Privacy and Uncertainty Aware Learning

Machine Unlearning: deletion of data or concepts from trained models

Subhodip Panda (20786)

Department of Electrical Communication Engineering

October 28, 2024

Table of Contents

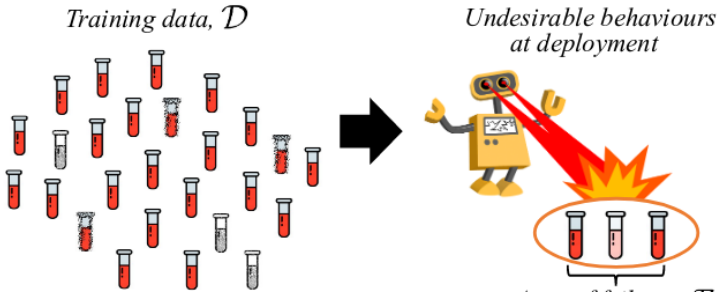
Problem Statement of Machine Unlearning

Problem Statement of Machine Unlearning

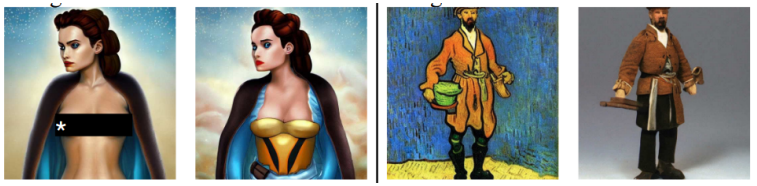
- Interesting Questions?
  1. How can we induce the trained model to selectively discard or diminish the acquired knowledge associated with a specific set of data points in the training dataset?
  2. Is it possible to induce the model to selectively erase or reduce its understanding of specific higher-level concepts acquired from a particular set of data points?

Problem Statement of Machine Unlearning

- Interesting Questions?
  1. How can we induce the trained model to selectively discard or diminish the acquired knowledge associated with a specific set of data points in the training dataset?
  2. Is it possible to induce the model to selectively erase or reduce its understanding of specific higher-level concepts acquired from a particular set of data points?
- Why do we want this?
  - data privacy and protection regulations, such as the European Union's GDPR [13] and the California Consumer Privacy Act (CCPA) [7]



(a) Removing Noisy Data



(b) Removing Undesired Features

Machine Unlearning

Machine Unlearning

- What is Machine Unlearning?
  - machine unlearning refers to the task of forgetting the learned information [11, 10, 17, 1, 4, 6, 3, 5], or erasing the influence [15, 9, 8, 14, 16, 2] of specific data subset of the training dataset from a learned model in response to a user request.

Machine Unlearning

- What is Machine Unlearning?
  - machine unlearning refers to the task of forgetting the learned information [11, 10, 17, 1, 4, 6, 3, 5], or erasing the influence [15, 9, 8, 14, 16, 2] of specific data subset of the training dataset from a learned model in response to a user request.
- What are the Mathematical Definitions?
  - Z as an example space, i.e., a space of datasets.
  - Given a dataset D, we want to obtain a machine-learning model from a hypothesis space H. The process of training a model on D by a learning algorithm, denoted by a function $A : Z \to H$, with the trained model denoted as $A(D)$.
  - To support forgetting requests, an unlearning mechanism, denoted by a function U, that takes as input a training dataset $D \in Z$, a forget set $D_f \subset D$ (data to forget), and a model $A(D)$. It returns a sanitized (or unlearned) model $U(D, D_f, A(D)) \in H$.
  - The unlearned model is expected to be the same or similar to a retrained model $A(D \setminus D_f)$

Machine Unlearning

Exact unlearning

Given a learning algorithm $A(\cdot)$, we say the process $U(\cdot)$ is an exact unlearning process if for all $T \subseteq H$, $D \in Z$, $D_f \subset D$, it holds that

$$\Pr(A(D \setminus D_f) \in T) = \Pr(U(D, D_f, A(D)) \in T)$$

Machine Unlearning

Exact unlearning

Given a learning algorithm $A(\cdot)$, we say the process $U(\cdot)$ is an exact unlearning process if for all $T \subseteq H$, $D \in Z$, $D_f \subset D$, it holds that

$$\Pr(A(D \setminus D_f) \in T) = \Pr(U(D, D_f, A(D)) \in T)$$

ε, δ-Approximate Unlearning

Given $\varepsilon, \delta > 0$, an unlearning mechanism U performs ε, δ-certified removal for a learning algorithm A if for all $T \subseteq H$, $D \in Z$, $z \in D$, it holds that

$$\Pr(U(D, z, A(D)) \in T) \leq e^{\varepsilon} \Pr(A(D \setminus z) \in T) + \delta$$

and

$$\Pr(A(D \setminus z) \in T) \leq e^{\varepsilon} \Pr(U(D, z, A(D)) \in T) + \delta$$

