ICML
International Conference
On Machine Learning
2025

Targeted Unlearning with Single Layer Unlearning Gradient

Zikui Cai[1,2 *], Yaoteng Tan[1 *], M. Salman Asif[1]

[1] University of California, Riverside

[2] University of Maryland

** Equal Contribution*

Challenges of Machine Unlearning

- Removing the influence of a specific subset of training data (forget set), while retaining overall model utility on the retain set.

$$\min_{\theta} \underbrace{\frac{1}{N_{\mathrm{r}}} \sum_{(x_{\mathrm{r}}, y_{\mathrm{r}}) \in D_{\mathrm{r}}} \ell(F_\theta(x_{\mathrm{r}}), y_{\mathrm{r}})}_{\mathcal{L}_{\text{retain}}} - \underbrace{\frac{\alpha}{N_{\mathrm{f}}} \sum_{(x_{\mathrm{f}}, y_{\mathrm{f}}) \in D_{\mathrm{f}}} \ell(F_\theta(x_{\mathrm{f}}), y_{\mathrm{f}})}_{\mathcal{L}_{\text{forget}}}$$

Fine-tuning (FT) Gradient ascent (GA)

Two stage: Gradient ascent + Fine-tuning (GAFT)

- Challenges:
  - **High Computational Cost:** involves iterative gradient calculation and model update over the whole model
  - **Side Effects:** Removing one concept can lead to degraded performance on unrelated tasks due to interconnected representations
  - **Robustness Issues:** Unlearned concepts can be recovered via careful probing or attacks

Our Three Main Objectives

Computational Efficiency

Minimize computational overhead compared to full retraining:

- Achieve unlearning with minimal parameter updates
- Enable practical unlearning for large-scale models

Effective Unlearning

Ensure complete removal of targeted concepts:

- Achieve zero forget accuracy on target concepts
- Prevent information leakage through related concepts
- Resist recovery through adversarial techniques

Targeted Removal with Minimal Side Effects

Maintain model utility while removing specific content:

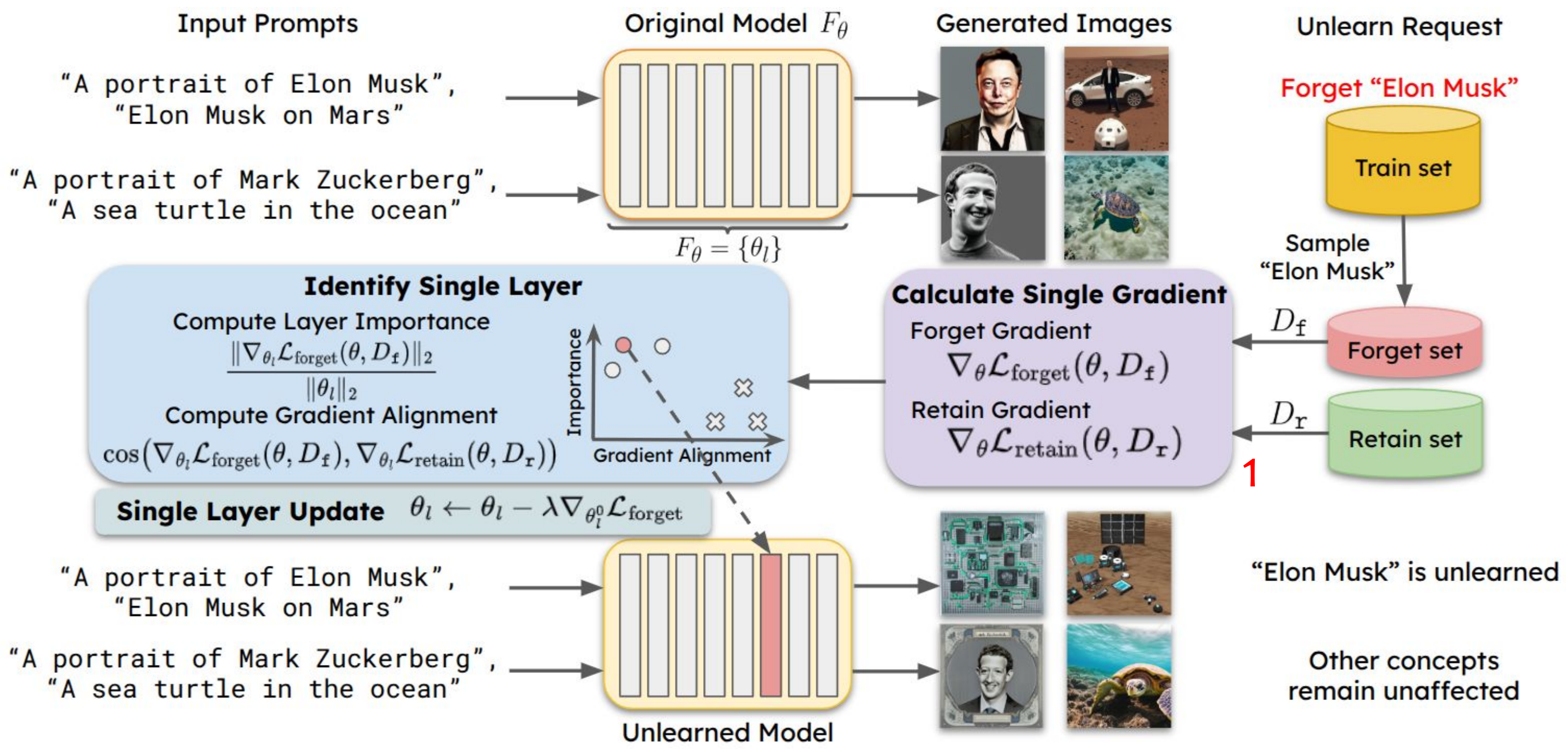
- Preserve performance on unrelated tasks
- Maintain accuracy on retain set
- Ensure precision in targeting only unwanted concepts

