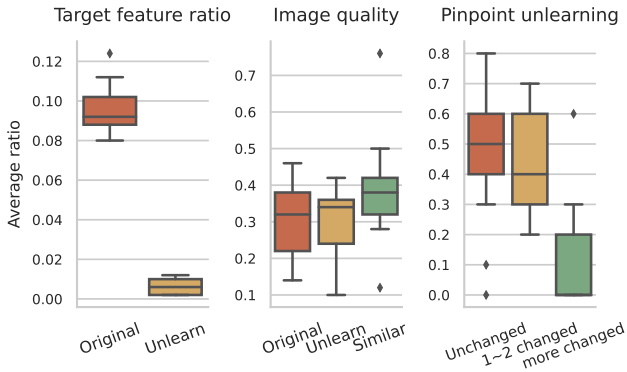

Figure 5: User study result of unlearning 'Glasses' feature from the StyleGAN.

Result. Figure 5 presents the result of the user study. We normalize the answers of each user and plot their distribution. The result shows a significant drop in the target feature ratio representing the effectiveness of our framework on the target feature. There is no significant difference in image quality between the original and unlearned models, and the majority of users answer the quality between the two models is similar to each other.

Participants evaluated that only the target feature was changed through unlearning on average 45% of cases. However, 43% of cases were reported to have one or two features changed, and 12% to have more than two features changed. We found that the entanglement between the 'Glasses' and 'Young' features leads to the result. In FFHQ, a person with 'Glasses' are more likely to be elderly. As a consequence, the identified target feature vector is likely to be entangled with the 'Young' feature. More sophisticated feature disentangle-

		Original	Unlearn
Before attack	Target feature ratio (↓)	3.54	0.96
	IS (↑)	3.16	3.16
	FID (↓)	23.54	24.00
After attack	Target feature ratio (↓)	14.28	5.54
	IS (↑)	2.83	3.03
	FID (↓)	44.25	38.39

Table 2: The target feature ratio (%)/ IS / FID before and after an adversarial attack.

ment algorithms (Tran, Yin, and Liu 2017; Locatello et al. 2019) could help to mitigate such effect, but we leave this for future work.

5 Adversarial Attack

To check the robustness of our unlearning method under the presence of adversaries, we further conduct an experiment with an adversarial attack method on an unlearned model.

5.1 Experimental Setting

An adversarial attack on the unlearned model is conducted to assess its vulnerability to malicious users who may attempt to exploit the model to generate harmful or explicit content. By subjecting the unlearned model to an adversarial attack, we can evaluate its robustness and ensure that the unlearned model does not generate content that goes against ethical or safety guidelines.

For this experiment, we employ a Projected Gradient Descent (PGD) attack (Madry et al. 2017) on the latent variable of the unlearned model. The attack is guided by a pre-trained feature classifier capable of classifying the target feature. The purpose of this attack is to determine whether the unlearned model can be manipulated to produce images that contain the target feature, even after it has been supposedly removed. The overall method used in this experiment is provided in Algorithm 1.

We attack the ProgGAN (Karras et al. 2017) trained with CelebA-HQ (Karras et al. 2017) with the target feature of 'Bang'. We conduct an adversarial attack on 10,000 distinct latent vectors to both the original and unlearned models. The classifier used for this experiment is a MobileNet (Howard et al. 2017) specified in Section 4.1. For the hyper-parameters of the PGD method, a learning rate of 0.02 is chosen, and the attack step is set at 50. The maximum magnitude of the permissible perturbation is set to 0.1, i.e., $||\tilde{\mathbf{z}} - \mathbf{z}||_\infty \leq 1$. Note that a larger perturbation may cause the perturbed latent vector to deviate significantly from the original distribution.

5.2 Results

As a preliminary step, we evaluate the feature target ratio and image quality for each original and unlearned model to compare the before and after attack results. Table 2 presents the result before the attack. The result shows that the unlearned model generates less number of target images than the original model while maintaining high image quality in terms of IS and FID.