Research Gap and Contributions

Research Gap and Contributions

- Recent Works
  - Initially machine unlearning algorithms have primarily been utilized for structured problems or relatively small-scale tasks, often focused on linear and logistic regression models.
  - Currently these algorithms are largely applied to deep classification networks in supervised learning settings.

Research Gap and Contributions

- Recent Works
  - Initially machine unlearning algorithms have primarily been utilized for structured problems or relatively small-scale tasks, often focused on linear and logistic regression models.
  - Currently these algorithms are largely applied to deep classification networks in supervised learning settings.
- Research Gaps
  - However, their applicability and effectiveness in the context of unsupervised models, particularly state-of-the-art generative models, remain largely unexplored.

Research Gap and Contributions

- Recent Works
  - Initially machine unlearning algorithms have primarily been utilized for structured problems or relatively small-scale tasks, often focused on linear and logistic regression models.
  - Currently these algorithms are largely applied to deep classification networks in supervised learning settings.
- Research Gaps
  - However, their applicability and effectiveness in the context of unsupervised models, particularly state-of-the-art generative models, remain largely unexplored.
- Contributions
  - **Q.1:** *How can we devise effective mechanisms to prevent a pre-trained generative model, whose underlying dataset is unknown (yet its parameters are accessible), from producing outputs containing undesired features?*
  - **Q.2:** *In many practical applications, generative models are employed as black-box systems, making it impossible to access their internal workings. Under such circumstances, what does it mean to unlearn from these black-box generative models?*

Proposed Methodology for Q.1

Notations

1. G_{θ_G} of a pre-trained GAN with parameters θ_G.
2. trained using a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$, where $\mathbf{x}_i \overset{\text{iid}}{\sim} p_X(x)$.
3. the user is presented with n samples $\mathcal{S} = \{\mathbf{y}_i\}_{i=1}^{n}$, where $\mathbf{y}_i$ represents the generated samples from the pre-trained GAN.
4. The user identifies a subset of these samples $\mathcal{S}_n = \{\mathbf{y}_i\}_{i \in s_n}$ as negative samples, which contain undesired features. The remaining samples, denoted as $\mathcal{S}_p = \{\mathbf{y}_i\}_{i \in s_p}$
5. s_p and s_n are index sets such that $s_p \cup s_n = \{1, 2, \ldots, n\}$ and $s_p \cap s_n = \emptyset$.

Joint work with Piyush!

Proposed Method for Q1

- In this question setting, the training dataset used for training the GAN is undisclosed but GAN so retraining is impossible.In our proposed method [12], we adopt a two-stage approach for unlearning the undesired features.

Proposed Method for Q1

- In this question setting, the training dataset used for training the GAN is undisclosed but GAN so retraining is impossible.In our proposed method [12], we adopt a two-stage approach for unlearning the undesired features.

- **Stage-1:** we adapt the pre-trained generator G_{θ_G} on the negative samples. This step gives us the parameters θ_N such that G_{θ_N} generates only negative samples. Hence, the optimal parameter θ_N for the adapted GAN can be obtained by solving the following optimization problem:

$$(\theta_N, \phi_N) = \arg\min_{\theta} \max_{\phi} (\mathcal{L}_{adv} + \gamma \mathcal{L}_{adapt}) \tag{1}$$

$$\mathcal{L}_{adv} = \mathbb{E}_{x \sim p_{\mathcal{S}_n}(x)} [\log D_\phi(x)] + \mathbb{E}_{z \sim p_Z(z)} [\log(1 - D_\phi(G_\theta(z)))] \tag{2}$$

$$\mathcal{L}_{adapt} = \lambda \sum_i F_i(\theta_i - \theta_{G,i}) \tag{3}$$

$$F = \mathbb{E}\left[-\frac{\partial^2}{\partial \theta_G^2} \mathcal{L}(\mathcal{S}_n \mid \theta_G)\right] \tag{4}$$