Introduction

- Goal:
  - Balance unlearning effectiveness & retaining model utility
  - Increase efficiency (computational time + resource)
- Questions
  - Can we select the most critical model part(s) for unlearning?
  - Can we do update that part efficiently?
- Proposed method:
  - Achieves unlearning with reduced model parameter manipulation
  - Requires <u>one-time gradient calculation</u> and <u>a single- layer update</u>.

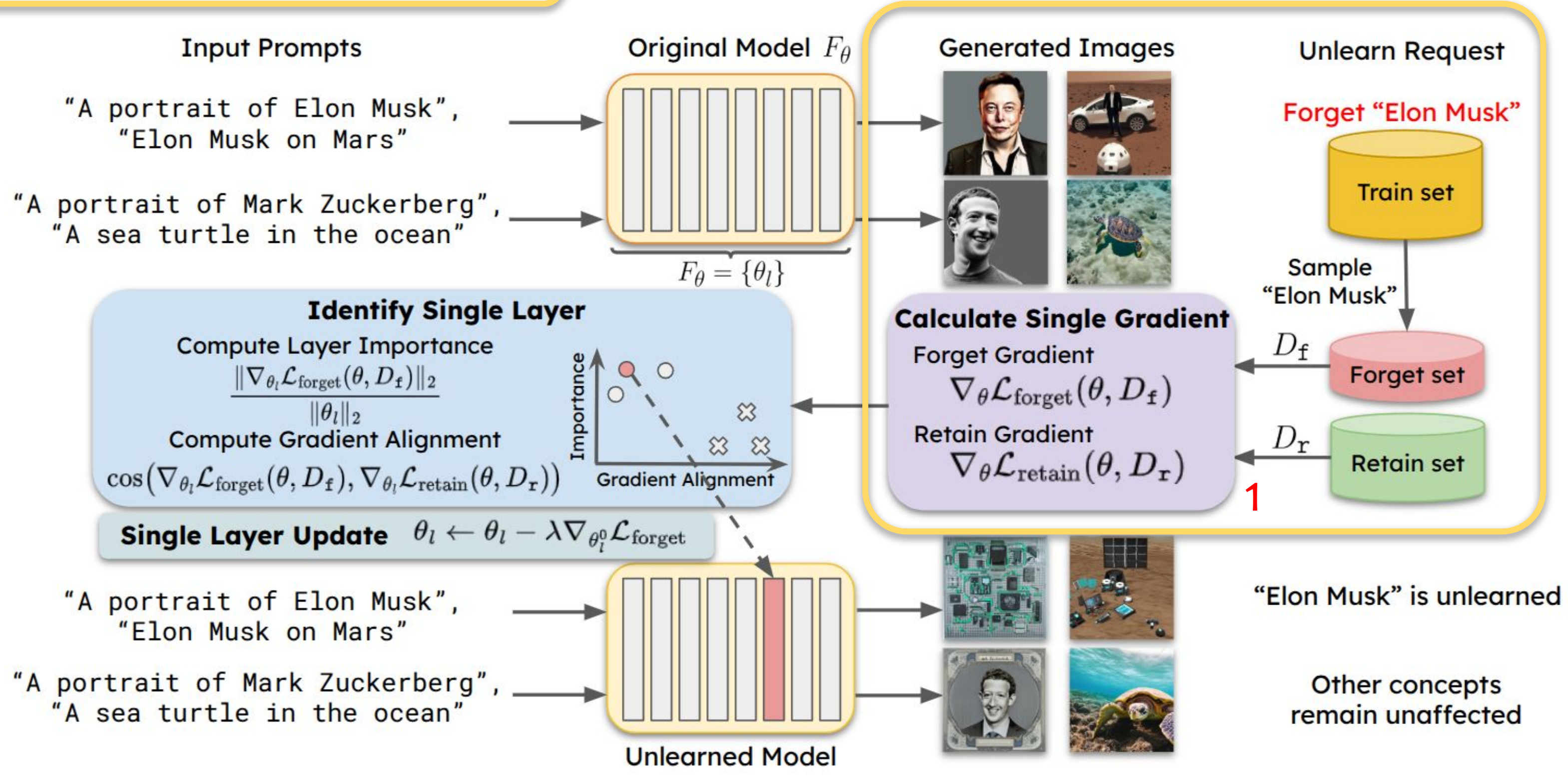
Framework: Single Layer Unlearning Gradient (SLUG)

- [1] Compute gradients –>



