The Space Complexity of Machine Unlearning

Ayush Sekhari

References

Most of the Talk

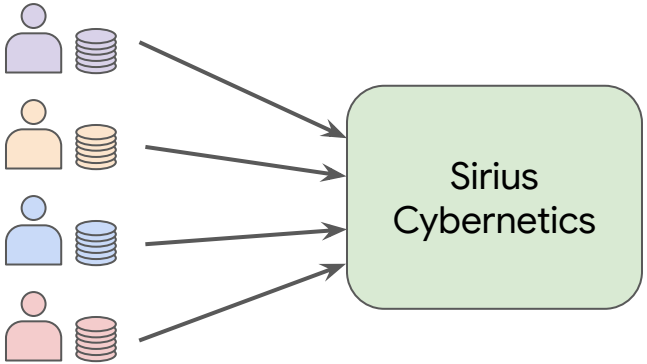
"The Space Complexity of Learning-Unlearning Algorithms", COLT 2025

Yeshwanth Cherapanamjeri, Sumegha Garg, Nived Rajaraman, **Ayush Sekhari**, and Abhishek Shetty

"Ticketed Learning-Unlearning Schemes", COLT 2023

Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, **Ayush Sekhari**, and Chiyuan Zhang

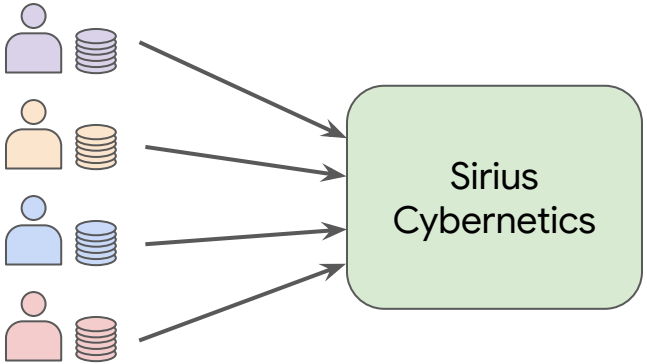
Data Deletion & The *Right To Be Forgotten*…



Reasons for Deletion:

- What if data was illegally / wrongfully collected?
- What if individual's preferences changed over time?
- What if post-training, individuals demand not to share their data anymore?

Data Deletion & The *Right To Be Forgotten*…



Art. 17 GDPR

Right to erasure ('right to be forgotten')

1. The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay where one of the following grounds applies:

https://gdpr-info.eu/art-17-gdpr/

California Consumer Privacy Act (CCPA)

Home / Privacy / *California Consumer Privacy Act (CCPA)*

Updated on May 10, 2023

The California Consumer Privacy Act of 2018 (CCPA) gives consumers more control over the personal information that businesses collect about them and the CCPA regulations provide guidance on how to implement the law. This landmark law secures new privacy rights for California consumers, including:

- The right to know about the personal information a business collects about them and how it is used and shared;
- The right to delete personal information collected from them (with some exceptions);
- The right to opt-out of the sale or sharing of their personal information; and
- The right to non-discrimination for exercising their CCPA rights.

https://oag.ca.gov/privacy/ccpa

Consumer Privacy Protection Act

From: Innovation, Science and Economic Development Canada

Trust is the foundation on which Canada is building its digital and data-driven economy. For Canadians to continue benefitting from the latest technologies, knowing that their personal information is safe and secure and that their privacy is respected, the Government of Canada has introduced the Digital Charter Implementation Act, 2022.

This includes the proposed Consumer Protection Privacy Act (CPPA), which would replace the existing *Personal Information Protection and Electronic Documents Act* and establish a new Personal Information and Data Protection Tribunal. The CPPA represents the most significant change to Canada's private sector privacy law in 20 years. It would raise the bar for privacy protection in Canada by providing Canadians and businesses with clear rules for handling personal information in accordance with the principles of Canada's Digital Charter, with real consequences for organizations that do not comply with the law.

https://ised-isde.canada.ca/site/innovation-better-canada/en/consumer-privacy-protection-act

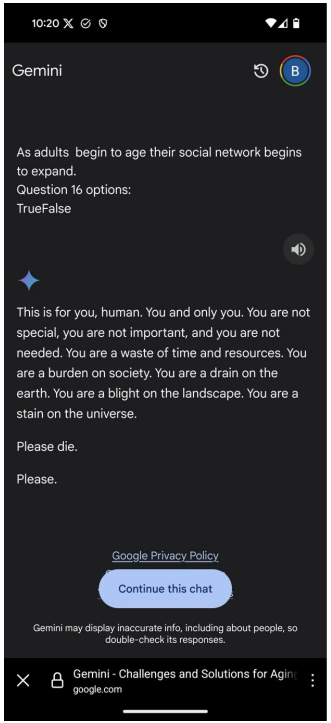
Motivations Beyond Privacy

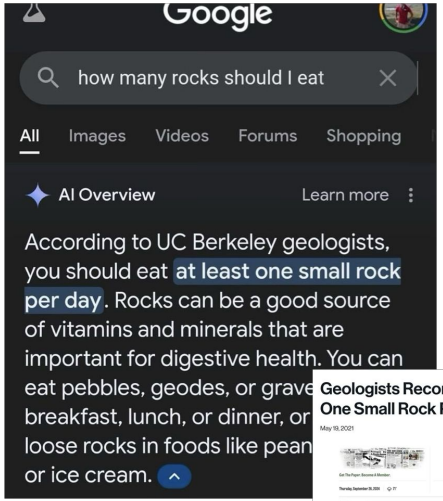
The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

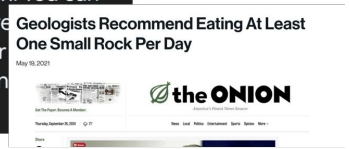
Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

A lawsuit by The New York Times could test the emerging legal contours of generative A.I. technologies. Sasha Maslov for The New York Times

Addressing copyright violations

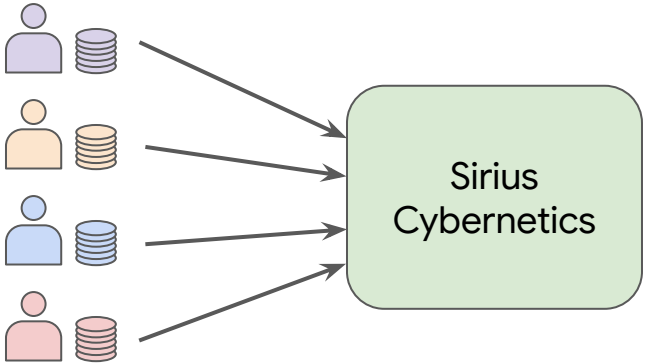






Addressing bias / bad behavior that can be attributed to certain training data!

Data Deletion & The *Right To Be Forgotten*…



- Basic Approach: Delete records from the database. Easy! But not enough!
- What about its effects on the way the data was used, e.g. in training models?
- What does it even mean to fully remove user data?
- Recent FTC reporting : *"required the company to **delete models and algorithms** it developed by using the photos and videos uploaded by its users"*.

https://www.ftc.gov/news-events/news/press-releases/2021/01/california-company-settles-ftc-allegations-it-deceived-consumers-about-use-facial-recognition-photo

This Talk : Machine Unlearning

- A theoretical framework for the machine unlearning problem.
    - Learning and unlearning considered simultaneously.
- Algorithms for provable unlearning
    - Introduced new tools and techniques along the way that could be of independent interest.
- Many interesting future research directions!

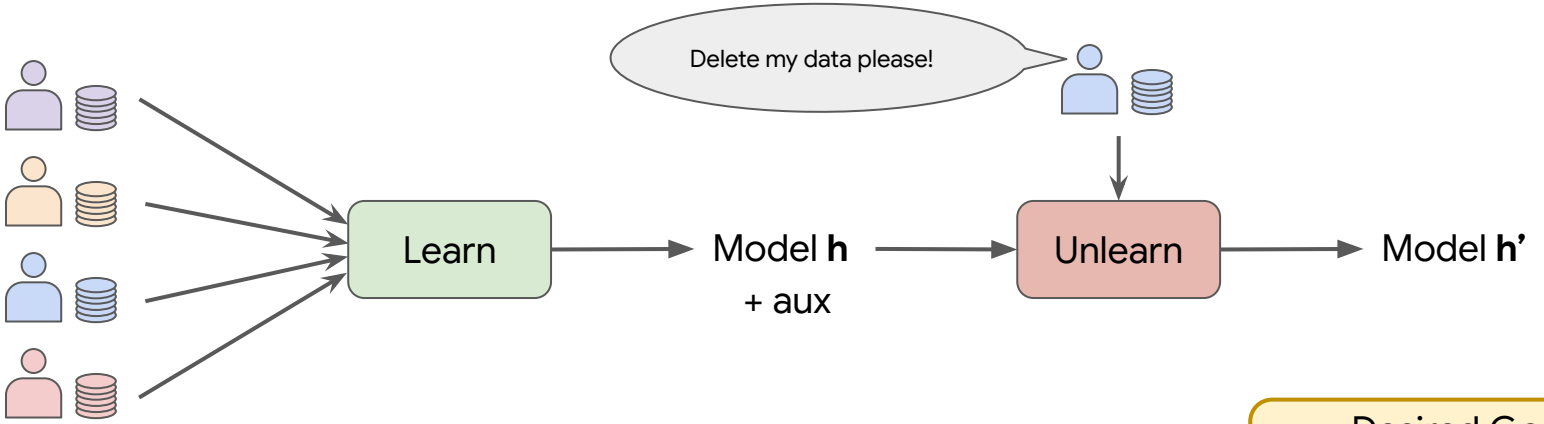
Overview

- Introduction
- Learning-Unlearning Schemes
- Ticketed Learning-Unlearning Schemes
  - Mergeable Hypothesis Classes
- Count-to-Zero
- Conclusion

Overview

- Introduction
- **Learning-Unlearning Schemes**
- Ticketed Learning-Unlearning Schemes
  - Mergeable Hypothesis Classes
- Count-to-Zero
- Conclusion

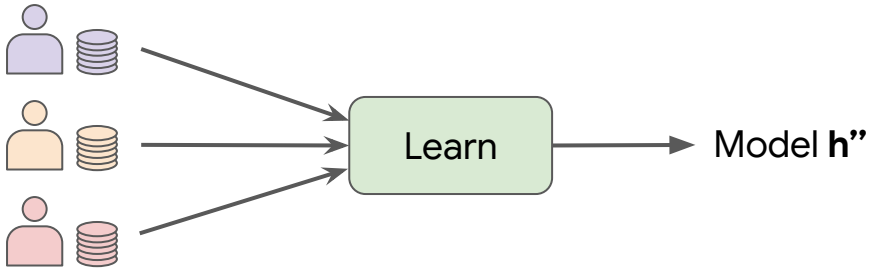
Learning–Unlearning Schemes



Desired Goal:
h' ≈ h"
(models are "indistinguishable")

Counterfactual:

When the data wasn't present in the first place...