Proposed Method for Q.1

- **Stage-2**, we actually unlearn the undesired feature by training the original generator G_{θ_G} on positive samples using the usual adversarial loss while adding an additional regularization term that makes sure that the learned parameter is far from θ_N. We call this regularization term *repulsion* loss as it repels the learned parameters from θ_N. Mathematically, we obtain the parameters after unlearning θ_P, ϕ_P by solving the following optimization problem:

$$\theta_P, \phi_P = \arg\min_{\theta} \max_{\phi} \mathcal{L}_{adv}^{'} + \gamma \mathcal{L}_{repulsion} \tag{5}$$

$$\mathcal{L}_{adv}^{'} = \mathop{\mathbb{E}}_{x \sim p_{\mathcal{S}_p}(x)} [\log D_\phi(x)] + \mathop{\mathbb{E}}_{z \sim p_Z(z)} [\log(1 - D_\phi(G_\theta(z)))] \tag{6}$$

Proposed Method for Q.1

- **Stage-2**, we actually unlearn the undesired feature by training the original generator G_{θ_G} on positive samples using the usual adversarial loss while adding an additional regularization term that makes sure that the learned parameter is far from θ_N. We call this regularization term *repulsion* loss as it repels the learned parameters from θ_N. Mathematically, we obtain the parameters after unlearning θ_P, ϕ_P by solving the following optimization problem:

$$\theta_P, \phi_P = \arg\min_{\theta} \max_{\phi} \mathcal{L}_{adv}^{'} + \gamma \mathcal{L}_{repulsion} \tag{5}$$

$$\mathcal{L}_{adv}^{'} = \mathop{\mathbb{E}}_{x \sim p_{\mathcal{S}_p}(x)} [\log D_\phi(x)] + \mathop{\mathbb{E}}_{z \sim p_Z(z)} [\log(1 - D_\phi(G_\theta(z)))] \tag{6}$$

- Particularly, we explore three choices for repulsion loss:

$$\mathcal{L}_{repulsion}^{\text{IL2}} = \frac{1}{||\theta - \theta_N||_2^2}, \quad \mathcal{L}_{repulsion}^{\text{NL2}} = -||\theta - \theta_N||_2^2,$$
$$\mathcal{L}_{repulsion}^{\text{EI2}} = \exp(-\alpha||\theta - \theta_N||_2^2) \tag{7}$$

Proposed Method for Q.2

- More stringent: the pertinent model parameters, architectures, and the underlying dataset remain entirely inaccessible.
- Due to the complete lack of access to model parameters and architectures, conventional unlearning procedures or the prospect of perturbing the model parameters becomes an unattainable endeavor.

Proposed Method for Q.2

- More stringent: the pertinent model parameters, architectures, and the underlying dataset remain entirely inaccessible.
- Due to the complete lack of access to model parameters and architectures, conventional unlearning procedures or the prospect of perturbing the model parameters becomes an unattainable endeavor.