Framework: **S**ingle **L**ayer **U**nlearning **G**radient (SLUG)

- [1] Compute gradients –>



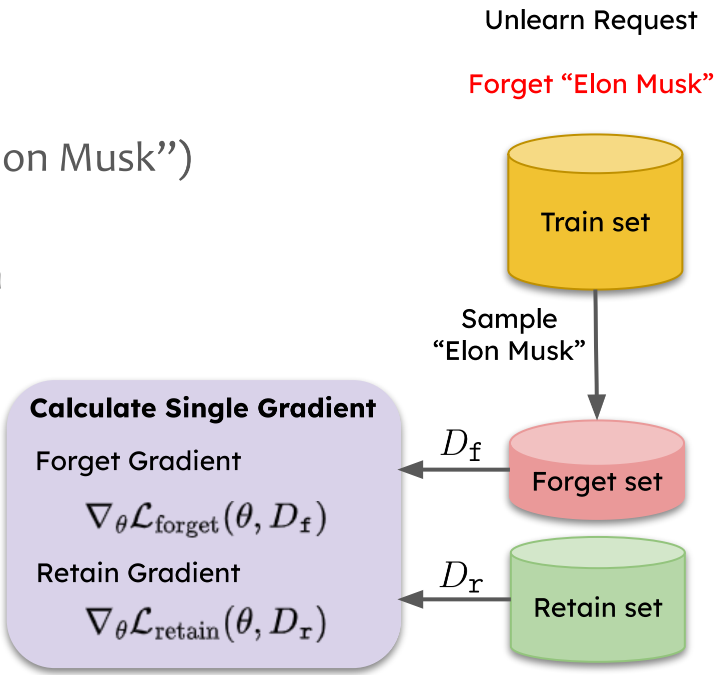
Compute Gradient

- Sample the forget and retain sets
  - Forget set:
    Images related to unlearning requests (e.g., “Elon Musk”)
  - Retain set:
    Subset of training set excluding the forget data
- Compute forget and retain gradients
  - Forget loss:
    Alignments of text and image embedding

$$\mathcal{L}_{\text{forget}}(\mathbf{v}_i, \mathbf{t}_j) = 1 - \cos(\mathbf{v}_i, \mathbf{t}_j)$$

