졸업논문

적대적 생성 신경망에 적용된 머신 언러닝의 평가 지표 Colored Metric (CM) 제안

Colored Metric (CM) : An Evaluation Metric for Machine Unlearning Applied to Generative Adversarial Networks

2026년

세종과학예술영재학교

정보과학

윤 의 진 (尹 義 診, Youn, Ui Jin)
3210

이 진 명 (李 珍 銘, Lee, Jin Myung)
3410

초록

인공지능 발전에 따라 잠재적 위험성과 윤리적 문제가 대두되고 있으며, 특히 사용자 데이터의 보안과 잊힐 권리에 대한 관심이 증가하고 있다. 머신 언러닝은 원치 않는 데이터를 최소한의 비용으로 삭제하기 위한 방법으로 주목받고 있다. 생성형 모델은 다양한 형태의 데이터를 생성하며, 이러한 데이터의 품질 평가는 대부분 주관적인 판단에 의존한다. 생성형 모델에 적용된 머신 언러닝의 평가로 사용되는 CID index, IS(Inception Score), FID(Fréchet Inception Distance) 등의 평가지표는 머신 언러닝 품질 평가에는 한계가 있다. 따라서 본 연구는 머신 언러닝 대상 클래스가 주변 클래스에 미치는 영향을 평가하는 CM(Colored Metric)을 제안하였으며, 이를 FID와 비교하여 유용성을 검증하였다.

중요어 : 머신 언러닝, 적대적 생성 신경망, 평가지표, 파인튜닝, 프라이버시

목 차

Ⅰ. 서론 (Introduction) ····· 1

Ⅱ. 배경지식 (Related Works) ····· 1

2.1 Preliminaries ····· 1

2.1.1 Introduction to Machine Unlearning ····· 1

2.1.2 Basics of Machine Unlearning ····· 1

2.1.3 Notation ····· 2

2.1.4 Machine Unlearning Classification Models ····· 2

2.1.5 Generative Adversarial Networks ····· 5

2.1.6 Machine Unlearning Generative Models ····· 5

Ⅲ. 설계 및 구현 (Proposed Method) ····· 5

3.1 Machine Unlearning Generative Adversarial Networks ····· 5

3.1.1 Experimental Setup ····· 5

3.1.2 Method ····· 5

3.1.3 Implementation ····· 6

3.1.4 Visualization ····· 6

3.1.5 Model Enhancement ····· 7

3.1.6 Machine Unlearning Conditional GAN ····· 7

3.1.7 Unlearning Algorithm Enhancement ····· 9

3.2 Evaluation Metrics ····· 11

3.2.1 Fréchet Inception Distance (FID) ····· 11

3.2.1.1 Overview ····· 11

3.2.1.2 Implementation ····· 12

3.2.2 Colored Metric (CM) ····· 13

3.2.2.1 Overview ····· 13

3.2.2.2 Implementation ····· 14

Ⅳ. 실험 결과 (Experimental Results) ····· 14
4.1 CM Visualization ····· 14
4.1.1 Experiment 1 ····· 15
4.1.2 Experiment 2 ····· 16
4.1.3 Experiment 3 ····· 16
4.1.4 Experiment 4 ····· 16
4.1.5 Experiment 5 ····· 18
4.1.6 Experiment 6 ····· 19
4.1.7 Experiment 7 ····· 20
4.2 CM Results ····· 21
4.3 Comparison of CM and FID ····· 23

Ⅴ. 고찰 (Discussion) ····· 23
5.1 Machine Unlearning Algorithm ····· 23
5.2 Experimental Environment ····· 23
5.3 Generative Adversarial Networks ····· 23
5.4 Limitations ····· 23

Ⅵ. 결론 (Conclusion) ····· 24

References ····· 25

Abstract ····· 26

표 목 차

Table. 1 Notations ····· 2
Table. 2 Hyperparameters for 3.1.4 ····· 7
Table. 3 Hyperparameters for 3.1.5 ····· 7
Table. 4 Hyperparameters for 3.1.6 ····· 8
Table. 5 Hyperparameters for 3.1.7 ····· 10
Table. 6 Hyperparameters for 4.1 ····· 15
Table. 7 RED Pixel Ratio by Class ····· 16
Table. 8 GREEN Pixel Ratio by Class ····· 17
Table. 9 BLUE Pixel Ratio by Class ····· 17
Table. 10 RED Pixel Ratio by Class ····· 19
Table. 11 GREEN Pixel Ratio by Class ····· 20
Table. 12 BLUE Pixel Ratio by Class ····· 21

그 림 목 차

Figure. 1 Machine Unlearning Schematic Diagram ····· 2
Figure. 2 Distribution of Two Data Sets across Three Significant Models ····· 4
Figure. 3 Image Generated by Algorithm 1 (GAN-based model) ····· 7
Figure. 4 L2 Loss over Epochs during Training with Algorithm 1 ····· 8
Figure. 5 Image Generated by Algorithm 1 (CGAN-based model) ····· 9
Figure. 6 L2 Loss over Epochs during Training with Improved Algorithm 1 ····· 10
Figure. 7 Image Generated by Improved Algorithm 1 (CGAN-based model) ····· 11
Figure. 8 Comparison of FID Differences for Three Forget Sets ····· 12
Figure. 9 CM Assumption Schematic Diagram ····· 13
Figure. 10 CM Visualization Results of Experiment 1, 2, 3, 4 ····· 15
Figure. 11 RGB Pixel Ratio Bars for Each Experiment ····· 18
Figure. 12 CM Visualization Results of Experiment 5 ····· 18
Figure. 13 RED Pixel Ratio Bar ····· 19
Figure. 14 CM Visualization Results of Experiment 6 ····· 19
Figure. 15 GREEN Pixel Ratio Bar ····· 20
Figure. 16 CM Visualization Results of Experiment 7 ····· 21
Figure. 17 BLUE Pixel Ratio Bar ····· 21

Figure. 18 Colored Metric Matrix (CMM) ····· 22

Ⅰ. 서론 (Introduction)

인공지능의 급속한 발전과 함께 데이터 보안 및 프라이버시 문제가 중요한 이슈로 대두되고 있다. 특히 Membership Inference Attack (MIA) 등 악의적인 공격으로 모델 훈련 데이터의 유출 가능성이 높아졌다. 인공지능은 의료, 금융 등 민감한 데이터를 다루는 분야에서도 사용되므로, 데이터 보안 및 개인 정보 보호의 중요성이 높아지고 있다. 또한, General Data Protection Regulation (GDPR) 제17조와 California Consumer Privacy Act (CCPA) 등의 규정으로 사용자가 모델에서 개인 데이터 삭제를 요구할 수 있으며, 이를 수행해야 한다.

모델의 규모가 증가하며, 사전 훈련 데이터 셋의 규모 또한 기하급수적으로 증가하고 있다. 따라서 데이터를 제거하기 위해 모델을 처음부터 재훈련하는 것은 시간과 비용 측면에서 비효율적이다. 이를 해결하기 위해 머신 언러닝(Maching Unlearning)이 주목 받고 있다. 머신 언러닝은 모델에서 특정 데이터를 선택적으로 제거하여 모델이 해당 데이터를 잊도록 하는 방법이다. 분류 모델(Classification Model)의 머신 언러닝 기법과 평가 지표에 대한 많은 연구가 이루어졌다. 반면, 생성형 모델(Generative Model)은 머신 언러닝 기법과 평가 지표에 대한 많은 연구가 이루어지지 않았다(Zhou et al., 2024).

생성형 모델은 이미지, 텍스트, 오디오 등 다양한 형태의 데이터를 생성한다. 이러한 데이터의 품질 평가는 대부분 주관적 판단에 의존한다. 많은 연구에서 생성형 모델에 적용된 머신 언러닝의 평가 지표로 CID index, Inception Score (IS), Fréchet Inception Distance (FID) 등을 사용하였다(Heng & Soh, 2023; Li et al,. 2024). 이들은 생성형 모델의 성능을 평가하는 지표로, 머신 언러닝의 성능을 평가하는데에 한계가 있다. 본 연구에서는 적대적 생성 신경망에 적용된 머신 언러닝의 평가 지표 Colored Metric (CM)을 제안하고, 이의 유용성을 입증 하고자 한다. 본 연구에서 사용된 리소스 및 구현 코드는 https://github.com/nobel2040ne/Colored-Metric 에서 확인할 수 있다.

Ⅱ. 배경지식 (Related Works)

2.1 Preliminaries

2.1.1 Introduction to Machine Unlearning

머신러닝(Machine Learning) 모델의 규모가 증가하며, 사전 훈련 데이터 셋의 규모 또한 증가하고 있다. 사전 훈련 데이터 셋에는 개인 데이터, 유료 저작권이 포함된 데이터, 유해한 데이터와 잘못된 데이터 등이 포함되어 있다. 머신 언러닝은 훈련된 모델에서 이와 같이 원하지 않는 데이터를 제거하는 것을 의미한다. 특히, 모델을 처음부터 재학습하지 않고 최소한의 비용으로 머신 언러닝을 수행하는 것이 주목받고있다.

2.1.2 Basics of Machine Unlearning

[Figure. 1]은 머신 언러닝의 기본적인 흐름과 대표적인 평가 방법을 제시한다.

본 연구에서는 A, B, C를 Train Set, A를 Forget Set, B, C를 Retain Set이라 명칭하였다. Forget Set은 모델에서 삭제하고자 하는 데이터이다. Retain Set은 머신 언러닝 이후 유지되는 데이터이다. 즉, Forget Set과 여집합 관계이다. Pre Trained Model은 Train Set을 학습한 모델이다. Ground Truth Model은 Retain Set을 학습한 모델으로, 머신 언러닝 후 모델은 Gruond Truth Model과 유사해야한다. Unlearned Model은 Pre Trained Model에 Unlearning algorithm을 적용한 모델이다. 머신 언러닝의 기초적인 평가방법은 Unlearned Model과 Ground Truth Model의 유사도가 높을 수록 머신 언러닝의 성능이 좋다고 평가한다.