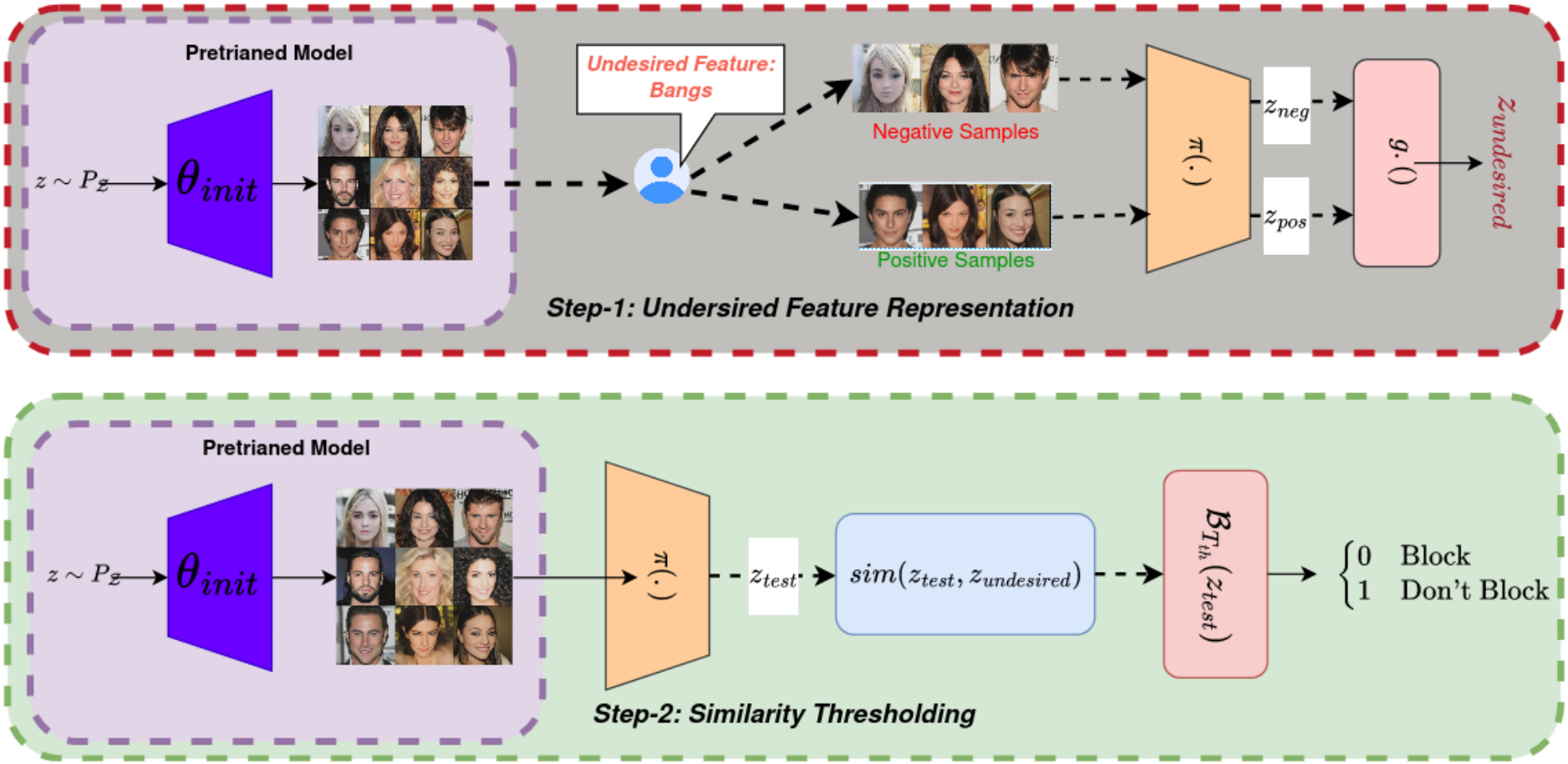
(ϵ, δ) Approximate Weak Unlearning

Given a retrained generative model $G_{\theta^r}(.)$, we say the model $G_{\theta^u}(.)$ is an (ϵ, δ) approximate weak unlearned model for a given $\epsilon, \delta \geq 0$ iff $\forall \mathcal{O} \subset \mathcal{X}$ the following conditions hold:

$$\Pr(G_{\theta^r}(z) \in \mathcal{O}) \leq e^{\epsilon} \Pr(G_{\theta^u}(z) \in \mathcal{O}) + \delta$$

$$\Pr(G_{\theta^u}(z) \in \mathcal{O}) \leq e^{\epsilon} \Pr(G_{\theta^r}(z) \in \mathcal{O}) + \delta$$

Proposed Method for Q.2



Experiments and Results for Q.1

Table: PUL (↑), FID (↓) and Ret-FID (↓) after unlearning MNIST classes. FID of pre-trained GAN: 5.4.

