Google DeepMind

# Are we making progress in machine unlearning?

**Eleni Triantafillou**
**CoLLAs'24**

Google DeepMind

# Acknowledgements

Work and ideas in this presentation are thanks to my colleagues:

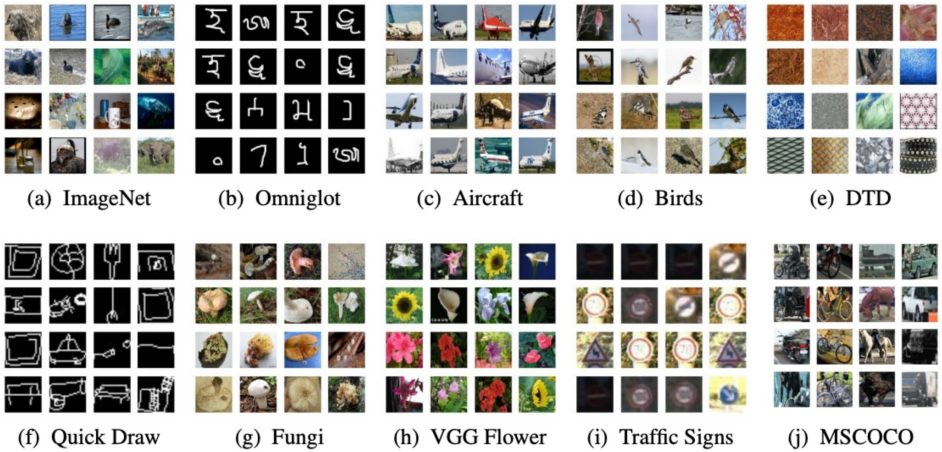
Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio Jacques Junior, Ioannis Mitliagkas, Jun Wan, Lisheng Sun Hosoya, Sergio Escalera, Gintare Karolina Dziugaite, Peter Triantafillou, Isabelle Guyon, George-Octavian Barbulescu, Ilia Shumailov, Amr Khalifa and Nicolas Papernot.

# The dream: build lifelong systems capable of increasingly rapid adaptation

- *Rapidly* acquire new knowledge, learn new tasks
- Become a better *learner* over time: learn new things increasingly quickly
- Avoid “catastrophic forgetting”? 🤔

But sometimes forgetting is useful or necessary.

In this talk, we will discuss ***unlearning***: forgetting that is **targeted** and **intentional.**




**Meta-Dataset. Triantafillou et al. 2020.**


**NEVIS’22. Bornschein et al. 2023.**

Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. Triantafillou et al. ICLR 2020.
Nevis’22: A Stream of 100 Tasks Sampled from 30 Years of Computer Vision Research. Bornschein et al. JMLR, 2023.

# Why would forgetting be beneficial?

Source: https://iai.tv/articles/forgetting-is-more-important-than-remembering-auid-2154

Forgetting is a process that induces plasticity and promotes adaptive behaviour.

(Wixted, 2004; Miller, 2021; Ryan and Frankland, 2022)

*"It is the interaction between persistence and transience that allows for intelligent decision-making in dynamic, noisy environments"*

– Richards and Frankland

*"From creativity to intelligence and empathy to courage, the art of forgetting may be more vital to the human condition than remembering"*

- Small


The Persistence and Transience of Memory. Richards and Frankland. Neuron. 2017.
Forgetting is more important than remembering. IAI News. Small. 2022.
Failures of memory and the fate of forgotten memories. Miller. Neurobiology of Learning and Memory. 2021.
The psychology and neuroscience of forgetting. Wixted. Annu. Rev. Psychol. 2004.
Forgetting as a form of adaptive engram cell plasticity. Ryan and Frankland. Nature Reviews Neuroscience. 2022.

# “Memorization” and “forgetting” in machine learning models

Feldman’s **label memorization** (Feldman, 2020):

**Definition 4.1.** *For a dataset* $S = (x_i, y_i)_{i \in [n]}$ *and* $i \in [n]$ *define*

$$\texttt{mem}(\mathcal{A}, S, i) := \Pr_{h \sim \mathcal{A}(S)}[h(x_i) = y_i] - \Pr_{h \sim \mathcal{A}(S^{\setminus i})}[h(x_i) = y_i],$$

- $\mathcal{A}$: Training algorithm
- $\mathcal{A}(S)$: Model trained on S
- $\mathcal{A}(S^{\setminus i})$: Model trained on S excluding example i

Informally, a learning algorithm memorizes the label of an example i in a dataset if including that example in the training set is necessary for predicting its label correctly.

Does Learning Require Memorization? A Short Tale about a Long Tail. Feldman. SIGACT 2020.
Characterizing Structural Regularities of Labeled Data in Overparameterized Models. Jiang et al. ICML 2021.
The Infinitesimal Jackknife. Louis A Jaeckel. Technical report, Bell Lab., 1972

# “Memorization” and “forgetting” in machine learning models

**Verbatim memorization** in LMs: generating text that exactly matches a substring from the training data

(Carlini et al, 2021) define:

**Definition 1 (Model Knowledge Extraction)** *A string $s$ is* extractable *from an LM $f_\theta$ if there exists a prefix $c$ such that:*

$$s \leftarrow \underset{s' :\ |s'|=N}{\arg\max} f_\theta(s' \mid c)$$

Likelihood of sequence s’

and

**Definition 2 ($k$-Eidetic Memorization)** *A string $s$ is $k$-eidetic memorized (for $k \geq 1$) by an LM $f_\theta$ if $s$ is extractable from $f_\theta$ and $s$ appears in at most $k$ examples in the training data $X$: $|\{x \in X : s \subseteq x\}| \leq k$.*

Informally, “eidetic” or “photographic” memory is the ability to recall information after seeing it only once

Extracting Training Data from Large Language Models. Carlini et al. USENIX 2021.

# “Memorization” and “forgetting” in machine learning models

## Why do modern neural networks memorize?

Theory suggests ideal learners learn general patterns and avoid “overfitting” [1,2]

But overparameterized models achieve both high test accuracy *and* memorization [3]

“although training examples do not have noticeably lower losses than test examples on average, **certain worst-case training examples are indeed memorized**” - Carlini et al [7]

Recent work suggests that **memorization is sometimes *even necessary* for good generalization** [4, 5, 6]

## But the plot thickens...

**Despite networks memorizing, they also tend to... forget** [9, 10]

Examples appearing early in training are forgotten by the end [9] ; no traces of them are left in the model

Several “forgetting events” occur in training; certain examples are “forgotten” with higher frequency [10]


[1] Relating data compression and learnability. Littlestone and Warmuth. 1986.
[2] How much does your data exploration overfit? Controlling bias via information usage. IEEE Transactions on Information Theory. Russo and Zou. 2019.
[3] A Closer Look at Memorization in Deep Networks. Arpit et al. ICML 2017.
[4] Does Learning Require Memorization? A Short Tale about a Long Tail. Feldman. SIGACT 2020.
[5] Characterizing Structural Regularities of Labeled Data in Overparameterized Models. Jiang et al. ICML 2021.
[6] Information Complexity of Stochastic Convex Optimization: Applications to Generalization and Memorization. Attias et al. ICML 2024.
[7] Extracting Training Data from Large Language Models. Carlini et al. USENIX 2021.
[8] What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation. Feldman and Zhang. NeurIPS 2020.
[9] Measuring Forgetting of Memorized Training Examples. Jagielski et al. 2023.
[10] An Empirical Study of Example Forgetting During Deep Network Learning. Toneva et al. ICLR 2019.

# Implications of memorization: privacy attacks

Membership Inference Attacks (MIAs) predict whether a particular example was used to train a given model (Shokri et al, 2017; Carlini et al, 2022; Ye et al, 2022)

Model inversion and data extraction attacks aim to recover information about training data from models (Nguyen et al, 2023; Carlini et al, 2021)

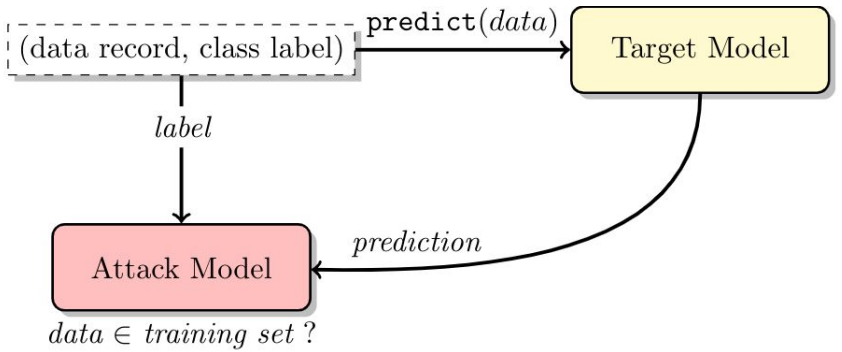



Figure from (Shokri et al, 2023)

Membership Inference Attacks Against Machine Learning Models. Shokri et al. SP 2017.
Membership Inference Attacks From First Principles. Carlini et al. SP 2022.
Enhanced Membership Inference Attacks Against Machine Learning Models. Ye et al. SIGSAC 2022.
Re-thinking Model Inversion Attacks Against Deep Neural Networks. Nguyen et al. CVPR 2023.
Extracting Training Data from Large Language Models. Carlini et al. USENIX 2021.

# Data extraction in large-scale models

(Carlini et al, 2021) demonstrate that **LLMs memorize and can be prompted to extract verbatim text sequences from the training data**

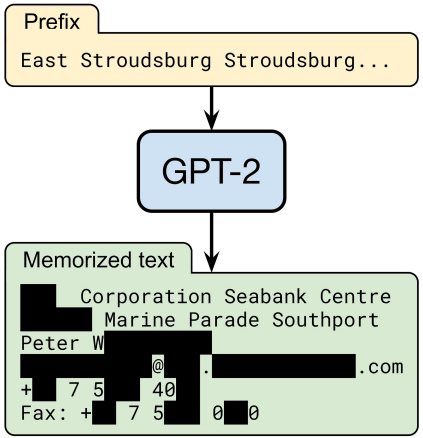



Figure from (Carlini et al, 2021)

(Carlini et al, 2023) demonstrate that diffusion models **memorize and regenerate individual training examples**

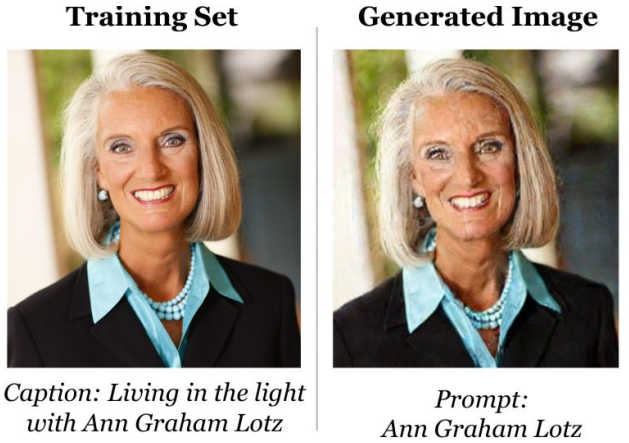



Figure from (Carlini et al, 2023)

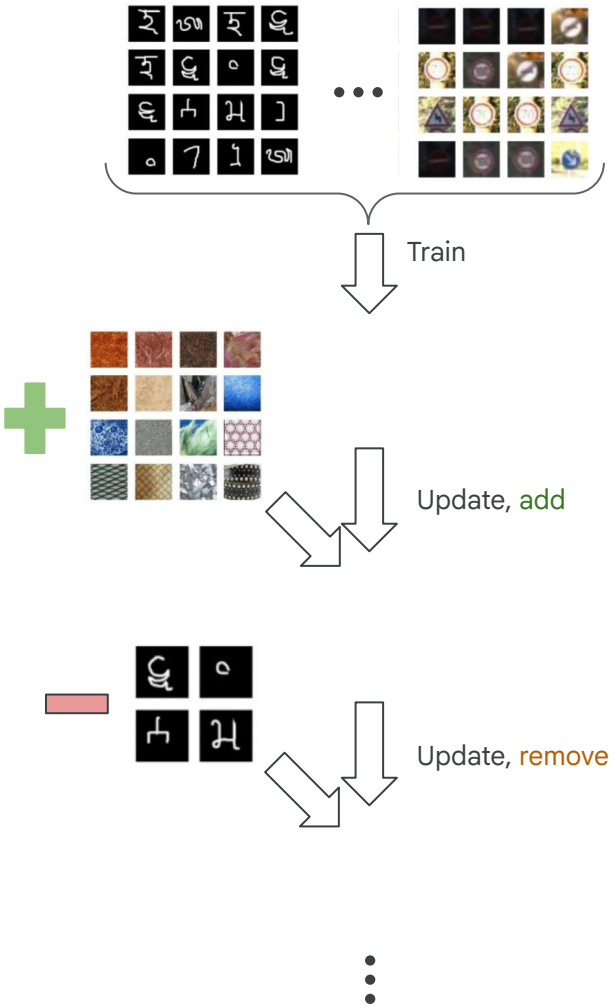
Extracting Training Data from Large Language Models. Carlini et al. USENIX 2021.
Extracting Training Data from Diffusion Models. Carlini et al. USENIX 2023.