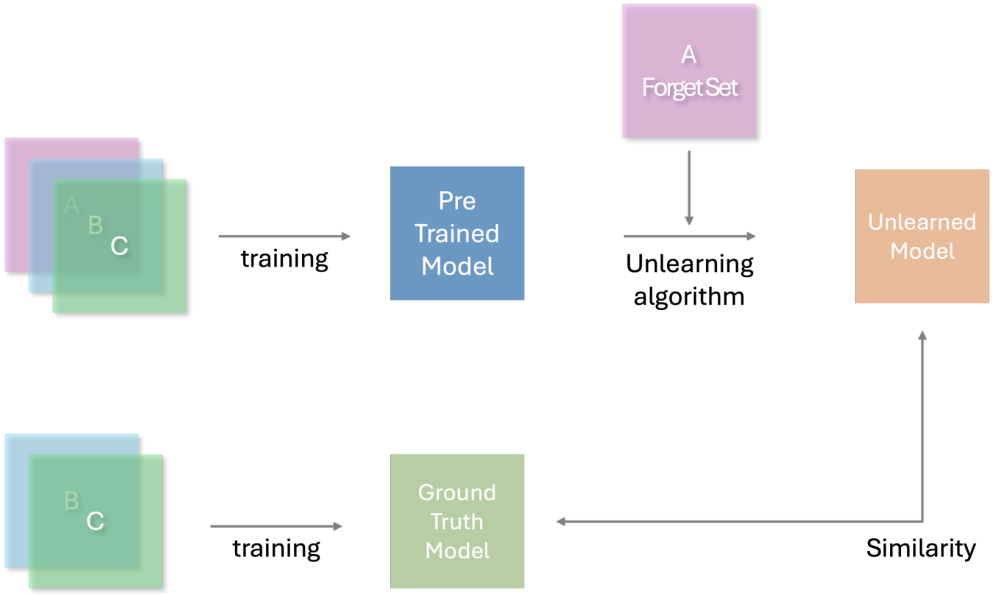


Figure. 1 Machine Unlearning Schematic Diagram

2.1.3 Notation

Table. 1 Notations

Notations	Descriptioins
A	Model
R	All Data Set
D	Data Set
S	Forget Set
U	Unlearning Algorithm
Pr	Probability
$EFF_{i,\ j}$	The Impact of Class i on Class j in Machine Unlearning
EFF_i	Average(%) of the Influence of Class i on the Remaining Classes in Maching Unlearning

2.1.4 Machine Unlearning Classification Models

분류 모델의 머신 언러닝에는 다양한 방법이 제시되고 있다. 첫 번째, Random

Forest의 변형인 DaRE Forests를 통해 최소한의 재훈련으로 데이터를 제거하며, 성능 저하 없이 데이터 삭제 속도를 향상시켰다(Brophy & Lowd, 2021). 두 번째, 경사 하강법을 사용해 모델 성능에 큰 영향을 미치는 데이터를 효율적으로 식별하고 삭제하는 약한 삭제를 제안하였다(Neel et al., 2021). 세 번째, 모델의 일반성을 높이기 위해 노이즈를 추가하는 Error Minimizing-Maximizing Noise와 지식 전이를 조절하는 Gated Knowledge Transfer를 제안하였다(Foster et al., 2024). 또한, Starting kit for the NeurIPS 2023 Machine Unlearning Challenge는 분류 모델에서의 머신 언러닝 알고리즘의 기초적인 기법과 평가 지표를 제공한다(NeurIPS 2023 Machine Unlearning Challenge, 2023). Resnet-18과, CIFAR10 데이터셋을 사용하고, 언러닝 알고리즘 중 가장 기초적인 파인튜닝(Fine-Tuning) 기법을 적용하였다. 평가에 사용된 수식은 (1)와 같이 표현된다.

$$\Pr[A(D\backslash S) \in R] \leq e^{\varepsilon} \Pr[U(A(D), S, D) \in R] + \delta \tag{1}$$

좌변 확률 항은 Ground Truth Model의 오차 분포를 의미하며, 우변 확률 항은 Unlearned Model의 오차 분포를 의미한다. 임의의 양수 ϵ에 대하여 부등식이 성립하므로 (1)을 통하여 두 모델의 유사도를 비교하여 머신 언러닝의 성능을 평가할 수 있다. MIA 또한 머신 언러닝 성능 평가에 사용된다. MIA는 주어진 데이터가 모델 학습에 사용되었는지를 추론하여 모델의 보안 취약성을 평가하는 방법 중 하나이다. $MIA_{accuracy}$는 0에서 1사이의 값으로 나타나며, 이 값이 1에 가까울수록 추론된 데이터가 모델 학습에 사용되었을 확률이 높다고 판단한다. 머신 언러닝 이후 Forget Set이 모델에서 제거되어 $MIA_{accuracy}$가 낮아질 것을 기대할 수 있다. 또한, Unlearned Model과 Ground Truth Model의 $MIA_{accuracy}$를 비교하면 (2)와 같은 관계로 나타난다.

$$MIA_{accuracy}(U(A(D),\ D,\ S)) > MIA_{accuracy}\ (A(D\backslash S)) \tag{2}$$

이는 Ground Truth Model에 Forget Set이 전혀 없으므로 MIA 추론이 더 어렵다.

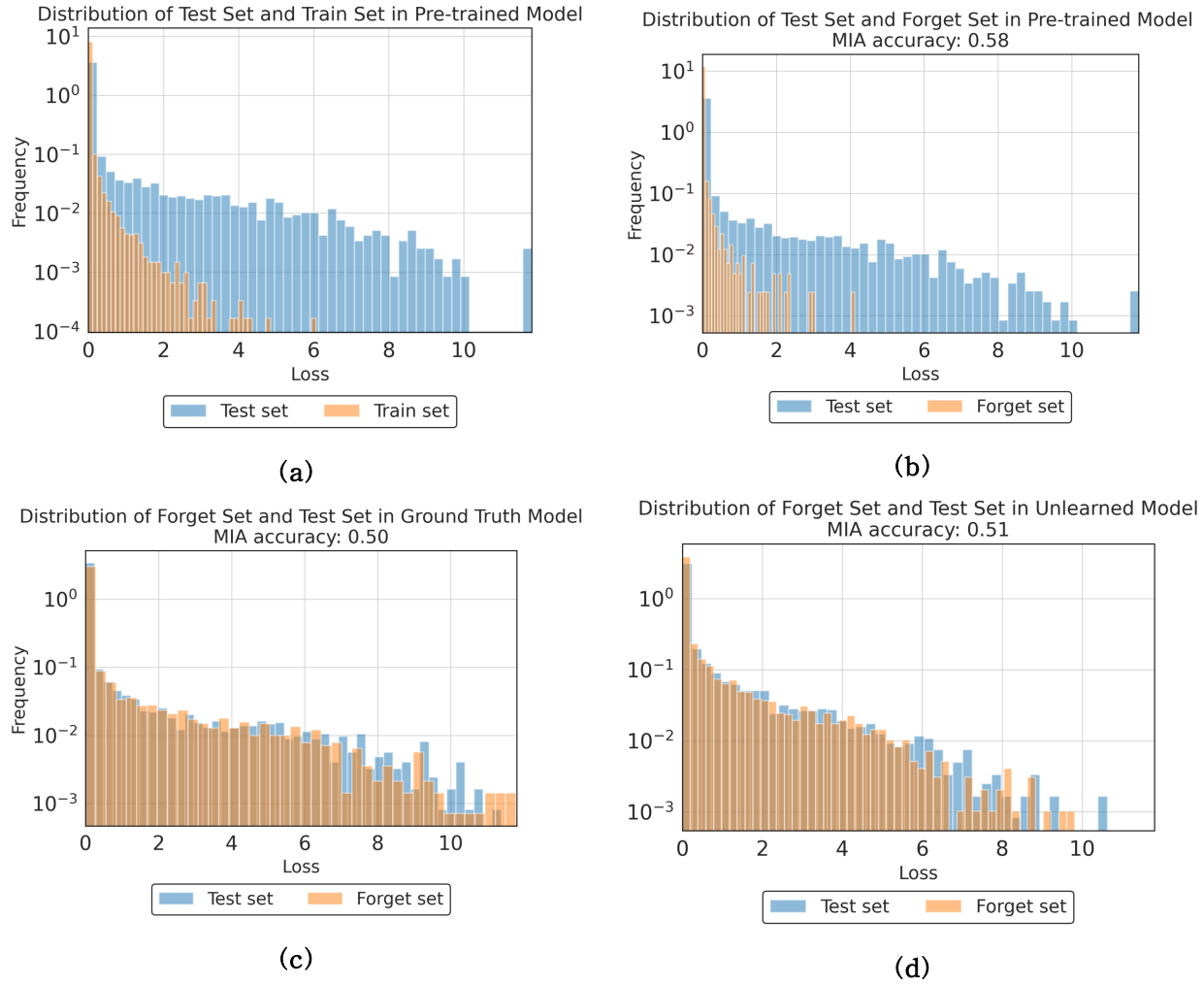


Figure. 2 Distribution of Two Data Sets across Three Significant Models **(a)** Distribution of Test Set and Train Set in Pre-trained Model **(b)** Distribution of Test Set and Forget Set in Pre-trained Model **(c)** Distribution of Forget Set and Test Set in Unlearned Model **(d)** Distribution of Forget Set and Test Set in Ground Truth Model

히스토그램 [Figure. 2]에서 x축은 오차(Loss), y축은 빈도(Frequency)이다. 즉, MIA 결과에서 오차가 x인 판단이 y의 빈도로 일어남을 의미한다. 첫째, [Figure. 2 (a)]는 모델의 Test Set과 Train Set 대한 MIA 결과이다. Train Set은 모델 훈련에 사용되었으므로 오차가 작다. 반대로, Test Set은 오차가 크다. 둘째, [Figure. 2 (b)]과 [Figure. 2 (c)]를 비교하면, 머신 언러닝 이후 Forget Set에 대한 MIA 결과가 Train Set이 아닌 Test Set과 유사하다. 즉, 머신 언러닝 이후 Forget Set이 모델에 학습되지 않은 Test Set과 유사함을 통해 머신 언러닝이 수행되었음을 알 수 있다. 셋째, [Figure. 2 (d)]는 Ground Truth Model의 Test Set과 Forget Set에 대한 MIA 결과이다. 즉, 머신 언러닝 이후 모델이 가져야할 이상적인 결과이다. [Figure. 2 (c)]와 [Figure. 2 (d)]의 결과는 유사하므로 머신 언러닝이 이루어졌음을 확인할 수 있다. 정리하자면, Unlearned Model과 Ground Truth Model의 MIA 결과가 유사할수록 머신 언러닝이 잘 수행되었음을 의미한다. 이에 대한 정량적 수식이 (1)이다.

2.1.5 Generative Adversarial Networks

적대적 생성 신경망(Generative Adversarial Networks, GAN)은 두 개의 신경망인 생성자(Generator)와 판별자(Discriminator)가 서로 경쟁하며 학습하는 구조를 가진다. 생성자는 원본 데이터와 유사한 새로운 데이터를 생성하고, 판별자는 이 데이터가 원본인지 새로 생성된 데이터인지 판별한다. 이와 같은 학습 과정은 최소-최대 문제(Min-max Problem)로 표현되며, 두 신경망이 최적의 균형 상태에 도달하기 위한 안장점(Saddle Point)을 찾는 것이 핵심이다.

2.1.6 Machine Unlearning Generative Models

현재 진행된 생성형 모델에서의 머신 언러닝 연구는 다음과 같다. 첫 번째, SISA Framework의 문제점을 지적하고, Substitution Mechanism과 Fake Label을 활용한 Cascaded Unlearning 방법을 제안하여 Modified National Institute of Standards and Technology database (MNIST) 데이터셋에서 모델 재학습 시간을 185배, CIFAR10에서 284배 단축시켰다(Sun et al., 2023). 두 번째, 확산 모델(Diffusion Model)과 VQ-GAN을 사용한 머신 언러닝 프레임워크를 제시하였고, 이미지에 마스킹을 적용한 후 언러닝 전후의 결과를 비교하였다(Li et al., 2024).

Ⅲ. 설계 및 구현 (Proposed Method)

3.1 Machine Unlearning Generative Adversarial Networks

3.1.1 Experimental Setup

본 연구에서 모든 실험은 Google Colaboratory Pro에서 진행되었으며, GPU를 사용하여 모델을 학습하였다. 모델은 적대적 생성 신경망을 사용하였으며, 머신 언러닝 기법으로 파인 튜닝을 사용하였다. 데이터 셋은 MNIST를 사용하였다. 머신 언러닝시 Forget Set은 클래스 0, Retain Set은 그 외의 클래스로 설정하였다.