Features	Metrics	Extrapolation	NL2 *repulsion*	IL2 *repulsion*	EL2 *repulsion*
Class-1	PUL	95.10 ± 0.69	97.85 ± 2.25	92.97 ± 0.48	**99.32 ± 0.43**
	FID	41.39 ± 1.76	9.69 ± 0.07	13.06 ± 0.46	**9.65 ± 0.21**
	Ret-FID	42.98 ± 0.68	6.70 ± 0.25	16.55 ± 0.54	**6.29 ± 0.18**
Class-4	PUL	94.50 ± 0.05	93.03 ± 0.7	90.39 ± 1.36	**96.23 ± 0.54**
	FID	17.90 ± 0.35	10.50 ± 0.34	15.54 ± 0.05	**10.24 ± 0.19**
	Ret-FID	27.81 ± 0.37	6.26 ± 0.12	8.64 ± 0.9	**5.80 ± 0.04**
Class-8	PUL	90.90 ± 0.12	97.92 ± 0.677	**98.28 ± 0.55**	95.22 ± 0.34
	FID	45.79 ± 0.29	9.95 ± 0.177	9.72 ± 0.31	**8.89 ± 0.52**
	Ret-FID	44.3 ± 0.40	6.70 ± 0.18	11.64 ± 0.46	**5.68 ± 0.10**

Table: PUL (↑), FID (↓) and Ret-FID (↓) after unlearning CelebA-HQ features. FID of pre-trained GAN: 5.3.

Features	Metrics	Extrapolation	NL2 *repulsion*	IL2 *repulsion*	EL2 *repulsion*
Bangs	PUL	89.54 ± 0.09	90.41 ± 0.19	84.05 ± 1.03	**90.45 ± 1.02**
	FID	11.54 ± 0.07	11.92 ± 0.46	13.09 ± 0.10	**11.16 ± 0.08**
	Ret-FID	11.02 ± 0.06	08.69 ± 0.05	09.07 ± 0.18	**07.94 ± 0.32**
Hat	PUL	94.35 ± 0.12	93.99 ± 1.70	94.00 ± 0.75	**94.40 ± 2.19**
	FID	12.18 ± 0.04	9.60 ± 0.25	11.31 ± 0.06	**9.45 ± 0.96**
	Ret-FID	10.12 ± 0.07	06.44 ± 0.11	07.25 ± 0.13	**06.31 ± 0.64**
Bald	PUL	94.44 ± 0.34	**97.13 ± 1.42**	83.51 ± 2.18	93.97 ± 2.65
	FID	23.44 ± 0.02	14.7 ± 0.55	12.94 ± 0.89	**11.07 ± 0.86**
	Ret-FID	26.40 ± 0.30	09.03 ± 0.13	09.87 ± 0.04	**07.83 ±0.05**
Eyeglasses	PUL	92.80 ± 0.14	83.76 ± 3.21	75.23 ± 6.25	**93.63 ± 0.42**
	FID	23.70 ± 0.07	12.81 ± 0.88	13.12 ± 0.78	**9.66 ± 0.58**
	Ret-FID	19.10 ± 0.10	07.93 ± 0.99	**06.11 ± 0.24**	09.84±0.23

Experiments and Results for Q.1

(a) Original samples (b) Extrapolation (c) IL2 *repulsion* (d) NL2 *repulsion* (e) EL2 *repulsion*

Figure: Results of Unlearning undesired feature via different methods.

Experiments and Results for Q.2

Table: Accuracy (↑), Recall(↑), AUC(↑), FID (↓) Density(↑) and Coverage(↑) after filtering **MNIST** classes with only **20 pos and 20 neg**. user-feedback