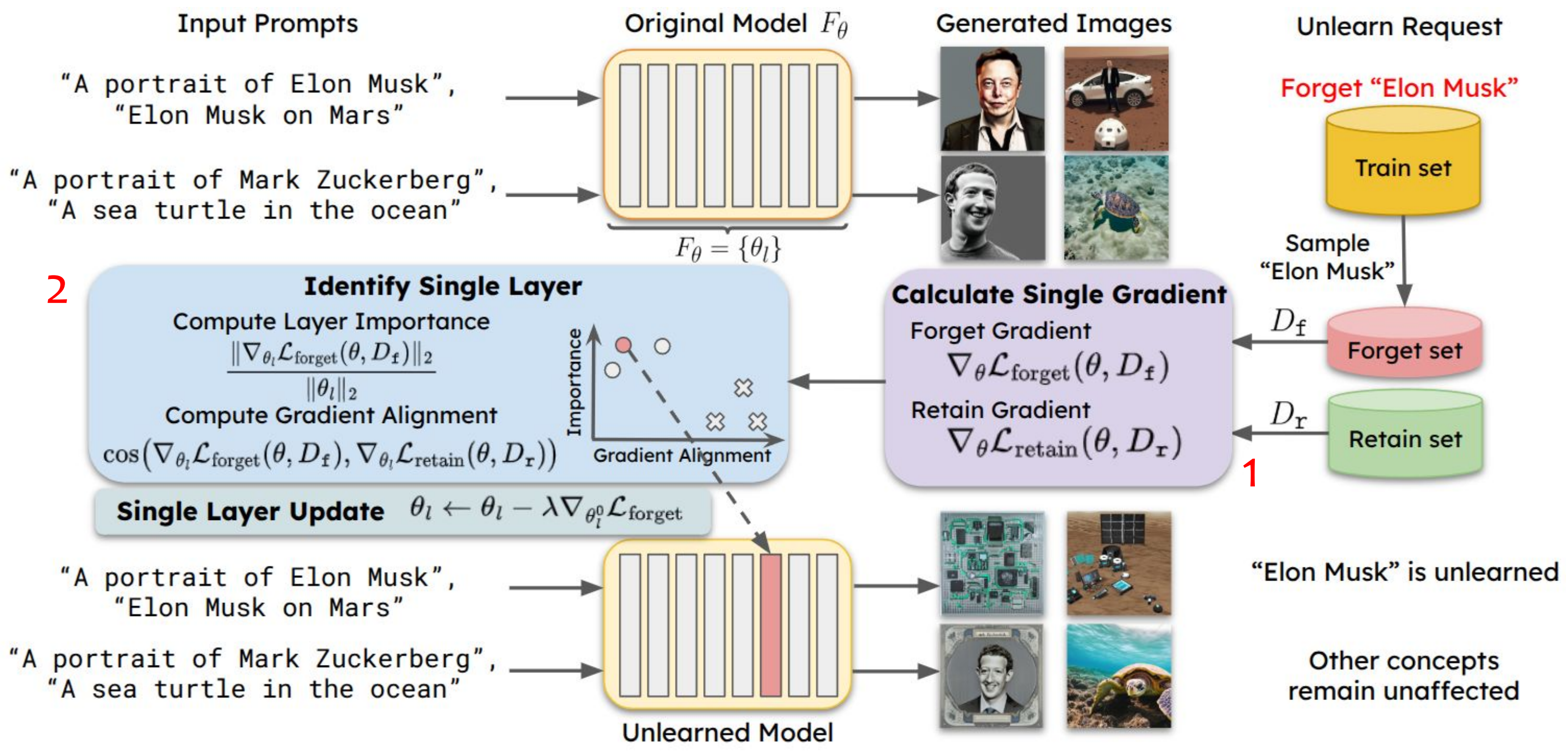
  - Retain loss:
    Image and Text contrastive loss (CLIP training)

$$\mathcal{L}_{\text{retain}} = \frac{1}{2N} \sum_{i=1}^{N} (\ell_{i2t}(i) + \ell_{t2i}(i))$$