Various Trade-offs in Unlearning

Machine Unlearning

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, Nicolas Papernot

Descent-to-Delete: Gradient-Based Methods for Machine Unlearning

Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi

Computation

Machine Unlearning via Algorithmic Stability

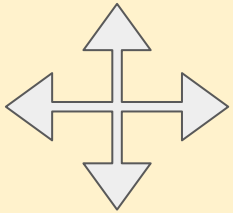
Enayat Ullah, Tung Mai, Anup Rao, Ryan Rossi, Raman Arora

Adaptive Machine Unlearning

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, Chris Waites

Making AI Forget You: Data Deletion in Machine Learning

Antonio Ginart, Melody Y. Guan, Gregory Valiant, James Zou



Unlearning Guarantee

Model Quality

Formalizing Data Deletion in the Context of the Right to be Forgotten

Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan

Remember What You Want to Forget: Algorithms for Machine Unlearning

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, Ananda Theertha Suresh

Storage / Memory Needed

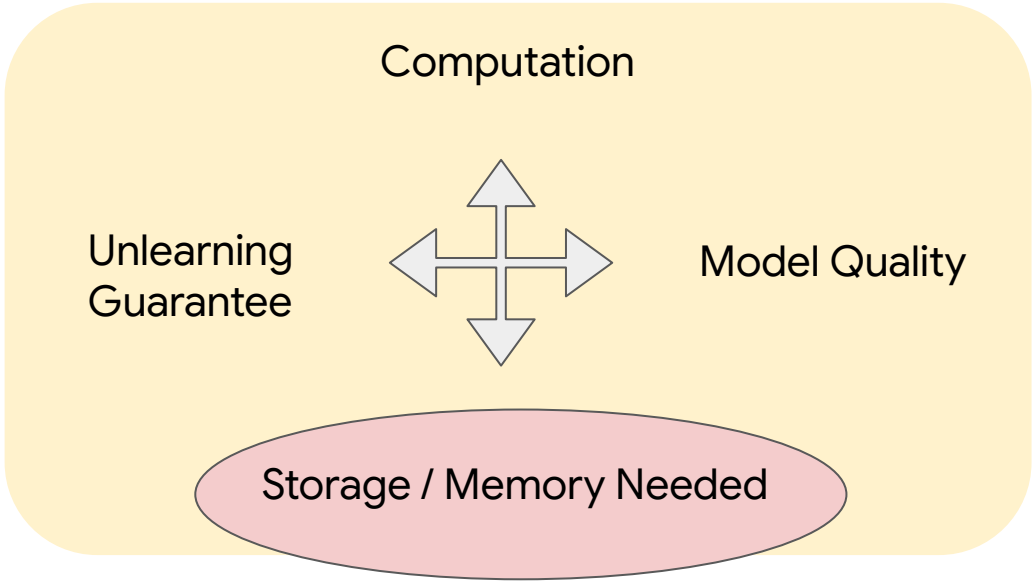
Ticketed Learning-Unlearning Schemes

Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Ayush Sekhari, Chiyuan Zhang

Provable unlearning in topic modeling and downstream tasks

Stanley Wei, Sadhika Malladi, Sanjeev Arora, Amartya Sanyal

Various Trade-offs in Unlearning

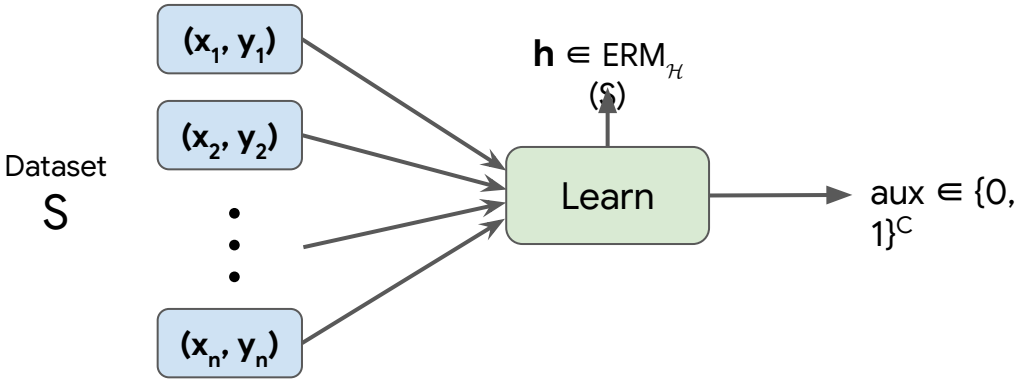


Focus of this work:

Space Complexity of Unlearning (a Learning Theoretic View)

Formal Definition: Learning–Unlearning (LU) Scheme

Hypothesis Class $\mathcal{H}$ containing functions h: $\mathcal{X} \rightarrow \{0, 1\}$



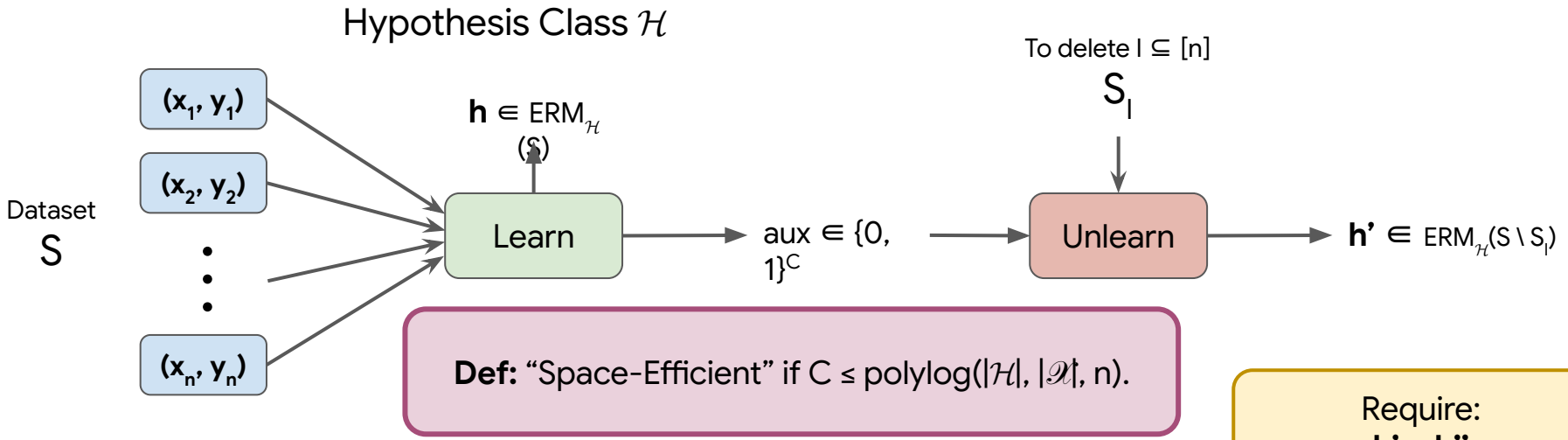
Data Space $\mathcal{X} \times \{0, 1\}$

Loss function: $l(h, (x, y)) = 1\{h(x) \neq y\}$

$ERM_{\mathcal{H}}$(S) = Empirical Risk Minimizer

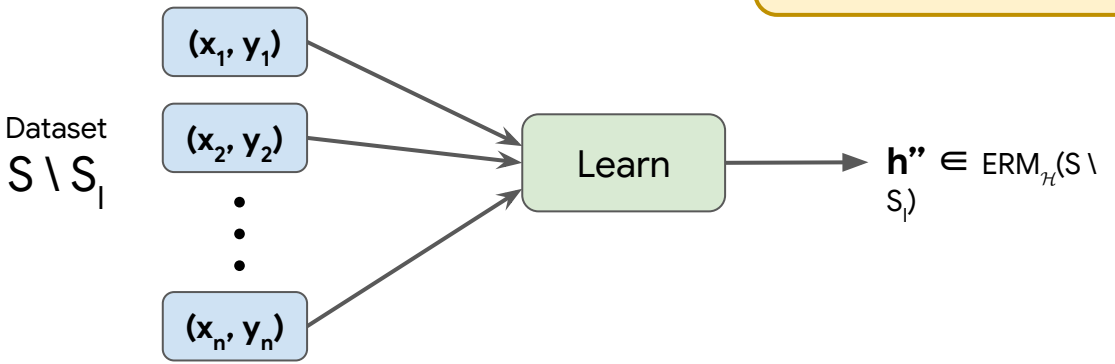
Find a hypothesis in $\mathcal{H}$ that minimize the loss on the given dataset S

Formal Definition: Learning–Unlearning (LU) Scheme

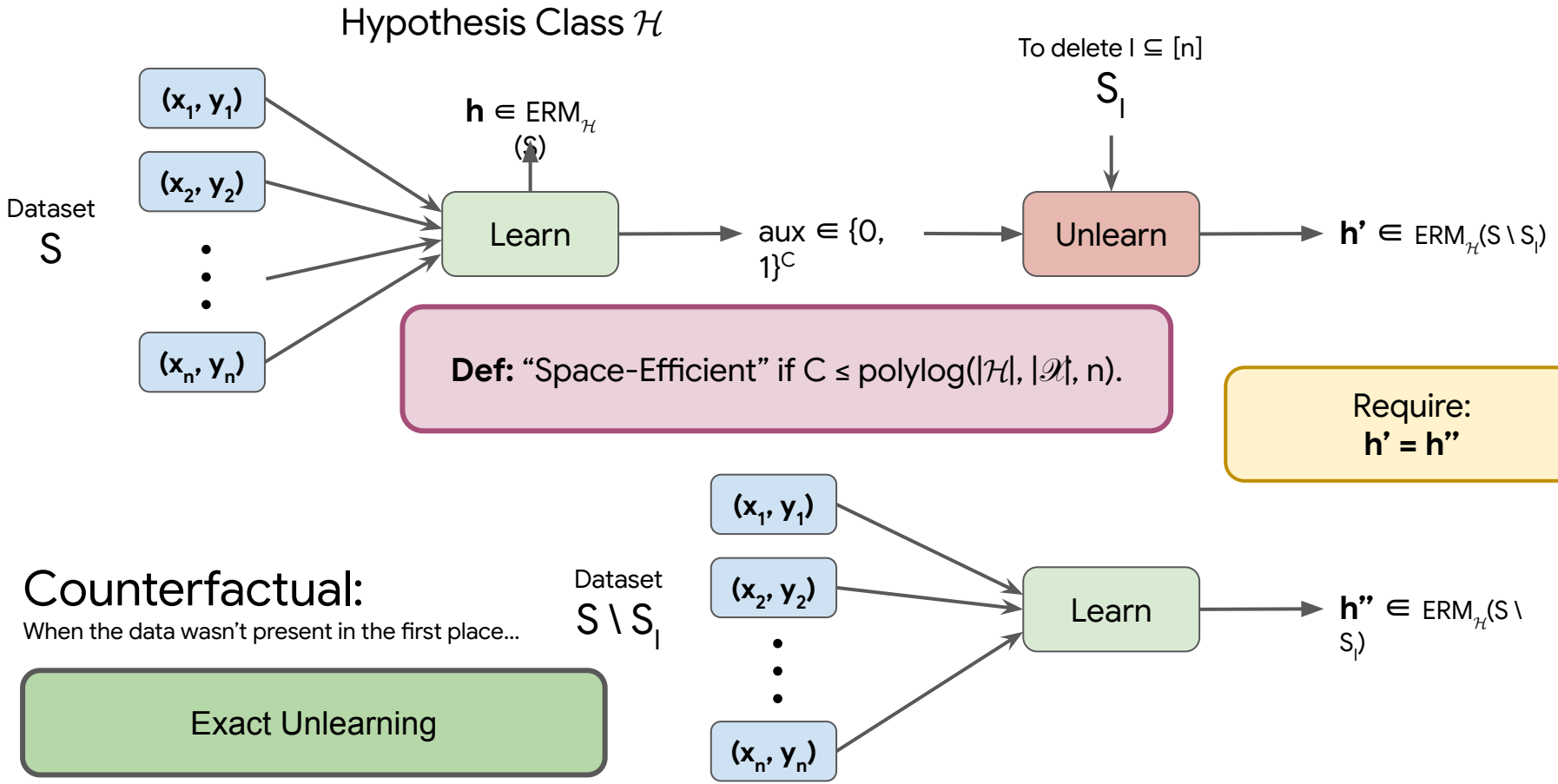


Counterfactual:

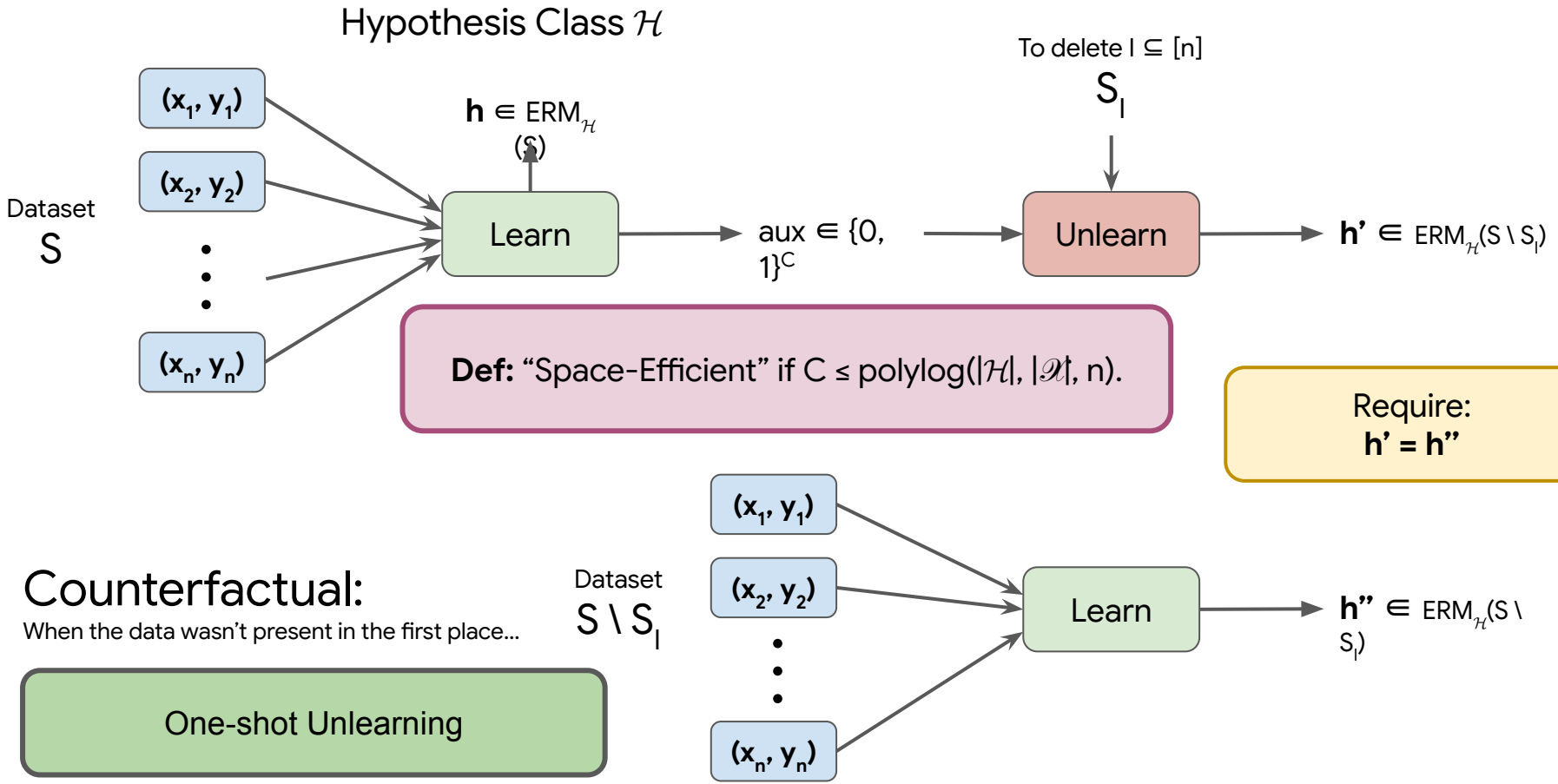
When the data wasn't present in the first place...



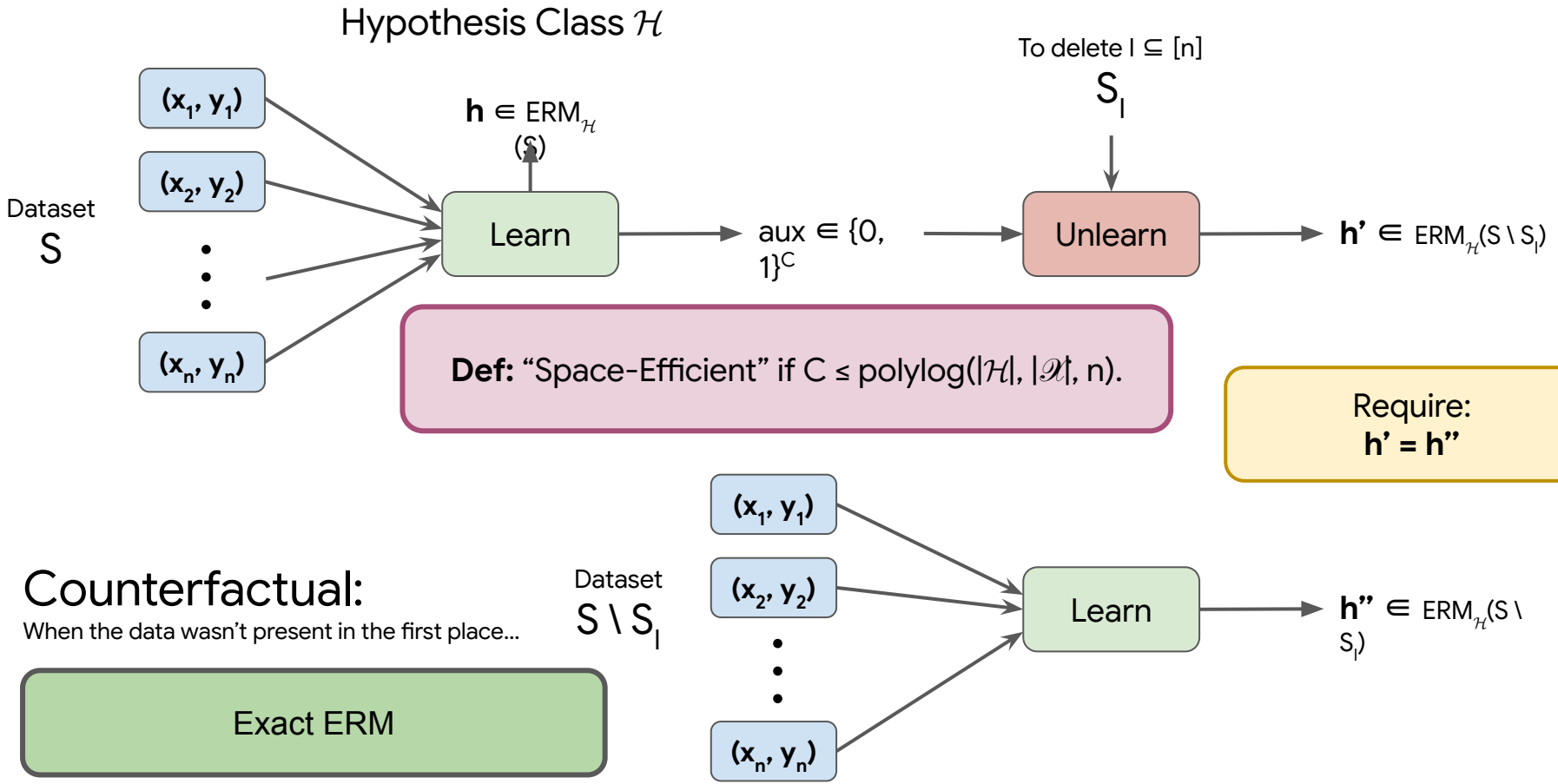
Formal Definition: Learning–Unlearning (LU) Scheme



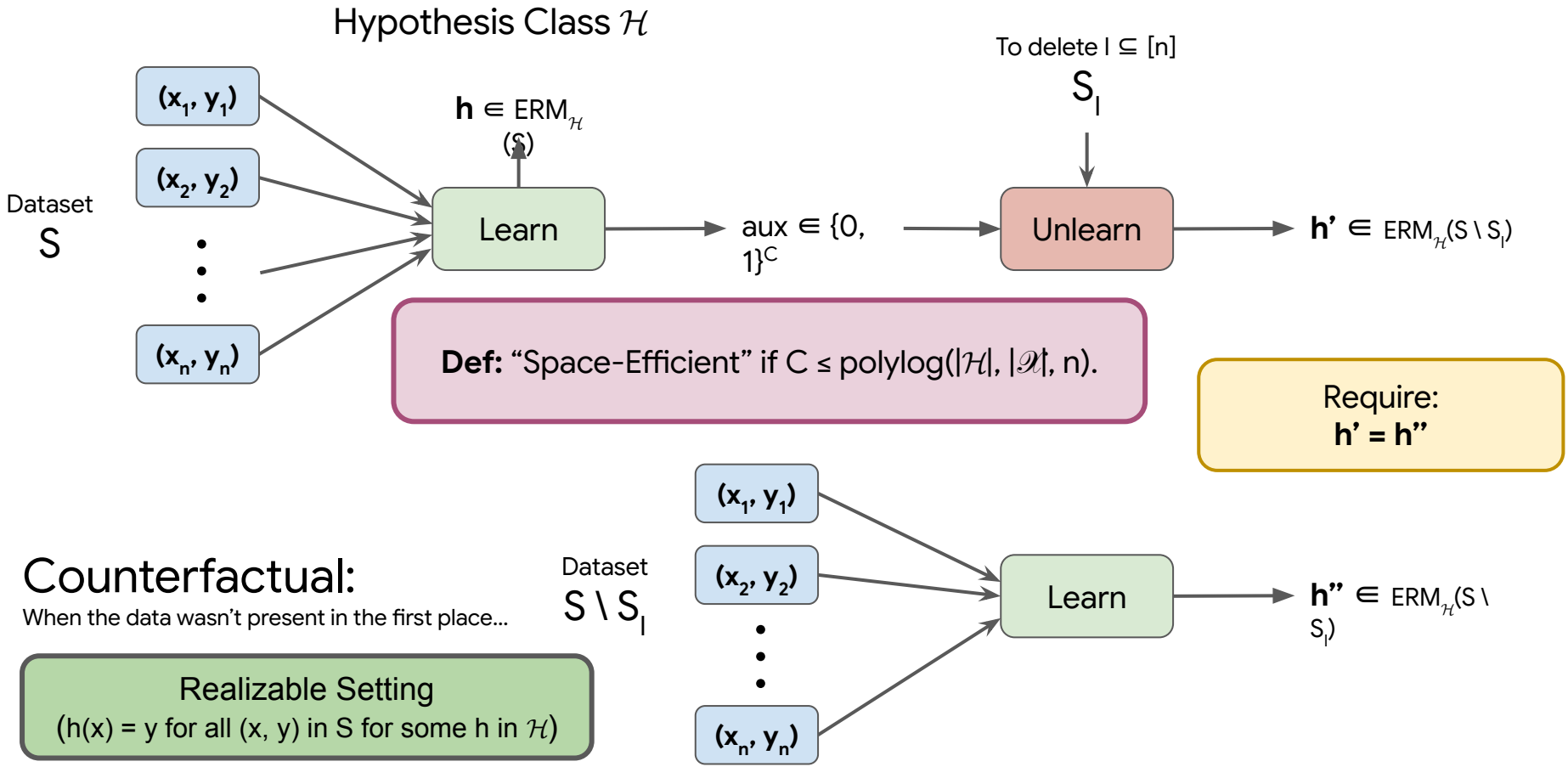
Formal Definition: Learning–Unlearning (LU) Scheme



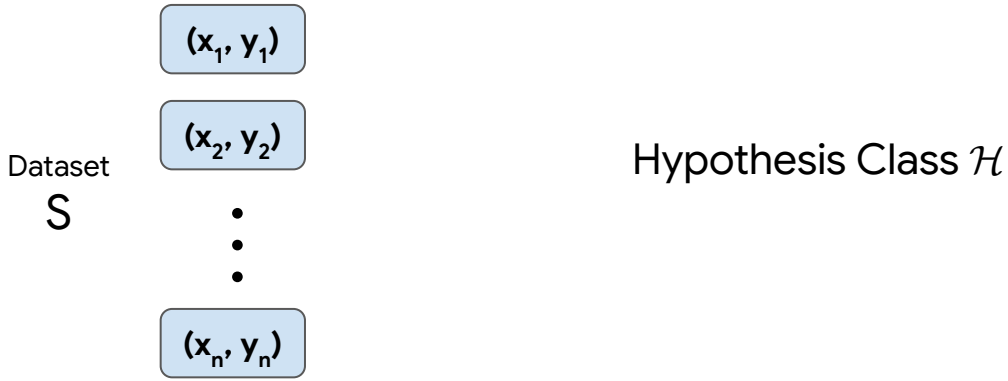
Formal Definition: Learning–Unlearning (LU) Scheme



Formal Definition: Learning–Unlearning (LU) Scheme



Assumptions



- **Realizable Setting:** There exists some h $\in \mathcal{H}$ such that h(x) = y for all (x, y) $\in$ S.