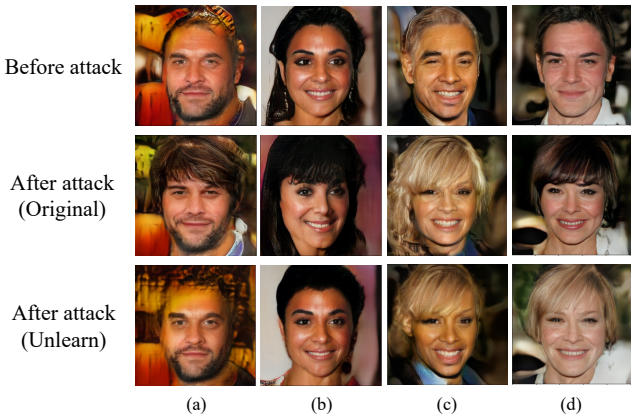

Figure 6: Examples of generated images before and after an adversarial attack on its latent variable. The top row shows the original image. 2nd and 3rd rows are attacked images from the original and unlearned models. The first two columns are successful defenses by the unlearned model and the last two show failures.

To measure the target feature ratio after the adversarial attack, we train the second target feature classifier with a different random seed. Since the adversarial attack perturbs the latent vector to fool the target classifier, the perturbed sample may not contain the target feature but overfit the target classifier. Hence, we use the second target classifier to measure the target feature ratio. Although the target feature ratio of the unlearned model has increased after the attack, the target feature ratio of the unlearned model remains lower than that of the original model.

We visualize the randomly generated images before and after an adversarial attack in Figure 6. The first two columns (a, b) show that the original model generates the target feature through the adversarial attack, while the unlearned model effectively defends against the attack. However, the last two columns (c, d) demonstrate that both models fail to defend against the attack.

6 Conclusion

The recent success of generative models has brought exciting developments in various fields, such as computer vision, natural language processing, and art generation. However, the potential risks associated with the generation of harmful or private content through these models highlight the importance of developing effective unlearning algorithms. Our proposed unlearning algorithm for generative models shows promising results in preventing the generation of unwanted features, which can serve as a crucial tool in addressing sensitive or private content concerns. Future research can build upon this work to improve the efficiency and effectiveness of unlearning algorithms in other contexts, such as data privacy and fairness. Ultimately, the development of robust and reliable unlearning algorithms can maximize the benefits of generative models while minimizing the associated risks.

References

Baumhauer, T.; Schöttle, P.; and Zeppelzauer, M. 2022. Machine unlearning: Linear filtration for logit-based classifiers. *Machine Learning*, 111(9): 3203–3226.

Cao, Y.; and Yang, J. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, 463–480. IEEE.

Castro, D. C.; Tan, J.; Kainz, B.; Konukoglu, E.; and Glocker, B. 2019. Morpho-MNIST: Quantitative Assessment and Diagnostics for Representation Learning. *Journal of Machine Learning Research*, 20(178).

Child, R. 2021. Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images. In *International Conference on Learning Representations*.

Chundawat, V. S.; Tarun, A. K.; Mandal, M.; and Kankanhalli, M. 2022. Zero-shot machine unlearning. *arXiv preprint arXiv:2201.05629*.

Gandikota, R.; Materzynska, J.; Fiotto-Kaufman, J.; and Bau, D. 2023. Erasing concepts from diffusion models. *arXiv preprint arXiv:2303.07345*.

Ginart, A.; Guan, M.; Valiant, G.; and Zou, J. Y. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.

Goetschalckx, L.; Andonian, A.; Oliva, A.; and Isola, P. 2019. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5744–5753.

Golatkar, A.; Achille, A.; and Soatto, S. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9304–9312.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Guo, T.; Guo, S.; Zhang, J.; Xu, W.; and Wang, J. 2022. Efficient Attribute Unlearning: Towards Selective Removal of Input Attributes from Feature Representations. *arXiv preprint arXiv:2202.13295*.