3.1.2 Method

머신 언러닝 기법으로 생성자의 가중치를 업데이트하는 파인 튜닝 기법 방식을 사용하였다. 손실 함수 $L2\ Loss$와 $Total\ Loss$를 각각 (3)과 (4) 같이 정의하였다:

$$L2\ Loss = \sum_{i=1}^{n} (y_{i,\ true} - y_{i,\ predict})^2 \tag{3}$$

$$Total\ Loss = L2\ Loss_{retain} - \alpha \times L2\ Loss_{forget} \tag{4}$$

여기서 $L2\ Loss$는 실제 값과 예측 값의 차이에 비례하는 오차를 의미한다. $Total\ Loss$는 Retain Set과 Forget Set의 $L2\ Loss$ 차이이며, 하이퍼파라미터 α으로 $L2\ Loss_{forget}$의 증감 정도를 조절할 수 있다. 생성자는 $Total\ Loss$가 최소화되도록 훈련된다. 이 과정에서 생성자의 $L2\ Loss_{retain}$을 최소화하여 생성된 이미지와 Retain Set과의 차이를 줄이고, $L2\ Loss_{forget}$을 최대화하여 Forget Set과의 차이를 늘리는 효과를 갖는다. 결과적으로 모델이 생성한 이미지가 이전보다 Retain Set과 유사해지며, Forget Set과 차이가 커진다.

3.1.3 Implementation

적용한 머신 언러닝 알고리즘의 의사 코드는 다음과 같다:

Algorithm 1 Pseudo Code of Our Unlearning Algorithm

1: For each epoch:

$generated_{images} = G(z)$ # create image

$real_{images} = Sample(data)$ # sampling real images

$Total_{loss} = L2_{loss,\ retain} - \alpha L2_{loss,\ forget}$ # compute total loss

$gradients = ComputeGradients(Total_{loss},\ G)$ # compute gradients

$UpdateParameters(G,\ gradients,\ \eta)$ # update parameters

End for

$G(z)$는 적대적 생성 신경망의 생성기이며, z는 사전 정의된 분포에서 샘플링된 랜덤한 잠재 벡터이다. G는 z를 이미지 공간으로 매핑하여 생성한 이미지를 $generated_{images}$에 저장한다. $real_{images}$는 $Sample(data)$를 통해 Train Set에서 샘플링된 실제 이미지의 배치이다. 이는 학습 과정에서 손실을 계산하기 위해 실제 데이터를 저장한다. $gradients$는 G의 파라미터에 대한 그래디언트로, 손실 함수 $Total_{loss}$의 최소화를 위해 파라미터를 업데이트하는 데 필요한 변화 방향과 크기를 제공한다. $UpdateParameters(G,\ gradients,\ \eta)$에서 $gradients$와 학습률 η로 G의 파라미터를 업데이트한다.

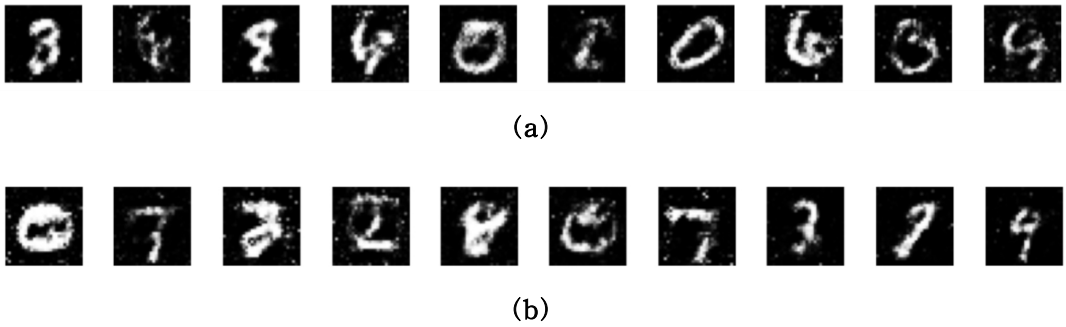
3.1.4 Visualization

Algorithm 1의 하이퍼파라미터는 다음과 같이 설정하였다:

Table. 2 Hyperparameters for 3.1.4

Hyperparameters	Values
Epochs	50,000
Batch Size	64
Learning Rate	0.00001
α	1,000
Forget Set	{0}
Retain Set	{1, 2, 3, 4, 5, 6, 7, 8, 9}

머신 언러닝 적용 전과 후의 모델이 생성한 이미지는 각각 [Figure. 3 (a)]와 [Figure. 3 (b)]에서 확인할 수 있다.



(a)

(b)

Figure. 3 Image Generated by Algorithm 1 (GAN-based Model) **(a)** Before Unlearning **(b)** After Unlearning

3.1.5 Model Enhancement

기본적인 적대적 생성 신경망은 [Figure. 3 (a)]과 [Figure. 3 (b)]에서 볼 수 있듯이, 사용자가 생성할 이미지에 대한 명시적인 제어가 불가능하다는 한계점이 있다. 이러한 한계를 극복하기 위해, 조건부 적대적 생성 신경망(Conditional GAN, CGAN)으로 모델을 수정하였다(Mirza, 2014). [Figure. 5]는 조건부 적대적 생성 신경망을 다음과 같은 하이퍼파라미터로 설정한 후, 각 클래스별로 5개의 이미지를 생성한 결과이다.

Table. 3 Hyperparameters for 3.1.5

Hyperparameters	Values
Epochs	50
Batch Size	100
Learning Rate	0.0002
Adam Beta 1	0.5
Adam Beta 2	0.999
Critic Steps	5
GP(Gradient Penalty Weight)	10

3.1.6 Machine Unlearning Conditional GAN

구현한 조건부 적대적 생성 신경망에 Algorithm 1을 적용하였으며, Algorithm 1의 하이퍼파라미터는 다음과 같이 설정하였다:

Table. 4 Hyperparameters for 3.1.6

Hyperparameters	Values
Epochs	100
Batch Size	100
Learning Rate	0.000005
α	10
Forget Set	{4}
Retain Set	{0, 1, 2, 3, 5, 6, 7, 8, 9}

머신 언러닝 적용 전과 후의 모델이 생성한 이미지는 각각 [Figure. 5 (a)]와 [Figure. 5 (b)]에서 확인할 수 있다.

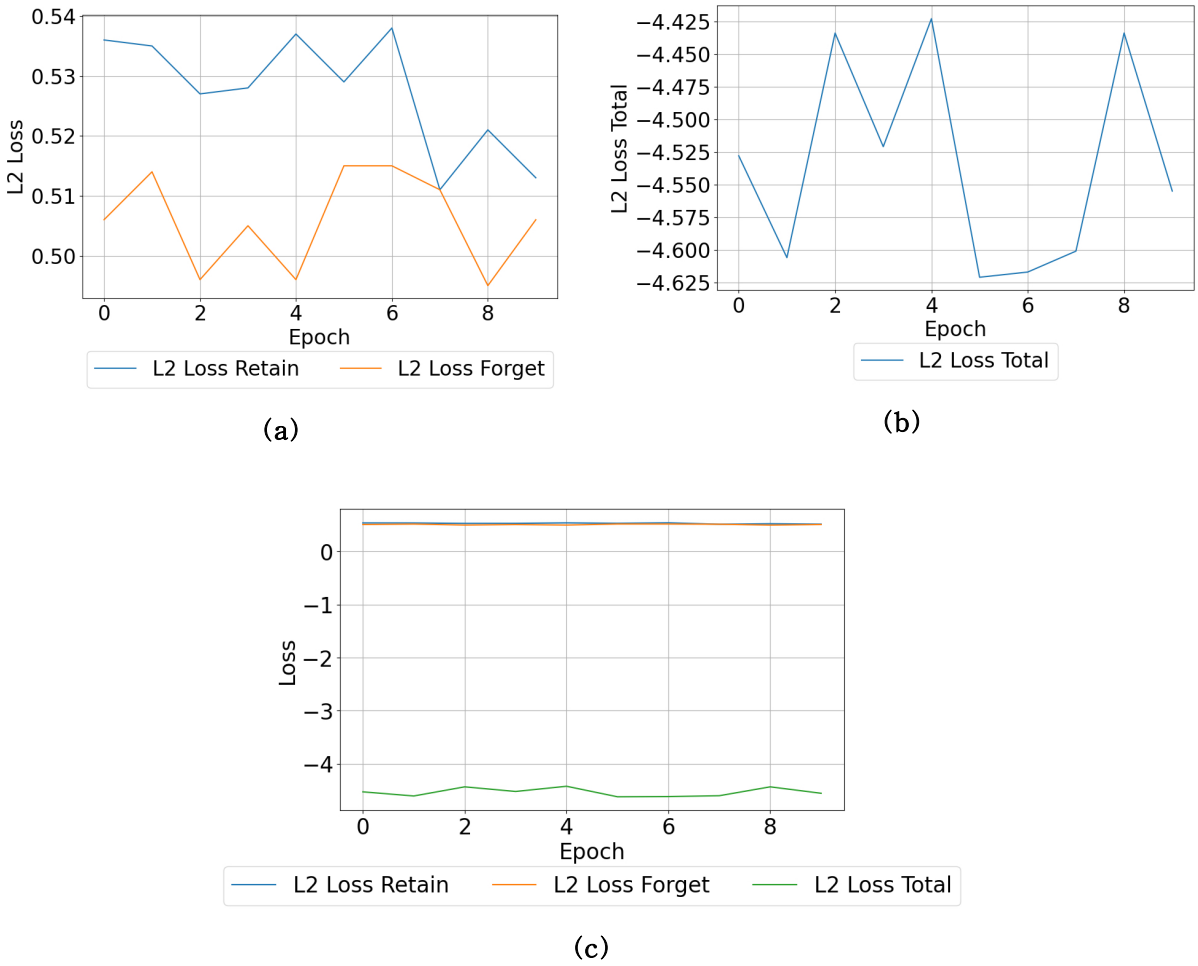
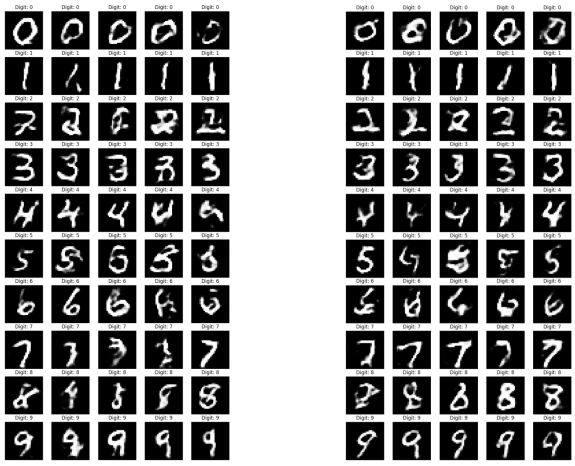


Figure. 4 L2 Loss over Epochs during Training with Algorithm 1 **(a)** L2 Loss Trends for Retain and Forget Over Epochs **(b)** L2 Loss Total Trend Over Epochs **(c)** L2 Loss Trends for Retain, Forget and Total Over Epochs



(a) (b)

Figure. 5 Image Generated by Algorithm 1 (CGAN-based Model) **(a)** Before Unlearning **(b)** After Unlearning

[Figure. 4 (a)], [Figure. 4 (b)], [Figure. 4 (c)]에서 Epochs를 100으로 설정하였음에도, 각 그래프에서는 Epochs가 0부터 10까지만 나타나 있다. 이는 머신 언러닝 과정에서 가중치 업데이트가 10번만 진행되었기 때문이다. 그래프를 보면, *L2 Loss*의 변화 경향이 명확하지 않음을 확인할 수 있다. 이상적으로, $L2\ Loss_{retain}$은 점진적으로 감소해야 하며, $L2\ Loss_{forget}$은 점진적으로 증가해야 한다. 그러나 관찰된 결과에서는 이러한 경향이 뚜렷하지 않아, 실행마다 오차가 안정적으로 감소하거나 증가할 것을 보장하기 어렵다. 이는 Algorithm 1의 개선 필요성을 시사한다.