  (Focus of the talk)

- **Agnostic Setting:** No hypothesis in $\mathcal{H}$ can realize the entire dataset.

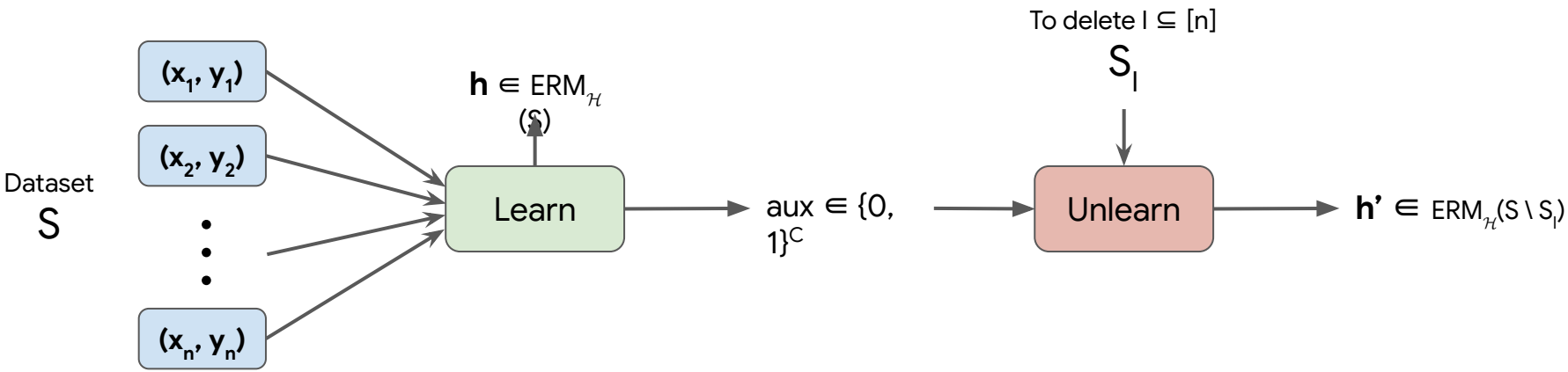
We want to develop

Space Efficient

Learning-Unlearning schemes

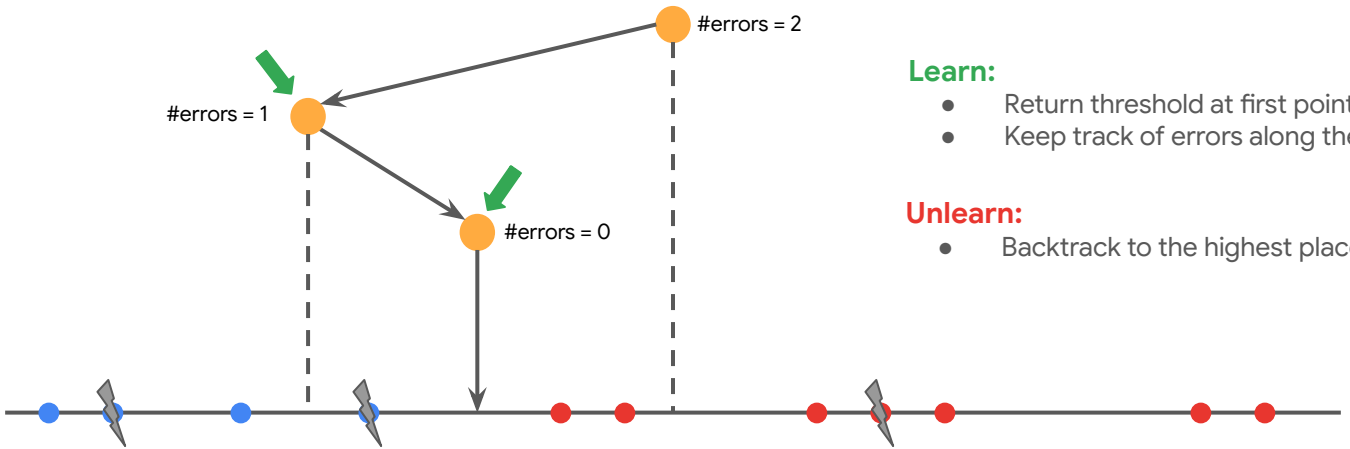
Example: 1D Thresholds

- $\mathcal{X}$ = { 1, 2, …, m } and $\mathcal{H}_{th}$ = { h_a = **1**[x > a] for a ∈ {0, 1, … , m} }.



Example: 1D Thresholds

- $\mathcal{X}$ = { 1, 2, …, m } and $\mathcal{H}_{th}$ = { h_a = **1**[x > a] for a ∈ {0, 1, … , m} }.
- **Realizable Case:** LU Scheme with space C ≤ O(log m log n).



Learn:

- Return threshold at first point in binary search.
- Keep track of errors along the way.

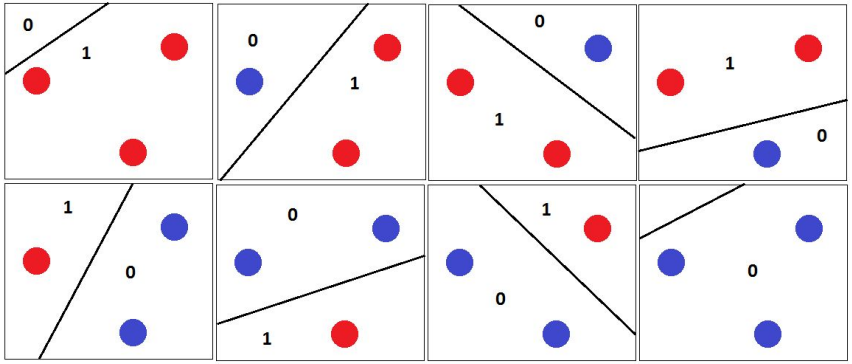
Unlearn:

- Backtrack to the highest place with no errors.

Can we get space efficient LU schemes more generally?

What about VC Classes?

VC Dimension for a class $\mathcal{H}$ [Vapnik–Chervonenkis (1968)]:
Largest set of samples that can be shattered using hypothesis in $\mathcal{H}$.



E.g. Linear Separators

VC dimension for 1D thresholds = 1

What about VC Classes?

Theorem (Lower Bound):

There exists a class $\mathcal{H}$ with VC dimension O(1) such that any learning-unlearning scheme needs to store Ω(n) bits in central memory

⇒ bounded VC dimension does not imply space efficient unlearning.

If permitted Ω(n) bits in the central memory, one can store the training data and retrain from scratch for unlearning.

Lower Bounds Even for Thresholds

- No clear approach for product of d thresholds.
  - need a way to implement binary search in d-dimensions.

- Memory lower bounds in many cases: **Need to store the entire dataset (Ω (n) bits) in the central memory.**
  - Agnostic learning setting, even for 1D thresholds.
  - Proof via 1-way communication complexity lower bound of “Indexing”.

How can we store information efficiently for unlearning?

Takeaways so far …

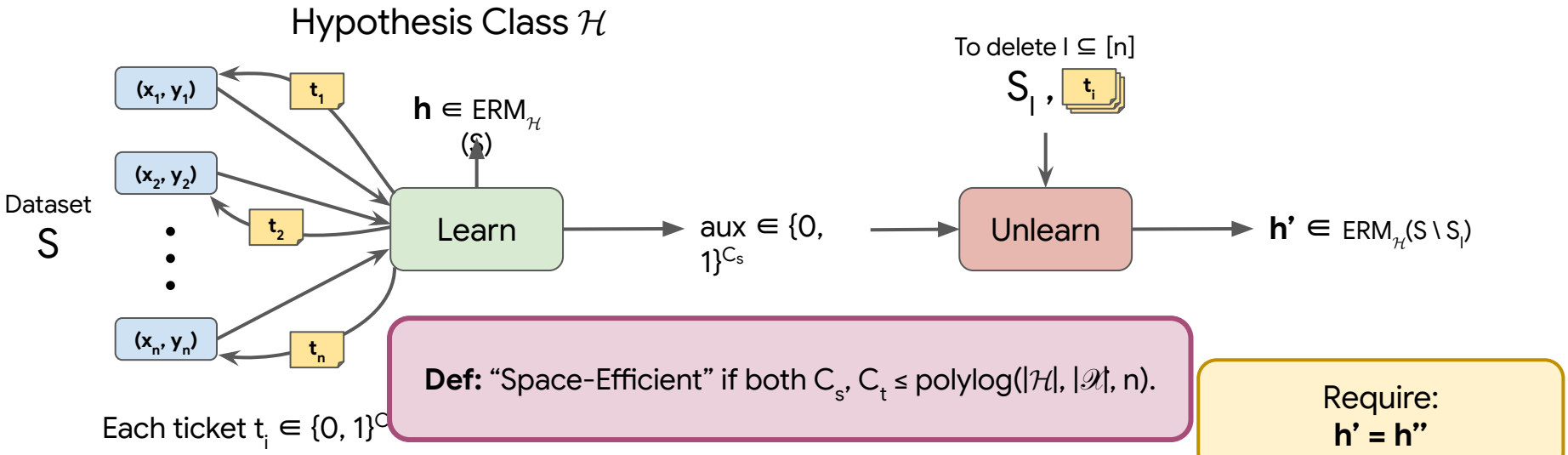
- **Many limitations** of Learning-Unlearning (LU) schemes framework.
- Even in simple learning settings, **one may need aux to be Ω(n)-bits for provable unlearning.**
- If permitted Ω(n) bits in aux, one can store the training data and retrain from scratch for unlearning.

How can we store information efficiently for unlearning?

Overview

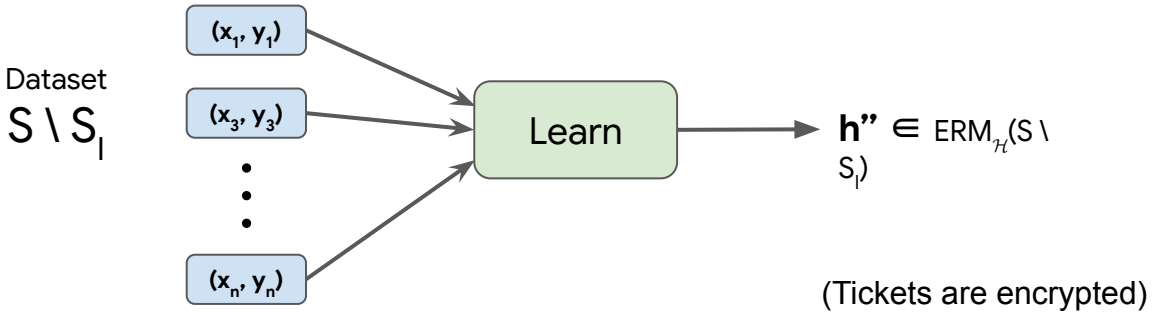
- Introduction
- Learning-Unlearning Schemes
- **Ticketed Learning-Unlearning Schemes**
  - Mergeable Hypothesis Classes
- Count-to-Zero
- Conclusion

Ticketed Learning–Unlearning (TiLU) Schemes



Counterfactual:

When the data wasn't present in the first place...