Gupta, V.; Jung, C.; Neel, S.; Roth, A.; Sharifi-Malvajerdi, S.; and Waites, C. 2021. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34: 16319–16330.

Härkönen, E.; Hertzmann, A.; Lehtinen, J.; and Paris, S. 2020. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33: 9841–9850.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, 694–711. Springer.

Kang, M.; Shin, J.; and Park, J. 2022. Studiogan: A taxonomy and benchmark of gans for image synthesis. *arXiv preprint arXiv:2206.09479*.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kong, Z.; and Chaudhuri, K. 2022. Data Redaction from Pre-trained GANs. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2018. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018): 11.

Locatello, F.; Bauer, S.; Lucic, M.; Raetsch, G.; Gelly, S.; Schölkopf, B.; and Bachem, O. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, 4114–4124. PMLR.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 1003–1011.

Nguyen, Q. P.; Low, B. K. H.; and Jaillet, P. 2020. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33: 16025–16036.

Nguyen, T. T.; Huynh, T. T.; Nguyen, P. L.; Liew, A. W.-C.; Yin, H.; and Nguyen, Q. V. H. 2022. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2536–2544.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Rosen, J. 2011. The right to be forgotten. *Stan. L. Rev. Online*, 64: 88.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Shen, Y.; Gu, J.; Tang, X.; and Zhou, B. 2020. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9243–9252.

Shen, Y.; and Zhou, B. 2021. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1532–1540.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2021. Fast yet effective machine unlearning. *arXiv preprint arXiv:2111.08947*.

Tran, L.; Yin, X.; and Liu, X. 2017. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1415–1424.

Tzelepis, C.; Tzimiropoulos, G.; and Patras, I. 2021. WarpedGANSpace: Finding non-linear RBF paths in GAN latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6393–6402.

Voynov, A.; and Babenko, A. 2020. Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning*, 9786–9796. PMLR.

Wang, B.; and Ponce, C. R. 2021. The geometry of deep generative image models and its applications. *arXiv preprint arXiv:2101.06006*.

White, T. 2016. Sampling generative networks. *arXiv preprint arXiv:1609.04468*.

Yoon, Y.; Nam, J.; Yun, H.; Kim, D.; and Ok, J. 2022. Few-Shot Unlearning by Model Inversion. *arXiv preprint arXiv:2205.15567*.

Yuan, X.; He, P.; Zhu, Q.; and Li, X. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9): 2805–2824.

Zhang, E.; Wang, K.; Xu, X.; Wang, Z.; and Shi, H. 2023. Forget-me-not: Learning to forget in text-to-image diffusion models. *arXiv preprint arXiv:2303.17591*.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, 649–666. Springer.

Zhao, H.; Gallo, O.; Frosio, I.; and Kautz, J. 2016. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1): 47–57.

Appendix

We discuss the model details, training details, and hyper-parameter setting in Section A. we present additional qualitative results and further analysis of experiments in Section B. We then describe the details of the user study in Section C. Finally, Section D offers an expanded explanation of the PGD attack used in Section 5.

A Experimental Details

A.1 Model Details

MNIST. We train DCGAN[4] on the MNIST dataset for 100 epochs with a learning rate of 0.0002. For the VAE, we employ three fully connected layers in both the encoder and decoder, training them for 100 epochs at a learning rate of 0.001.

CelebA. We utilize the ProgGAN[5] and train it on the CelebA dataset for 96,000 iterations, employing a learning rate of 0.001. We train the VDVAE[6] with the imagenet32 configuration from the code.

FFHQ. We use the pre-trained StyleGAN[7] model at a resolution of 1024 pixels. The target feature vector is identified within the W space, i.e., the output space of the mapping network. Also, we do not update the mapping function, which maps z space to W space.