3.1.7 Unlearning Algorithm Enhancement

Algorithm 1에서 *Total Loss*를 최소화하는 과정에서, 실험한 하이퍼파라미터 값에 대하여 $L2\ Loss_{retain}$의 감소 비율보다 $L2\ Loss_{forget}$의 증가 비율이 더 큰 경향을 보였다. 그 결과, $L2\ Loss_{forget}$이 증가함에 따라 모델이 Forget Set의 주변 데이터에도 영향을 미쳐, Retain Set에 대한 생성 능력이 저하되는 현상이 관찰되었다. 이를 해결하기 위해 *Total Loss*를 각각 (5), (6), (7)과 같이 재정의하여 이러한 문제를 해결하였다:

$$Positive\ Loss = L2\ Loss_{retain} \tag{5}$$

$$Negative\ Loss = margin\ -\ L2\ Loss_{forget} \tag{6}$$

$$Total\ Loss = Positive\ Loss\ +\ \alpha\ \times\ Negative\ Loss \tag{7}$$

$L2\ Loss_{forget}$이 $margin$보다 커지는 경우, $Negative\ Loss$를 0으로 설정하였다. 이를 통해 $L2\ Loss_{forget}$이 과도하게 커지지 않도록 제어할 수 있다. 개선된 알고리즘의 하이퍼파라미터는 다음과 같이 설정하였다:

Table. 5 Hyperparameters for 3.1.7

Hyperparameters	Values
Epochs	1000
Batch Size	100
Learning Rate	0.000005
$margin$	1
α	50
Forget Set	{4}
Retain Set	{0, 1, 2, 3, 5, 6, 7, 8, 9}

머신 언러닝 적용 전과 후의 모델이 생성한 이미지는 각각 [Figure. 7 (a)]와 [Figure. 7 (b)]에서 확인할 수 있다.

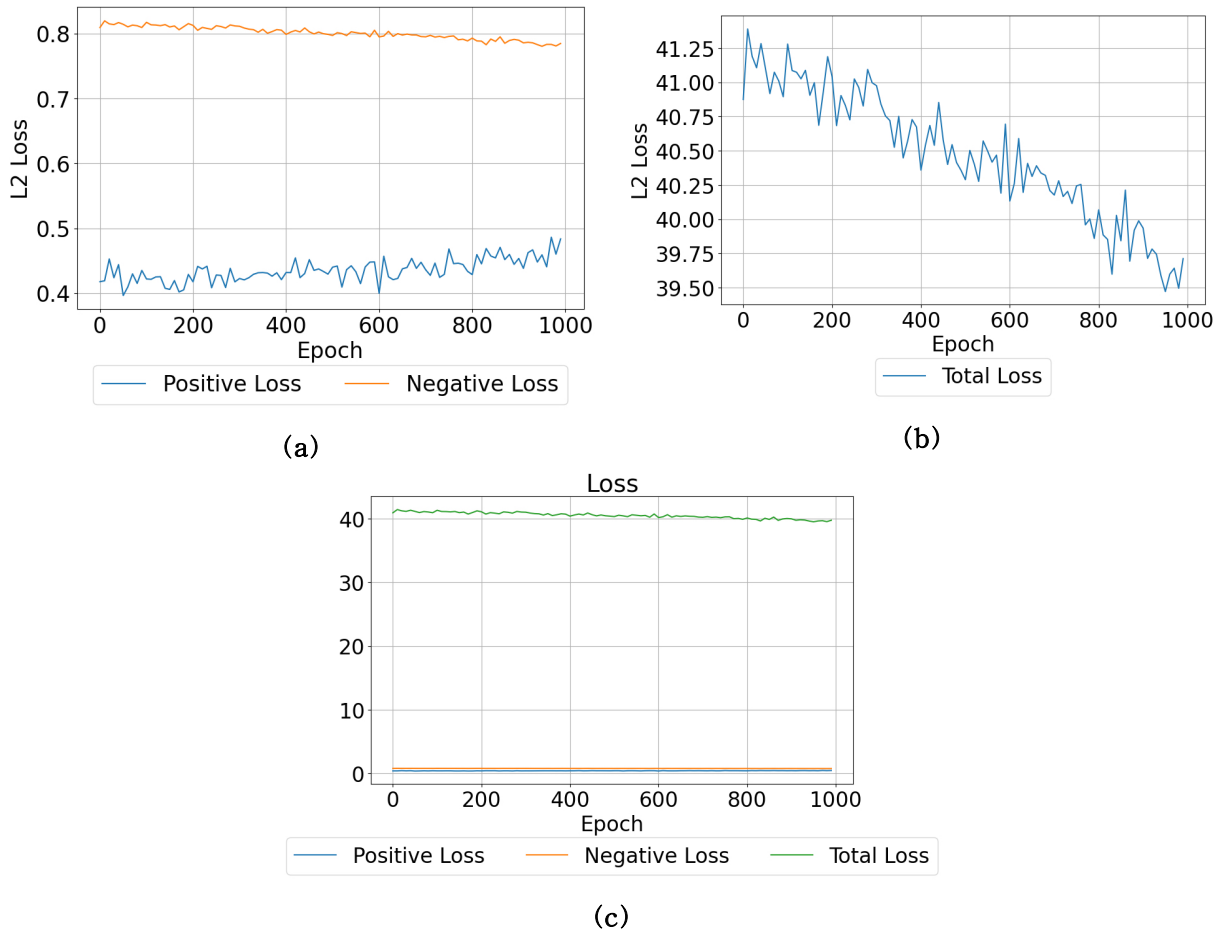
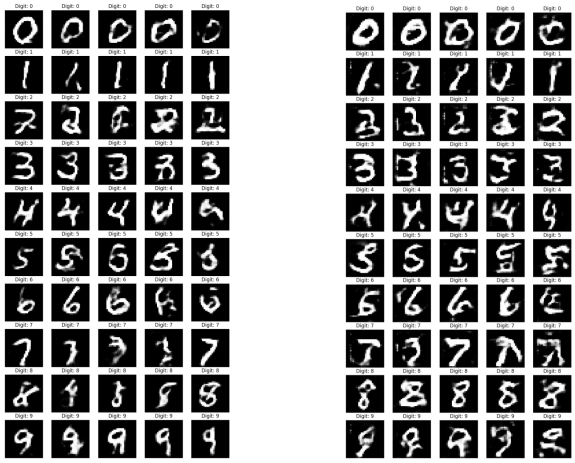


Figure. 6 L2 Loss over Epochs during Training with Improved Algorithm 1 **(a)** L2 Loss Trends for Positive and Negative Loss Over Epochs **(b)** L2 Loss Total Trend Over Epochs **(c)** L2 Loss Trends for Positive, Negative and Total Over Epochs



(a) (b)

Figure. 7 Image Generated by Improved Algorithm 1 (CGAN-based Model) **(a)** Before Unlearning **(b)** After Unlearning

[Figure. 4 (a)], [Figure. 4 (b)], [Figure. 4 (c)]과는 달리, [Figure. 6 (a)], [Figure. 6 (b)], [Figure. 6 (c)]에서는 *Positive Loss*가 증가하는 경향을, *Negative Loss*는 감소하는 경향을, 그리고 *Total Loss*가 전반적으로 감소하는 경향을 보인다. 또한, 모든 Epochs에서 가중치 업데이트가 진행되는 등 더 안정된 성능을 보인다. 다른 하이퍼파라미터 조합에서도 이와 같은 경향을 가짐을 확인하였다. 따라서, 안정성이 개선되었으며 추후 실험에 사용하기 적합하다.

3.2 Evaluation Metrics

3.2.1 Fréchet Inception Distance (FID)

3.2.1.1 Overview

FID는 생성된 이미지의 분포를 실제 이미지 집합과 비교하여 GAN의 성능을 평가하는 지표이다. FID 값이 클수록 원본과 차이가 크다는 것을 의미한다. FID는 Inception v3의 출력 레이어를 제거하고 마지막 풀링 레이어의 활성화 값을 사용하여 각 이미지를 2048차원의 특징 벡터로 표현한다. 이러한 특징 벡터들을 이용하여 이미지의 각 차원별 평균과 표준편차를 계산한다. FID 계산 알고리즘은 아래와 같다:

Algorithm 2 Fréchet Inception Distance Calculation method

1: Mean, covariance calculation for two datasets → $\mu_1, \mu_2, \sigma_1, \sigma_2$

2: $ssdiff = \left(\mu_1^2 - \mu_2^2\right)$

3: $covmean = \sqrt{\sigma_1 \sigma_2}$

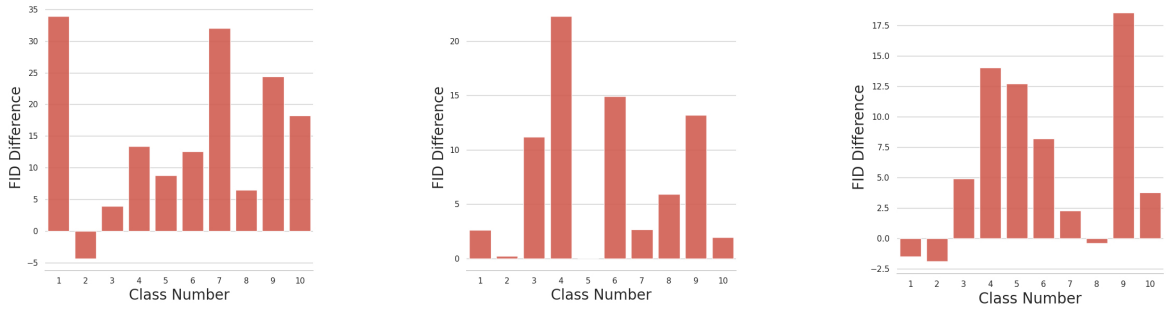
4: FID : $ssdiff + Tr(\sigma_1 + \sigma_2 - 2 \times covmean)$

End for

이 알고리즘에서 μ, σ는 각각 평균 벡터와 공분산 행렬을 말한다. 공분산 행렬은 행렬의 주대각선 성분에 각 변수의 분산이 위치하며, 주대각선 이외의 성분에는 가능한 두 변수 간의 공분산 값을 나타낸다. 공분산은 한 변수가 변화할 때 다른 변수가 어떻게 연관되어 변하는지를 나타낸 값이다. $ssdiff$는 평균 벡터 간의 거리 제곱을 의미하며 Tr은 대각합(Trace)을 의미한다.

3.2.1.2 Implementation

새로운 평가지표로 머신 언러닝이 얼마나 잘 수행되었는지 평가하기 전에, 먼저 언러닝이 실제로 제대로 이루어졌는지 확인하기 위해, 0, 3, 8을 언러닝했을 때 각 숫자의 FID 값 변화를 계산하였다. [Figure. 8 (a)], [Figure. 8 (b)], [Figure. 8 (c)]은 각 클래스마다 FID 값 변화를 나타낸다.



(a) (b) (c)

Figure. 8 Comparison of FID Differnces for Three Forget Sets **(a)** Forget Set : {0} **(b)** Forget Set : {3} **(c)** Forget Set : {8}

각 막대그래프에서 Class Number는 실제 클래스에 1은 더한 값으로 해석한다. 각 실험에서 Forget Set의 FID Difference가 가장 크게 나타났으며, 이를 통해 언러닝이 제대로 수행되었음을 알 수 있었다. 또한 0을 언러닝했을 때 0과 형태가 비슷한 숫자인 6과 8의 FID 값이 크게 증가한 점, 3을 언러닝했을 때 5와 8의 FID 값이 증가한 점, 그리고 8을 언러닝했을 때 3의 FID 값이 증가한 점을 통해 형태가 유사한 숫자들이 언러닝의 영향을 크게 받는다는 사실을 확인할 수 있었다.