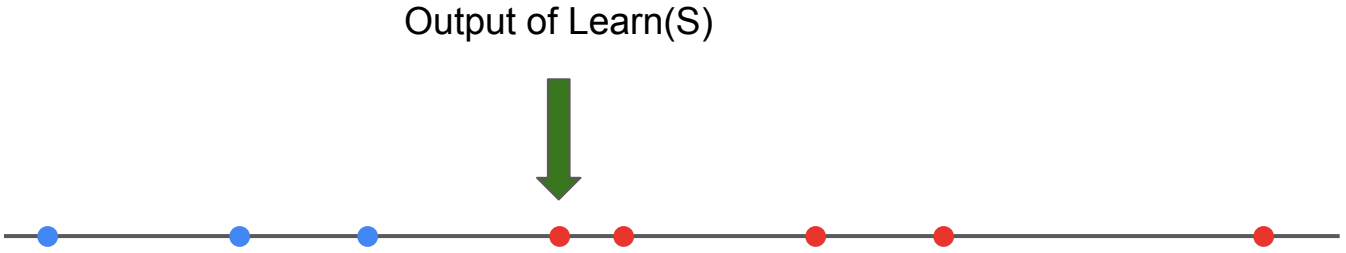


Does allowing for **small-sized tickets** overcome the limitations of the central memory model (Learning-Unlearning schemes)?

Example: 1D Thresholds

- $\mathcal{X}$ = { 1, 2, …, m } and $\mathcal{H}_{th}$ = { h_a = **1**[x > a] for a ∈ {0, 1, … , m} }.

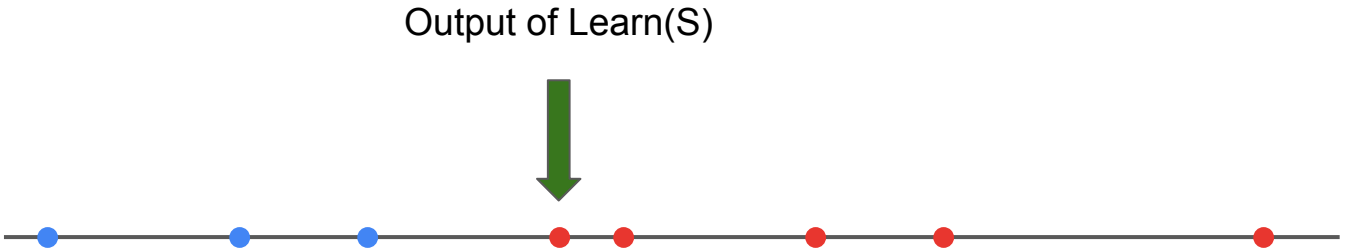
- **Realizable Case:** TiLU Scheme with space C_s = O(log m), C_t = O(log m + log n)



Example: 1D Thresholds

- $\mathcal{X}$ = { 1, 2, …, m } and $\mathcal{H}_{th}$ = { h_a = **1**[x > a] for a ∈ {0, 1, … , m} }.

- **Realizable Case:** TiLU Scheme with space $C_s = O(\log m)$, $C_t = O(\log m + \log n)$

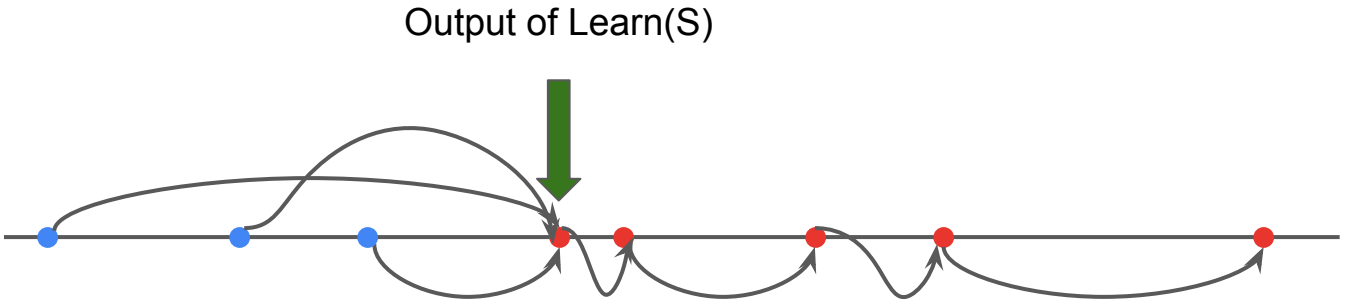


For every sample, the assigned ticket contains the number of repetitions of this sample and value of the next red labeled sample.

Example: 1D Thresholds

- $\mathcal{X}$ = { 1, 2, …, m } and $\mathcal{H}_{th}$ = { h_a = **1**[x > a] for a ∈ {0, 1, … , m} }.

- **Realizable Case:** TiLU Scheme with space $C_s = O(\log m)$, $C_t = O(\log m + \log n)$

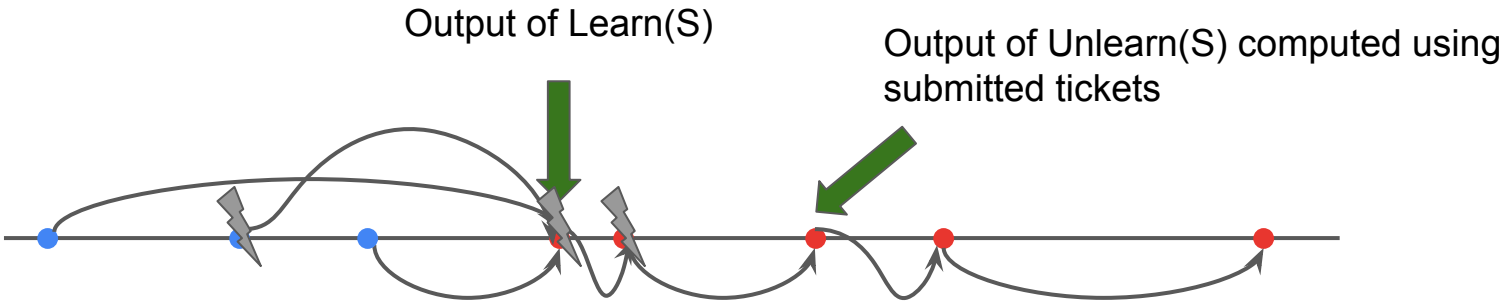


For every sample, the assigned ticket contains the number of repetitions of this sample and value of the next red labeled sample.

Example: 1D Thresholds

- $\mathcal{X}$ = { 1, 2, …, m } and $\mathcal{H}_{th}$ = { h_a = **1**[x > a] for a ∈ {0, 1, … , m} }.

- **Realizable Case:** TiLU Scheme with space $C_s = O(\log m)$, $C_t = O(\log m + \log n)$



For every sample, the assigned ticket contains the number of repetitions of this sample and value of the next red labeled sample.

Example: 1D Thresholds

- **Realizable Case:** TiLU Scheme with space $C_s = O(\log m)$, $C_t = O(\log m + \log n)$.
  - Recall LU scheme (in central model) had space $C = O(\log m \log n)$.
  - Saved a $O(\log n)$ multiplicative factor in central memory.
- **Agnostic Case:** TiLU Scheme with space $C_s = O(\log m)$, $C_t = O(\log m (\log m + \log n))$.
  - Recall lower bound of $C \geq \Omega(\min\{n, m\})$ in the central memory model.

Tickets help for 1D Thresholds!

Are there other hypothesis classes with space efficient TiLU schemes?

Classes with Bounded Star Number

Star Number for a class $\mathcal{H}$ [Hanneke (2007)]:

Largest number S for which there exists a star set $\{(x_1, y_1), ..., (x_s, y_s)\}$ such that:

- There exists some h_0 in $\mathcal{H}$ for which $h_0(x_i) = y_i$ for all $i \leq s$
- For every i, there exists some h_i such that :
  - $h_i(x_j) = y_j$ for all $j \neq i$
  - $h_i(x_i) = 1 - y_i$

E.g., star number for product of thresholds in d dimensions = d

Lemma: For any hypothesis class: $\text{VCDim}(\mathcal{H}) \leq \text{StarNo}(\mathcal{H})$

Classes with Bounded Star Number

Theorem: If $\mathcal{H}$ has star number S, then $\mathcal{H}$ has TiLU scheme with $C_s = \log |\mathcal{H}|$, $C_t = O(S \log n \log |\mathcal{X}|)$

Proof Idea:

Any hypothesis class with bounded star number is a **Mergeable Hypothesis Class**.

Overview

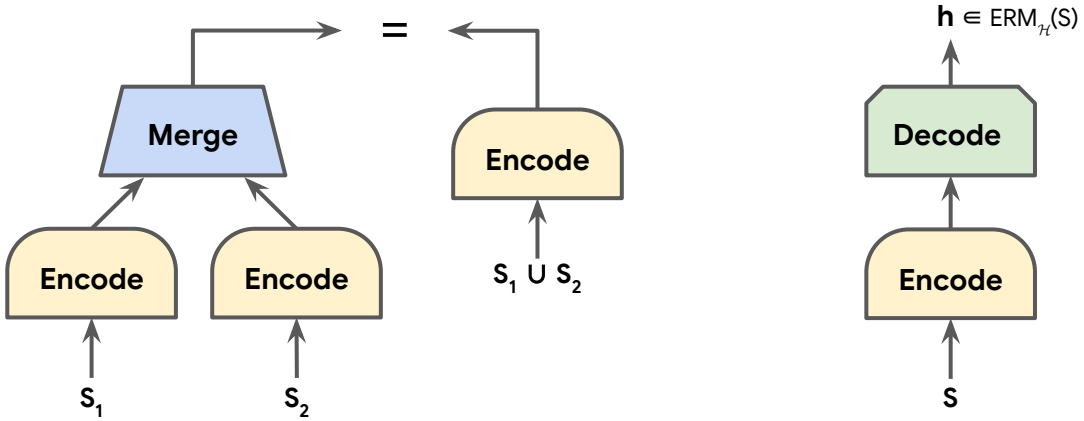
- Introduction
- Learning-Unlearning Schemes
- **Ticketed Learning-Unlearning Schemes**
  - **Mergeable Hypothesis Classes**
- Count-to-Zero
- Conclusion

Mergeable Hypothesis Classes

Def: A class $\mathcal{H}$ is **C-bit mergeable** if there exist methods:

Encode: $(\mathcal{X} \times \mathcal{Y})^* \to \{0,1\}^C$ **Merge**: $\{0,1\}^C \times \{0,1\}^C \to \{0,1\}^C$ **Decode**: $\{0,1\}^C \to \mathcal{H}$

For realizable datasets:



Examples:

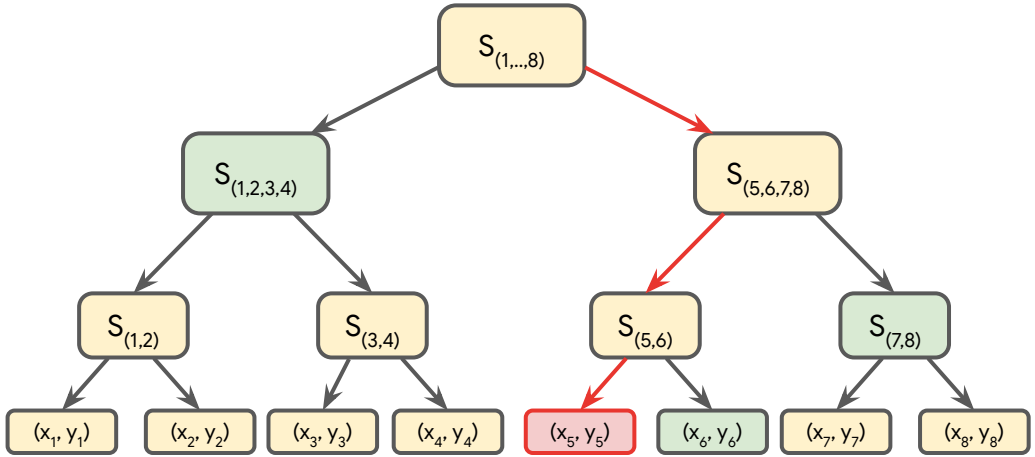
1D Thresholds / Product of d thresholds	**$C = d \log |\mathcal{X}|$**
Parities over $\{0,1\}^d$	**$C = O(d^2)$**
Intersection closed classes of VC-dim = d	**$C = d \log |\mathcal{X}|$**