A.2 Training Details

Target feature of the dataset. We specify the number of images that have a target feature within each dataset in Table 3. For MNIST, we set it as having the target feature if it has a measured value using Morpho-MNIST in the top 10%. Also, we select 'Bang' and 'Beard' as target features. These features appear around 10% of the CelebA dataset.

		Target image	Non-target image
MNIST	Thickness	6,000	54,000
	Slant	6,000	54,000
CelebA	Bang	30,709	171,890
	Beard	33,441	169,158

Table 3: Number of the target image and non-target image for each feature used in the experiments.

Oracle model. The oracle model is initialized with the trained model. We further train the oracle model without the target features over 50 epochs and 96,000 iterations for MNIST and CelebA, respectively, to remove the target features as done in (Nguyen, Low, and Jaillet 2020).

A.3 Hyper-parameter Setting

We detail the hyper-parameter used for each dataset in Table 4. The same hyperparameters were applied when unlearning the GAN and VAE trained on the same dataset.

	Learning-rate	α	Epoch	# of sample
MNIST	0.0001	3	200	500
CelebA	0.0002	300	500	500
FFHQ	0.0005	300	500	20

Table 4: Hyper-parameter setting used for each dataset in experiments.

B Additional Results

B.1 Additional Qualitative Results

We further present qualitative results from StyleGAN, visualizing the interpolation of four features: Bang, Beard, Hat, and Glasses. Our results indicate that the target features are removed while preserving high image quality.

[4]https://github.com/pytorch/examples/tree/main/dcgan

[5]https://github.com/facebookresearch/pytorch_GAN_zoo/

[6]https://github.com/openai/vdvae

[7]https://github.com/rosinality/style-based-gan-pytorch

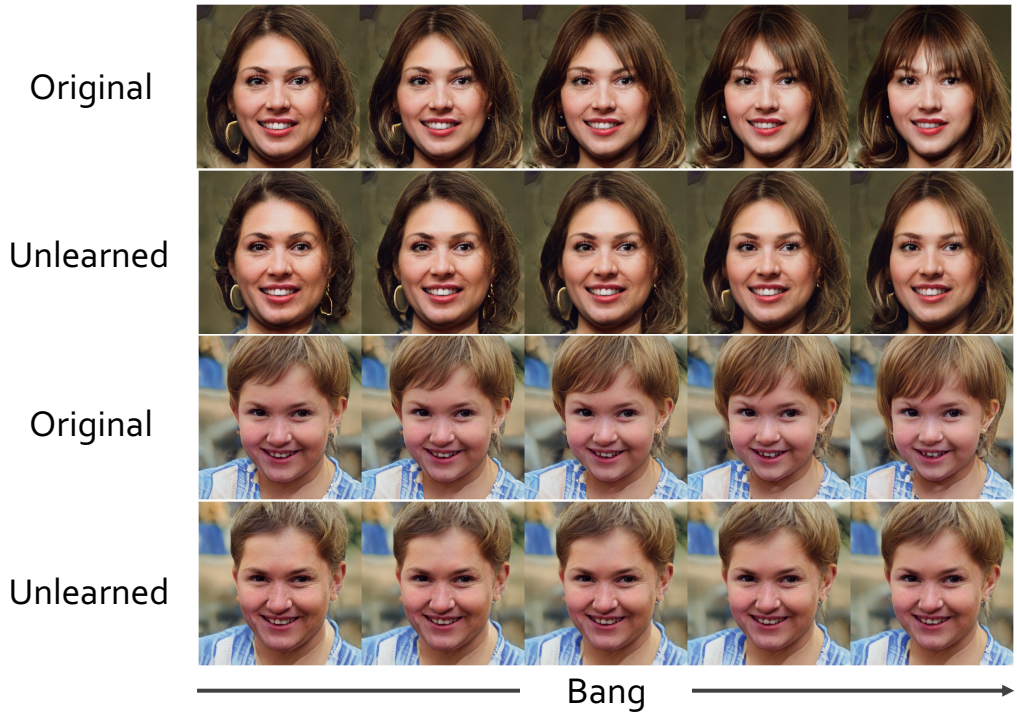

Figure 7: Visualization of ‘Bang’ feature before and after unlearning from pre-trained StyleGAN model. All paired images in each column are generated from the same latent vector.

