MACHINE UNLEARNING

Based on: https://arxiv.org/pdf/1912.03817.pdf

SISA Training

Sharded, Isolated, Sliced, and Aggregated

Gene Olafsen

CREDIT

- Authors: Lucas Bourtoule, Varun Chandrasekaran, Christopher Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, Nicolas Papernot
- Institutions: University of Toronto, Vector Institute, University of Wisconsin-Madison

THE "RIGHT TO BE FORGOTTEN"

- General Provision: "Right to be forgotten" mandates that companies take “reasonable steps” to achieve “the erasure of personal data concerning” an individual.

LEGISLATION

- European Union: General Data Protection Regulation (GDPR)
- US: California Consumer Privacy Act
- Canada: PIPEDA privacy legislation

A MACHINE LEARNING ISSUE

- Removing information from a trained model is very difficult, and usually involves removing the data element(s) from the training set and retraining the model(s).
- Need to remove the contribution to model parameter updates of an individual training point.
- This isn't like removing a 'row' from some database tables and having the deletion occur in 'realtime' from the hosted application.

ERASER

- The procedure of removing traces of an individual's data is more a 'sanitizing' operation.
- A person may leave fingerprints, DNA, hair, etc. when visiting a room or other physical location.
- In a sense, the same thing happens to a machine learning model. Unlearning assures the user, that the model is no longer influenced in any way, by data which the user elected to erase. It's as if the user never entered the room.

REMOVE AND RETRAIN

- Naïve approach: Remove the data in question and retrain the model from scratch.
- This approach suffers from two major drawbacks:
  - Large computational cost.
  - A certain abount of training time necessary for the model to converge.

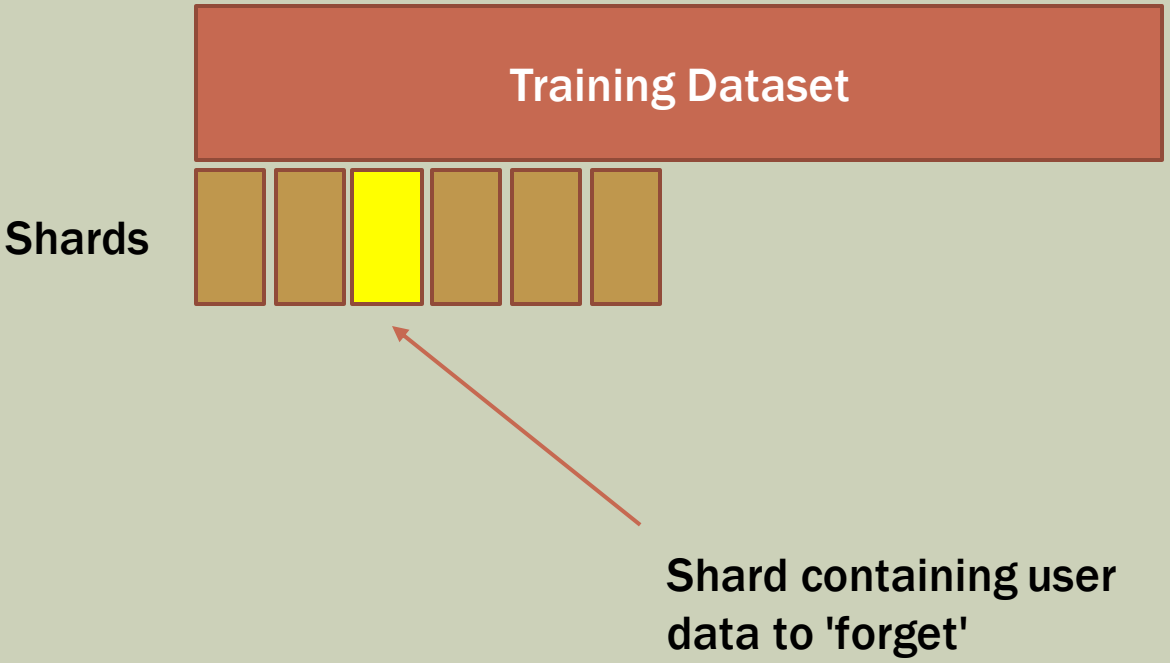
ENTER 'SISA'

- SISA is a framework that decreases the number of model parameters affected by an unlearning request and caches intermediate outputs of the training algorithm to limit the number of model updates that need to be computed to have these parameters unlearn.