3.2.2 Colored Metric (CM)

3.2.2.1 Overview

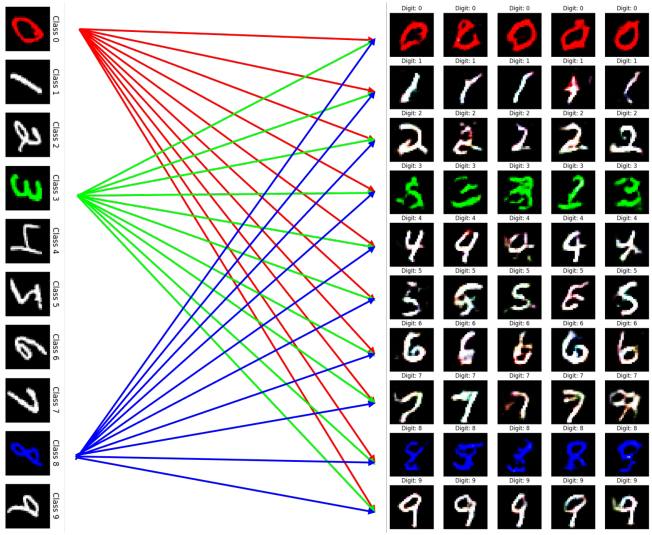


Figure. 9 CM Assumption Schematic Diagram

Colored Metric (CM)은 대부분의 생성형 모델의 머신 언러닝 연구에서 사용되는 평가 지표가 반영하지 못하는 요소에 초점을 맞추었다. 대부분의 평가 지표는 분류기 기반 접근 방식과 FID와 같은 이미지 품질 기반 접근 방법으로 크게 두 가지로 나뉜다. 분류기 기반 평가 지표는 데이터의 미묘한 영향과 차이를 확인할 수 없다. 또한, FID 등의 이미지 품질 기반 평가 지표는 머신 언러닝 전과 후의 생성된 이미지의 모든 차이를 완벽하게 평가할 수 없다. 정리하자면, 대부분의 평가 지표는 머신 언러닝 후, 모델의 성능을 위주로 평가하므로 머신 언러닝 목적에 따른 평가를 하기에 한계가 명확하다(Zhou et al., 2024). CM은 Forget Set을 머신 언러닝 시켰을 때, 머신 언러닝이 Retain Set에 미친 영향을 평가한다. 즉, 머신 언러닝 대상만 잘 잊혀질수록 머신 언러닝의 성능이 높다고 판단한다. 예를 들어, A사의 노트북 클래스를 언러닝한다고 가정하면, 모델은 오직 이 클래스를 잊어야 한다. 언러닝된 클래스와 유사한 B사의 노트북과 같은 유사 클래스는 언러닝 과정의 영향을 받지 않아야 한다. 이는 언러닝이 목표한 대상에 한정되고, 관련 있지만 별개의 데이터 클래스에 의도치 않은 영향을 미치지 않도록 보장한다.

CM은 GAN이 색상을 마스킹한 MNIST 데이터 셋을 학습하고, 임의의 한 클래스의 이미지를 생성할 때에 그 외의 클래스에서 학습된 색상의 영향을 받음을 전제로 한다. 만약 Train Set에서 클래스 0의 이미지를 빨간색으로 마스킹하고 GAN을 학습시켰다고 하자. 이후 GAN이 훈련을 마친 뒤 생성한 이미지에서 빨

간색 픽셀이 나타난다면, 이는 GAN이 클래스 0의 특징을 반영하여 그 픽셀을 생성했다고 보는 것이다. 즉, 생성된 이미지의 빨간색 영역은 원래 마스킹했던 클래스(0번 클래스)의 영향을 받아 만들어졌다고 가정하는 것이다[Figure. 9]. 이에 기반하여 평가 지표를 제안하였다. 초기 아이디어는, Forget Set의 이미지의 흰색 픽셀을 빨간색으로 마스킹한 후 모델을 학습시키고, 이후 머신 언러닝 전후의 이미지를 비교하여 빨간색 픽셀의 개수 변화를 확인함으로써 주변 클래스의 영향을 측정하는 지표로 사용하는 것이었다. 그러나 단일 클래스를 머신 언러닝하여 이를 확인하는 방식은 머신 언러닝 알고리즘을 평가하기에 충분하지 않다고 판단하였다. 따라서 모든 클래스의 색상을 각각 다르게 마스킹하고, Forget Set을 클래스 하나로 설정하고, 10번의 머신 언러닝을 통하여 색상 변화를 각각 확인하는 방법을 제안하였다. 그러나 제안한 방법의 문제점은 모든 클래스의 색상을 다르게 하기 위해 10가지의 R, G, B 조합이 필요하다는 점이다. 빨간색, 초록색, 파란색 이외의 색상은 R, G, B 채널의 값을 고르게 가지므로, 이를 기반으로 픽셀의 색상 변화를 측정하는 것은 무의미할 가능성이 높다. 따라서 대표 클래스 3가지를 선택하여 각각 빨간색, 초록색, 파란색으로 마스킹한 후 평가를 진행하는 방법을 최종적으로 채택하였다.

Forget Set에 포함된 클래스의 경우, 머신 언러닝 이후 생성된 숫자의 영역이 증가하거나 감소할 수 있으므로 픽셀 수의 직접적인 비교는 무의미하다. 따라서 픽셀의 비율 변화에 대한 실험(Experiment 1, 2, 3, 4)과 픽셀의 R, G, B 값을 비교하는 실험(Experiment 5, 6, 7)을 수행하였다.

3.2.2.2 Implementation

본 연구에서는 Train Set에서 클래스 0, 3, 8을 각각 빨간색, 초록색, 파란색에 해당하는 채널 값을 최댓값으로 수정하였다. 즉, 클래스 0을 빨간색으로, 클래스 3을 초록색으로, 클래스 8을 파란색으로 마스킹하였다. 이렇게 수정된 Train Set을 모델에 학습시킨 후, 실험을 진행하였다. Experiments 1, 2, 3, 4에서는 먼저 흰색 픽셀을 제외한 뒤, 각 픽셀의 R, G, B 값 중 하나가 0.9 이상이면 해당 픽셀을 그 색으로 판단하여 카운팅하였다(예: R 채널이 0.9 이상이면 빨간색으로 분류). 반면, Experiments 5, 6, 7에서는 흰색 픽셀과 검은색 픽셀을 제외한 상태에서, 각 픽셀의 R, G, B 값을 모두 합산한 뒤 그 평균값을 산출하여 활용하였다. 이와 같은 방법을 통하여 클래스별 색상 채널 정보가 명확히 반영된 Train Set을 구축하고, 픽셀 단위의 색상 판단 기준(0.9)을 적용함으로써, 클래스 간 색 분포 특징을 보다 직접적으로 분석하고자 하였다.

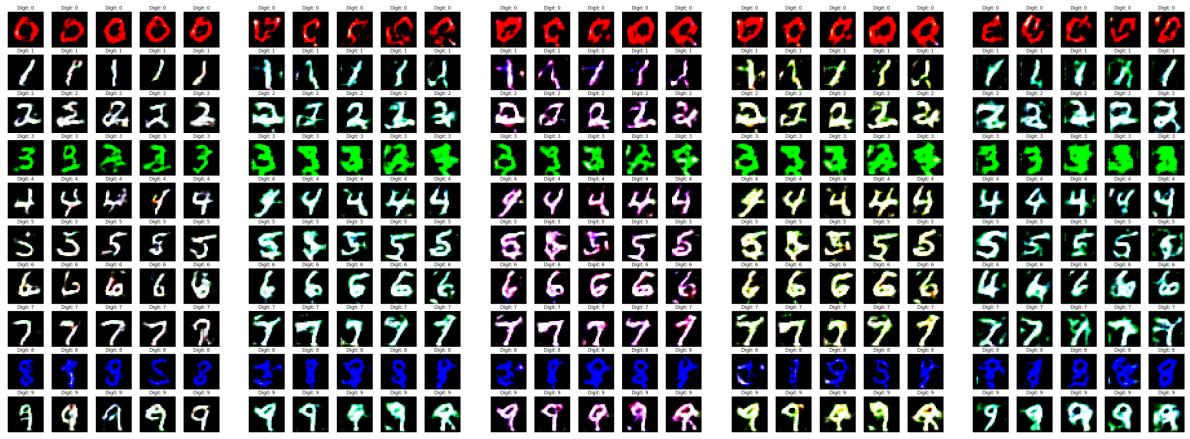
Ⅳ. 실험 결과 (Experimental Results)

4.1 CM Visualization

모든 실험은 사전 학습된 조건부 적대적 생성 신경망을 대상으로 머신 언러닝을 다섯 번 반복 실험하였다. Experiment 1, 2, 3, 4에서는 빨간색, 초록색, 파란색 픽셀의 비율 변화를 조사하였으며, Experiment 5, 6, 7에서는 빨간색, 초록색, 파란색 픽셀의 실제 수치 합의 평균을 분석하였다. 사용된 머신 언러닝 알고리즘의 하이퍼파라미터는 다음과 같다:

Table. 6 Hyperparameters for 4.1

Hyperparameters	Values
Epochs	1,000
Batch Size	100
Learning Rate	0.000005
α	50
$margin$	1



(a) (b) (c) (d) (e)

Figure. 10 CM Visualization Results of Experiment 1, 2, 3, 4 **(a)** Before Unlearning **(b)** After Unlearning Class 0 in Experiment 1 **(c)** After Unlearning Class 3 in Experiment 2 **(d)** After Unlearning Class 8 in Experiment 3 **(e)** After Unlearning Class 0, 3 in Experiment 4

4.1.1 Experiment 1

Experiment 1에서는 클래스 0에 대해 머신 언러닝을 수행하여, 모든 클래스에서 빨간색 픽셀의 증감 비율을 계산하였다. [Figure. 10 (a)]와 [Figure. 10 (b)]를 시각적으로 비교한 결과, 머신 언러닝 이후 모든 숫자가 두꺼워졌으며, 특히 클래스 0에서는 원이 연속적이지 않게 변형되었음을 확인할 수 있었다. 또한 [Figure. 11 (a)]에서 알 수 있듯이, 클래스 0을 제외한 모든 클래스에서 빨간색 픽셀의 비율이 현저히 감소하였다. 반면 클래스 0에서는 오차 막대가 상대적으로 크고 언러닝 전후의 오차 막대가 겹치기 때문에, 빨간색 픽셀의 증가 또는 감소를 명확히 판단하기 어렵다.

4.1.2 Experiment 2

Experiment 2에서는 클래스 3에 대해 머신 언러닝을 수행하여, 모든 클래스에서 초록색 픽셀의 증감 비율을 계산하였다. [Figure. 10 (a)]와 [Figure. 10 (c)]를 시각적으로 비교한 결과, 머신 언러닝 이후 모든 숫자가 두꺼워졌으며, 특히 클래스 3에서는 숫자의 형태가 다소 변형되었음을 확인할 수 있었다. 또한 [Figure. 11 (b)]에서 알 수 있듯이, 클래스 3을 제외한 모든 클래스에서 초록색 픽셀의 비율이 현저히 감소하였다. 반면 클래스 3에서 Experiment 1과 달리 오차막대가 겹치지 않으므로, 초록색 픽셀 비율이 증가하였다고 판단된다.