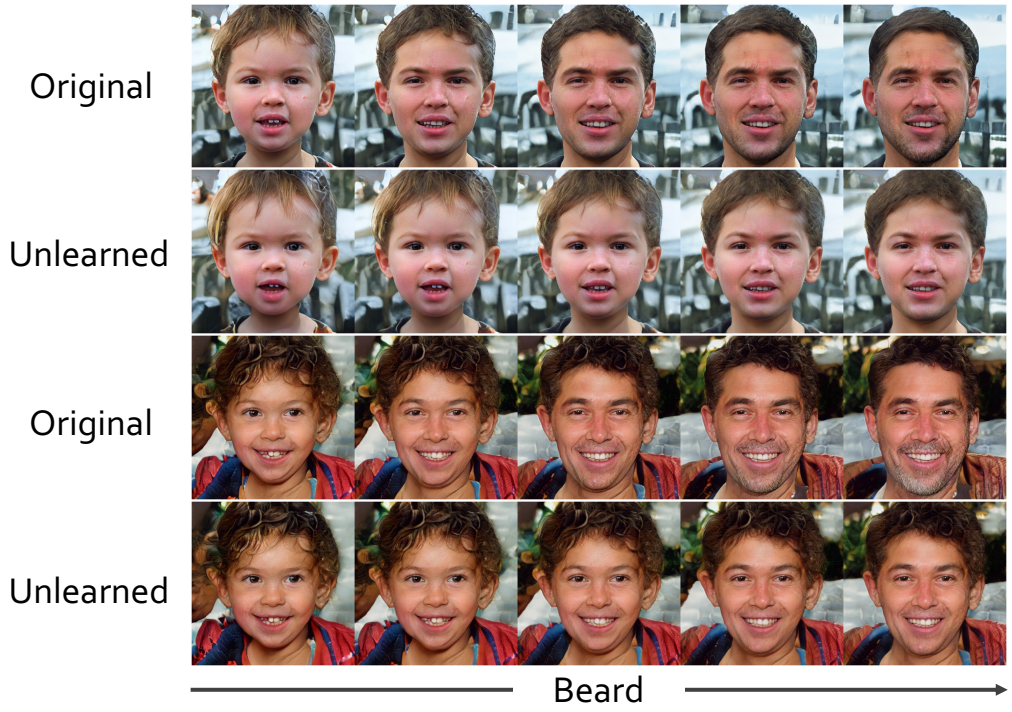

Figure 8: Visualization of ‘Beard’ feature before and after unlearning from pre-trained StyleGAN model. All paired images in each column are generated from the same latent vector.