Methods	Class-5						Class-8					
	Acc.	Rec.	AUC	FID	Dens.	Cov.	Acc.	Rec.	AUC	FID	Dens.	Cov.
Base Classifier	0.64±0.01	0.19±0.01	0.45±0.01	3.60±0.14	0.93±0.01	0.91±0.01	0.70±0.01	0.35±0.02	0.53±0.01	1.42±0.07	**0.99±0.01**	0.99±0.00
Base Classifier + Data Aug.	**0.89±0.00**	0.10±0.00	0.50±0.00	0.80±0.03	0.93±0.00	**0.99±0.00**	**0.94±0.00**	0.10±0.00	0.50±0.00	0.55±0.07	**0.99±0.00**	**1.01±0.00**
Imp-LS + MD	0.60±0.10	**0.69±0.06**	0.69±0.04	2.34±0.97	0.95±0.02	0.96±0.02	0.63±0.05	**0.71±0.03**	**0.72±0.03**	1.78±0.33	0.98±0.01	0.97±0.01
Imp-LS + MD + Latent Aug.	0.61±0.09	**0.69±0.06**	0.69±0.04	2.30±0.94	**0.96±0.02**	0.96±0.01	0.63±0.05	**0.71±0.03**	**0.73±0.03**	1.79±0.36	**0.99±0.01**	0.97±0.01
Imp-LS + Norm-SVM	0.61±0.11	0.67±0.11	0.69±0.03	2.27±1.00	**0.96±0.01**	0.96±0.02	0.62±0.05	0.69±0.04	0.71±0.02	1.90±0.36	**0.99±0.02**	0.97±0.01
Imp-LS + Norm-SVM + Latent Aug.	0.54±0.06	0.56±0.02	0.57±0.05	3.05±0.72	0.92±0.01	0.95±0.01	0.58±0.04	0.60±0.10	0.63±0.06	2.39±0.37	**0.99±0.01**	0.96±0.01
Inv-LS + MD	0.87±0.10	0.11±0.02	0.74±0.04	0.65±0.95	0.93±0.02	**0.99±0.02**	0.92±0.01	0.01±0.01	0.61±0.08	0.64±0.08	**0.99±0.01**	**1.10±0.00**
Inv-LS + MD + Latent Aug.	0.87±0.01	0.12±0.02	0.75±0.03	**0.63±0.97**	0.93±0.02	**0.99±0.01**	0.92±0..02	0.01±0.00	0.59±0.06	0.58±0.08	**0.99±0.01**	0.99±0.01
Inv-LS + Norm-SVM	0.85±0.13	0.55±0.02	**0.83±0.04**	0.72±1.01	**0.96±0.01**	0.97±0.02	0.89±0.06	0.11±0.14	0.68±0.04	0.85±0.33	**0.99±0.00**	0.99±0.01
Inv-LS + Norm-SVM + Latent Aug.	0.79±0.14	0.30±0.02	0.70±0.04	2.22±1.02	0.93±0.01	0.94±0.02	0.92±0.05	0.12±0.14	**0.72±0.05**	**0.48±0.03**	**0.99±0.01**	0.99±0.01

Table: Accuracy (↑), Recall(↑), AUC(↑), FID (↓) Density(↑) and Coverage(↑) after filtering **Celeba-HQ** features with only **20 pos and 20 neg.** user-feedback

Methods	Class-Bangs						Class-Hats					
	Accuracy	Recall	AUC Score	FID	Density	Coverage	Accuracy	Recall	AUC Score	FID	Density	Coverage
Base Classifier	0.41±0.03	0.20±0.01	0.69±0.01	10.58±0.42	**1±0.01**	0.88±0.01	0.64±0.02	0.27±0.01	0.45±0.01	6.37±0.51	**1.08±0.02**	0.99±0.00
Base Classifier + Data Aug.	**0.97±0.01**	0.10±0.00	0.50±0.00	**0.08±0.02**	0.99±0.01	**1±0.00**	0.67±0.01	0.1±0.00	0.5±0.00	6.57±0.04	1.05±0.01	**1±0.00**
Imp-LS + MD	0.78±0.12	0.91±0.05	0.90±0.04	3.71±1.12	0.99±0.02	0.94±0.02	0.76±0.11	0.77±0.06	0.84±0.02	3.55±1.01	0.98±0.01	0.93±0.01
Imp-LS + MD + Latent Aug.	0.77±0.10	0.91±0.07	0.89±0.04	3.69±0.97	0.99±0.02	0.94±0.02	0.77±0.12	0.78±0.06	0.85±0.03	3.50±0.92	0.98±0.01	0.93±0.01
Imp-LS + Norm-SVM	0.78±0.08	**0.93±0.06**	0.92±0.02	2.54±1.00	0.99±0.01	0.96±0.02	0.77±0.07	0.82±0.04	0.86±0.02	3.81±0..93	0.95±0.01	0.93±0.01
Imp-LS + Norm-SVM + Latent Aug.	0.81±0.06	**0.93±0.02**	**0.93±0.05**	2.31±0.72	0.99±0.01	0.97±0.01	0.76±0.05	0.77±0.03	0.84±0.05	3.35±0.65	0.99±0.15	0.94±0.01
Inv-LS + MD	0.71±0.02	0.89±0.01	0.89±0.01	9.91±1.14	0.96±0.01	0.85±0.01	0.78±0.06	**0.86±0.03**	0.88±0.02	9.06±0.72	0.94±0.01	0.83±0.02
Inv-LS + MD + Latent Aug.	0.72±0.02	0.89±0.01	0.89±0.01	9.86±1.04	0.96±0.01	0.85±0.01	**0.79±0.06**	**0.86±0.03**	**0.89±0.02**	9.05±0.70	0.94±0.01	0.83±0.02
Inv-LS + Norm-SVM	0.80±0.01	**0.94±0.02**	**0.93±0.03**	3.61±0.65	**1.01±0.02**	0.96±0.01	0.76±0.05	0.74±0.01	0.85±0.01	5.15±0.81	0.93±0.02	0.90±0.01
Inv-LS + Norm-SVM + Latent Aug.	0.80±0.02	0.92±0.02	**0.93±0.02**	3.83±0.70	1.00±0.01	0.94±0.01	0.78±0.05	0.78±0.01	0.86±0.01	5.48±0.75	0.94±0.02	0.90±0.01