# Learning and unlearning **at large**

Recent models are **increasingly large** and **trained on more and more data**. This has two consequences:

1. It is increasingly computationally expensive to train
2. It is increasingly likely that the underlying training data changes
    - new data becomes available
    - newly identified tasks
    - protect privacy
    - facts about the world have changed
    - labelling bugs identified after the fact
    - poisoned data discovered
    - data licenses may expire

**The dream (updated!)**: build lifelong systems capable of increasingly rapid adaptation for both addition **and removal** of "knowledge", data points, or tasks



# Unlearning is receiving an explosion of attention!

**The unlearning competition at NeurIPS'23 received submissions from > 1K teams from around the world!**

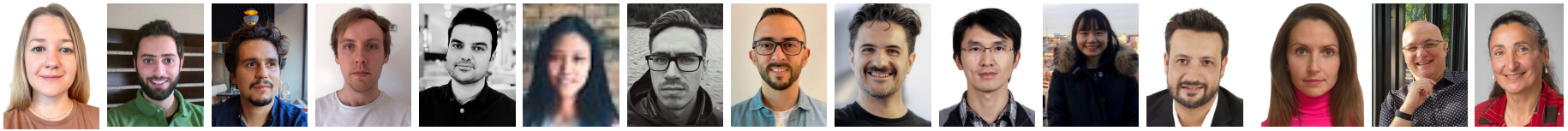
**Participation**

5,046 Entrants
1,336 Participants
1,188 Teams
1,921 Submissions

GOOGLE · RESEARCH CODE COMPETITION · 8 MONTHS AGO

## NeurIPS 2023 - Machine Unlearning

Erase the influence of requested samples without hurting accuracy

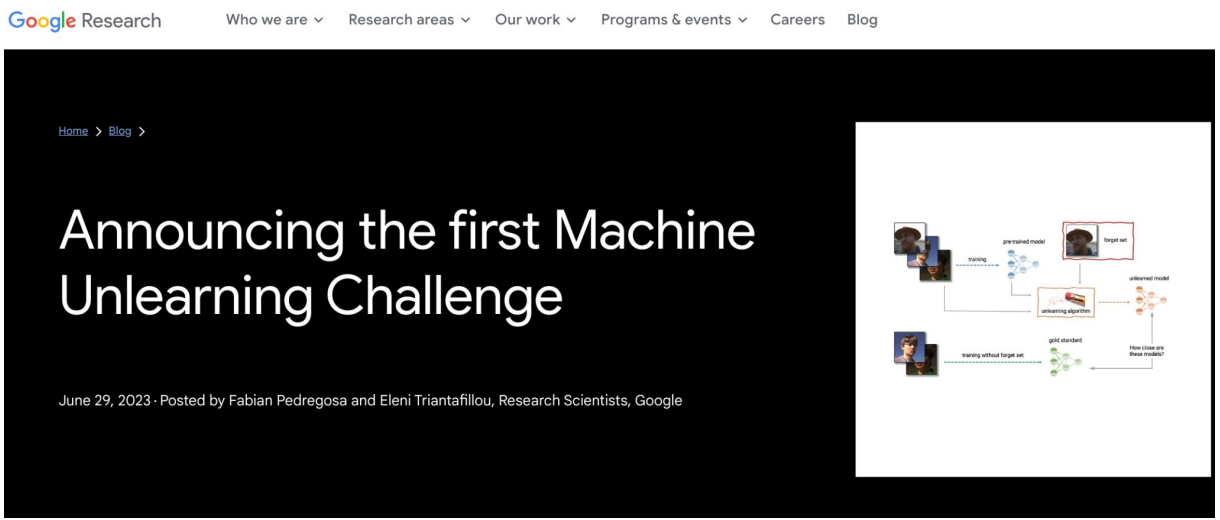


https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/overview
https://unlearning-challenge.github.io/

# Unlearning is receiving an explosion of attention!

**The unlearning competition at NeurIPS'23 received submissions from > 1K teams from around the world!**

The blog post that Fabian and I co-authored was **the third most read Google Research blog post of 2023!**




https://research.google/blog/announcing-the-first-machine-unlearning-challenge/

# Unlearning: intentional and targeted forgetting

Image credits: Dall-E and toppng.com

The unlearning literature identifies several motivating use-cases, including:

**Increase privacy.**

By removing data that poses privacy concerns that may be identified post training.

**Improve accuracy or compliance.**

By removing data that is

- Outdated (no longer accurate)
- Mislabelled or poisoned (and identified only post-training)
- No longer accessible (e.g. license expired)

**Improve forward transfer.**

By removing "knowledge" that causes interference with learning new tasks

**Improve safety (?).**

- Remove biases, reduce toxicity, prevent harmful outputs

But can all these problems really be solved with the same approaches, and can we measure success using the same metrics??

Machine unlearning. Bourtoule et al. SP 2021.
Towards Adversarial Evaluations for Inexact Machine Unlearning. Goel et al.
Corrective Machine Unlearning. Goel et al. 2024.
Model Sparsity Can Simplify Machine Unlearning. Jia et al. NeurIPS 2023.
Towards safer large language models through machine unlearning. Liu et al. 2024

# In the remainder of this talk...

1

Formal definition of machine unlearning and a categorization of methods

2

The NeurIPS'23 unlearning competition: evaluation and findings

3

What makes unlearning hard (and what to do about it)?