SISA OVERVIEW

- SISA is designed to minimally impact existing learning pipelines.
- 1. Divide the training set into multiple *shards*. Ideally a training point only exists in a single shard- or at least a small number of shards.
- 2. Train isolated models on each shard. A training point's influence is only contained in the models trained with the shard(s).
- 3. Unlearning requires only the models with containing the affected shards to be retrained.

USER DATA SHARD



PROPOSED ADVANTAGE

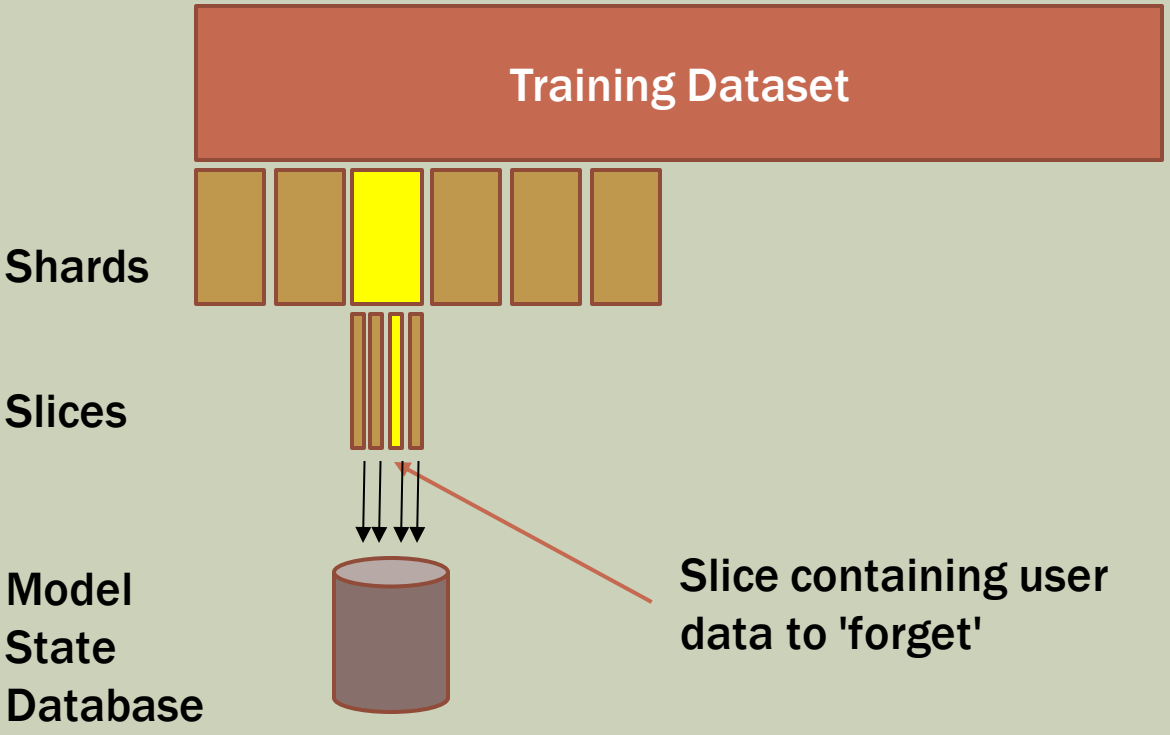
- Training time is decreased over using the full training set, because shards are smaller than the entire training set.

SLICING A SHARD

- It is possible to further reduce training time by dividing a shard into smaller pieces or 'slices'.