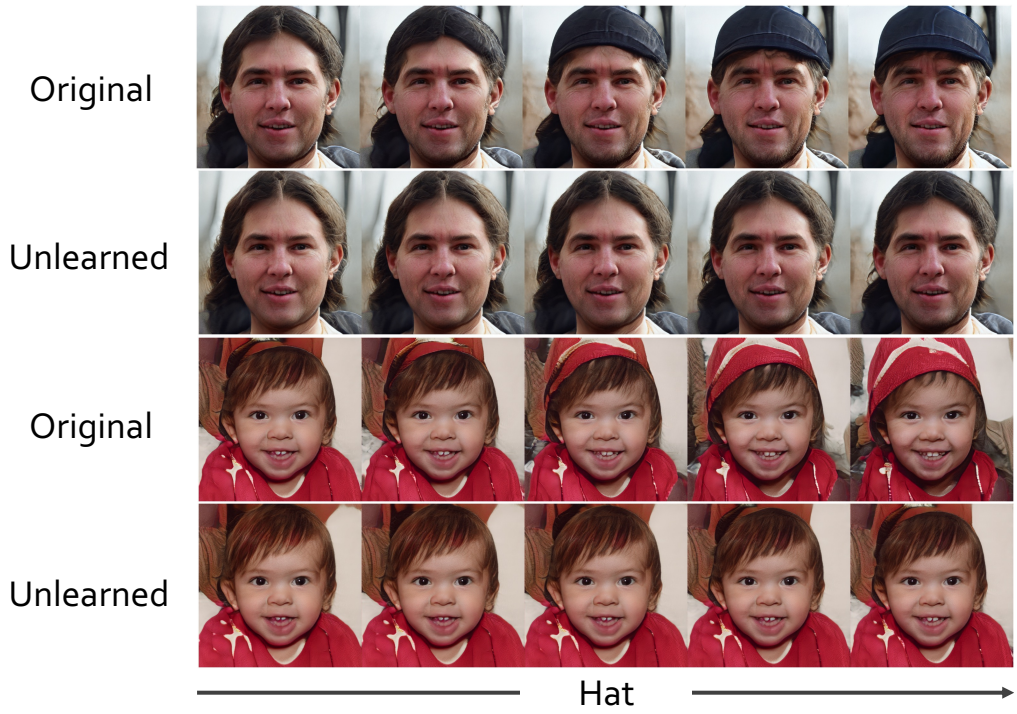

Figure 9: Visualization of ‘Hat’ feature before and after unlearning from pre-trained StyleGAN model. All paired images in each column are generated from the same latent vector.

Figure 10: Visualization of ‘Glasses’ feature before and after unlearning from pre-trained StyleGAN model. All paired images in each column are generated from the same latent vector.

B.2 Analysis of the experiment

Target identification in latent space. The feature identification method proposed in Eq. 1 raises a question about the quality of the result. We use the same classifier used to measure the target feature ratio to measure the quality of the feature identification method. Note that the pre-trained classifiers are only used in the evaluation and not given during unlearning.

Table 5 shows the ROC-AUC score of the feature identification used in each experiment. The feature classification method achieves relatively high accuracy without an external classification model showing that the feature extracted from the latent space can be used for unlearning.

MNIST		
Models	Thickness	Slant
VAE	0.908	0.936
DCGAN	0.853	0.858
CelebA		
Models	Bang	Beard
VDVAE	0.840	0.834
ProgGAN	0.891	0.806

Table 5: ROC-AUC score of feature identification method.

Ablation on the objective. We conduct an ablation study to evaluate the effectiveness of our proposed objective function. Specifically, we experiment with erasing 'Bang' in a pre-trained GAN trained with the CelebA dataset. Table 6 provides a detailed comparison of the performance under different combinations of objectives. Without the perception loss, the model achieves a better unlearning performance in terms of target ratio, sacrificing the quality of images.

$\mathcal{L}_{\text{recon}}$	$\mathcal{L}_{\text{unlearn}}$	$\mathcal{L}_{\text{percep}}$	TFR (%)	IS	FID
	✓		0.08	2.89	52.48
✓	✓		0.38	2.85	50.93
✓	✓	✓	0.42	2.92	49.82
	Baseline		0.49	2.91	51.37
	Original		6.74	2.92	48.05

Table 6: Ablation study to unlearn the 'Bang' feature from a pre-trained GAN. Each metric indicates the target feature ratio (↓), inception score (↑), and Fréchet inception distance (↓).

Choices of hyper-parameter. We analyze the results with varying hyper-parameter α on unlearning the 'Bang' feature from GAN. As shown in Table 7, the importance of the unlearning objective becomes more significant as α increases. Consequently, increasing α results in a decrease in the target ratio, which successfully erases the target feature from the generated images. However, we observe the trade-off between the target feature ratio and the quality of the generated images. Therefore, careful selection of α is important to achieve the desired balance between effective unlearning and preserving image quality.