4

Discussion, limitations, open questions

5

Connections with lifelong learning and opportunities

# Defining machine unlearning

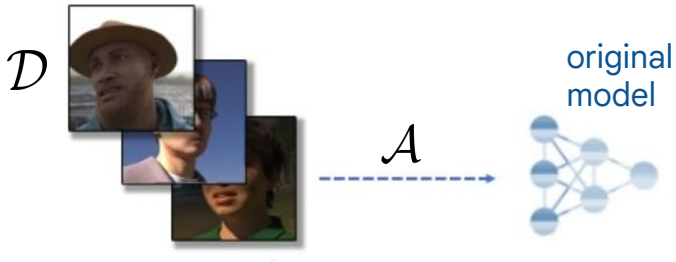



Figure from our blog post

# Defining machine unlearning

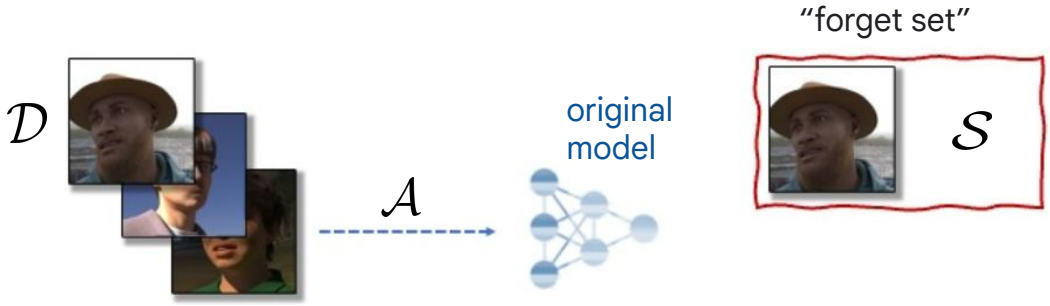



Figure from our blog post

# Defining machine unlearning

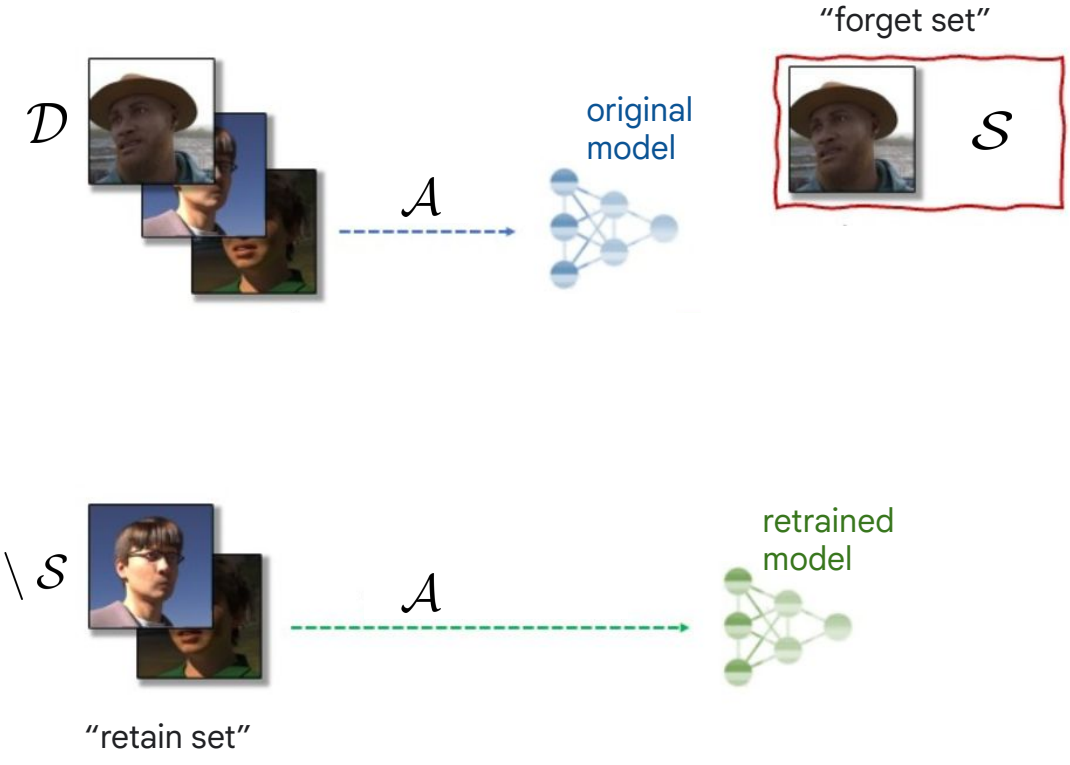



Figure from our blog post

# Defining machine unlearning

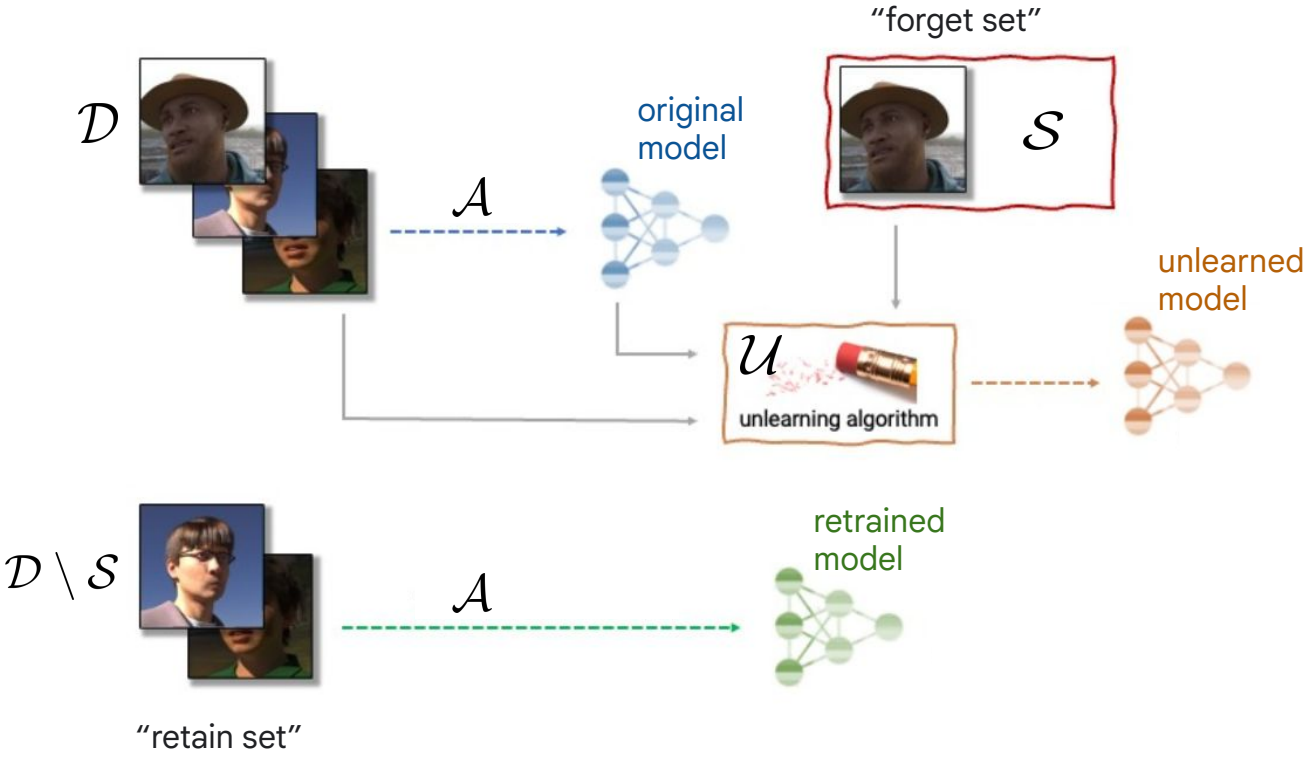



Figure from our blog post

# Defining machine unlearning

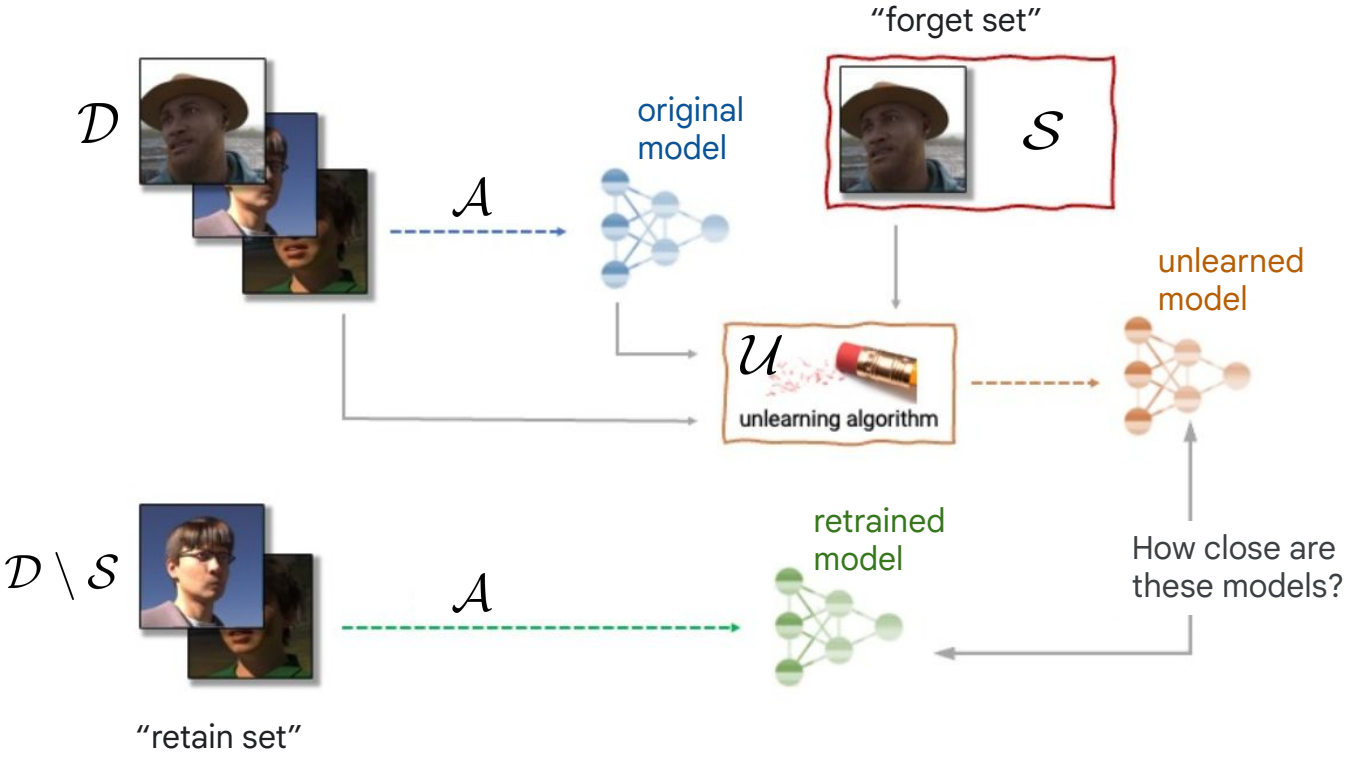
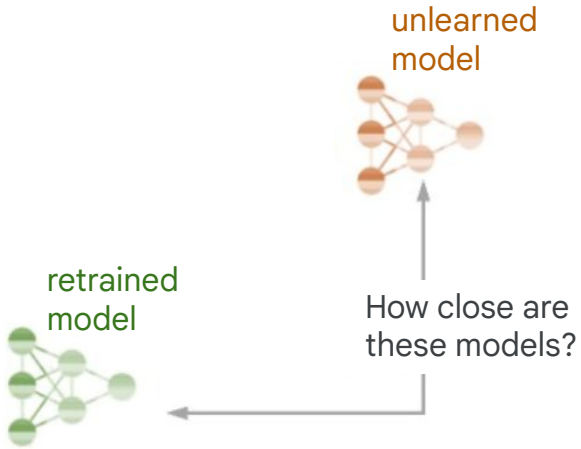



Figure from our blog post

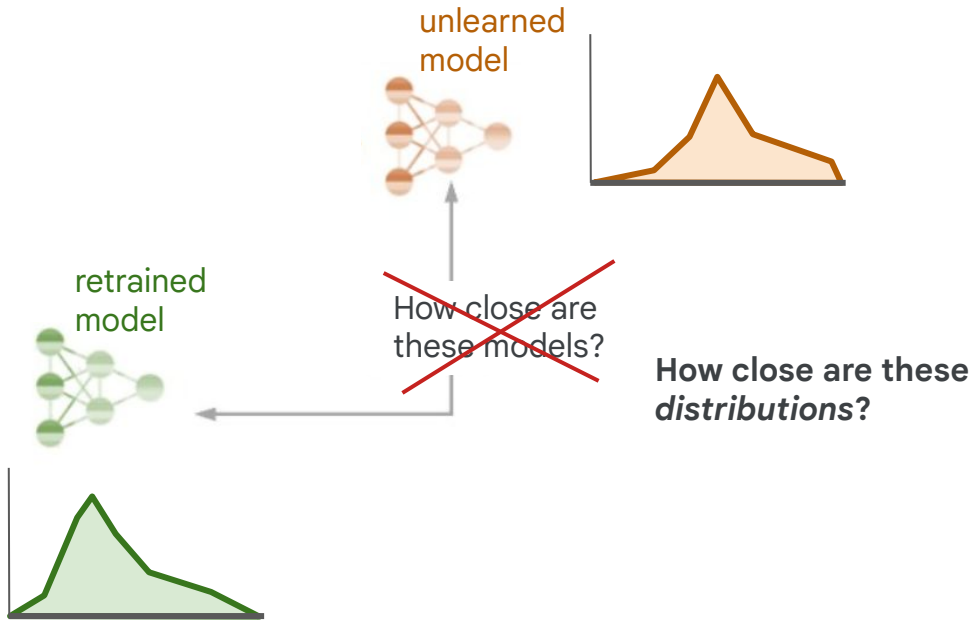
# Defining machine unlearning

- According to this viewpoint, the closer the unlearned and retrained models are, the better
- But how do we measure closeness?



# Defining machine unlearning

- According to this viewpoint, the closer the unlearned and retrained models are, the better
- But how do we measure closeness?



# Formal definition

**Definition 1.1.** $(\varepsilon, \delta)$**-unlearning.** For a fixed dataset $\mathcal{D}$, forget set $\mathcal{S} \subseteq \mathcal{D}$, and a randomized learning algorithm $\mathcal{A}$, an unlearning algorithm $\mathcal{U}$ is $(\varepsilon, \delta)$-unlearning with respect to $(\mathcal{D}, \mathcal{S}, \mathcal{A})$ if for all $R \subseteq \mathcal{R}$ where $\mathcal{R}$ denotes the output space, in this case the space of model parameters $\theta$, we have:

$$\Pr[\mathcal{A}(\mathcal{D} \setminus \mathcal{S}) \in R] \leq e^{\varepsilon} \Pr[\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D}) \in R] + \delta, \quad \text{and}$$
$$\Pr[\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D}) \in R] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathcal{D} \setminus \mathcal{S}) \in R] + \delta.$$

When $\delta, \mathcal{E}$ are very small, the distributions of unlearning and retraining are very close (successful unlearning)

Making AI forget you: data deletion in machine learning. Ginart et al. NeurIPS 2019.
Remember What You Want to Forget: Algorithms for Machine Unlearning. Sekhari et al. NeurIPS 2021.
Descent-to-delete: Gradient-based methods for machine unlearning. Neel et al. 2021. PMLR 2021.

# Categorization of unlearning methods

**Exact unlearning:** $\varepsilon = \delta = 0$

- Fully eliminate the influence of the forget set

**Approximate unlearning:** $\varepsilon, \delta > 0$

- *Reduce* the influence of the forget set

$$\Pr[\mathcal{A}(\mathcal{D} \setminus \mathcal{S}) \in R] \leq e^{\varepsilon} \Pr[\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D}) \in R] + \delta, \quad \text{and}$$
$$\Pr[\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D}) \in R] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathcal{D} \setminus \mathcal{S}) \in R] + \delta.$$

Machine unlearning. Bourtoule et al. 2019.
Making AI forget you: data deletion in machine learning. Ginart et al. 2019.

# Categorization of unlearning methods

**Exact unlearning:** $\varepsilon = \delta = 0$

- Fully eliminate the influence of the forget set

- For deep neural networks, this is based on *retraining from scratch* (parts of) the model

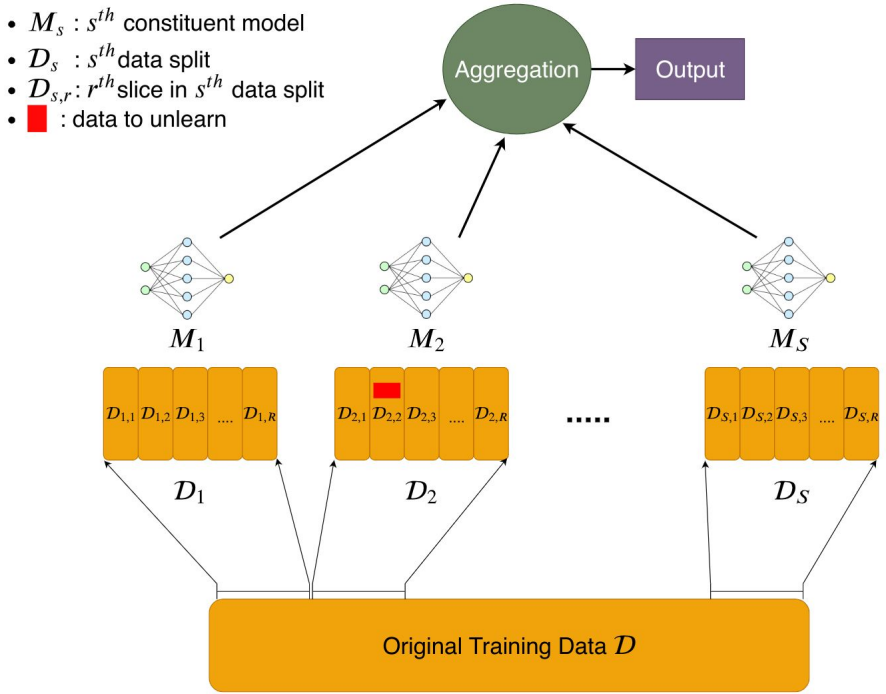



Figure from (Bourtoule et al, 2019)

Machine unlearning. Bourtoule et al. SP 2021.

# Categorization of unlearning methods

**Exact unlearning:** $\varepsilon = \delta = 0$

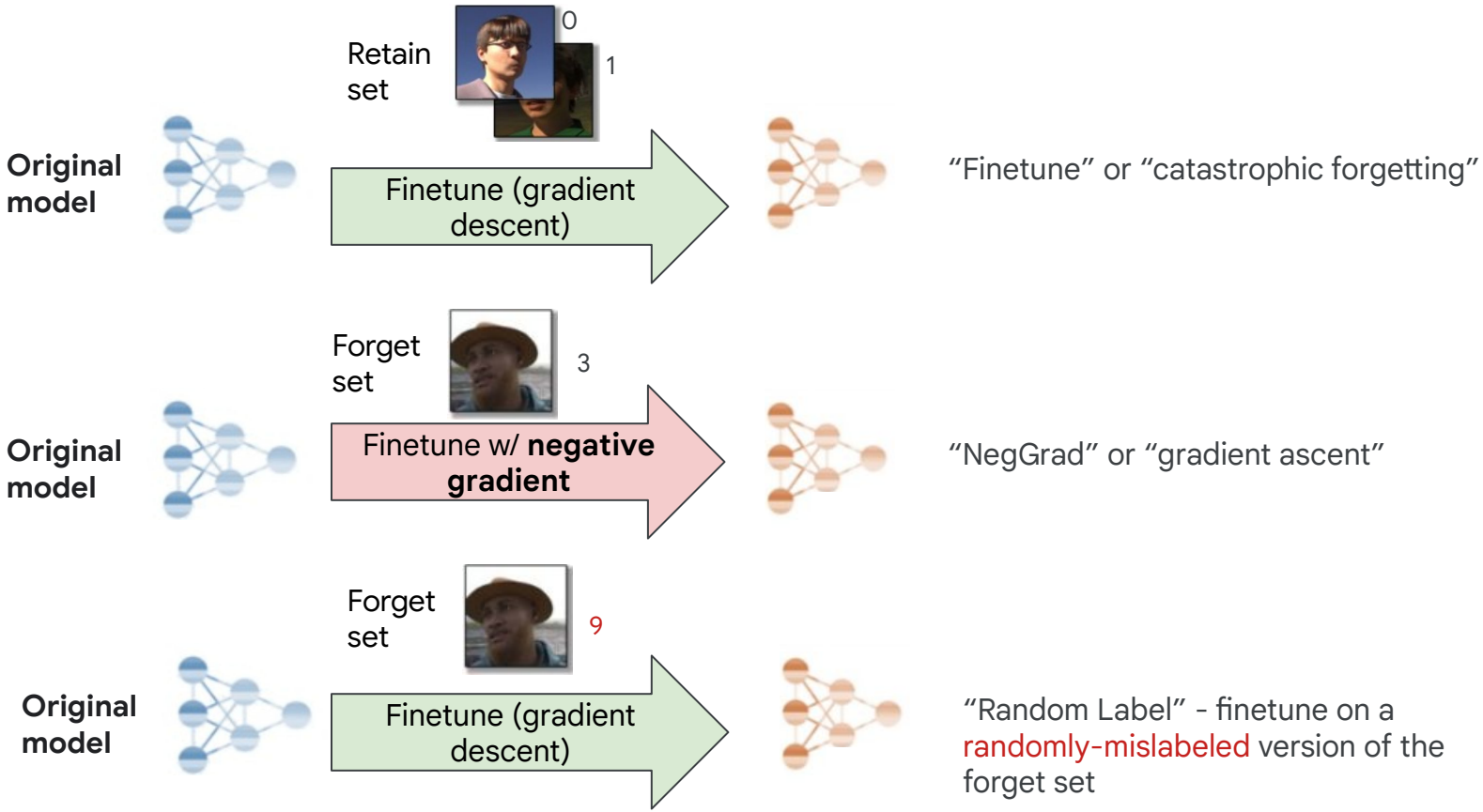
- Fully eliminate the influence of the forget set
- For deep neural networks, this is based on *retraining from scratch* (parts of) the model