- 1. Divide a Shard further into Slices
- 2. Save model parameter state between training a slice.
- 3. When 'unlearning' is required, instead of randomly initialing the model state, apply the model state as it was before the 'slice' containing the unlearnd data point.

USER DATA SLICE



INFERENCING

- Aggregate the predictions of the models trained on each shard to determine the label for each point.

DATASETS

- Two datasets were utilized in the development of the SISA techniques.
  - SVHN
    - Google streetview house numbers
  - Purchase
    - Online shopping purchase decisions

SVHN

- http://ufldl.stanford.edu/housenumbers/
- SVHN is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. It can be seen as similar in flavor to MNIST (e.g., the images are of small cropped digits), but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved, real world problem (recognizing digits and numbers in natural scene images). SVHN is obtained from house numbers in Google Street View images.

- 10 classes, 1 for each digit. Digit '1' has label 1, '9' has label 9 and '0' has label 10.
- 73257 digits for training, 26032 digits for testing, and 531131 additional, somewhat less difficult samples, to use as extra training data
- Comes in two formats:
  - 1. Original images with character level bounding boxes.
  - 2. MNIST-like 32-by-32 images centered around a single character (many of the images do contain some distractors at the sides).

PURCHASE

- https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset
- Online Shoppers Purchasing Intention Dataset Data Set
- Abstract: Of the 12,330 sessions in the dataset, 84.5% (10,422) were negative class samples that did not end with shopping, and the rest (1908) were positive class samples ending with shopping.
- The dataset consists of feature vectors belonging to 12,330 sessions.
- The dataset consists of 10 numerical and 8 categorical attributes.

SHARD PERFORMANCE

- Compared to retraining from scratch, the researchers find that sharding the training data into 20 shards, provides a speed-up of 3.13X and 1.658X for the Purchase and SVHN dataset respectively.
- Assumptions:
  - The number of unlearning requests is 0.003% of the total dataset sizes.
  - Requests are made uniformly across the dataset.

SHARD ACCURACY

- When configured using the most accuracy-preserving settings, SISA training can handle orders of magnitude more unlearning requests than what Google expects would be required to implement the right-to-be-forgotten process.

DEMONSTRATES

- The combination of sharding and slicing does not impact model accuracy.

SLICING

- When the researches added in slicing, they find an additional speed-up of 1.428X and 1.176X for the Purchase and SVHN datasets respectively.
- Assumptions:
  - The number of unlearning requests is 0.001% of the total dataset sizes.

SLICING ACCURACY

- It is validated experimentally that slicing has no impact on accuracy, as it merely reorders data analyzed during training.