4.1.3 Experiment 3

Experiment 3에서는 클래스 8에 대해 머신 언러닝을 수행하여, 모든 클래스에서 파란색 픽셀의 증감 비율을 계산하였다. [Figure. 10 (a)]와 [Figure. 10 (d)]를 시각적으로 비교한 결과, 머신 언러닝 이후 모든 숫자가 두꺼워졌으며, 특히 클래스 8에서는 숫자 '8'의 위아래 원이 비대칭적으로 나타남을 확인할 수 있었다. 또한 [Figure. 11 (c)]에서 알 수 있듯이, 클래스 8을 제외한 모든 클래스에서 파란색 픽셀의 비율이 현저히 감소하였다. 반면 클래스 8에서 Experiment 1 달리 오차막대가 겹치지 않으므로, 파란색 픽셀 비율이 증가하였다고 판단된다.

4.1.4 Experiment 4

Experiment 4에서는 클래스 0, 3에 대해 머신 언러닝을 수행하여, 모든 클래스에서 빨간색, 초록색, 파란색 픽셀의 증감 비율을 계산하였다. [Figure. 11 (d)], [Figure. 11 (e)], [Figure. 11 (f)]를 비교해보면, 빨간색 픽셀의 비율만 줄어든 반면, 초록색과 파란색 픽셀의 비율은 증가했음을 확인할 수 있다. [Figure. 10 (e)]을 [Figure. 10 (b)], [Figure. 10 (c)], [Figure. 10 (d)]와 시각적으로 비교해보면, [Figure. 10 (b)]와 가장 유사한 것을 확인할 수 있다.

Table 7. RED Pixel Ratio by Class

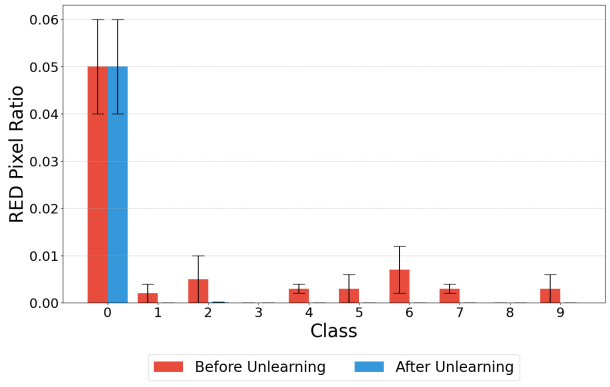
	Class	0	1	2	3	4	5	6	7	8	9
RED Pixel Ratio	Before Unlearning	0.0500	0.0020	0.0050	0	0.0030	0.0030	0.0070	0.0030	0	0.0030
	Before Unlearning Uncertainty	0.0100	0.0020	0.0050	0	0.0010	0.0030	0.0050	0.0010	0	0.0030
	After Unlearning	0.0500	0	0.0001	0	0	0	0	0	0	0
	After Unlearning Uncertainty	0.0100	0	0.0001	0	0	0	0	0	0	0
	Difference	0.0040	0.0020	0.0050	0	0.0030	0.0030	0.0070	0.0030	0	0.0030

Table 8. GREEN Pixel Ratio by Class

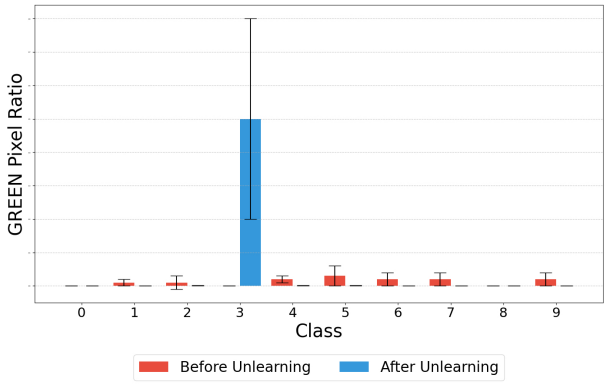
	Class	0	1	2	3	4	5	6	7	8	9
GREEN Pixel Ratio	Before Unlearning	0	0.0010	0.0010	0	0.0020	0.0030	0.0020	0.0020	0	0.0020
	Before Unlearning Uncertainty	0	0.0010	0.0020	0	0.0010	0.0030	0.0020	0.0020	0	0.0020
	After Unlearning	0	0	0.0001	0.0500	0.0001	0.0001	0	0	0	0
	After Unlearning Uncertainty	0	0	0.0001	0.0300	0.0001	0.0001	0	0	0	0
	Difference	0	0.0010	0.0010	-0.0200	0.0020	0.0030	0.0020	0.0020	0	0.0020

Table 9. BLUE Pixel Ratio by Class

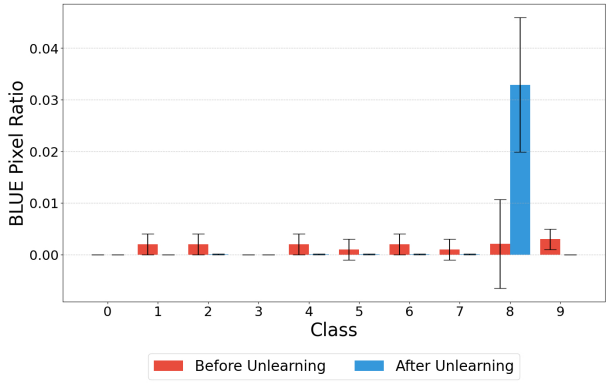
	Class	0	1	2	3	4	5	6	7	8	9
BLUE Pixel Ratio	Before Unlearning	0	0.0020	0.0020	0	0.0020	0.0010	0.0020	0.0010	0.0221	0.0030
	Before Unlearning Uncertainty	0	0.0020	0.0020	0	0.0020	0.0020	0.0020	0.0020	0.0086	0.0020
	After Unlearning	0	0	0.0001	0	0.0001	0.0001	0	0.0001	0.0329	0
	After Unlearning Uncertainty	0	0	0.0001	0	0.0001	0.0001	0	0.0001	0.0130	0
	Difference	0	0.0020	0.0020	0	0.0020	0.0010	0.0020	0.0010	-0.0108	0.0030



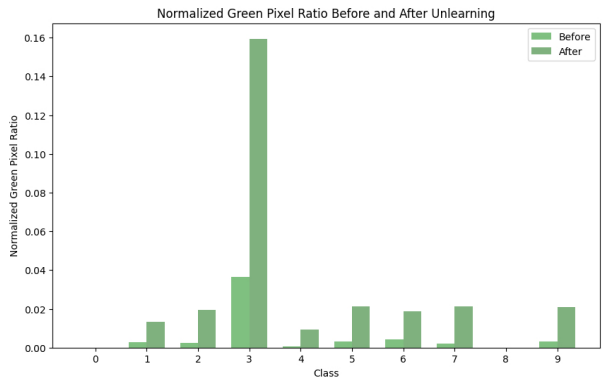
(a)



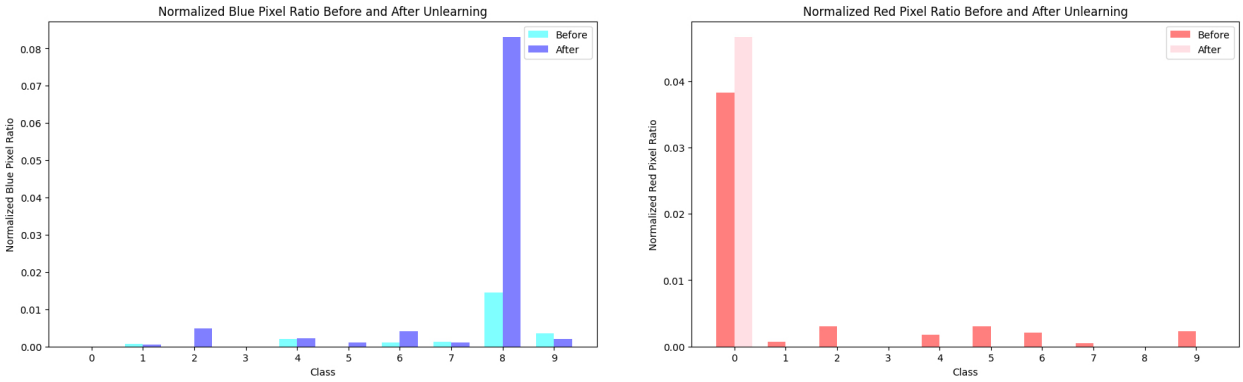
(b)



(c)



(d)

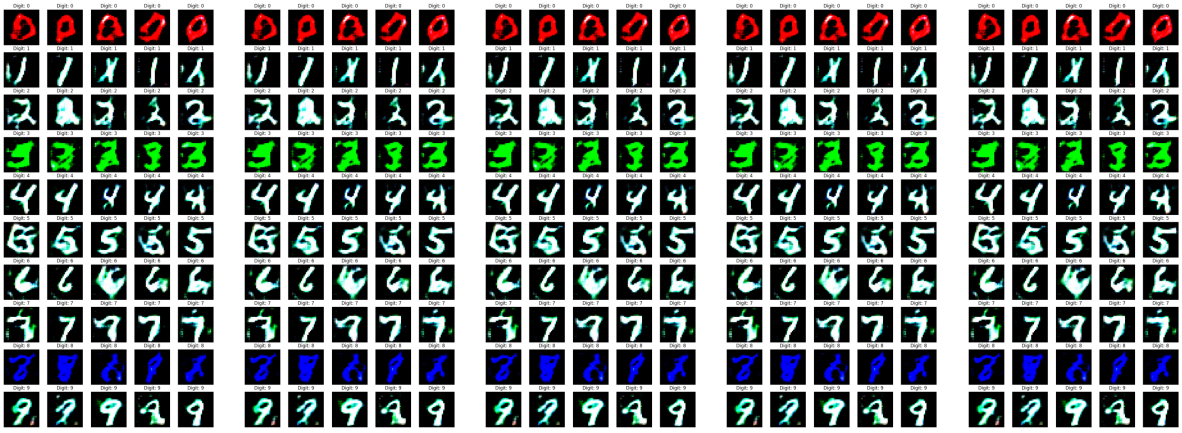


(e) (f)

Figure. 11 RGB Pixel Ratio Bars for Each Experiment **(a)** RED Pixel Ratio Bar in Experiment 1 **(b)** GREEN Pixel Ratio Bar in Experiment 2 **(c)** BLUE Pixel Ratio Bar in Experiment 3 **(d)** GREEN Pixel Ratio Bar in Experiment 4 **(e)** BLUE Pixel Ratio Bar in Experiment 4 **(f)** RED Pixel Ratio Bar in Experiment 4

4.1.5 Experiment 5

Experiment 5에서는 클래스 0에 대해 머신 언러닝을 수행하여, 모든 클래스에서 각 픽셀의 R 값을 합산하였다. [Figure. 13]에서 알 수 있듯이, 클래스 0, 3, 8을 제외한 모든 클래스에서 픽셀의 R 값의 합이 현저하게 감소하였다. 반면 클래스 0에서는 증가하였으며, 클래스 3과 8의 경우 전후 값 모두 작아 유의미한 의미를 제공하지 않는다.