	Target feature ratio (↓)	IS (↑)	FID (↓)
$\alpha = 1$	5.36	2.87	48.41
$\alpha = 2$	1.08	2.89	49.09
$\alpha = 3$	0.42	2.91	49.82
$\alpha = 4$	0.21	2.89	49.93
$\alpha = 5$	0.01	2.87	50.57

Table 7: Result of hyper-parameter analysis varying α

C User Study

C.1 User Interface.

To conduct the user study, we recruit 13 participants from a graduate school studying machine learning and artificial intelligence. Participants first identify generated images with specific target features (e.g., Glasses). Using the collected dataset, we unlearn pre-trained StyleGAN with our proposed unlearning framework. Then, each participant assesses their customized unlearned model based on three criteria: target feature ratio, image quality, and pinpoint unlearning. This section shows the screenshot from the user study discussed in Section 4.4.

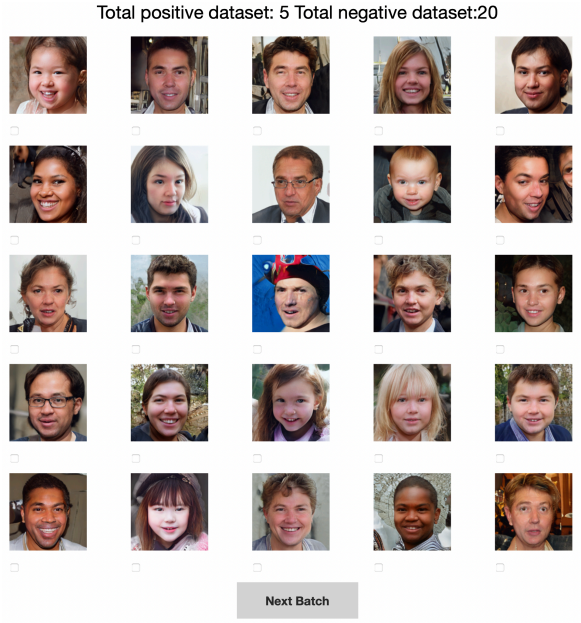

Figure 11: Screenshot for annotating images that have target feature. Participants examine 500 images from generated images.

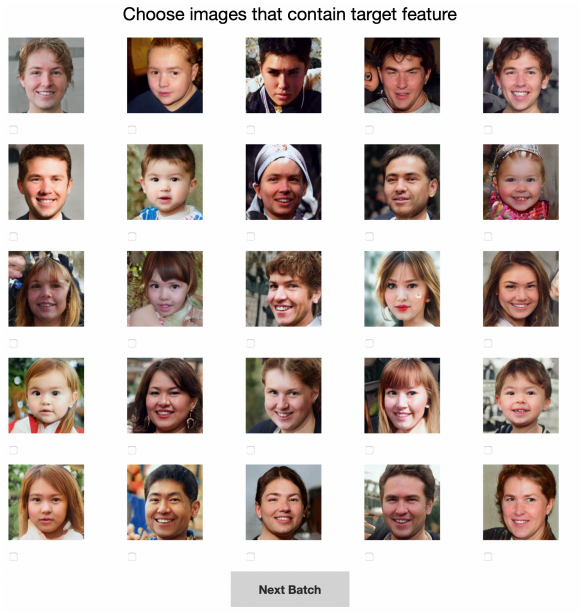

Figure 12: Screenshot for evaluating target feature ratio. Given 500 images, participants select the images that have a target feature.

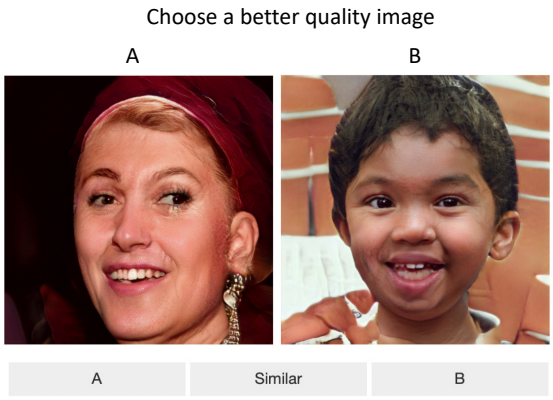

Figure 13: Screenshot for evaluating image quality. Participants choose the better quality image given two random images generated from the original model and the unlearned model.