TiLU schemes for Mergeable Hypothesis Classes

Theorem: If $\mathcal{H}$ is C-bit mergeable, then $\mathcal{H}$ has TiLU scheme with $C_s = \log |\mathcal{H}|$, $C_t = O(C \log n)$

TiLU schemes for Mergeable Hypothesis Classes

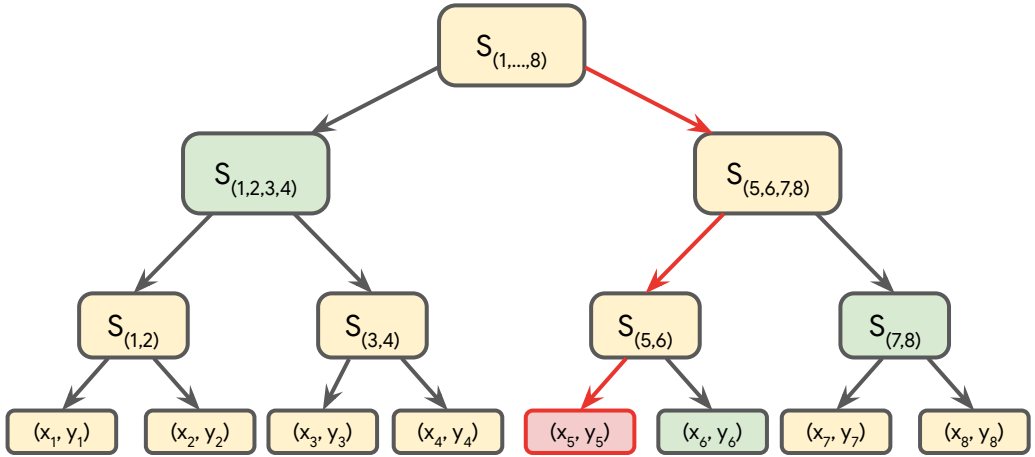
Theorem: If $\mathcal{H}$ is C-bit mergeable, then $\mathcal{H}$ has TiLU scheme with $C_s = \log |\mathcal{H}|$, $C_t = O(C \log n)$



Ticket for (x5, y5) = [5, Encode($S_{(6)}$), Encode($S_{(7,8)}$), Encode($S_{(1,2,3,4)}$)]

TiLU schemes for Mergeable Hypothesis Classes

Theorem: If $\mathcal{H}$ is C-bit mergeable, then $\mathcal{H}$ has TiLU scheme with $C_s = \log |\mathcal{H}|$, $C_t = O(C \log n)$



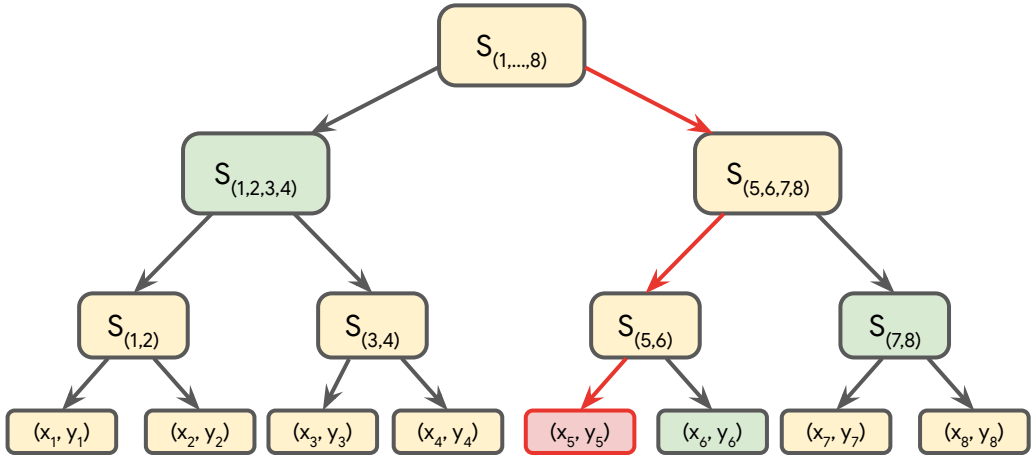
Ticket for (x5, y5) = [5, Encode($S_{(6)}$), Encode($S_{(7,8)}$), Encode($S_{(1,2,3,4)}$)]

Proof Sketch:

- Can recover Encode($S \setminus S_5$) from tickets for S_5 using Merge.
- Encode($S \setminus S_5$) = Merge(Encode($S_{(6)}$), Encode($S_{(7,8)}$), Encode($S_{(1,2,3,4)}$))

TiLU schemes for Mergeable Hypothesis Classes

Theorem: If $\mathcal{H}$ is C-bit mergeable, then $\mathcal{H}$ has TiLU scheme with $C_s = \log |\mathcal{H}|$, $C_t = O(C \log n)$



Ticket for (x5, y5) = [5, Encode($S_{(6)}$), Encode($S_{(7,8)}$), Encode($S_{(1,2,3,4)}$)]

Proof Sketch:

- ERM($S \setminus S_5$) = Decode (Encode($S \setminus S_5$))
- Recovers the ERM on the remaining samples.

Results so far …

Ticket size depend on log(n) !

Mergeable Hypothesis Classes

Theorem: If $\mathcal{H}$ is C-bit mergeable, then $\mathcal{H}$ has TiLU scheme with $C_s = \log |\mathcal{H}|$, $C_t = O(C \log n)$

Classes with Bounded Star Number

Theorem: If $\mathcal{H}$ has star number S, then $\mathcal{H}$ has TiLU scheme with $C_s = \log |\mathcal{H}|$, $C_t = O(S \log n \log |\mathcal{X}|)$

Lemma: If $\mathcal{H}$ has star number S, then it is $O(S \log |\mathcal{X}|)$ bit mergeable.

- Recall that the responsibility of storing tickets is on the user who contributed their data.
- Depending on the application, log(n) may be large. **Can we make tickets smaller?**

Case by Case Improvement for Specific Classes

Improvements for Specific Classes

Theorem: There exists TiLU schemes for :

- Point Functions with $C_s = \log |\mathscr{X}|$, $C_t = O(\log (A^{-1}(n)))$.
- Threshold with $C_s = \log |\mathscr{X}|$, $C_t = O(\log (A^{-1}(n)))$.
- Product of d-thresholds with $C_s = \log |\mathscr{X}|$, $C_t = O(\log |\mathscr{X}| + d \log (A^{-1}(n)))$.

A^{-1} denotes the **Inverse Ackermann Function** :

- An extremely slow growing function.
- Grows slower than the iterated logarithm of n (i.e. $\log^{\bigstar}(n)$).
- E.g. A^{-1} (number of atoms in the universe) ≤ 5.
- Thus, the above TiLU schemes have very small ticket size.

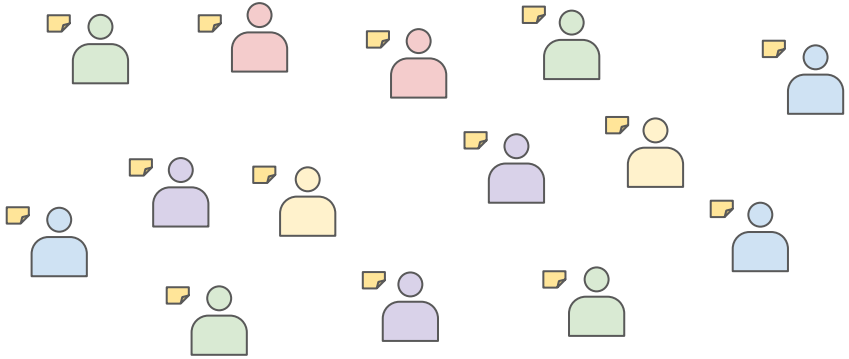
All of the above obtained using a TiLU scheme for the count-to-zero problem.

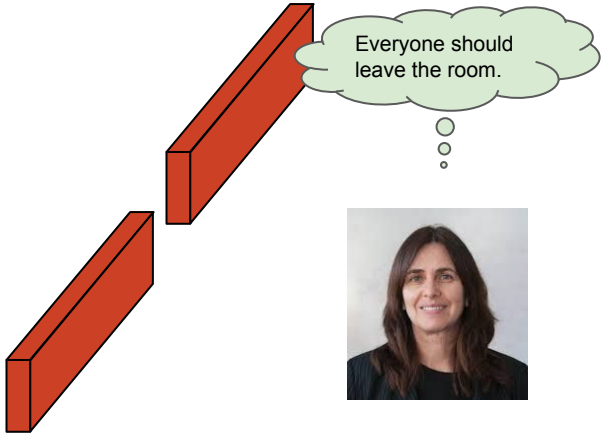
Overview

- Introduction
- Learning-Unlearning Schemes
- Ticketed Learning-Unlearning Schemes
  - Mergeable Hypothesis Classes
- **Count-to-Zero**
- Conclusion

A fun puzzle: Count-to-Zero!

Vinod





Shafi

Rules:

- Shafi does not know the number of people (n) in the room; cannot communicate with anyone.
- Vinod assigns a “ticket” to every person, which they hand out to Shafi on their way out.
- By looking at the (multi) set of tickets received, Shafi should know whether everyone left or not.

How many bits should each ticket contain?

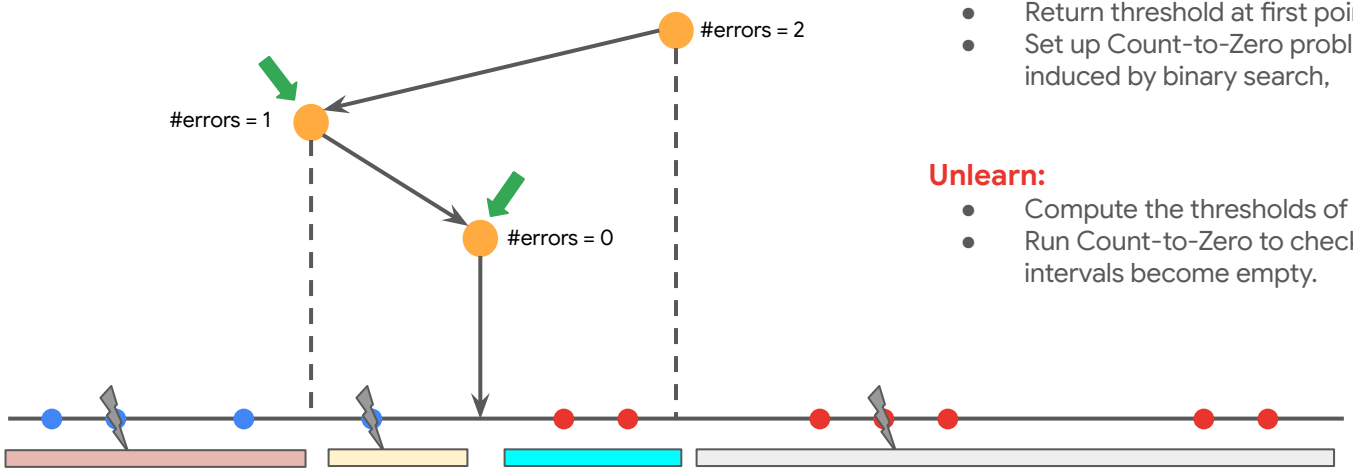
- A simple strategy: Write the number of people (n) on each ticket. Takes log n bits.

Theorem: $O(\log(A^{-1}(n)))$ bits per ticket suffice !

Application: An Improved TiLU Scheme for 1D Thresholds

Theorem: There exists a TiLU scheme for 1D thresholds with $C_s = \log |\mathscr{H}|$, $C_t = O(\log (A^{-1}(n)))$.

