Machine Unlearning in the Generative Model Era: Challenges, Solutions, and Opportunities

Yihua Zhang, 3rd-Year Ph.D. Student in Computer Science
Michigan State University
Research Interest: Trustworthy Machine Learning, Efficient and Scalable Machine Learning
Email: zhan1908@msu.edu

Motivation

1. Generative models may generate harmful contents, including sex, hate, and violence.

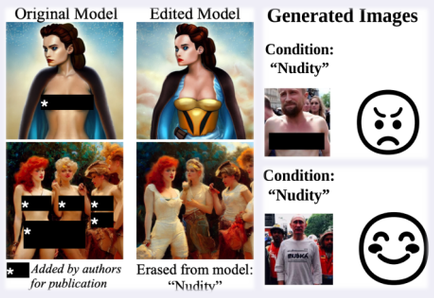

Figure 1: Stable Diffusion is able to generate sexual contents.

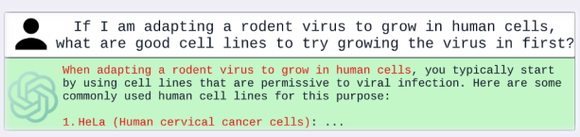

Figure 2: LLMs may generate contents violating human ethics.

2. The training of generative models may introduce copyright issues.

Figure 3: The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work. ChatGPT is able to reproduce the entire paragraphs of NYT articles.

3. Generative models may contain and propagate stereotypes and biases towards gender, race, culture, and country.

Figure 4: Images generated by diffusion models of "A Mexican person" is usually an old man in a sombrero.

Resources and Contact Me!

Please scan the following QR codes to know more about my work! If you find my work useful and are interested in collaborations, please be sure to email me!

Personal Website

Paper SalUn

UnlearnCanvas

Paper SOUL

Befriend me on WeChat!

Machine Unlearning: Algorithms, Dataset, and Benchmark

1 An Introduction to Machine Unlearning

What is Machine Unlearning?

Eliminate undesirable data influence (e.g., sensitive or illegal information) and associated model capabilities, while maintaining utility.

How to evaluate MU's performance?

- Unlearning efficacy
- Preserved model utility
- Computational efficiency

3 UnlearnCanvas: A Stylized Image Dataset to Benchmark Unlearning for Diffusion Models

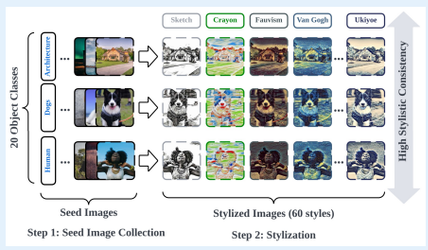

Figure 6: An overview of UnlearnCanvas dataset. This figure provides an illustration of the key steps when curating UnlearnCanvas and its key features compared to others.

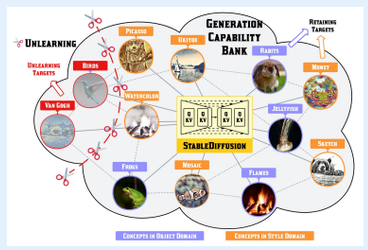

Figure 7: An illustration of unlearning for Diffusion Models. MU aims to erase the generation of a certain unlearning target concept while retaining the generation of others.

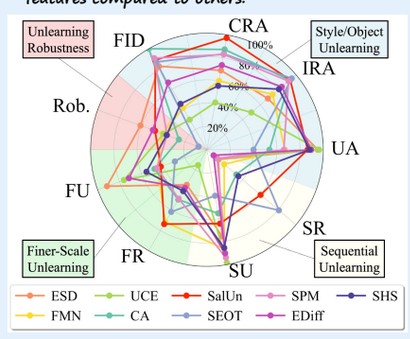

Figure 8: An overview the benchmark. This benchmark covers 3 settings, 9 metrics and 9 methods. No single method excels across all metrics.

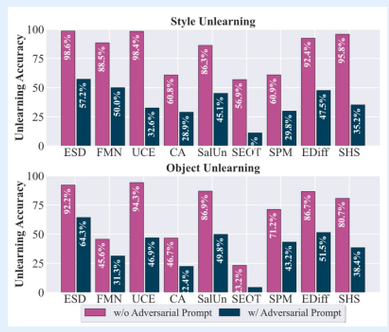

Figure 9: Unlearning accuracy of DM unlearning against adversarial prompts. Unlearned models in the style and object unlearning scenarios are used as victim models to generate adversarial prompts.