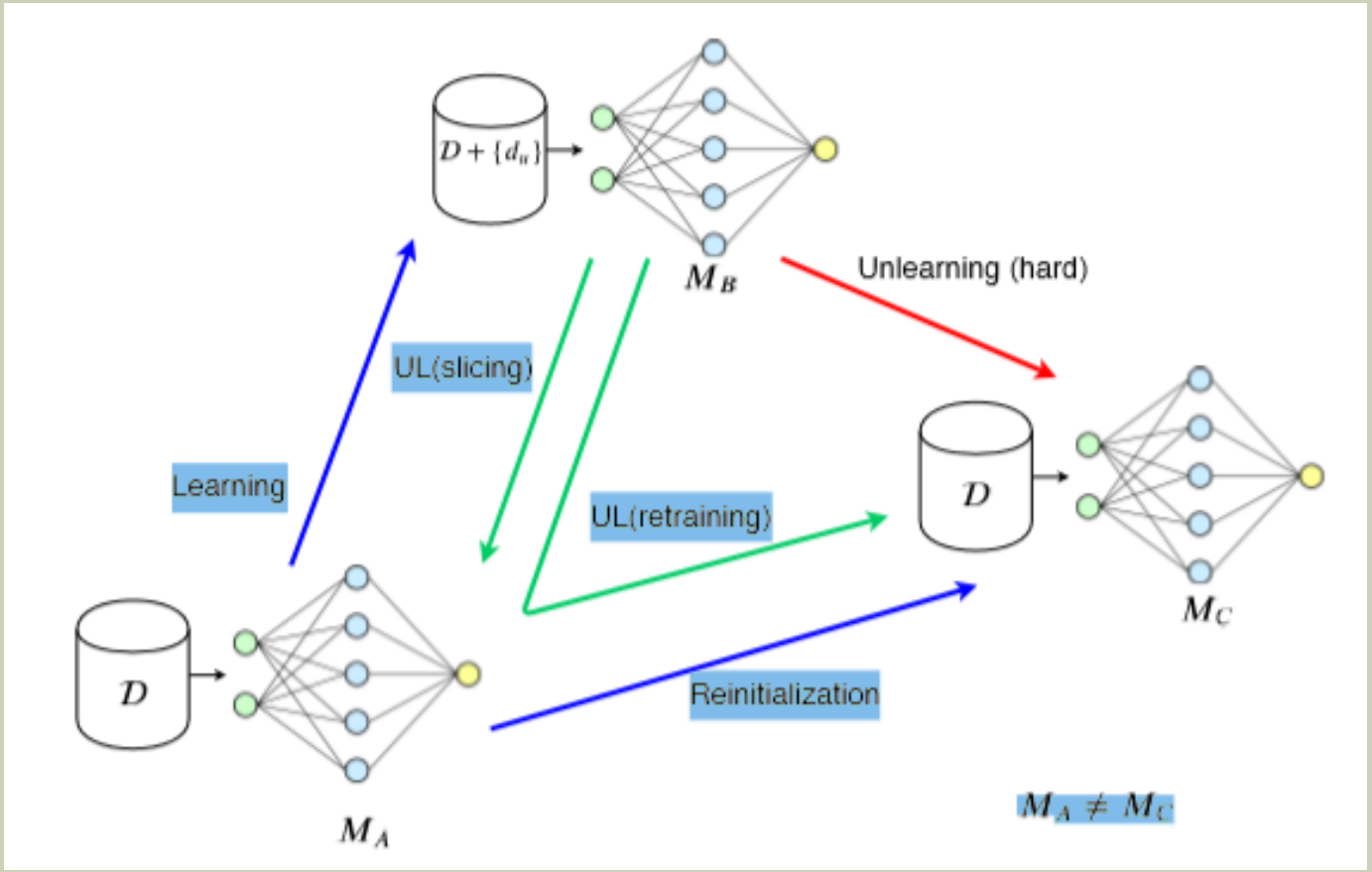
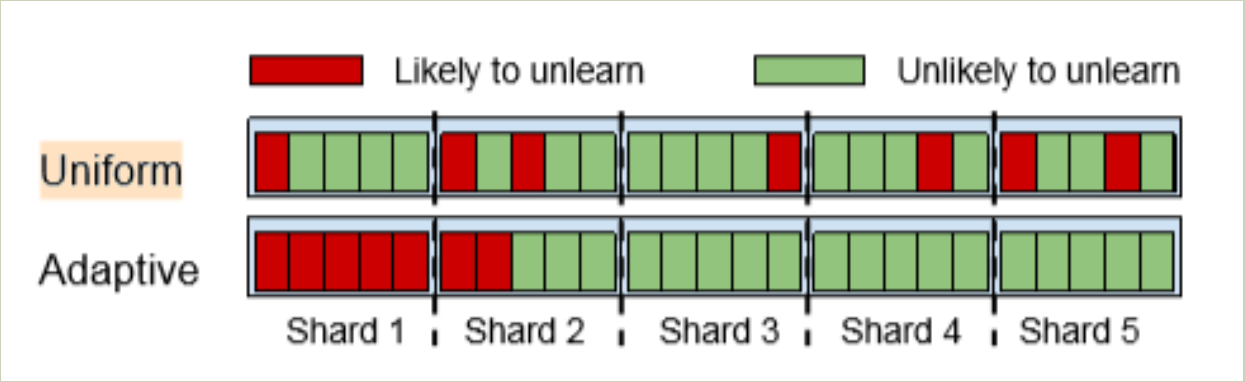