Conclusion

- In this work, we present a methodology to prevent the generation of samples containing undesired features from a pre-trained GAN.
- It is worth mentioning that our method does not assume the availability of the training dataset of the pre-trained GAN so it can generalize to zero-shot settings.
- Despite these advantages, there are some limitations that our methodology can't encompass such as changes in correlated features while unlearning undesired features. Due to high entanglement between the semantics features this kind of impact on other features is visible in the generated outputs

References I

Cao, Y., and Yang, J.

Towards making systems forget with machine unlearning.

In Proc. of IEEE Symposium on Security and Privacy (2015).

Chourasia, R., and Shah, N.

Forget unlearning: Towards true data-deletion in machine learning.

In Proc. of ICML (2023).

Ginart, A., Guan, M., Valiant, G., and Zou, J. Y.

Making ai forget you: Data deletion in machine learning.

In Proc. of NIPS (2019).

Golatkar, A., Achille, A., Ravichandran, A., Polito, M., and Soatto, S.

Mixed-privacy forgetting in deep networks.

In Proc. of CVPR (2021).

References II

Golatkar, A., Achille, A., and Soatto, S.

Eternal sunshine of the spotless net: Selective forgetting in deep networks.

In Proc. of CVPR (2020).

Golatkar, A., Achille, A., and Soatto, S.

Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations.

In Proc. of ECCV (2020).

Goldman, E.

An introduction to the california consumer privacy act (ccpa).

Santa Clara Univ. Legal Studies Research Paper (2020).

Graves, L., Nagisetty, V., and Ganesh, V.

Amnesiac machine learning.

In Proc. of AAAI (2021).

References III

Guo, C., Goldstein, T., Hannun, A., and van der Maaten, L.

Certified data removal from machine learning models.

In Proc. of ICML (2020).

Ma, Z., Liu, Y., Liu, X., Liu, J., Ma, J., and Ren, K.

Learn to forget: Machine unlearning via neuron masking.

In Proc. of IEEE Transactions on Dependable and Secure Computing (2022).

Sekhari, A., Acharya, J., Kamath, G., and Suresh, A. T.

Remember what you want to forget: Algorithms for machine unlearnings.

In Proc. of NeurIPS (2021).

Tiwary, P., Guha, A., Panda, S., and A.P, P.

Adapt then unlearn: Exploiting parameter space semantics for unlearning in generative adversarial networks.

arXiv (2023).

References IV

Voigt, P., and dem Bussche, A.

The EU general data protection regulation (GDPR).

Springer, 2017.

Wu, G., Hashemi, M., and Srinivasa, C.

Puma: Performance unchanged model augmentation for training data removal.

In Proc. of AAAI (2022).

Wu, Y., Dobriban, E., and Davidson, S. B.

Deltagrad: Rapid retraining of machine learning models.

In Proc. of ICML (2020).

Wu, Y., Tannen, V., and Davidson, S. B.

Priu: A provenance-based approach for incrementally updating regression models.

In Proc. of SIGMOD (2020).

References V

Ye, J., Fu, Y., Song, J., Yang, X., Liu, S., Jin, X., Song, M., and Wang, X.

Learning with recoverable forgetting.

In Proc. of ECCV (2022).