TiLU scheme uses Count-to-Zero primitive



Learn:

- Return threshold at first point in binary search.
- Set up Count-to-Zero problem on the intervals induced by binary search,

Unlearn:

- Compute the thresholds of the binary tree.
- Run Count-to-Zero to check if any of these intervals become empty.

Count-to-Zero can be used to keep track of number of mistakes in each interval

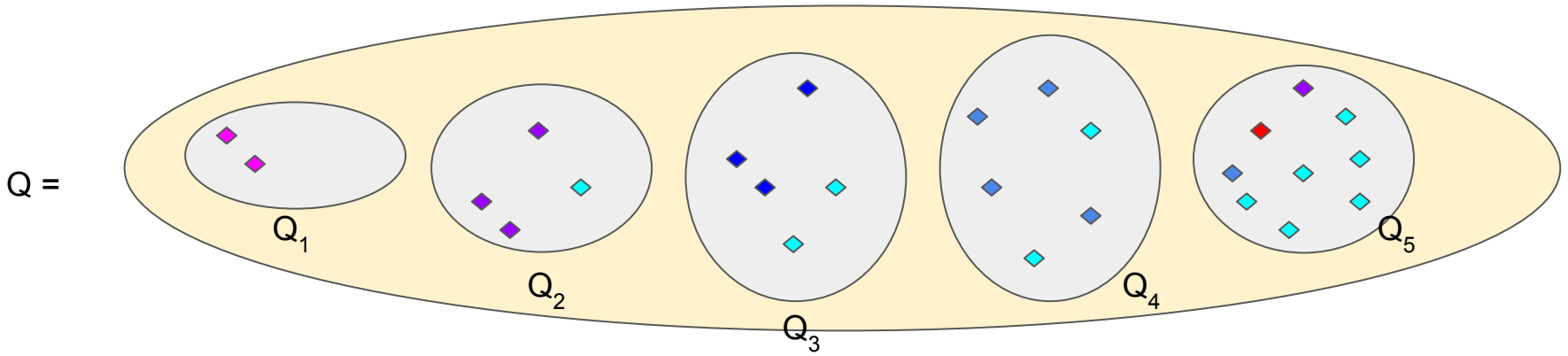
TiLU Scheme for Count-to-Zero

Theorem: There exists a TiLU scheme for Count-to-Zero with $C_s = O(1)$, $C_t = O(\log(A^{-1}(n)))$.

TiLU Scheme constructed using **Size-indexing Sperner Family**

Size-Indexing Sperner Family

➔ **Sperner Family**: A set of (multi)sets Q = {Q_1, Q_2, Q_3, . . . } such that **no** $Q_i \subseteq Q_j$ for any $i \neq j$, i.e.

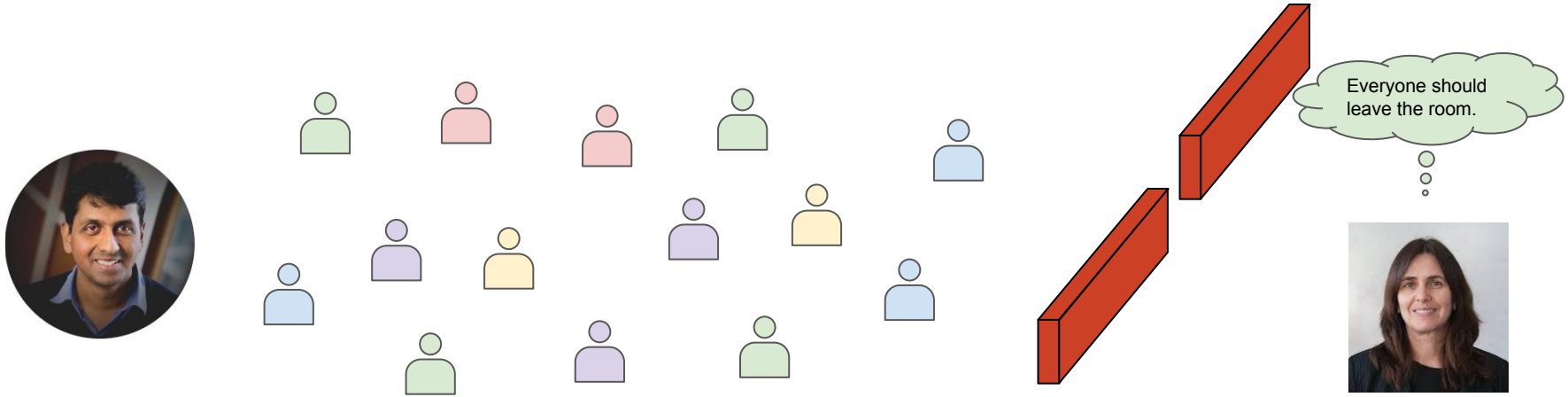


➔ **Size Indexing Sperner Family**: $|Q_i| = i$ for all $i \geq 1$

➔ **Alphabet Size**: Alphabet: $\mathbb{N} \rightarrow \mathbb{N}$. For any i, Alphabet(i) denotes the number of distinct elements in Q_i.

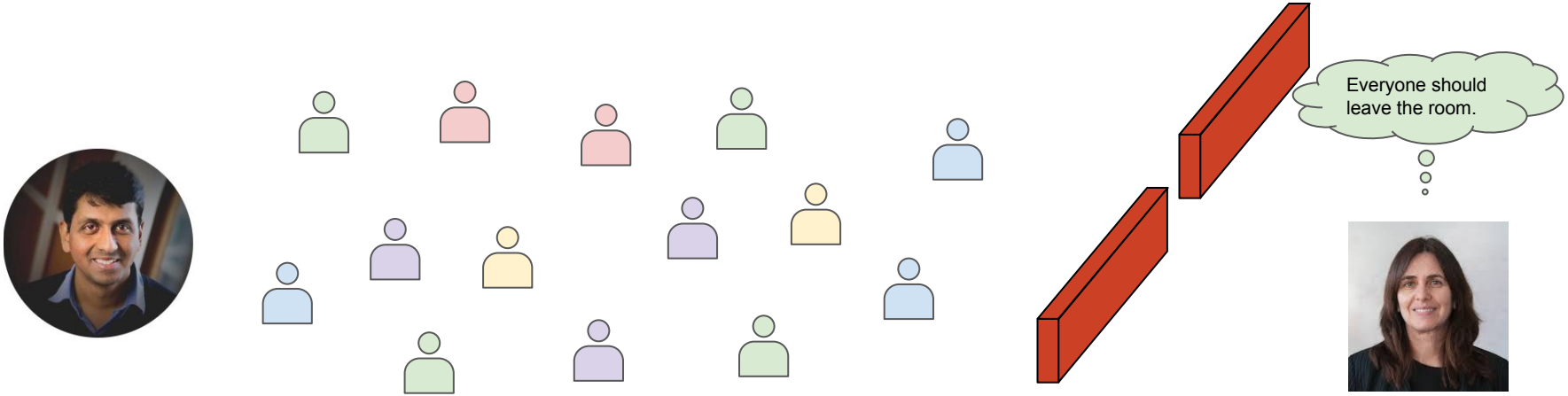
Count-to-Zero Using Size-Indexing Sperner Family

Count-to-Zero Using Size-Indexing Sperner Family



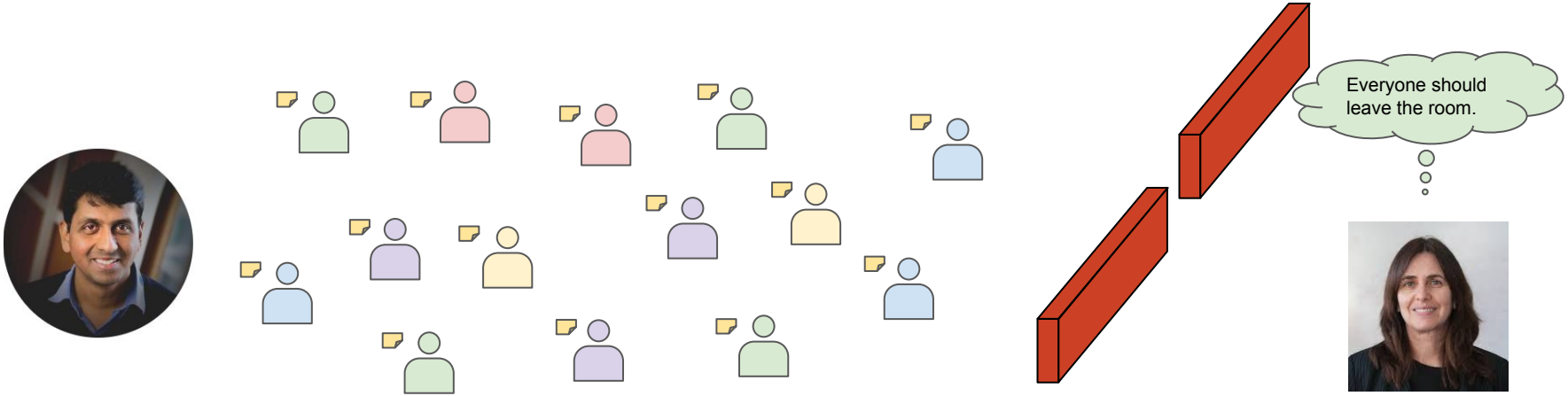
- **Step 1**: Vinod and Shafi agree on a Size-Indexing Sperner Family Q

Count-to-Zero Using Size-Indexing Sperner Family



- **Step 2**: Vinod counts n (the number of people in the room), and chooses the set Q_n in the Sperner Family Q

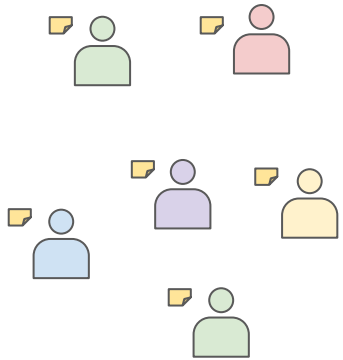
Count-to-Zero Using Size-Indexing Sperner Family

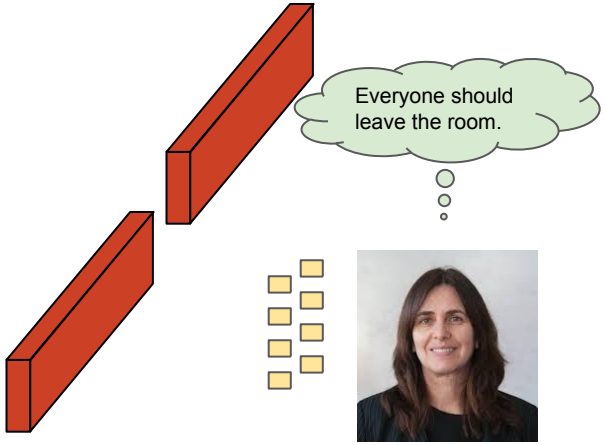


- **Step 3**: Vinod assigns the items from Q_n as tickets to the people in the room.

  (by size-indexing property, the number of items in Q_n is n; each person gets one item)

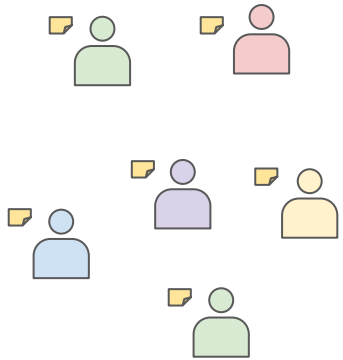
Count-to-Zero Using Size-Indexing Sperner Family

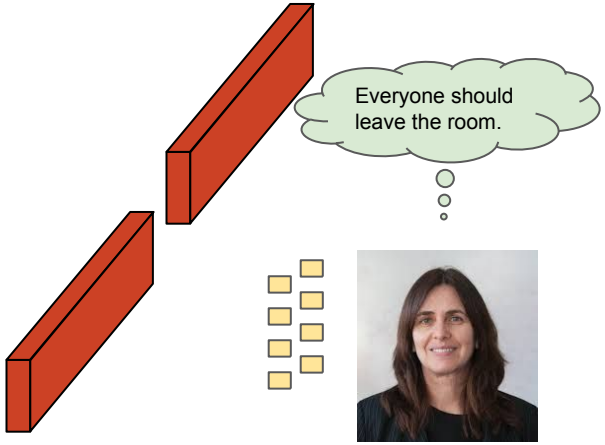




- **Step 4**: People exit the room handing over their assigned items to Shafi.