$D + \{d_u\}$
M_B
Unlearning (hard)
UL(slicing)
Learning
UL(retraining)
D
M_C
D
Reinitialization
M_A
$M_A \neq M_C$

FIGURE DESCRIPTION

- From the paper:
- Unlearning (red arrow) is hard because there exists no function that measures the influence of augmenting the dataset D with point du and fine-tuning a model MA already trained on D to train (left blue arrow) a model MB forD+{du}.
- This makes it impossible to revert to model MA without saving its parameter state before learning about du. We call this model slicing (short green arrow).
- In the absence of slicing, one must retrain (curved green arrow) the model without du, resulting in a model MC that is different from the original model MA.

NON-UNIFORM REQUESTS

- Consider a population split between two groups:
  - A group H having a high probability pH of being unlearned.
  - A group L having a low probability pL of being unlearned (i.e. this data is expected to remain in the model for an extended period of time)
- Uniform sharding will intermix groups H and L within a shard.
- Adaptive sharding concentrates group H in a few shards.



DISTRIBUTION AWARE SHARDING

- The creation of shards in a way so as to minimize the time required for retraining.
- Assumptions:
  - The distribution of unlearning requests is known precisely.
  - The distribution is relatively constant over a time interval.
- Data points are sorted in order of erasure probability.
- Shards of unequal size are created.

Input: Dataset $\mathcal{D}$, constant C

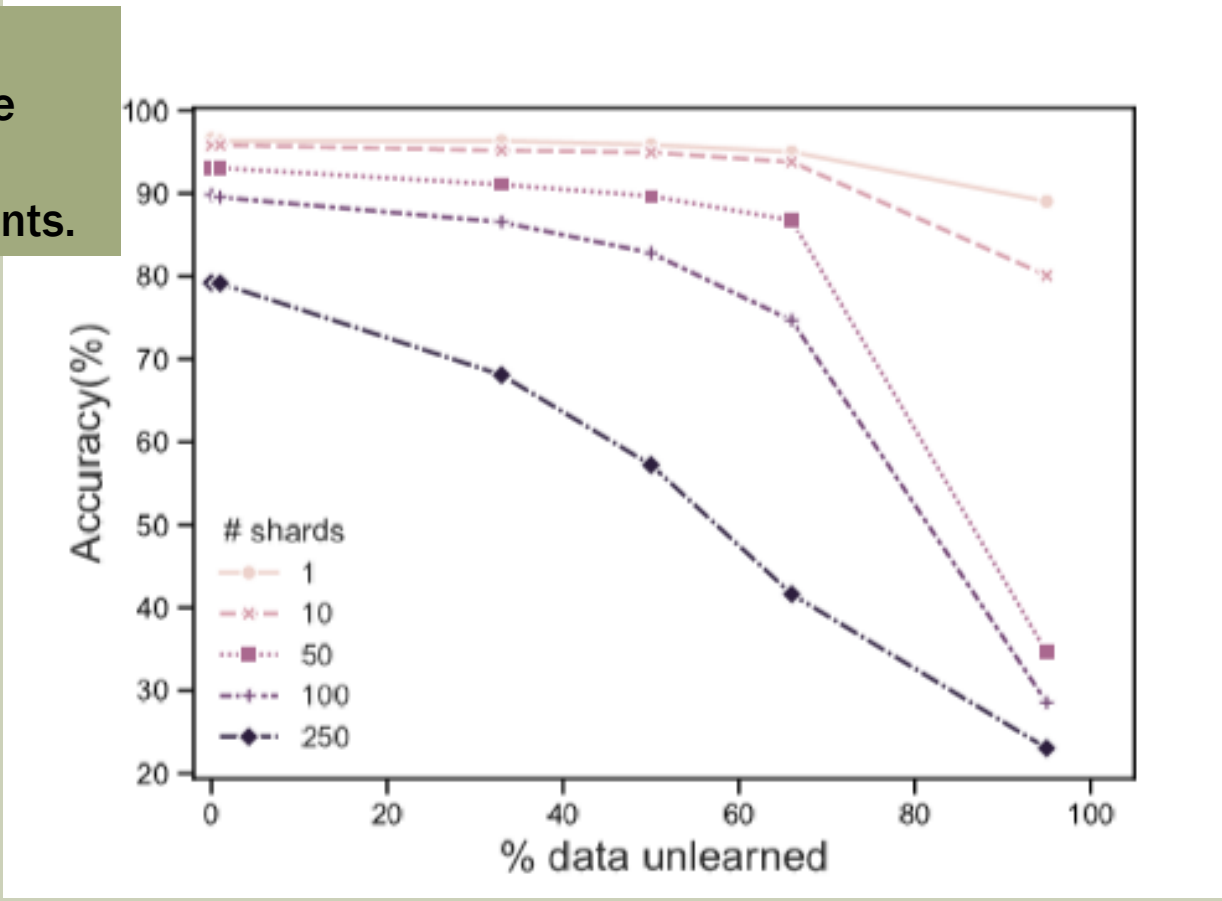
procedure $ShardData(\mathcal{D}, C)$
  sort $\{d_u\}_{i=1}^{|\mathcal{D}|}$ by $p(u)$
  $i \leftarrow 0$
  create empty shard $\mathcal{D}_i$
  for $j \leftarrow 0$ to $|\mathcal{D}|$ **do**
    remove d_u with lowest $p(u)$ from $\mathcal{D}$
    $\mathcal{D}_i = \mathcal{D}_i \cup d_u$
    if $\mathbb{E}[\chi_i] \geq C$ **then**
      $\mathcal{D}_i = \mathcal{D}_i \setminus d_u$
      $i \leftarrow i + 1$
      create empty shard $\mathcal{D}_i$
      $\mathcal{D}_i = \mathcal{D}_i \cup d_u$
    end if
  end for
end procedure