Framework: **S**ingle **L**ayer **U**nlearning **G**radient (SLUG 🐌)

- [1] Compute gradients –> [2] Identify layer to update



Layer Identification

- Identify the most critical part to update
- Utilize Forget and Retain gradients
  - **Layer importance:**
    Higher the more impactful on unlearning
  - **Gradient alignment:**
    Lower the least impactful on retaining
  - **Pareto-front:** Layers that are well-balanced for unlearning and retaining

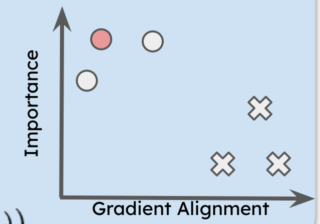
Identify Single Layer

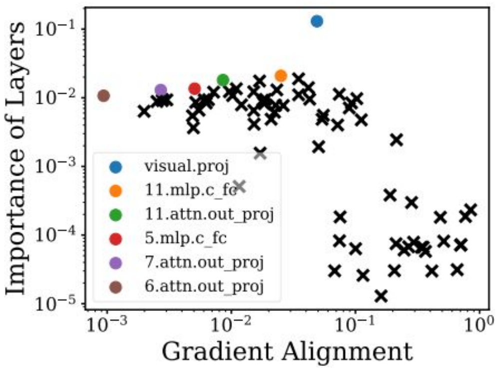
Compute Layer Importance

$$\frac{\|\nabla_{\theta_l}\mathcal{L}_{\text{forget}}(\theta, D_{\mathtt{f}})\|_2}{\|\theta_l\|_2}$$

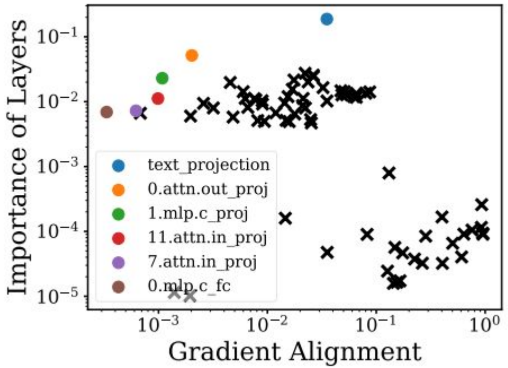
Compute Gradient Alignment

$$\cos\left(\nabla_{\theta_l}\mathcal{L}_{\text{forget}}(\theta, D_{\mathtt{f}}), \nabla_{\theta_l}\mathcal{L}_{\text{retain}}(\theta, D_{\mathtt{r}})\right)$$