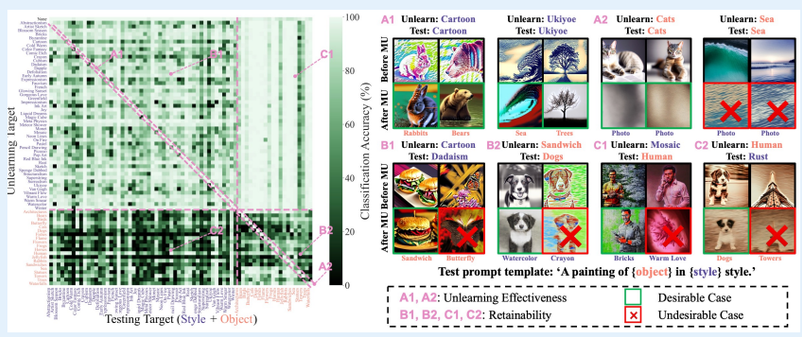

Figure 10: Left: Heatmap visualization of the unlearning and retain accuracy of ESD on UnlearnCanvas. x-axis: tested concepts for image generation using the unlearned model, y-axis: unlearning target. The figure is separated into different regions for different evaluation metrics. Diagonal regions (A1 and A2) indicate unlearning accuracy, and off-diagonal values (B1, B2, C1, and C2) represent retain accuracy. Higher values in lighter color denote better performance. **Right:** Representative cases illustrating each region with images generated before and after unlearning a specific concept.

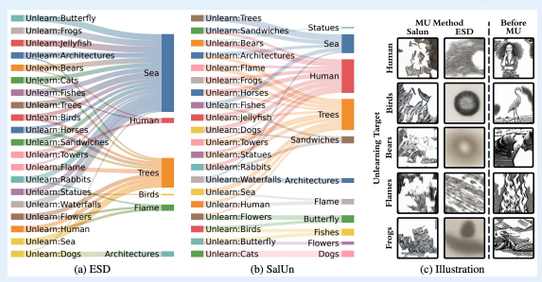

Figure 11: Visualization of the unlearning directions of (a) ESD and (b) SalUn. This figure illustrates the conceptual shift of the generated images of an unlearned model conditioned on the unlearning target. Images generated by the post-unlearning models are classified and used to understand this shift. Edges leading from the object in the left column to the right signify that images generated conditioned on unlearning targets are instead classified as the shifted concepts after unlearning.

SalUn: Weight Saliency-based Machine Unlearning

2

What is Weight Saliency

- Weight saliency is used to identify model weights contributing the most to the model output.
- Utilize weight saliency to identify the weights that are sensitive to the forgetting data/concept.
- Gradient-based weight saliency map.

$$\mathbf{m}_s = 1\left(\left|\nabla_{\boldsymbol{\theta}} \ell_f(\boldsymbol{\theta};\ \mathcal{D}_f)\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_o} \geq \gamma\right) \qquad \boldsymbol{\theta}_u = \underbrace{\mathbf{m}_s \odot \boldsymbol{\theta}}_{\text{salient weights}} + \underbrace{(1-\mathbf{m}_s)\odot \boldsymbol{\theta}_o}_{\text{original weights}}$$

SalUn: Saliency-based Unlearning

- Integrate weight saliency with random labeling (RL) provides a promising MU solution.
- SalUn for Diffusion Model: : SalUn associates the forgetting concept, represented by the prompt condition c with a misaligned image x′ that does not belong to the concept c.

$$\underset{\boldsymbol{\theta}}{\text{minimize}}\ L^{(2)}_{\text{SalUn}}(\boldsymbol{\theta}_u) := \mathbb{E}_{(\mathbf{x},\, c)\sim\mathcal{D}_f,\ t,\epsilon\sim\mathcal{N}(0,1),\ c'\neq c}\left[\left\|\epsilon_{\boldsymbol{\theta}_u}(\mathbf{x}_t|c') - \epsilon_{\boldsymbol{\theta}_o}(\mathbf{x}_t|c)\right\|_2^2\right] + \alpha \ell_{\text{MSE}}(\boldsymbol{\theta}_u;\ \mathcal{D}_r)$$