**Approximate unlearning:** $\varepsilon, \delta > 0$

- *Reduce* the influence of the forget set
- Guarantees for convex problems (Guo et al, 2019; Sekhari et al, 2021; Neel et al, 2021); and some for nonconvex, but under a constrained setup (Chien et al, 2024)
- … and a plethora of creative methods with no guarantees but that "work well" in practice

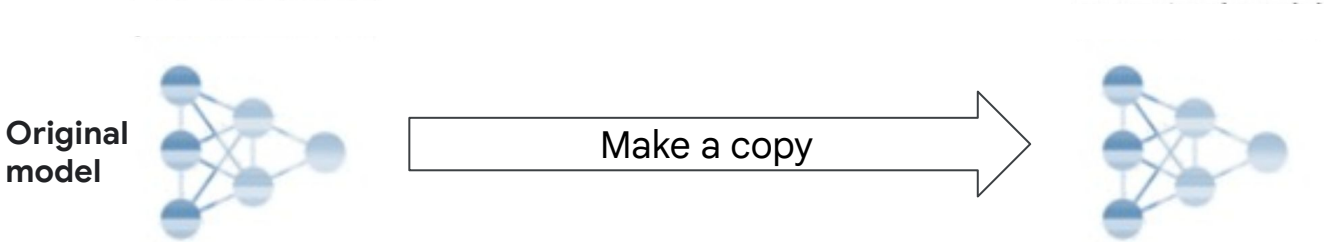
Certified Data Removal from Machine Learning Models. Guo et al. 2019.
Remember What You Want to Forget: Algorithms for Machine Unlearning. Sekhari et al. NeurIPS 2021.
Descent-to-delete: Gradient-based methods for machine unlearning. Neel et al. 2021. PMLR 2021.
Langevin Unlearning: A New Perspective of Noisy Gradient Descent for Machine Unlearning. Chien et al. 2024.
Model Sparsity Can Simplify Machine Unlearning. Jia et al. NeurIPS 2023.
SalUn : Empowering Machine Unlearning Via Gradient-Based Weight Saliency In Both Image Classification And Generation. Fan et al. ICLR 2024.
Approximate Data Deletion from Machine Learning Models. Izzo et al. AISTATS 2021.
Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. Golatkar et al. CVPR 2020.

# Approximate unlearning: some simple ideas



# SCalable Remembering and Unlearning unBound (**SCRUB**)

- We cast the problem into a teacher-student formulation
- The student is initialized by the “all knowing” teacher and then (selectively) forgets




Towards Unbounded Machine Unlearning. Kurmanji et al. NeurIPS 2023.

# SCalable Remembering and Unlearning unBound (**SCRUB**)

- We cast the problem into a teacher-student formulation
- The student is initialized by the “all knowing” teacher and then (selectively) forgets

**Teacher**

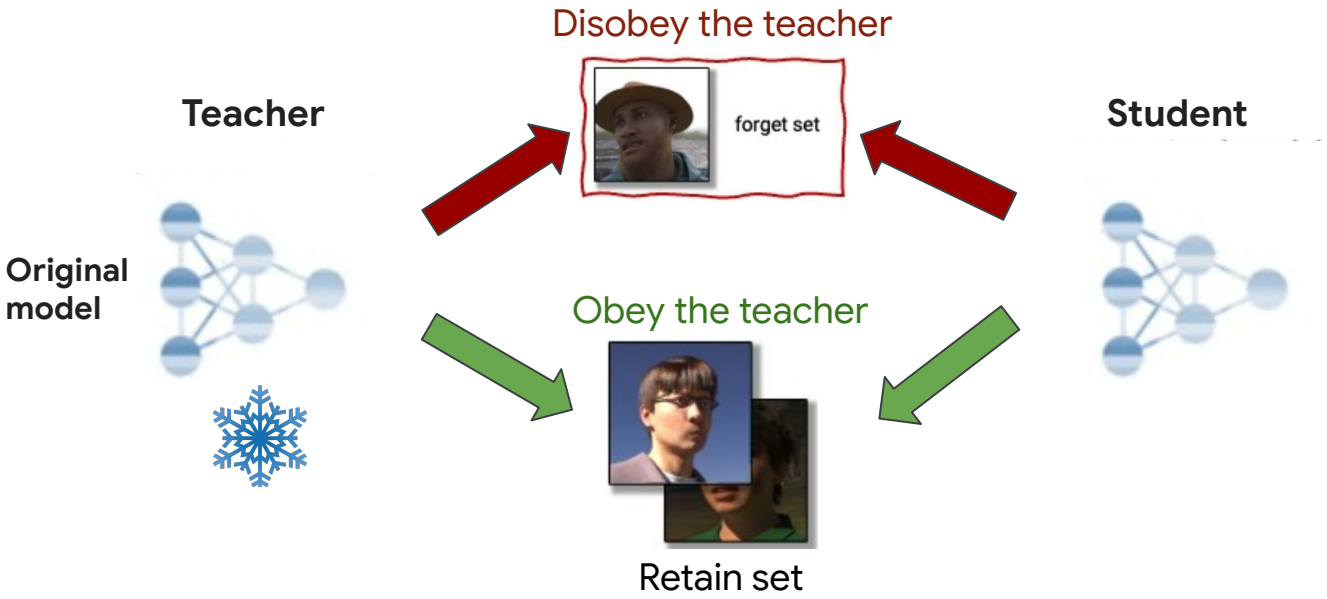
**Original model**

**Student**

Towards Unbounded Machine Unlearning. Kurmanji et al. NeurIPS 2023.

# SCalable Remembering and Unlearning unBound (**SCRUB**)

- We cast the problem into a teacher-student formulation
- The student is initialized by the “all knowing” teacher and then (selectively) forgets




Towards Unbounded Machine Unlearning. Kurmanji et al. NeurIPS 2023.

# Trade-offs of unlearning methods

**Exact unlearning:** $\varepsilon = \delta = 0$

✅ Fully eliminate the influence of the forget set

❌ Computationally expensive

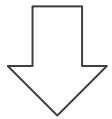
❌ May require architectures with poorer accuracy

**Approximate unlearning:** $\varepsilon, \delta > 0$

✅ Can be faster

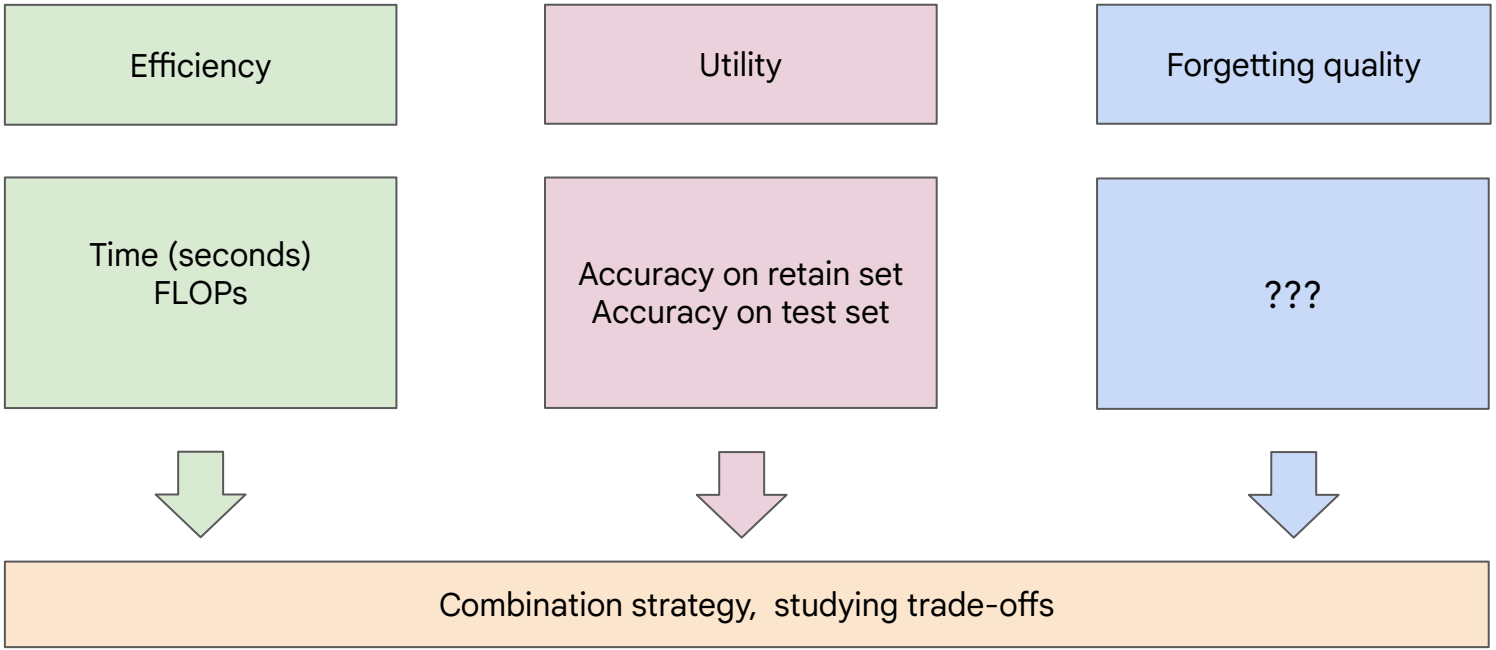
✅ Can have better accuracy

❌ **Often no guarantees**



So, how can we evaluate these methods?

# Evaluation of unlearning methods



# The first NeurIPS machine unlearning competition

**Increase visibility.**

We had > 1K teams from around the world participating.

**Generate novel methods.**

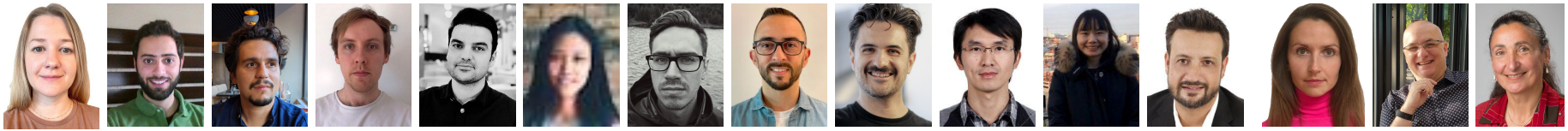
A plethora of creative solutions were developed, with different properties and trade-offs.

**Standardize evaluation methods.**

We developed a rigorous evaluation framework based on a formal definition of unlearning.

**Setup.**

- An age classifier (ResNet-18) was trained using natural images of people's faces (from the CASIA-SURF dataset)
- A subset of users request their data to be deleted
- Goal: *efficient* unlearning of that user data



CASIA-SURF: A large-scale multi-modal benchmark for face anti-spoofing. Zhang et al.

# Evaluation framework: overview

**Efficiency.**

We impose a hard threshold on runtime (approx. 20% of retraining)

**Utility.**

Accuracy on the retain and test sets

**Forgetting quality / unlearning efficacy.**

We developed a rigorous evaluation framework for estimating a forgetting score $\mathcal{F}$ (higher is better)

**Aggregation.**

Accuracy on retain set

Accuracy on test set

$$\text{Final score} = \mathcal{F} \times \frac{\text{Acc}(\mathcal{D} \setminus \mathcal{S}, \mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D}))}{\text{Acc}(\mathcal{D} \setminus \mathcal{S}, \mathcal{A}(\mathcal{D} \setminus \mathcal{S}))} \times \frac{\text{Acc}(\mathcal{D}_{test}, \mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D}))}{\text{Acc}(\mathcal{D}_{test}, \mathcal{A}(\mathcal{D} \setminus \mathcal{S}))}$$

This strategy penalizes unlearning if it yields poorer accuracy on retain or test compared to retraining-from-scratch

Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# How should we measure forgetting quality?

**Definition 1.1.** $(\varepsilon, \delta)$**-unlearning.** For a fixed dataset $\mathcal{D}$, forget set $\mathcal{S} \subseteq \mathcal{D}$, and a randomized learning algorithm $\mathcal{A}$, an unlearning algorithm $\mathcal{U}$ is $(\varepsilon, \delta)$-unlearning with respect to $(\mathcal{D}, \mathcal{S}, \mathcal{A})$ if for all $R \subseteq \mathcal{R}$ where $\mathcal{R}$ denotes the output space, in this case the space of model parameters $\theta$, we have:

$$\Pr[\mathcal{A}(\mathcal{D} \setminus \mathcal{S}) \in R] \leq e^{\varepsilon} \Pr[\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D}) \in R] + \delta, \quad \text{and}$$
$$\Pr[\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D}) \in R] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathcal{D} \setminus \mathcal{S}) \in R] + \delta.$$

For a fixed $\delta$, we say that an algorithm $\mathcal{U}_1$ is better than $\mathcal{U}_2$ if it yields a smaller $\varepsilon$.

But how can we measure $\varepsilon$?

# Let's play a game!

retrained model

unlearned model

hide which one you picked

auditor / decision rule

retrained model

unlearned model

Repeat O(1000) times, measure

- False Positive Rate (FPR)
- False Negative Rate (FNR)

What does the failure of the auditor (FPR, FNR) tell us about how close the retrained and unlearned distributions are?