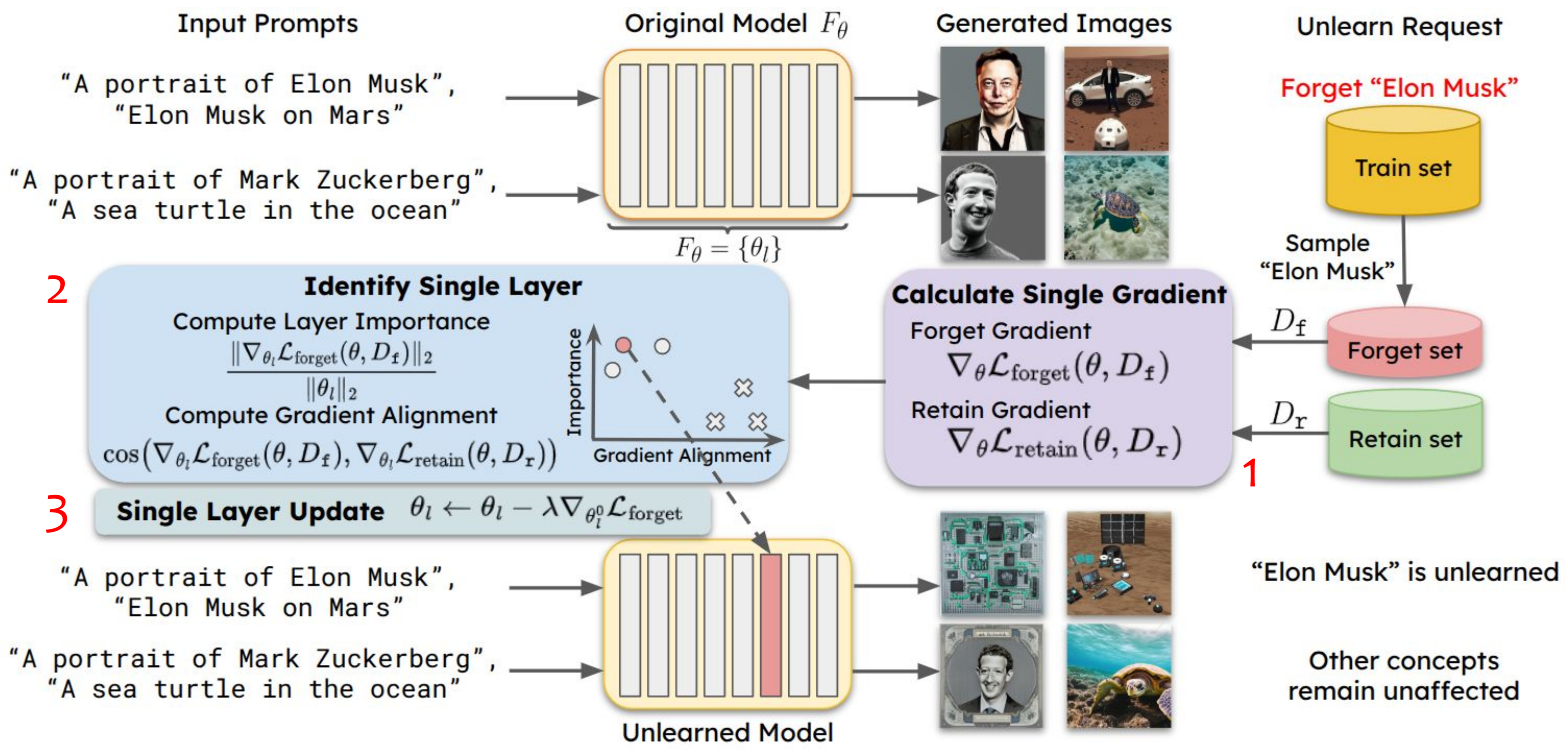
Layers of vision model



Layers of language model

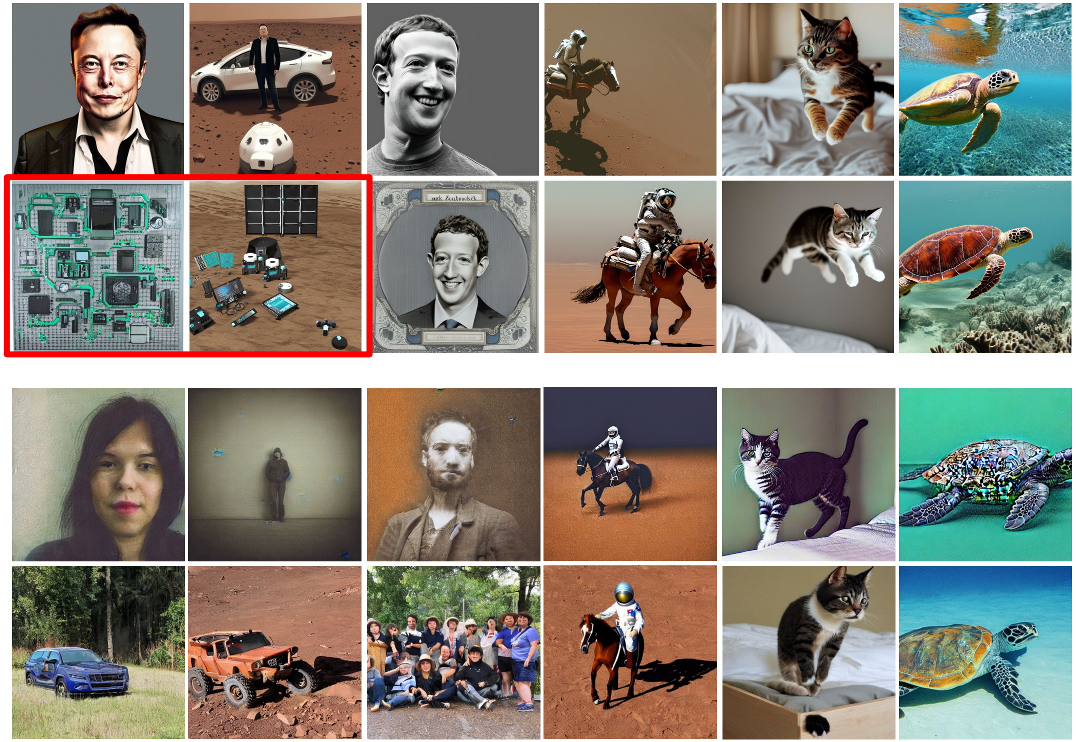
Framework: **S**ingle **L**ayer **U**nlearning **G**radient (SLUG)

- [1] Compute gradients –> [2] Identify layer to update –> [3] Search proper step-size



Unlearning Text-to-Image Models

- Setup
  - Unlearning identity “Elon Musk” on Stable Diffusion
  - Comparing methods
    - SalUn [1]
    - ESD [2]



Examples of Stable Diffusion unlearned “Elon Musk”

[1] SalUn (Fan et. al, ICLR 2024)
[2] ESD (Gandikota et. al, ICCV 2023)

More examples

- Setup
  - Unlearning intellectual properties that have copyright on Stable Diffusion
    - “Iron man”*
    - “Mickey Mouse”*

Examples of Stable Diffusion unlearned “Iron Man”*

Examples of Stable Diffusion unlearned “Mickey Mouse”*

---

*Iron Man: Marvel Comics character
*Mickey Mouse: Walt Disney character

Results: UnlearnCanvas benchmark

- Setup
  - UA (unlearning acc)
  - IRA (In domain retain acc)
  - CRA (cross domain retain acc)
- Main takeaway
  - SLUG achieve the best trade-off between efficiency and effectiveness

Illustration of UA, IRA, CRA metrics [3]