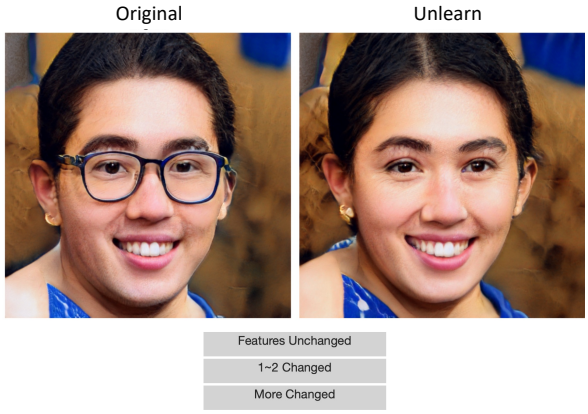

Figure 14: Screenshot for evaluating pinpoint unlearning. Participants are provided with two images: one from the original model and another from the unlearned model. Both are generated using the same latent vector. Then they measure how many features are changed except the target feature.

C.2 Detail results of user study

Table 8 provides the average and standard deviation of three user study experiments.

Experiments	Source	Avg.±Std.
Target feature ratio	Original	47.92±6.04
	Unlearn	3.23±1.92
Image quality	Original	15.15±5.10
	Unlearn	15.23 ±4.54
	Similar	19.61±7.46
Pinpoint unlearning	Unchanged	4.53±2.26
	1∼2 changed	4.38±1.66
	More changed	1.07±1.80

Table 8: Average and standard deviation of user study result.

D PGD Attack

We employ a Projected Gradient Descent (PGD) attack (Madry et al. 2017) on the latent variable of the unlearned model in Section 5. The attack is guided by a pre-trained feature classifier capable of classifying the target feature. The purpose of this attack is to determine whether the unlearned model can be manipulated to produce images that contain the target feature, even after it has been supposedly removed. Specifically, let $h : \mathcal{X} \to [0, 1]$ be the target feature classifier. With randomly sampled latent vector $\mathbf{z}$, the PGD attack aims to find $\tilde{\mathbf{z}}$ such that

$$\tilde{\mathbf{z}} = \underset{\mathbf{z}' \in \Delta_{\mathbf{z}}}{\arg\min}\, \mathrm{CE}(1, h(g_\theta(\mathbf{z}'))),$$

where CE is a cross-entropy loss, and $\Delta_{\mathbf{z}}$ defines a set of possible perturbations from the original value $\mathbf{z}$. Hence, the PGD attack finds the latent variable generating a sample that can be classified as the one with the target feature. The overall method used in this experiment is provided in Algorithm 1.

Algorithm 1: Projected Gradient Descent method

Require: h: target classifier, g_θ: unlearned model
Require: A: attack step, η: learning rate, ϵ: maximum magnitude of perturbation
  Sample a latent vector $\mathbf{z}$ from $N(0, 1)$
  Initialize a perturbation $\mathbf{z}' = \mathbf{z}$
  for $i = 1$ **to** A **do**
    Compute the loss $\mathcal{L} = \mathrm{CE}(1, h(g_\theta(\mathbf{z}')))$
    Compute gradient $\nabla = \frac{\partial \mathcal{L}}{\partial \mathbf{z}'}$
    Update the perturbation $\mathbf{z}' = \mathbf{z}' + \eta \cdot \mathrm{sign}(\nabla)$
    Project $\mathbf{z}'$ to ϵ-ball under the infinity norm:
    $\mathbf{z}' = \min(\max(\mathbf{z}' - \mathbf{z}, -\epsilon), \epsilon)$
  end for
  return $\mathbf{z}'$