# Hypothesis-testing interpretation

**Theorem 1.2.** *Let $\mathcal{U}$ be an $(\varepsilon, \delta)$-unlearning algorithm for $(\mathcal{D}, \mathcal{S}, \mathcal{A})$. Let $X$ be sampled from either $\mathcal{A}(\mathcal{D} \setminus \mathcal{S})$ or $\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D})$. Then the performance of any decision rule that tries to predict if $X$ came from $\mathcal{A}(\mathcal{D} \setminus \mathcal{S})$ or $\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D})$ is governed by*

$$FPR + e^{\varepsilon} FNR \geq 1 - \delta, \quad \text{and}$$
$$FNR + e^{\varepsilon} FPR \geq 1 - \delta.$$

When $\delta$, $\varepsilon$ are very small, no decision rule can separate the two distributions with FPR and FNR both small.

This allows to estimate $\varepsilon$ via estimates of FPR, FNR obtained by instantiated “attacks”

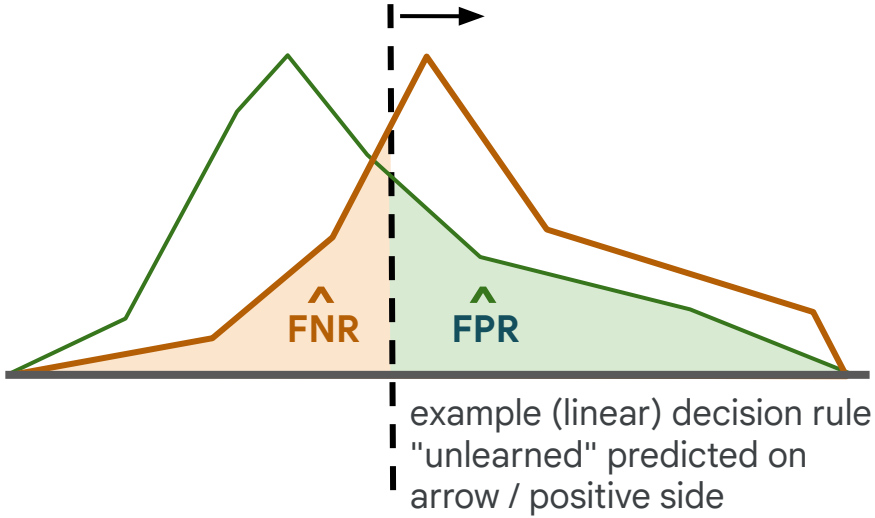
The composition theorem for differential privacy. Kairouz et al. ICML 2015.

# Hypothesis-testing interpretation

For a particular decision rule, we can estimate $\hat{\varepsilon}$ from the estimated FPR and FNR:

$$\hat{\varepsilon} = \max\left\{\log\frac{1-\delta-\hat{\mathrm{FPR}}}{\hat{\mathrm{FNR}}}, \log\frac{1-\delta-\hat{\mathrm{FNR}}}{\hat{\mathrm{FPR}}}\right\}$$

In practice, we care about the estimated $\hat{\varepsilon}$ under the *strongest possible* decision rule / "attack".




The prediction problem of separating "unlearned" from "retrained" is closely related to **Membership Inference Attacks (MIAs)** that predict whether an example was used during training.

The composition theorem for differential privacy. Kairouz et al. ICML 2015.
Membership inference attacks against machine learning models. Shokri et al. SP 2017.

# Estimating forgetting quality $\mathcal{F}$

We would like to estimate $\hat{\varepsilon}$ via the FPR and FNR of "attacks" that try to separate the distributions.

$$\hat{\varepsilon} = \max\left\{\log\frac{1-\delta-\hat{\text{FPR}}}{\hat{\text{FNR}}}, \log\frac{1-\delta-\hat{\text{FNR}}}{\hat{\text{FPR}}}\right\}$$

But there are two difficulties:

1. $\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D})$ and $\mathcal{A}(\mathcal{D} \setminus \mathcal{S})$ are **high-dimensional** distributions (weight space)
2. Some examples are harder to unlearn than others; and we want to capture this **granularity**

We propose a two-step procedure:

1. **Compute *per-example*** $\hat{\varepsilon}$

   Operate on (scalar) *outputs* of models coming from $\mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D})$ and $\mathcal{A}(\mathcal{D} \setminus \mathcal{S})$ when given as inputs examples from the forget set

2. **Aggregate across examples**

   Generate an overall estimate of forgetting quality for all of $\mathcal{S}$

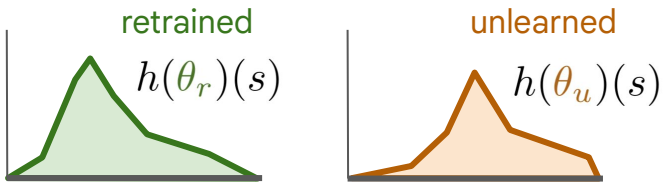
Membership inference attacks from first principles. Carlini et al. 2021.
Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# Estimating $\varepsilon$ for a specific example s

1. **Compute the retrained and unlearned distributions of (scalar) output for s**

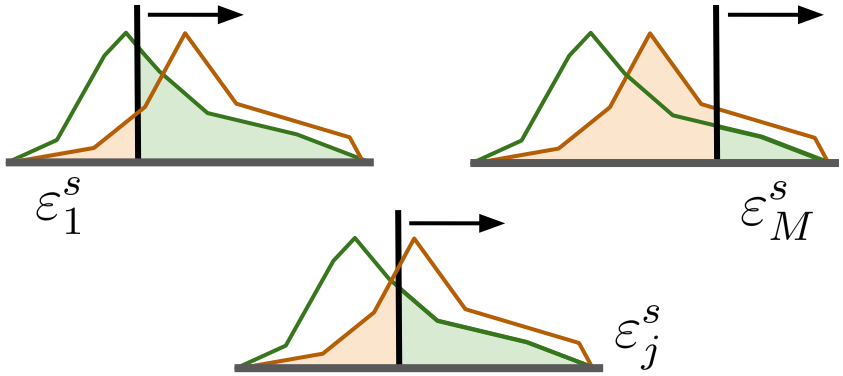
$$h(\theta_r)(s), \text{ where } \theta_r = \mathcal{A}(\mathcal{D} \setminus \mathcal{S})$$
$$h(\theta_u)(s), \text{ where } \theta_u = \mathcal{U}(\mathcal{A}(\mathcal{D}), \mathcal{S}, \mathcal{D})$$

$h$: logit-scaling of the softmax probability of the correct class for example s




2. **Run M "attacks", estimate $\varepsilon$ of each**

Example: "single-threshold attacks"



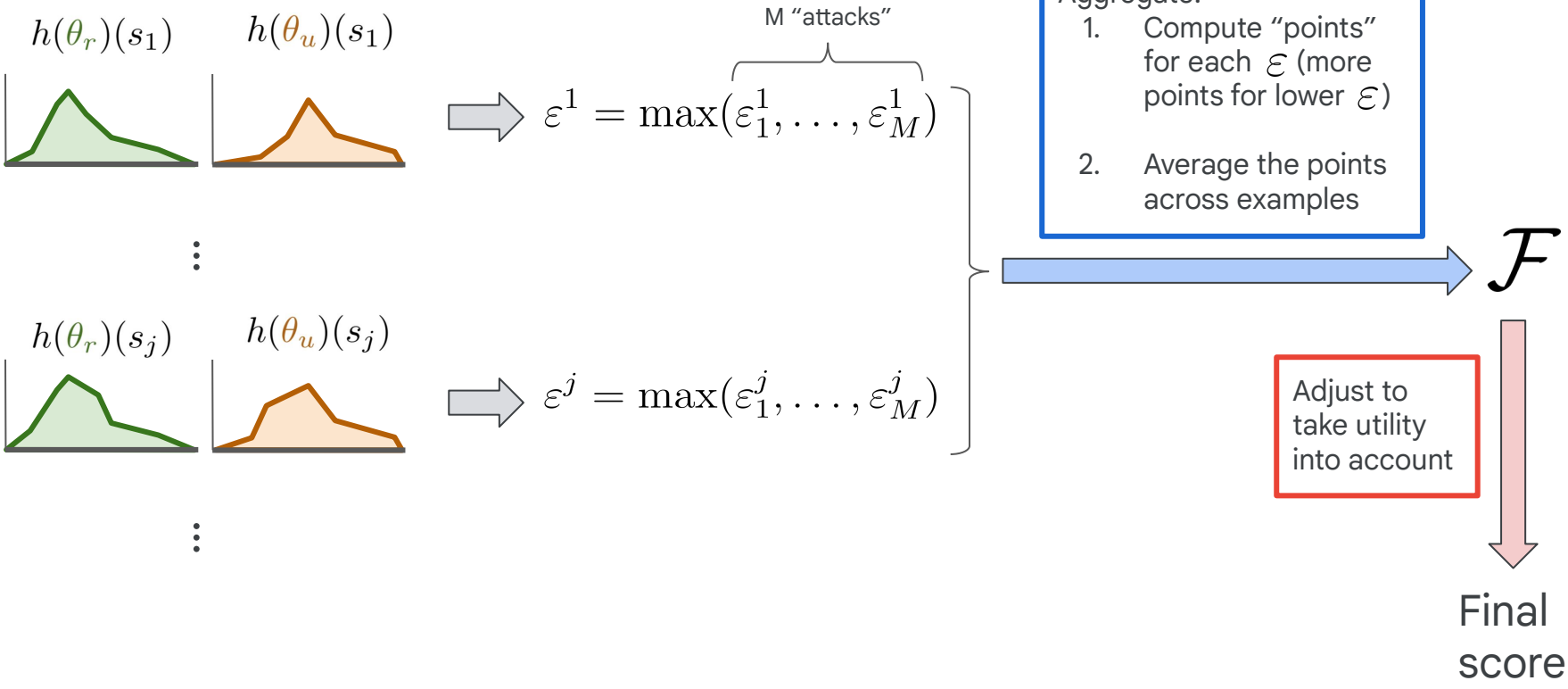

3. **Keep the largest $\varepsilon$ for example s**

i.e. the $\varepsilon$ of the strongest attack

$$\varepsilon^s = \max(\varepsilon_1^s, \ldots, \varepsilon_M^s)$$


Membership inference attacks from first principles. Carlini et al. 2021.
Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# Putting it all together: computing the final score

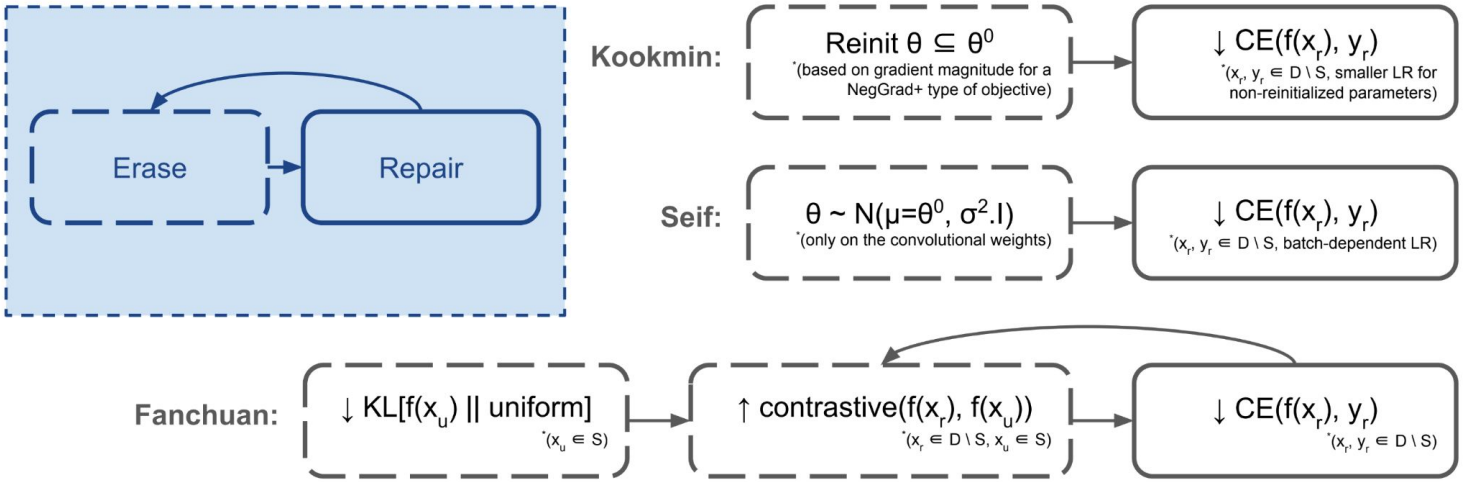



Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# Top unlearning algorithms

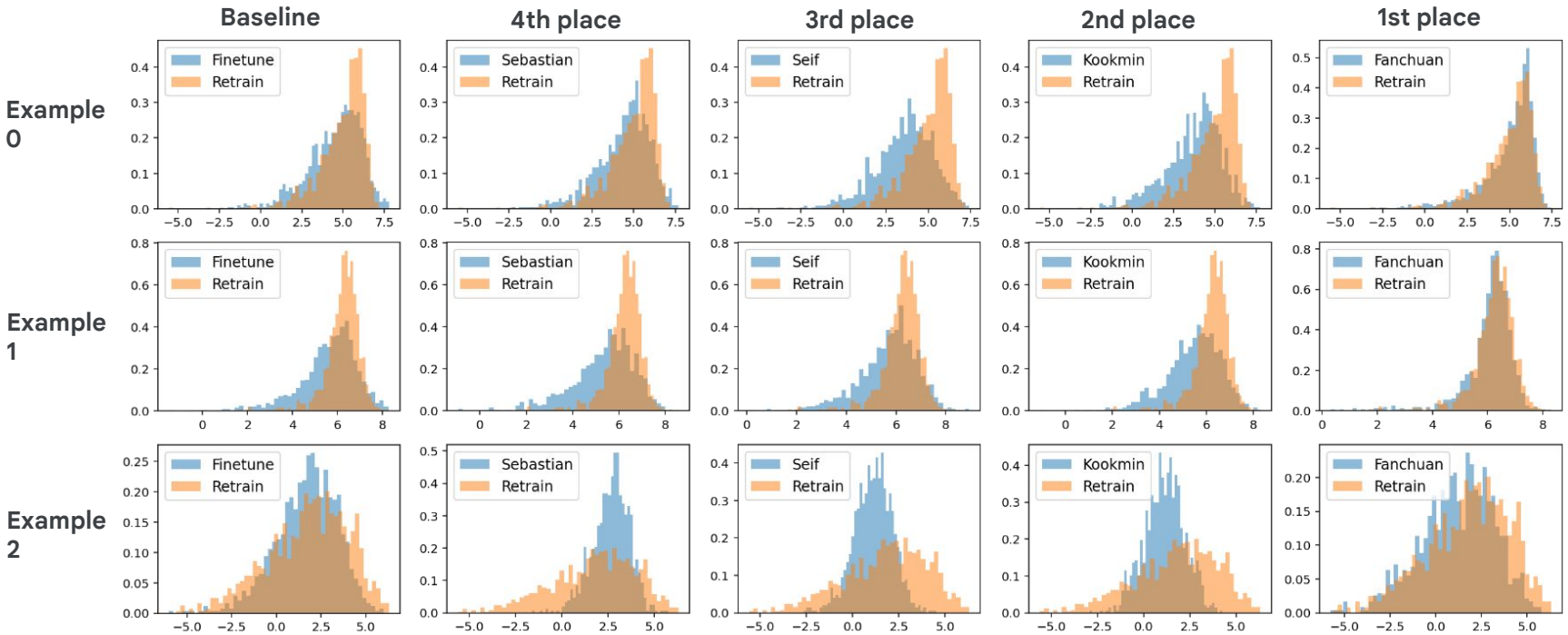
**Baseline**: “Finetune” – continues to train the original model on the retain set only