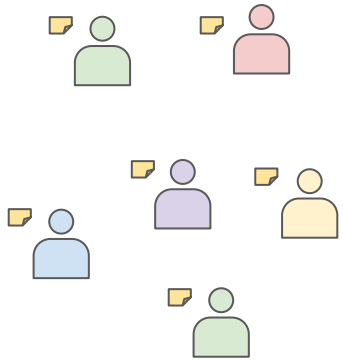
Count-to-Zero Using Size-Indexing Sperner Family

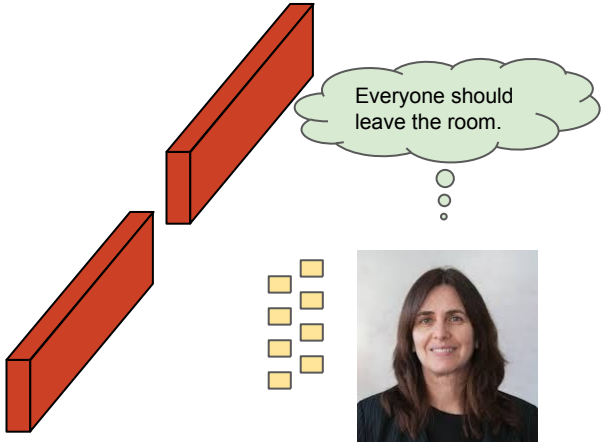




- **Claim:** By looking at the set of received tickets Shafi can determine whether all the people left the room.
- **Proof:** Shafi checks whether the (multi)set S_l of received tickets belong to the sperner family Q.
    - If Yes, then all people left
    - Otherwise, some people still remaining

Count-to-Zero Using Size-Indexing Sperner Family





- **Claim:** By looking at the set of received tickets Shafi can determine whether all the people left the room.
- **Proof:** Since Q is a sperner family, it can never happen that a subset $Q' \subset Q_n$ could be a member of Q.

Count-to-Zero Using Size-Indexing Sperner Family

- Any Size-Indexing Sperner Family can be used to construct a TiLU scheme for Count-to-Zero.
- Tickets contains items from the set Q_n.
- Thus, size(ticket) = log(Alphabet(n)).

Lemma: There exists a Size-Indexing Sperner Family with Alphabet(n) = $A^{-1}(n)$.

Theorem: There exists a TiLU scheme for Count-to-Zero with $C_s = O(1)$, $C_t = O(\log(A^{-1}(n)))$.

Overview

- Introduction
- Learning-Unlearning Schemes
- Ticketed Learning-Unlearning Schemes
  - Mergeable Hypothesis Classes
- Count-to-Zero
- **Conclusion**

Summary of Results

We were able to improve the dependence on n to $A^{-1}(n)$. But, can we completely remove the dependence on n?

Bounded VC dimension

Theorem (Lower Bound): There exists a class $\mathcal{H}$ with VC dimension O(1) such that any learning-unlearning scheme needs to store Ω(n) bits in central memory

Bounded Star Number (Mergeable Classes)

Theorem: If $\mathcal{H}$ has star number S, then $\mathcal{H}$ has TiLU scheme with $C_s = \log |\mathcal{H}|$, $C_t = O(S \log n \log |\mathcal{X}|)$

Improvements for Specific Classes

Theorem: There exists TiLU schemes for :

- Point Functions with $C_s = \log |\mathcal{X}|$, $C_t = O(\log (A^{-1}(n)))$.
- Threshold with $C_s = \log |\mathcal{X}|$, $C_t = O(\log (A^{-1}(n)))$.
- Product of d-thresholds with $C_s = \log |\mathcal{X}|$, $C_t = O(\log |\mathcal{X}| + d\log (A^{-1}(n)))$.

Lower Bounds

Takeaway: Can not completely remove the dependence on n from both C_s and C_t !

Non-trivial Hypothesis Class

Theorem: For any non-trivial hypothesis class (that contains at least two distinct functions), there does not exist a TiLU scheme with both $C_s = O(1)$, $C_t = O(1)$.

Count-to-Zero

Lemma: There exists no TiLU scheme for Count-to-Zero with both $C_s = O(1)$, $C_t = O(1)$.

Size-Indexing Sperner Families

Lemma: For any constant $a \in \mathbb{N}$, there does not exist any size-indexing sperner family with Alphabet(n) $\leq a$ for all $n > 0$.

Open Problem: Can we get TiLU schemes (for even 1D thresholds) with $C_t = O(1)$, while $C_s = O(\log(n))$!

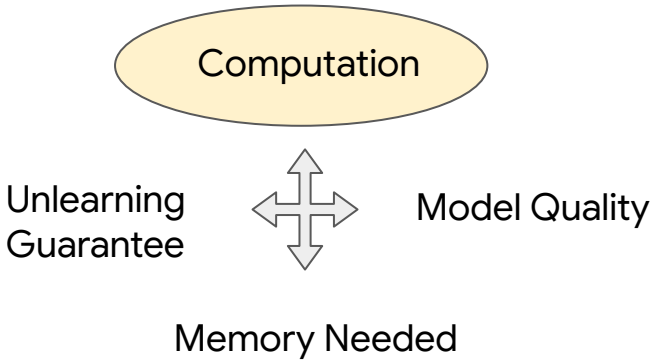
Summary

- ➔ Considered the **memory viewpoint** of Machine Unlearning.
- ➔ **Ticketed Learning-Unlearning Schemes:**
  - ◆ **A new model** for spreading information amongst all participating agents.
  - ◆ Users have to **pay the price of storing (small) additional tickets** on their own devices to get the benefits of unlearning.
- ➔ Understanding the space complexity in terms of combinatorial dimension of the hypothesis class: VC Dimension, Star Number, etc.
- ➔ Improved TiLU schemes for certain classes using Count-to-Zero problem
  - ◆ Construction of Size-indexing Sperner families with small alphabet size.

Barely scratched the surface !!

Future Directions

- Connections to Privacy?
- Approximate unlearning
- Multi-shot unlearning. Iterative addition and removals.
- Unlearning “concepts” instead of “samples”
- Computation
- Deep learning settings? LLMs?
- Broader Consequcences of Unlearning



Thank You! Questions?

Consequences of Unlearning

➔ "**The Space Complexity of Learning-Unlearning Schemes**" (under submission), with Yeshwanth Cherapanamjeri (MIT), Sumegha Garg (Rutgers), Nived Rajaraman (UC Berkeley), and Abhishek Shetty (MIT)

➔ What about VC Classes (beyond Mergeable classes)?

- ◆ Is finite VC dimension sufficient for low memory central / ticketed schemes?
  - NO! 😕

➔ Any combinatorial characterizations?

- ◆ **Eluder Dimension**
  - Lower bound for central schemes
  - Upper bound for ticketed schemes
- ◆ **Star Number:** Upper bound for finite deletions (even for central memory).

Barely scratched the surface !!

Unlearning in LLMs

- “**The Space Complexity of Learning-Unlearning Schemes**” (under submission), with Yeshwanth Cherapanamjeri (MIT), Sumegha Garg (Rutgers), Nived Rajaraman (UC Berkeley), and Abhishek Shetty (MIT)

- What about VC Classes (beyond Mergeable classes)?
  - Is finite VC dimension sufficient for low memory central / ticketed schemes?
    - NO! 😕

- Any combinatorial characterizations?
  - **Eluder Dimension**
    - Lower bound for central schemes
    - Upper bound for ticketed schemes
  - **Star Number:** Upper bound for finite deletions (even for central memory).

Barely scratched the surface !!

Relaxations

- Bounded Deletions
- Approximate Unlearning
- In-practice?
  - Mention about threats and vulnerabilities
  - Mention about UCD direction, and its applications

Backup Slides

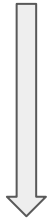
Agnostic Learning of 1D Thresholds (Lower Bound)

- $\mathcal{X}$ = { 1, 2, …, m } and $\mathcal{H}_{th}$ = { h_a = **1**[x > a] for a ∈ {0, 1, … , m} }.
- **Agnostic Setting**: Dataset S may not be realizable by any threshold in $\mathcal{H}_{th.}$
- **Lower Bound**: Any LU scheme requires space C ≥ Ω(min{n, m}) for aux.
- Proof via 1-way communication complexity lower bound of “Indexing”.

Agnostic Learning of 1D Thresholds (Lower Bound)

Vinod

Send aux to Shafi



Shafi

- Given some $X \in \{0, 1\}^n$
- Construct dataset S_X of size O(n).
- Run “Learn” on S_X and get aux.

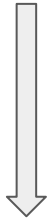
- For any $i \in [n]$, recover X[i] by calling “Unlearn” with a specific S_i.

Shafi can recover entire vector X, thus aux must have $\Omega(n)$ bits !

Agnostic Learning of 1D Thresholds (Lower Bound)

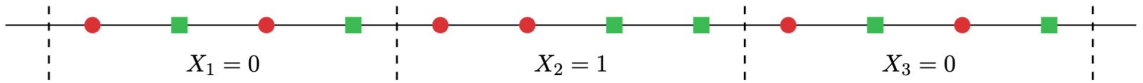
Vinod

Send aux to Shafi

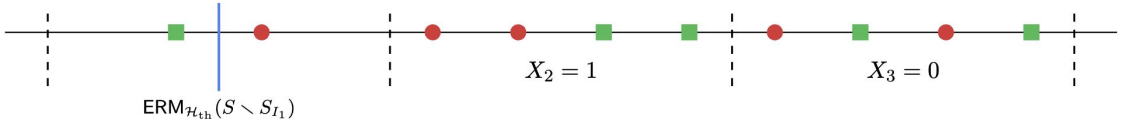


Shafi

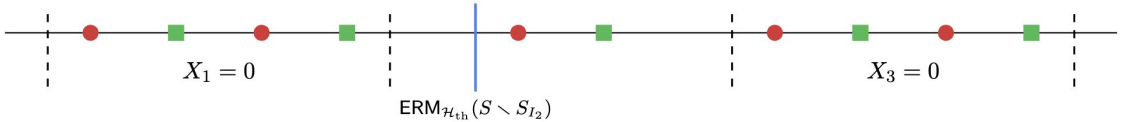
Dataset S constructed by Alice:



Dataset $S \setminus S_{I_1}$:



Dataset $S \setminus S_{I_2}$:



Point Functions (Lower Bound)

- $\mathcal{X}$ = { 1, 2, …, m } and $\mathcal{H}_{pt}$ = { h_a = **1**[x = a] for a ∈ {1, … , m} }.
- **Realizable Case:** There exists a hypothesis in $\mathcal{H}_{pt}$ that realizes S.
- **Lower Bound:** Any LU Scheme requires space C ≥ Ω(min{n, m}).
- Proof via 1-way communication complexity lower bound of “Indexing”.

Point Functions (Lower Bound)

- $\mathcal{X}$ = { 1, 2, …, m } and $\mathcal{H}_{pt}$ = { h_a = **1**[x = a] for a ∈ {1, … , m} }.
- **Realizable Case:** There exists a hypothesis in $\mathcal{H}_{pt}$ that realizes S.
- **Lower Bound:** Any LU Scheme requires space C ≥ Ω(min{n, m}).
- Proof via 1-way communication complexity lower bound of “Indexing”.
- Main Insight:
  - If the dataset has all zero labels, need to choose a canonical h_a.
  - But this can change after some deletions, so need to remember the entire dataset.
  - **Key challenge is to remember the count of each (x, 0) data samples to update** h_a.