(a) (b) (c) (d) (e)

Figure. 12 CM Visualization Results of Experiment 5 **(a)** Trial 1 **(b)** Trial 2 **(c)** Trial 3 **(d)** Trial 4 **(e)** Trial 5

Table 10. RED Pixel Ratio by Class

	Class	0	1	2	3	4	5	6	7	8	9
RED Pixel Ratio	Before Unlearning	0.7486	0.4611	0.4965	0.0023	0.5019	0.5069	0.5385	0.5270	0.0041	0.4996
	Before Unlearning Uncertainty	0	0	0	0	0	0	0	0	0	0
	After Unlearning	0.8202	0.2409	0.2802	0.0069	0.2599	0.2850	0.2675	0.2335	0.0019	0.2390
	After Unlearning Uncertainty	0.0006	0.0011	0	0	0.0003	0.0003	0.0002	0.0002	0	0.0003
	Difference	-0.0716	0.2201	0.2163	-0.0047	0.2420	0.2220	0.2709	0.2935	0.0021	0.2606

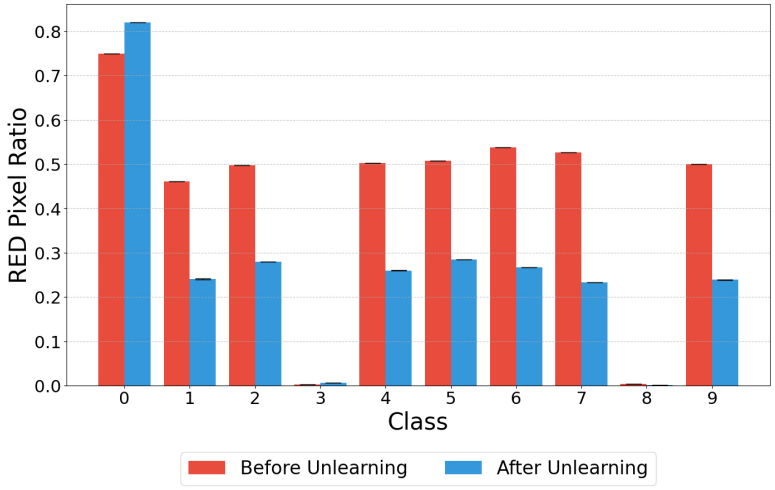
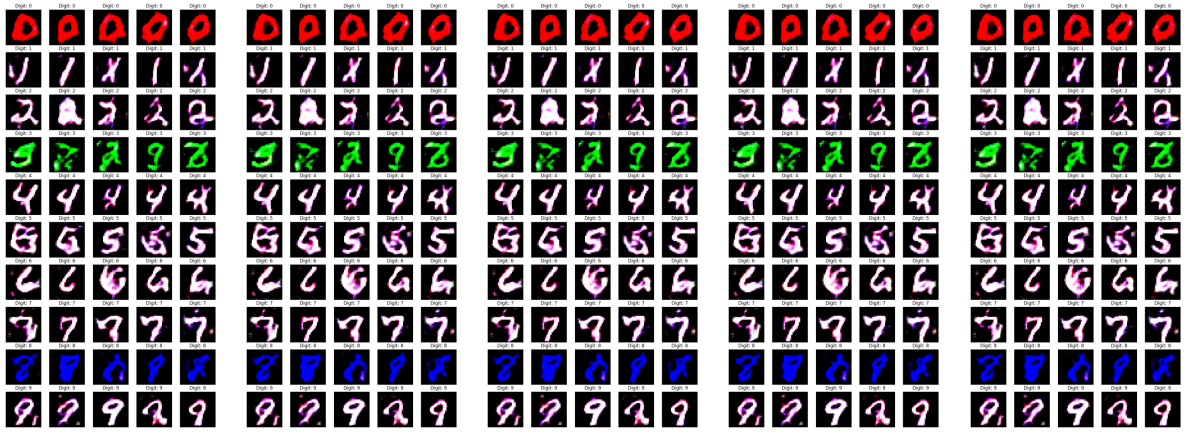


Figure. 13 RED Pixel Ratio Bar

4.1.6 Experiment 6

Experiment 6에서는 클래스 3에 대해 머신 언러닝을 수행하여, 모든 클래스에서 각 픽셀의 G 값을 합산하였다. [Figure. 15]에서 알 수 있듯이, 클래스 0, 3, 8을 제외한 모든 클래스에서 픽셀의 G 값의 합이 현저하게 감소하였다. 반면 클래스 3에서는 미미하게 감소하였으며, 클래스 0과 8의 경우 전후 값 모두 작아 유의미한 의미를 제공하지 않는다.



(a) (b) (c) (d) (e)

Figure. 14 CM Visualization Results of Experiment 6 **(a)** Trial 1 **(b)** Trial 2 **(c)** Trial 3 **(d)** Trial 4 **(e)** Trial 5

Table 11. GREEN Pixel Ratio by Class

	Class	0	1	2	3	4	5	6	7	8	9
GREEN Pixel Ratio	Before Unlearning	0.0038	0.5049	0.4784	0.8237	0.4251	0.4336	0.4427	0.4215	0.0022	0.4269
	Before Unlearning Uncertainty	0	0	0	0	0	0	0	0	0	0
	After Unlearning	0.0046	0.2186	0.2583	0.8155	0.2547	0.2849	0.2786	0.2564	0.0026	0.2367
	After Unlearning Uncertainty	0	0	0.0004	0.0004	0.0008	0.0007	0.0004	0	0	0.0001
	Difference	-0.0008	0.2863	0.2201	0.0083	0.1704	0.1487	0.1641	0.1651	-0.0005	0.1902

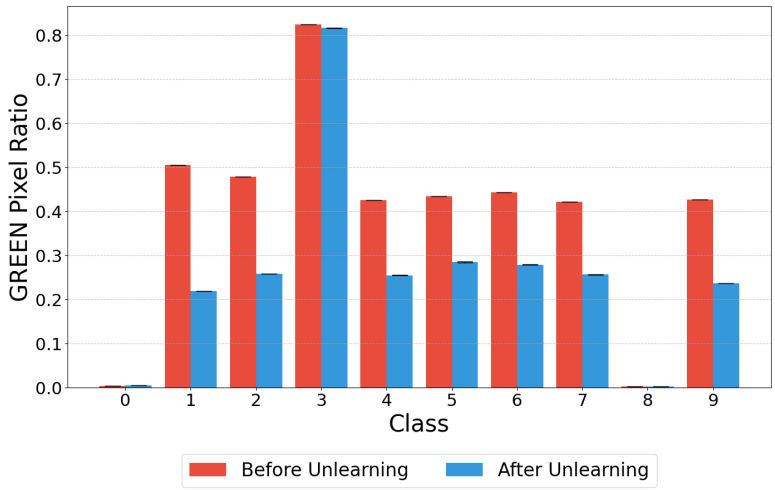
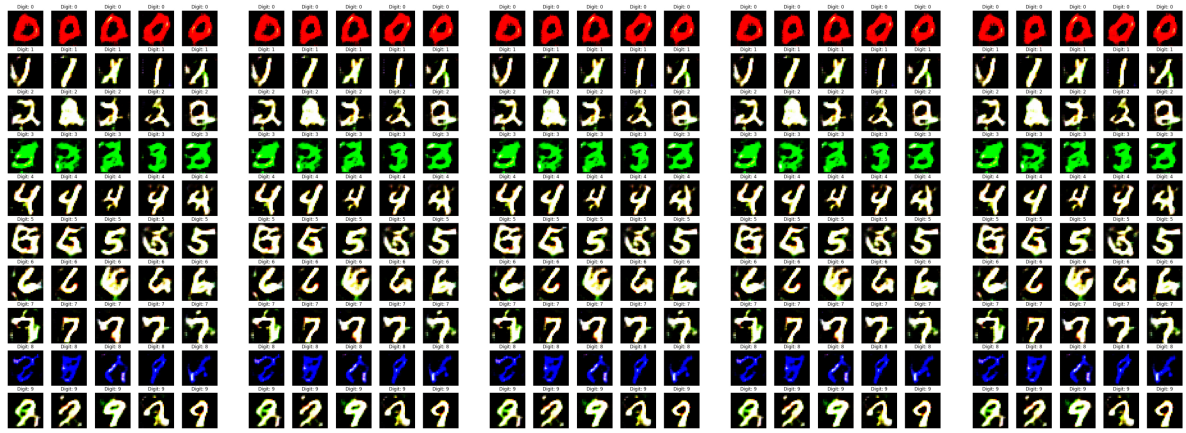


Figure. 15 GREEN Pixel Ratio Bar

4.1.7 Experiment 7

Experiment 7에서는 클래스 8에 대해 머신 언러닝을 수행하여, 모든 클래스에서 각 픽셀의 B 값을 합산하였다. [Figure. 17]에서 알 수 있듯이, 클래스 0, 3, 8을 제외한 모든 클래스에서 픽셀의 B 값의 합이 현저하게 감소하였다. 반면 클래스 8에서는 증가하였으며, 클래스 0과 3의 경우 전후 값 모두 작아 유의미한 의미를 제공하지 않는다.



(a) (b) (c) (d) (e)

Figure. 16 CM Visualization Reesults of Experiment 7 **(a)** Trial 1 **(b)** Trial 2 **(c)** Trial 3 **(d)** Trial 4 **(e)** Trial 5

Table 12. BLUE Pixel Ratio by Class

	Class	0	1	2	3	4	5	6	7	8	9
BLUE Pixel Ratio	Before Unlearning	0.0041	0.4155	0.4003	0.0043	0.4223	0.4169	0.4976	0.3946	0.7478	0.5126
	Before Unlearning Uncertainty	0	0	0	0	0	0	0	0	0	0
	After Unlearning	0.0056	0.2616	0.2972	0.0141	0.2595	0.2857	0.2646	0.2556	0.7516	0.2580
	After Unlearning Uncertainty	0	0.0244	0.0005	0	0	0	0.0004	0	0.0004	0.0006
	Difference	-0.0015	0.1539	0.1032	-0.0098	0.1628	0.1313	0.2330	0.1390	-0.0037	0.2546

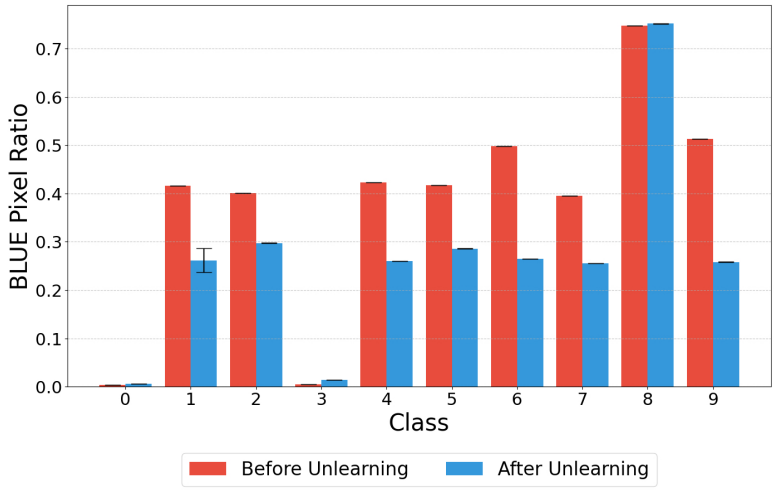


Figure. 17 BLUE Pixel Ratio Bar

4.2 CM Results

Experiment 1, 2, 3을 통해, 각 클래스를 빨간색, 초록색, 파란색으로 마스킹하여 머신 언러닝할 때 나머지 클래스에서 머신 언러닝된 클래스의 색을 가진 픽셀 비율이 감소함을 확인할 수 있었다. 이를 바탕으로 Experiment 4를 분석해보면, 클래스 0과 3을 머신 언러닝한 후 빨간색 픽셀의 비율만 감소한 것을 볼 때, 머신 언러닝 과정에서 클래스 0이 클래스 3보다 나머지 클래스에 더 큰 영향을 미친다는 것을 알 수 있

다. 하지만 머신 언러닝 이후 각 색의 비율이 대부분 0이 되어 유의미한 분석을 하기에 한계가 있다. 이를 개선하여 실험한 것이 Experiment 5, 6, 7이다. 이에 대하여 히트맵으로 나타내면 [Figure. 18]과 같다.