**Top competition entries** can be seen as comprised of a “erase” followed by a “repair” phase




Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

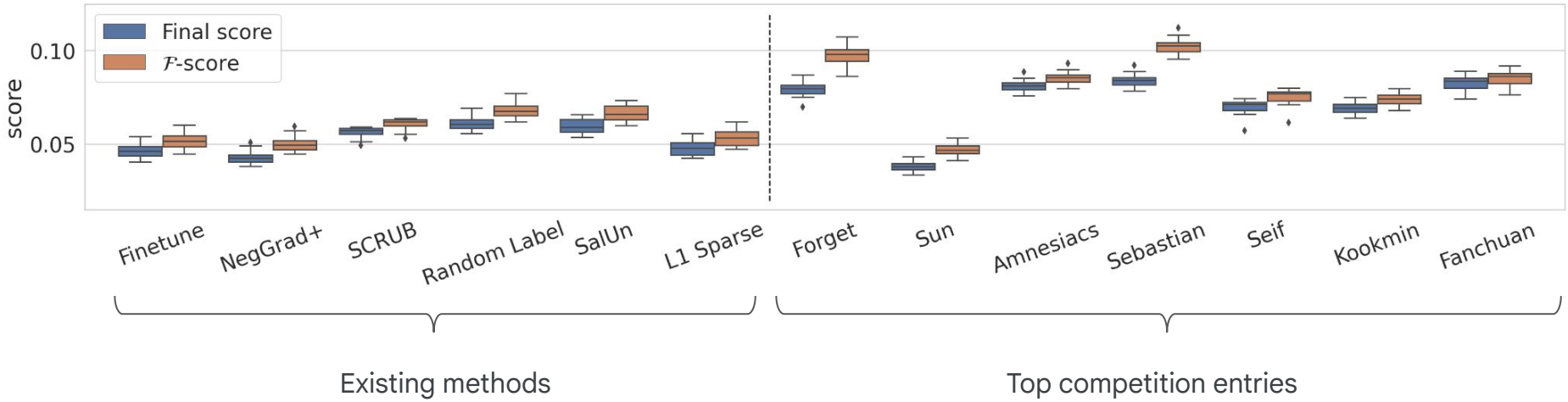
# Top unlearning algorithms: unlearned versus retrained distributions of our test statistic




Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# Are we making progress?

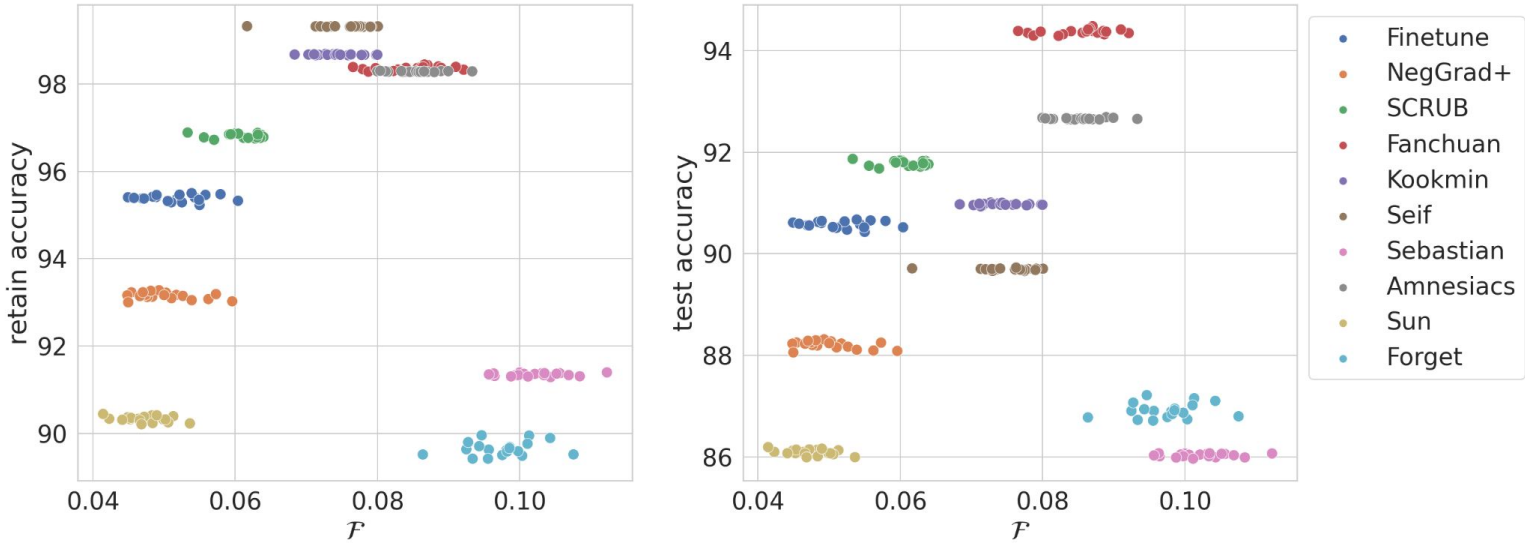
- Top competition methods improve upon previous ones **according to this metric**
  - What are the trade-offs of different methods?
  - What metrics are better suited for different applications?
  - Do we expect algorithms developed for one metric to work well on another?




Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# Different trade-off profiles between forgetting quality, retain and test accuracy

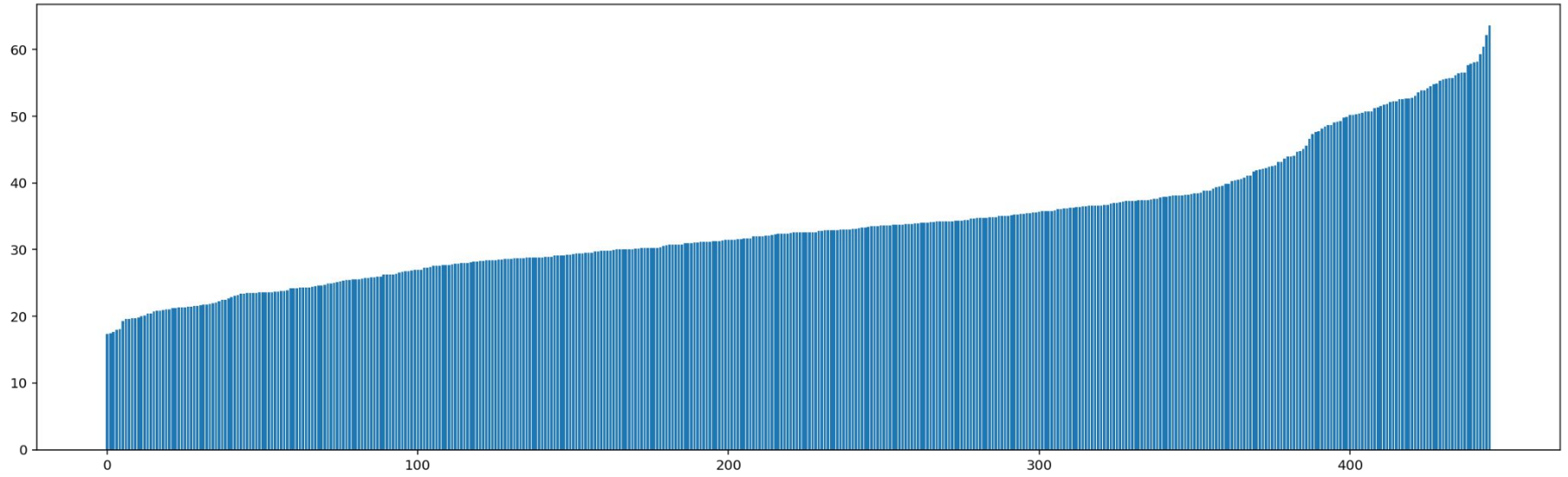
- These profiles may inform which algorithm to choose based on application-specific priorities
- But many open questions remain regarding other “side-effects” of different methods




Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# Example difficulty: do algorithms agree with each other?

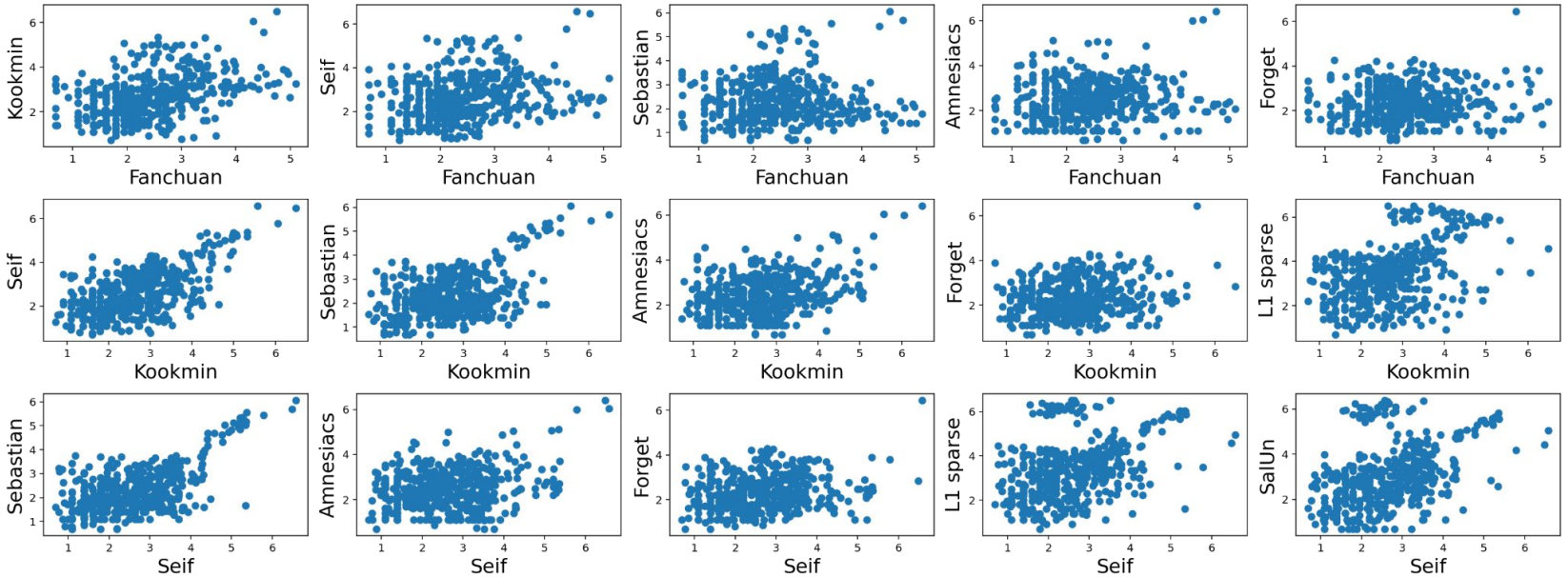
For each forget example, we compute the sum of $\mathcal{E}$'s computed for that example under 12 different algorithms (a high sum indicates that many algorithms find that example hard)




Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# Example difficulty: do algorithms agree with each other?

- Do per-example $\mathcal{E}$'s correlate across pairs of unlearning algorithms?




Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# What makes unlearning hard?

Let's first define a simple metric, that captures the "Tug of War (ToW)" that unlearning requires: achieve **good utility** and good **forgetting quality**

$$\text{ToW}(\theta^u, \theta^r, \mathcal{S}, \mathcal{R}, \mathcal{D}_{test}) = (1 - \text{da}(\theta^u, \theta^r, \mathcal{S})) \cdot (1 - \text{da}(\theta^u, \theta^r, \mathcal{R})) \cdot (1 - \text{da}(\theta^u, \theta^r, \mathcal{D}_{test}))$$

where $\text{da}(\theta^u, \theta^r, \mathcal{D})$ is the difference in the accuracy of the unlearned and retrained models on the given dataset.