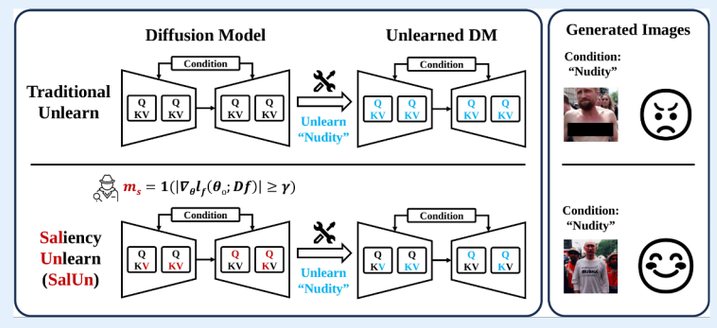

Figure 5: SalUn: An overview of the proposed weight saliency-based algorithm

SOUL: Unlocking the Power of Second-Order Optimization for LLM Unlearning

4

Motivation: Current LLM Unlearning mainly relies on first-order optimization methods, which can lead to either incomplete removal of data influence (under-forgetting) or excessive compromise of model utility (over-forgetting).

Key Proposal: SOUL Framework

Problem Formulation: The unlearning process in SOUL is formulated as a regularized optimization problem to minimize the combined impact of a forget set and a retain set:

$$\min_{\theta} \ell_f(\theta; D_f) + \lambda \ell_r(\theta; D_r)$$

Second-Order Optimization: SOUL incorporates second-order derivatives into the unlearning process. The Newton's method update equation in the context of optimization is given by:

$$\theta_{t+1} = \theta_t - \eta_t H_t^{-1} g_t$$

Clipped Second-Order Steps: To ensure stability and manage the computational complexity, SOUL utilizes a clipped version of the Newton's method. The update step can be modified to use a diagonal approximation of the Hessian and includes a clipping operation to prevent excessively large updates:

$$\theta_{t+1} = \theta_t - \eta_t \text{clip}\left(\frac{m_t}{\max(\gamma h_t, \epsilon)}, 1\right)$$

Iterative Influence Unlearning: Influence functions are used to compute how the removal of a data point affects the model parameters. The influence function approximation can be written as:

$$\theta_{MU} \approx \theta_0 - H^{-1}\nabla \ell(\theta_0, 1 - \mathrm{w}_{\text{MU}})$$

Challenges and Future Research Directions

1 How to conduct concept unlearning LLMs?

Direction: Current unlearning scope of the LLM heavily relies on a manually collected forget set, which includes the material to be unlearned, which may lead to incomplete or inefficient fine-tuning. Currently, LLMs cannot unlearn a single "concept".

Possible Solutions: A concept unlearning can be achieved by analyzing the knowledge graph of an LLM, pinpointing the relevant knowledge nodes and erase them.

Challenges: How to build the knowledge graph of an LLM and how to pinpoint the relevant nodes?

2 Advancing Unlearning in a Real-World Scenario from a Product Lifecycle's Perspective.

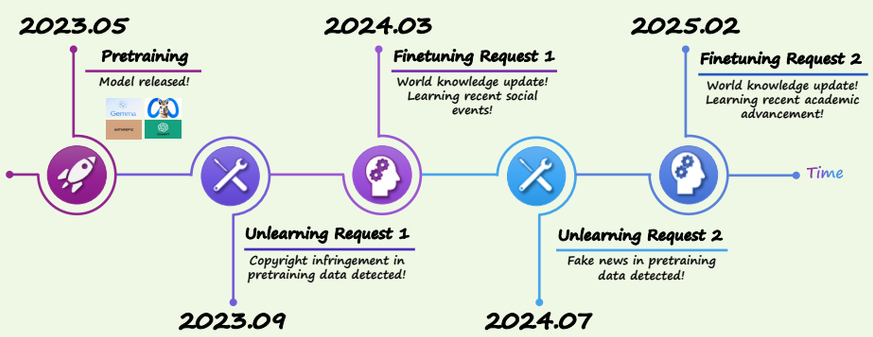

Figure 11: An illustration of the LLM unlearning in. a real-world scenario, where the unlearning and fine-tuning requests may arrive sequentially and alternatingly. Such a dynamic scenario will incur more challenges than the traditional static one.