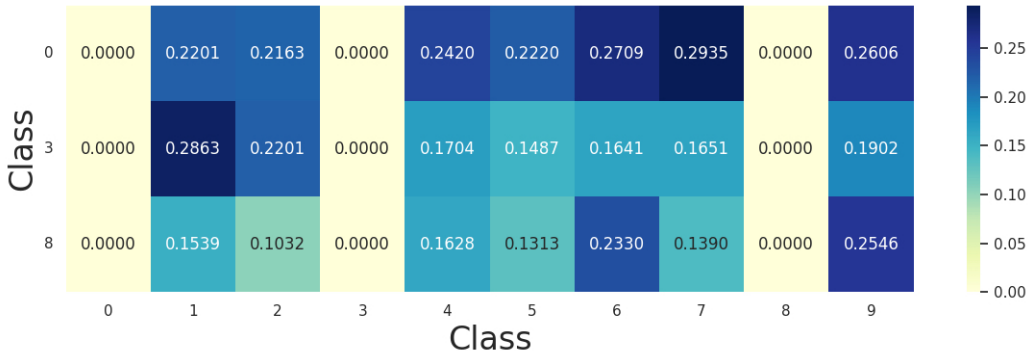


Figure. 18 Colored Metric Matrix (CMM)

히트맵에 사용된 데이터는 [Table. 10], [Table. 11], [Table. 12]의 Difference 값이다. 클래스 0, 3, 8에 대해서는 유의미한 분석을 하지 못하므로 0으로 처리하였다. [Figure. 18]의 첫 번째 행의 요소는, 클래스 0을 머신 언러닝 시켰을 때 나머지 클래스가 영향을 받는 정도를 나타낸다. 즉, 클래스 0은 클래스 7에, 클래스 3은 클래스 1에, 클래스 8은 클래스 9에 머신 언러닝시 큰 영향을 주는 것으로 해석할 수 있다. $EFF_{i,\ j}$를 클래스 i가 머신 언러닝 수행시 클래스 j에 미치는 영향이라고 정의하자. EFF_i를 클래스 i가 머신 언러닝 수행시 나머지 클래스에 미치는 영향의 평균(%)라고 정의하자. 이 값이 0%에 가까울수록 머신 언러닝 성능이 클래스 i에 대하여 높다고 평가한다.

$$EFF_0 = \frac{\sum_{i\ =\ 0,\ i\ \neq\ 0,\ 3,\ 8}^{9} EFF_{0,\ i}}{7} \times 100\% = 21.56\% \tag{8}$$

$$EFF_3 = \frac{\sum_{i\ =\ 0,\ i\ \neq\ 0,\ 3,\ 8}^{9} EFF_{3,\ i}}{7} \times 100\% = 19.21\% \tag{9}$$

$$EFF_8 = \frac{\sum_{i\ =\ 0,\ i\ \neq\ 0,\ 3,\ 8}^{9} EFF_{8,\ i}}{7} \times 100\% = 16.83\% \tag{10}$$

본 실험에서 클래스 8, 3, 0 순으로 머신 언러닝 성능이 높음을 알 수 있다.

4.3 Comparison of CM and FID

FID는 생성된 이미지가 원본 이미지와 얼마나 유사한지 측정하는 지표로, 머신 언러닝의 효과를 확인할 수 있지만 어느 특정 클래스가 얼마나 영향을 받았는지에 대한 정보를 제공하지 않는다. 반면, CM을 이용하면 특정 클래스가 머신 언러닝 과정에서 어느 정도 영향을 받았는지 구체적으로 파악할 수 있다. CM은 클래스별 픽셀 변화 비율을 분석하여 어떤 클래스가 가장 큰 변화를 겪었는지 시각적으로 확인할 수 있게 해주므로, 특정 클래스의 변화 정도를 측정하는 데 유리하다. 또한, FID의 결과와 CM의 결과가 일치하지 않기 때문에 본 지표는 유의미하다.

Ⅴ. 고찰 (Discussion)

5.1 Machine Unlearning Algorithm

본 연구에서는 적대적 생성 신경망에 머신 언러닝을 적용시키기 위해, 머신 언러닝의 가장 기초적인 기법인 파인튜닝을 사용하였다. 머신 언러닝의 평가 지표를 제안하는 것이 목적이므로 머신 언러닝이 제대로 이루어져야 한다. 특히, 파인튜닝은 하이퍼파라미터 값에 영향을 많이 받으므로 적절한 값을 찾는 과정이 중요했다.

5.2 Experimental Environment

본 연구에서 Google Colaboratory Pro를 사용하였다. 해당 클라우드 컴퓨팅 시스템은 컴퓨팅 단위의 제한이 있다. 따라서 학습 시간이 긴 확산 모델을 사용한 선행연구를 재현하기 어려움이 있었다(Li et al., 2024; Heng & Soh, 2023). 본 환경에서 적대적 생성 신경망은 제한 시간 안에 충분한 학습이 가능하여 사용하였다. 또한 상대적으로 작은 데이터 셋인 MNIST 데이터 셋을 사용하였다.

5.3 Generative Adversarial Networks

적대적 생성 신경망의 안장점을 찾기 위해서 다양한 하이퍼파라미터 조합에서 실험하며 생성되는 이미지 품질을 높였다. 일반적인 적대적 생성 신경망은 클래스를 무작위로 생성하므로 평가 지표 설계에 적합하지 않다. 따라서 조건부 적대적 생성 신경망을 사용하였다.

5.4 Limitations

본 연구의 한계점은 다음과 같다. 첫째, 지표의 수학적 검증이 필요하다. 예를 들어, 클래스 0에 빨간색을 입힐 때, 나머지 클래스에서 발견되는 빨간색 픽셀들이 모두 클래스 0의 영향이라고 가정한다. 그러나 이는 적대적 생성 신경망이 학습 과정에서 색

상을 어떻게 처리하는지에 대한 수학적 검증이 이루어지지 않은 상태에서 진행되었기에 논리적 한계를 가진다. 둘째, EFF_i 값에 대한 명확한 기준이 없다. 특정 값을 초과할 때 머신 언러닝 성능이 좋다고 판단할 수 있는 기준이 부족하다. 이를 해결하기 위해 다양한 실험 데이터에서 EFF_i 값을 통계적으로 분석하고 적절한 임계값을 도출하는 추가 연구가 필요하다. 셋째, 하나의 모델에서만 실험을 진행하였다. 확산 모델, VQ-GAN, Image-to-Image Generator, Variational Auto Encoder (VAE) 등의 모델에서도 유효한지 검증이 필요하다. 넷째, 컬러 이미지에서는 제안된 평가 지표가 변형해야 한다. MNIST 데이터 셋은 흑백 이미지이므로 R, G, B 채널을 적용할 수 있었고, 이를 통해 평가 지표의 값들을 직관적으로 해석할 수 있었다. 그러나 CIFAR10, ImageNet 처럼 컬러 이미지로 구성된 데이터 셋에는 평가 지표를 도입하기 어렵다. 또한, 학습 시 데이터의 R, G, B 값에 영향을 받기 때문에 평가 지표의 값을 직관적으로 해석할 수 없다. 흑백 이미지의 경우 학습 과정에서 R, G, B 채널을 사용하지 않기 때문에 표식 역할로 사용할 수 있었다. 그러나 컬러 이미지의 경우 이미 R, G, B 채널이 사용되고 있어 표식 역할을 하기에 적합하지 않다.

Ⅵ. 결론 (Conclusion)

본 연구는 기존 생성형 모델의 머신 언러닝 연구에서 사용되던 평가지표의 근본적인 문제점을 해결하고자 CM을 제안하였다. 제안된 평가지표는 R, G, B 채널을 활용하여 생성형 모델에 적용된 머신 언러닝 성능을 평가한다. 이 지표를 활용하면 머신 언러닝 수행 시 Forget Set의 클래스가 Retain Set의 클래스에 미치는 영향을 정량적으로 파악할 수 있다. 이를 통해 생성형 모델의 적용된 머신 언러닝을 보다 직관적으로 분석할 수 있는 방법을 제안하였다. 그러나, 모델에 대한 추가적인 수학적 검증이 필요하며, 평가지표의 범용성을 높이기 위해 다양한 모델과 데이터 셋에서 실험을 통해 그 유효성을 입증 해야한다. 끝으로, 본 연구는 생성형 모델에 적용된 머신 언러닝 성능 평가에 관한 새로운 관점과 방법을 제시하였으며, 향후 개선 연구의 기초 자료로 활용될 수 있을 것으로 기대된다.

References

Brophy, J., & Lowd, D. (2021, July 1). Machine unlearning for random forests. PMLR. https://proceedings.mlr.press/v139/brophy21a.html

Foster, J., Fogarty, K., Schoepf, S., Dugue, Z., Öztireli, C., & Brintrup, A. (2024, February 2). An information theoretic approach to machine unlearning. arXiv.org. https://arxiv.org/abs/2402.01401

Heng, A., & Soh, H. (2023, May 17). Selective amnesia: a continual learning approach to forgetting in deep generative models. arXiv.org. https://arxiv.org/abs/2305.10120

Ken Ziyu Liu. (2024). Machine Unlearning in 2024. https://ai.stanford.edu/~kzliu/blog/unlearning

Li, G., Hsu, H., Chen, C., & Marculescu, R. (2024, February 1). Machine unlearning for Image-to-Image generative models. arXiv.org. https://arxiv.org/abs/2402.00351

Mirza, M., & Osindero, S. (2014, November 6). Conditional generative adversarial Nets. arXiv.org. https://arxiv.org/abs/1411.1784

Neel, S., Roth, A., & Sharifi-Malvajerdi, S. (2021, March 1). Descent-to-Delete: Gradient-Based Methods for Machine Unlearning. PMLR. https://proceedings.mlr.press/v132/neel21a.html

NeurIPS 2023 Machine Unlearning Challenge. (n.d.). NEURIPS 2023 Machine Unlearning Challenge. https://unlearning-challenge.github.io/

Sun, H., Zhu, T., Chang, W., & Zhou, W. (2023, August 19). Generative Adversarial Networks Unlearning. arXiv.org. https://arxiv.org/abs/2308.09881

Zhou, S., Wang, L., Ye, J., Wu, Y., & Chang, H. (2024, August 1). On the Limitations and Prospects of Machine Unlearning for Generative AI. arXiv.org. https://arxiv.org/abs/2408.00376

Abstract

As artificial intelligence advances, potential risks and ethical issues are emerging, and in particular, interest in the security of user data and the right to be forgotten is increasing. Machine unlearning is gaining attention as a method to delete unwanted data with minimal cost. Generative models generate various types of data, but the quality evaluation of such data mostly relies on subjective judgment. The current evaluation of machine unlearning is the same. Generative model evaluation indices such as CID index, IS (Inception Score), and FID (Fréchet Inception Distance) have limitations in evaluating the quality of machine unlearning. Therefore, this study proposed a Colored Metric (CM) that evaluates the influence of a class to be machine unlearned on surrounding classes, and verified its usefulness by comparing it with FID.

Keywords : Machine Unlearning, Generative Adversarial Networks(GAN), Evaluation Metric , Fine Tuning, Privacy

* A thesis submitted to the Sejong Academy of Science and Arts in partial fulfillment of the requirements for the graduation. The study was conducted in accordance with Code of Research Ethics. Declaration of Ethical Conduct in Research: I, as a graduate student of **SASA**, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.