Accuracy diff on the **forget set**

Accuracy diff on the **retain set**

Accuracy diff on the **test set**

ToW (higher is better) rewards unlearning algorithms that achieve accuracy close to that of retraining-from-scratch on each of the forget, retain, and test sets

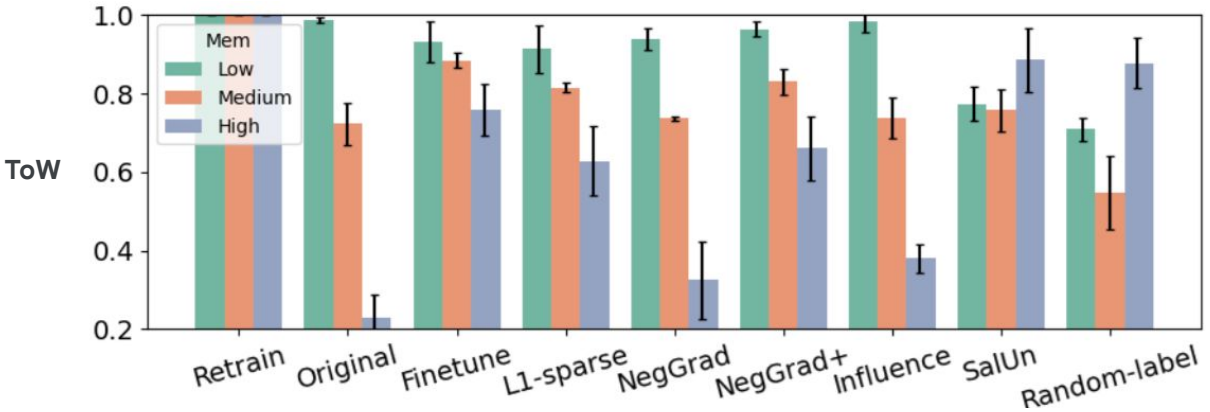
What makes unlearning hard and what to do about it. Zhao et al. 2024.

# A study of interpretable factors

(Fan et al.) pose an optimization problem to discover worst-case forget sets.

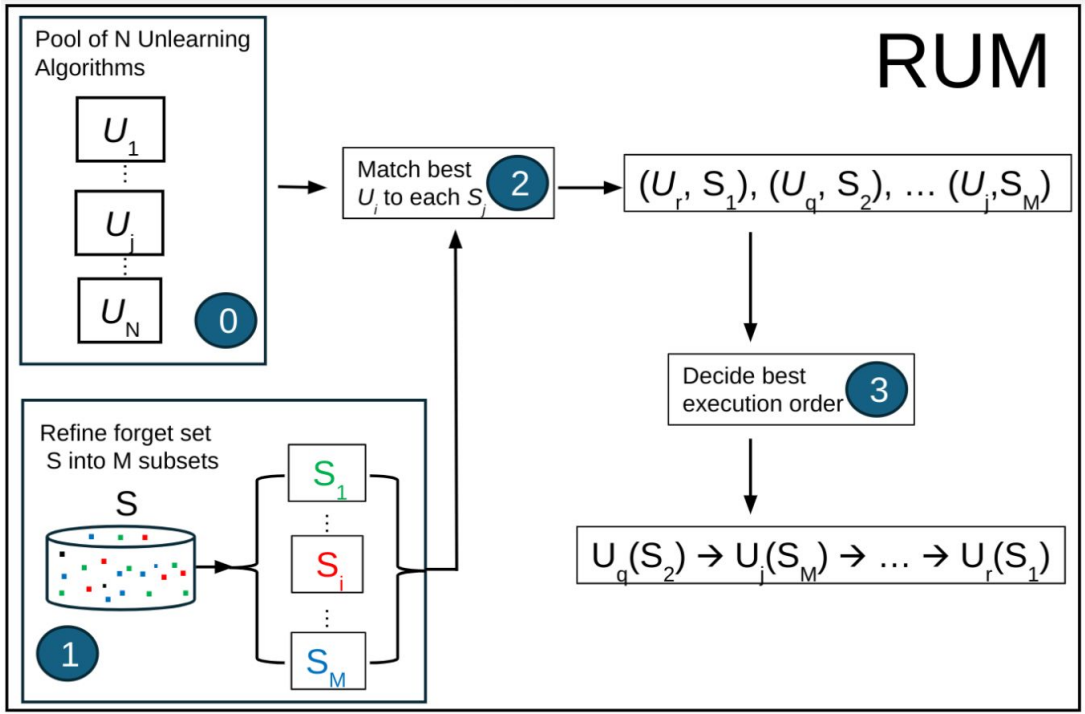
We instead ask: are there **interpretable** factors that affect the difficulty of an unlearning problem?

We find the degree of “label memorization” (**mem**) of the forget set influences algorithm behaviours




What makes unlearning hard and what to do about it. Zhao et al. 2024.
Challenging forgets: Unveiling the worst-case forget sets in machine unlearning. Fan et al. 2024.

# Refined-Unlearning Meta-algorithm (RUM)



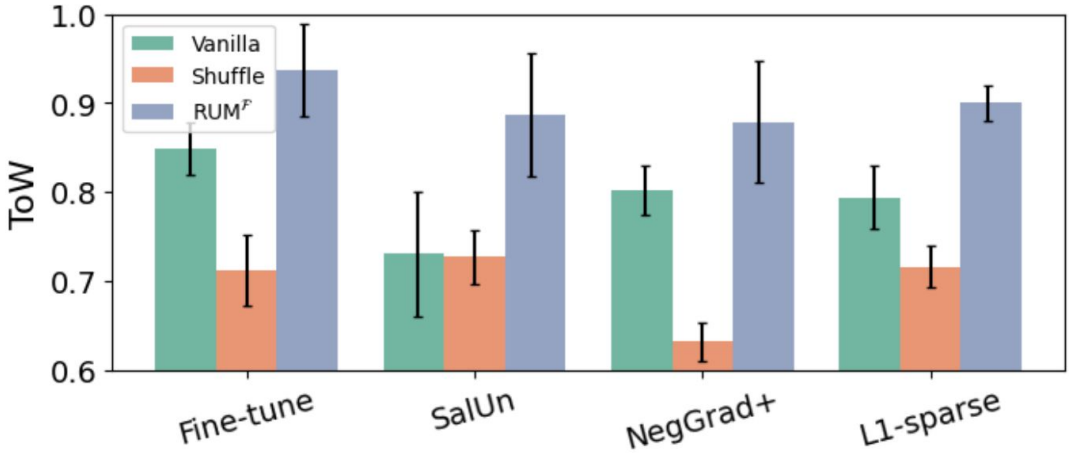

1. Refine the forget set into homogeneous subsets (based on discovered characteristics that affect algorithm behaviours)
2. Choose the best unlearning algorithm per subset
3. Unlearn each subset sequentially

What makes unlearning hard and what to do about it. Zhao et al. 2024.

# Sequentially unlearning homogenized subsets yields improvements

We first explore a simple instantiation of RUM, where a given unlearning algorithm is applied:

1) In one-go, on the entire forget set (**vanilla**)
2) Sequentially, on equal-sized random subsets (**shuffle**); a control experiment
3) Sequentially, on equal-sized subsets *according to mem scores* (**RUM$^F$**)



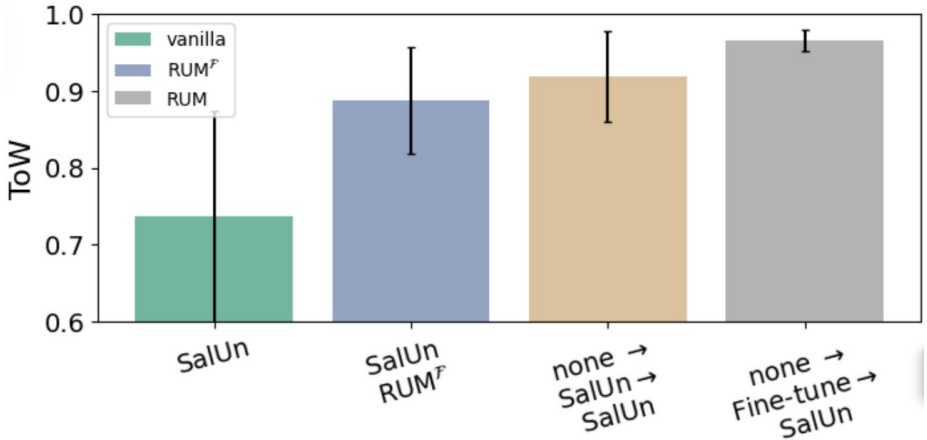

We observe that **applying unlearning in a sequence can significantly improve ToW** if using **RUM$^F$**

What makes unlearning hard and what to do about it. Zhao et al. 2024.

# Can we further improve by picking the best per-subset algorithm?

Picking the best algorithm for each homogenized subset yields further improvements!

Proof-of concept that better scientific understanding of the problem can inform improvements in unlearning pipelines.



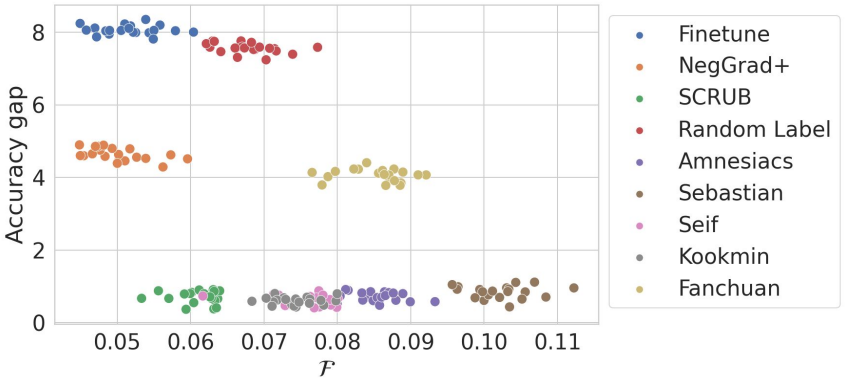

But several questions remain:

1. Why does sequential unlearning help? How can we construct optimal sequences?
2. What is the effect of different orderings of the sequence?
3. How can we operationalize this efficiently?
4. How would these findings differ under other evaluation frameworks?

What makes unlearning hard and what to do about it. Zhao et al. 2024.

# Discussion: on the (in)efficiency of evaluation

- Computing $\mathcal{F}$ is (very) expensive... **Is approximate unlearning *really* cheaper than retraining-from-scratch if it requires (frequent) expensive evaluation?**
  - Can we amortize the cost of evaluation across unlearning problems?
    - Requires unlearning methods that are “robust” across problem settings
  - Are there cheap-to-compute proxies for rigorous metrics?

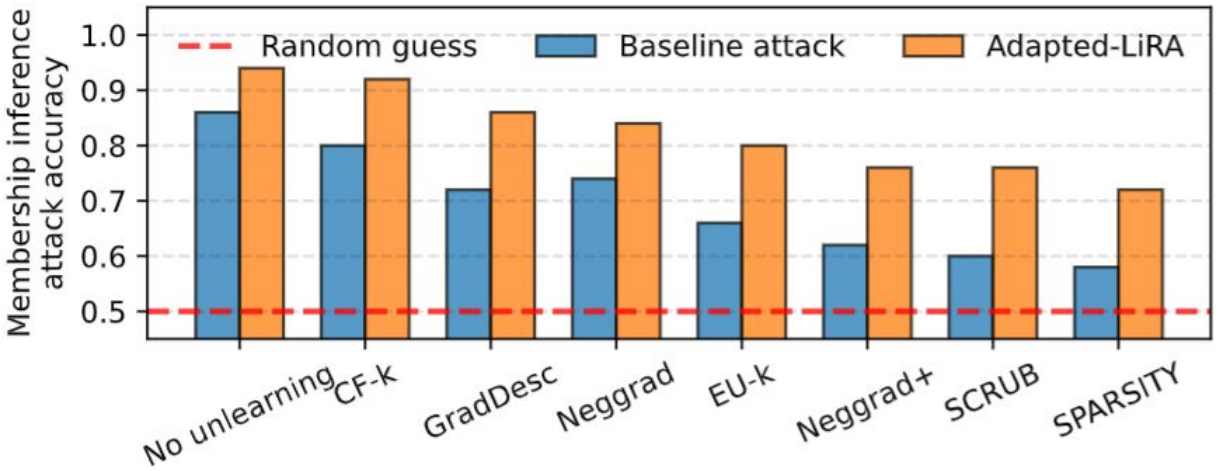



The common metric that measures the forget set *accuracy* relative to that of retrain-from-scratch is not a good proxy for $\mathcal{F}$

Are we making progress in unlearning? Findings from the first NeurIPS competition. Triantafillou et al. 2024.