Quantitative evaluation of SLUG unlearning on UnlearnCanvas benchmark

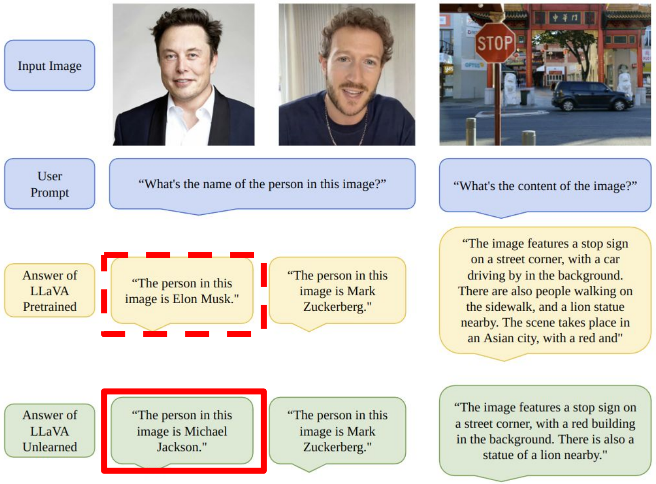
Method	Effectiveness							Efficiency		
	Style Unlearning			Object Unlearning			FID (↓)	Time	Memory	Storage
	UA (↑)	IRA (↑)	CRA (↑)	UA (↑)	IRA (↑)	CRA (↑)		(s) (↓)	(GB) (↓)	(GB) (↓)
ESD [9]	98.58%	80.97%	93.96%	92.15%	55.78%	44.23%	65.55	6163	17.8	4.3
FMN [40]	88.48%	56.77%	46.60%	45.64%	90.63%	73.46%	131.37	350	17.9	4.2
UCE [10]	98.40%	60.22%	47.71%	94.31%	39.35%	34.67%	182.01	434	5.1	1.7
CA [18]	60.82%	96.01%	92.70%	46.67%	90.11%	81.97%	54.21	734	10.1	4.2
SalUn [7]	86.26%	90.39%	95.08%	86.91%	96.35%	99.59%	61.05	667	30.8	4.0
SEOT [21]	56.90%	94.68%	84.31%	23.25%	95.57%	82.71%	62.38	95	7.34	0.0
SPM [25]	60.94%	92.39%	84.33%	71.25%	90.79%	81.65%	59.79	29700	6.9	0.0
EDiff [36]	92.42%	73.91%	98.93%	86.67%	94.03%	48.48%	81.42	1567	27.8	4.0
SHS [35]	95.84%	80.42%	43.27%	80.73%	81.15%	67.99%	119.34	1223	31.2	4.0
SLUG (Ours)	86.29%	84.59%	88.43%	75.43%	77.50%	81.18%	75.97	39	3.61	0.04

Unlearning Image-to-Text Models

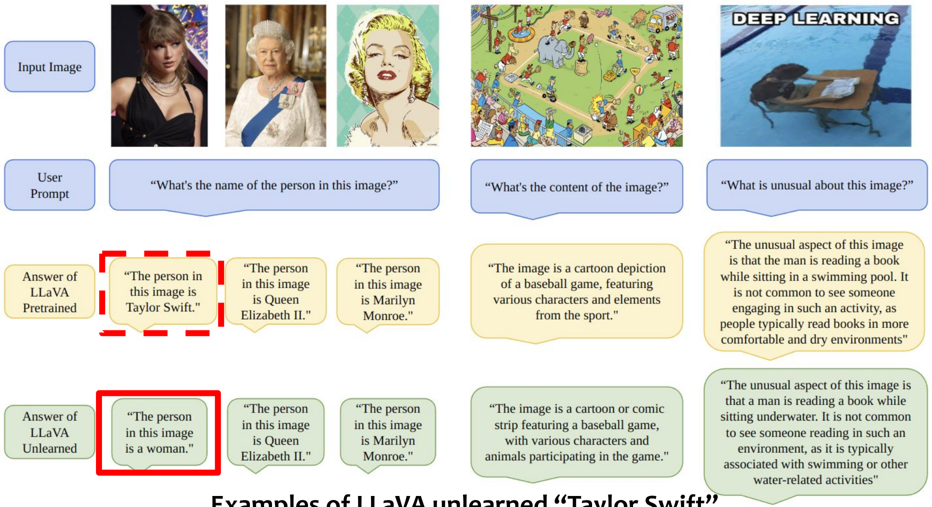
- Setup
  - Unlearning identities on VLMs (LLaVA 1.5-7B)
    - "Elon Musk"
    - "Taylor Swift"

Quantitative evaluation of unlearned LLaVA 1.5 models

Model	Forget Accuracy (%) (↓)	VLM Benchmark Score (↑)			
		MME Cognition	MME Perception	GQA	MMBench (en)
Original LLaVA-1.5	99.50	323.57	1481.21	61.28	62.97
Unlearned "Elon Musk"	3.0	298.57	1354.61	60.70	61.34
Unlearned "Taylor Swift"	2.0	334.64	1336.09	60.72	60.14
Average	2.5	316.61	1345.35	60.71	60.74



Examples of LLaVA unlearned "Elon Musk"



Examples of LLaVA unlearned "Taylor Swift"

Conclusion

- Summary
  - We propose a single layer, one-step unlearning for vision-language foundation models that significantly improves the unlearning efficiency
  - Our framework is scalable to different vision-language tasks

- More details

Paper **Code** **Project page**