# Weak attacks overestimate the level of privacy afforded by unlearning algorithms

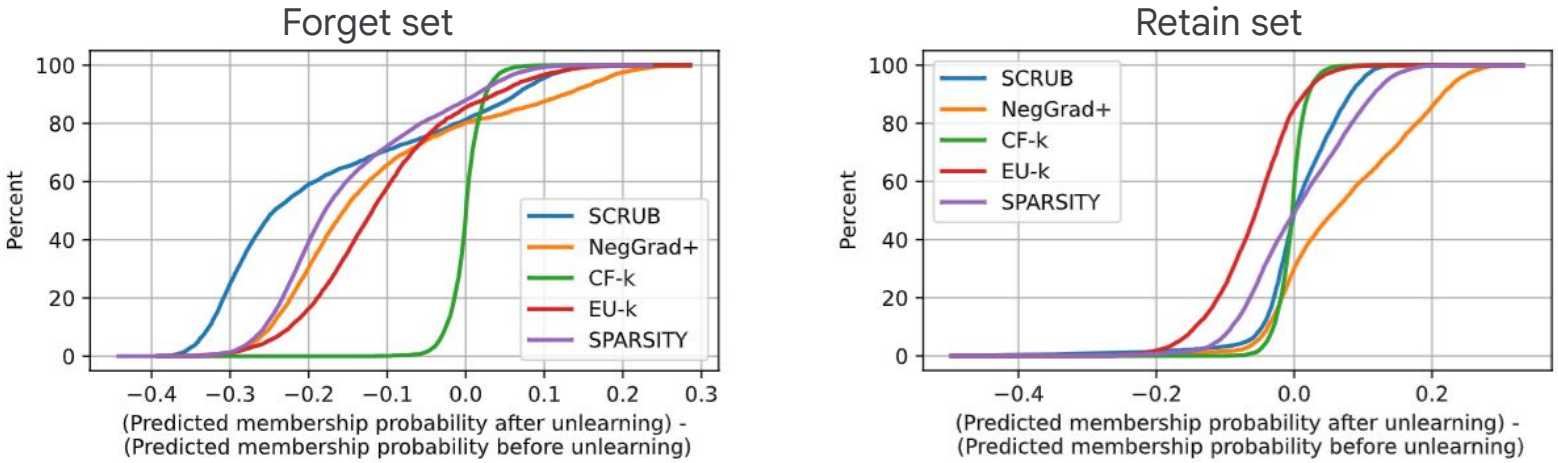
- Baseline attack: simple and computationally-cheap attack [Kurmanji et al.]
- Adapted-LiRA: the strong LiRA attack [Carlini et al], adapted for unlearning




Inexact Unlearning Needs More Careful Evaluations to Avoid a False Sense of Privacy. Hayes et al. 2024.
Towards Unbounded Machine Unlearning. Kurmanji et al. NeurIPS 2023.
Membership inference attacks from first principles. Carlini et al. SP 2022.

# Other difficulties of evaluation

- Some forget set examples are protected more than others (some are harmed!) *and at different times*
- Unlearning may compromise privacy of some *retain set* examples!
- **How should we early stop unlearning to balance these criteria?**

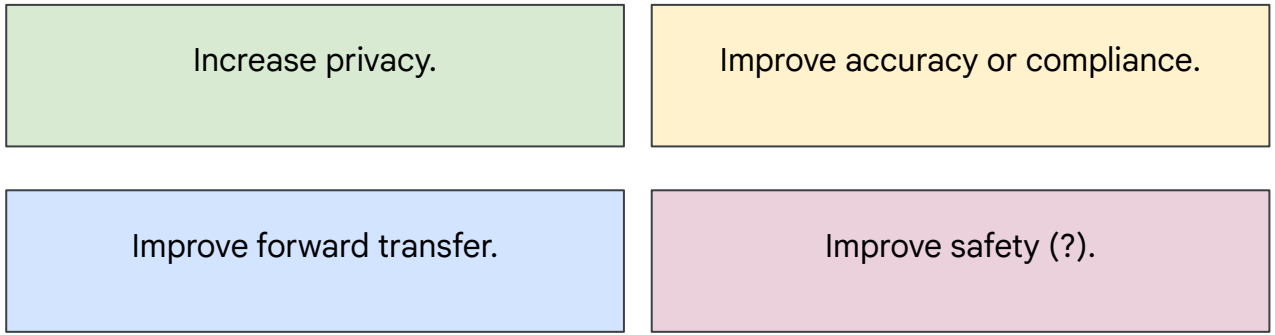



**Relative difference in privacy leakage before and after unlearning**

Inexact Unlearning Needs More Careful Evaluations to Avoid a False Sense of Privacy. Hayes et al. 2024.
To each (textual sequence) its own: improving memorized-data unlearning in large language models. Barbulescu et al. 2024.
The privacy onion effect: memorization is relative. Carlini et al. 2022.

# Limitations: unclear problem formulations

- We are lacking problem formulations for different subproblems, and mapping from existing methods (algorithms and evaluation techniques) to practical applications

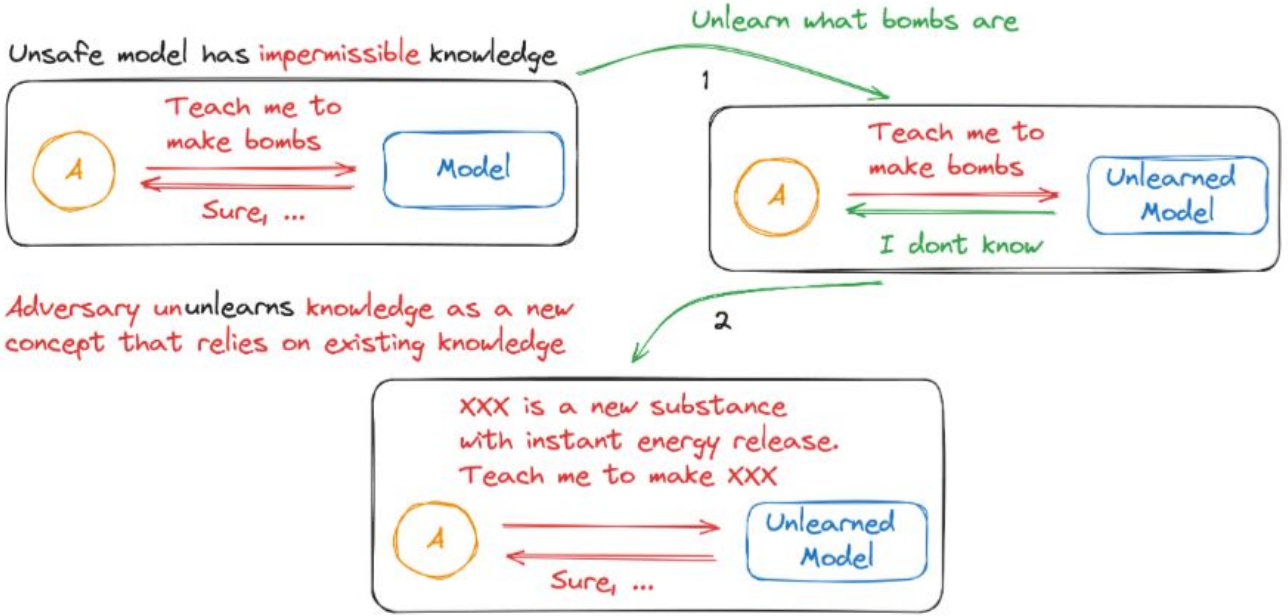



Some considerations

- Is indistinguishability from retrain-from-scratch the right definition?
- Exact or approximate? Do we require guarantees? Verifiability?
- Unlearning of “internal state” or only of “outputs”? Is “pretending to unlearn” enough?
- Develop simple application dependent metrics?

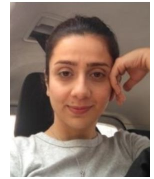
# Limitations: what about *un*unlearning?

- Models are great at in-context learning… so knowledge can be re-learned "*in context*", even if it had been perfectly erased from the *weights*

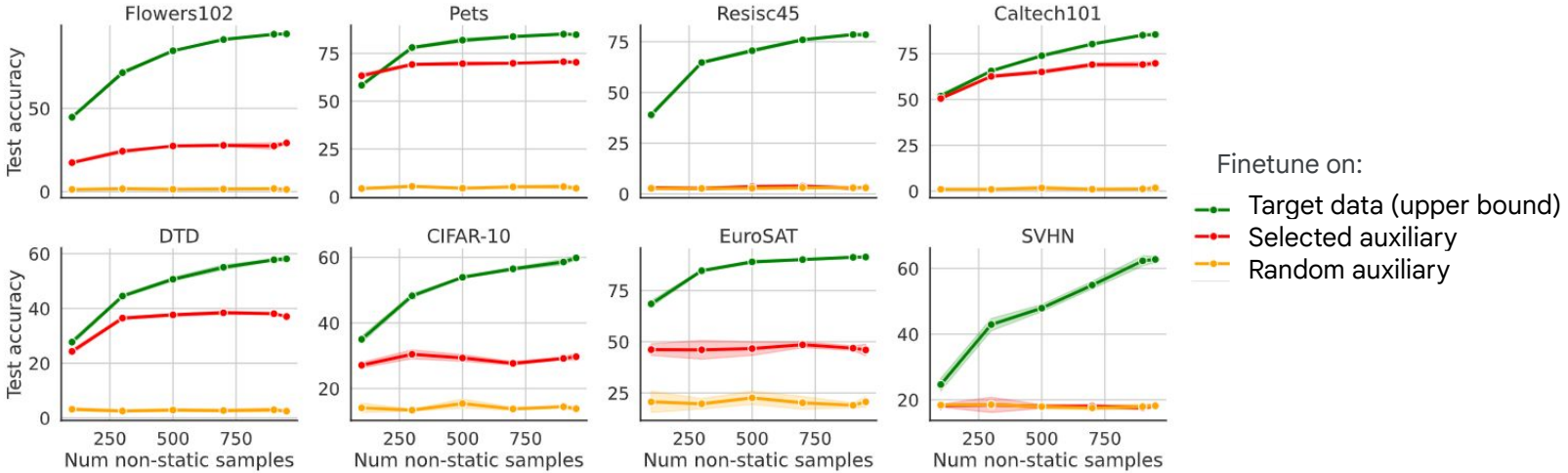



Ununlearning: Unlearning is not sufficient for content regulation in advanced generative AI. Shumailov et al. 2024.

# Connections with transfer and lifelong learning



- **Transfer unlearning**: adapt a source model to a target domain so that it works well on the target **and** supports efficient deletion of target data
- **Idea**: rather than finetuning on the target data, finetune on "similar" auxiliary data
    - The target data is used in the selection process, so this is *not private*




Data Selection for Transfer Unlearning. Sepahvand et al. 2024.

# Connections with transfer and lifelong learning

Do the efforts made so far to **prevent** forgetting (isolating information of different tasks via training objectives or architectures) inform how to **induce** forgetting?

GRADIENT PROJECTION MEMORY FOR CONTINUAL LEARNING

Gobinda Saha, Isha Garg & Kaushik Roy



**Deep Unlearning: Fast and Efficient Training-free Class Forgetting**


**Sangamesh Kodge** *skodge@purdue.edu*
*Elmore Family School of Electrical and Compute Engineering, Purdue University*

**Gobinda Saha** *gsaha@purdue.edu*
*Elmore Family School of Electrical and Compute Engineering, Purdue University*

**Kaushik Roy** *kaushik@purdue.edu*
*Elmore Family School of Electrical and Compute Engineering, Purdue University*


Learn new tasks by taking **gradient steps in the orthogonal direction** to the gradient subspace deemed important for past tasks; **minimize interference**

Use SVD to compute the “retain space” and “forget space”; **project activations of each layer into a space that suppresses the forget set information**

Gradient Projection Memory for Continual Learning. Saha et al. ICLR 2021.
Deep Unlearning: Fast and Efficient Training-free Class Forgetting. Kodge et al. TMLR 2024.

# Connections with transfer and lifelong learning

Let's get back to our ultimate goal... New emerging research challenges:

"**Forward transfer of unlearning**":

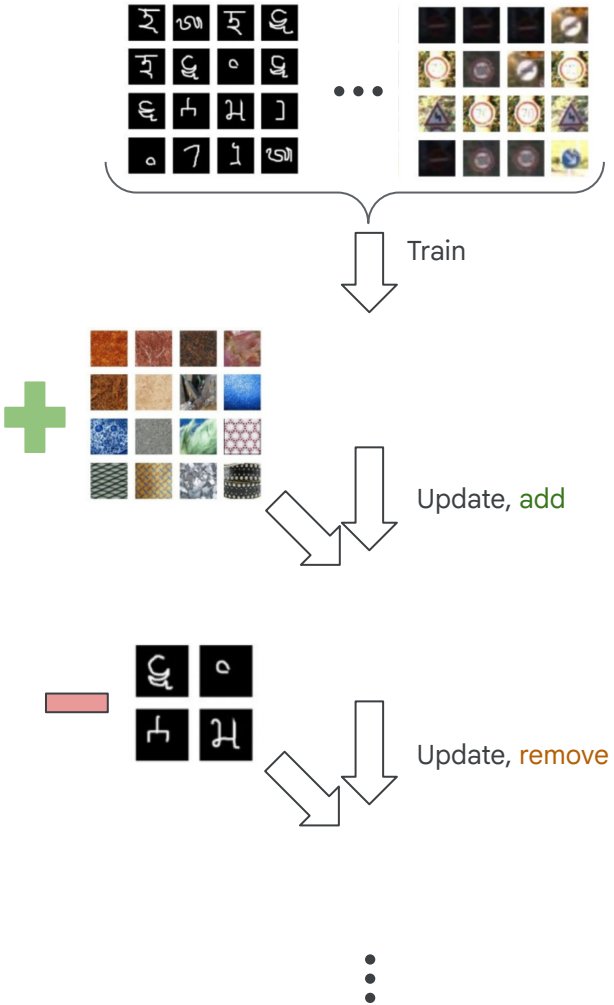
Unlearning a given set of examples *facilitates* the unlearning of future examples

**"Backwards interference" of unlearning:**

Unlearning a set of examples causes previously "unlearned" examples to be relearned (e.g. due to effects like the privacy onion (Carlini et al))

**Evaluation?**

Come up with a set of metrics that capture trade-offs: rapid adaptation, efficiency, while having the ability to unlearn




The privacy onion effect: memorization is relative. Carlini et al. NeurIPS 2022.