NON-UNIFORM RESULTS

- Expected: The distribution aware strategy decreases the number of points to be retrained.
- Distribution-aware sharding results in about 94.4% prediction accuracy in the regime of unlearning requests which were considered. This is one percent point lower than uniform sharding, at 95.7%.
- Distribution-aware sharding incurs a trade-off of accuracy for decreased unlearning overhead.

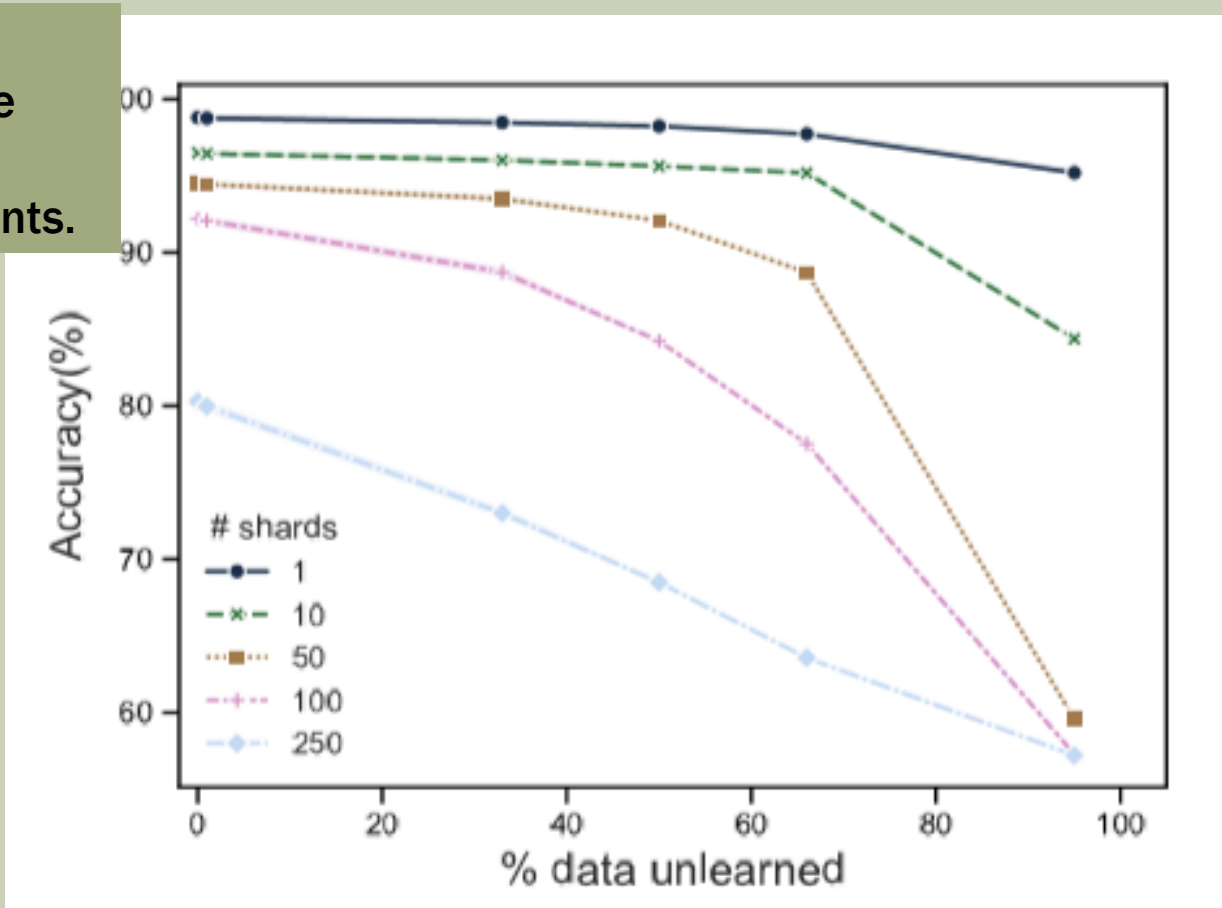
SVHN UNLEARN

Accuracy as a function of the number of unlearned points.

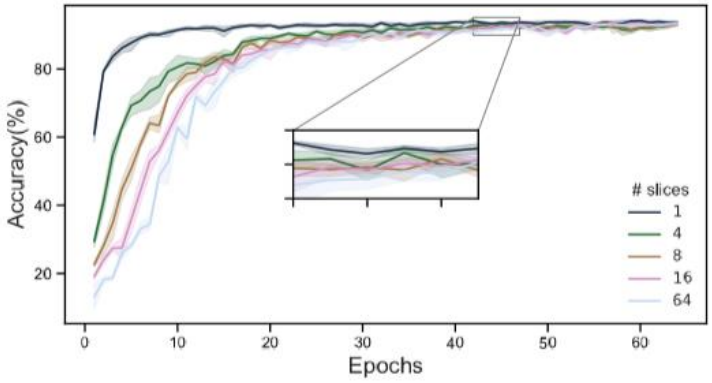


PURCHASE UNLEARN

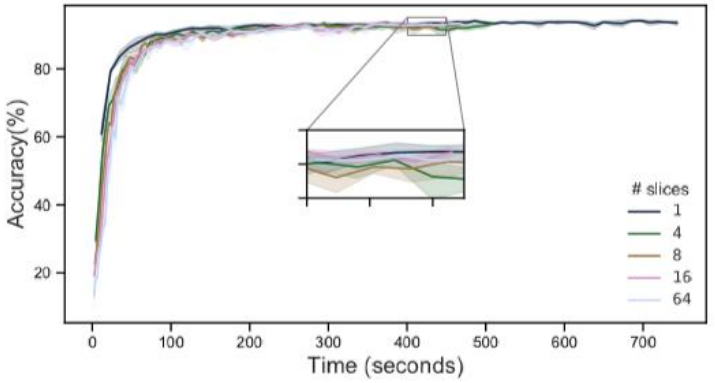
Accuracy as a function of the number of unlearned points.



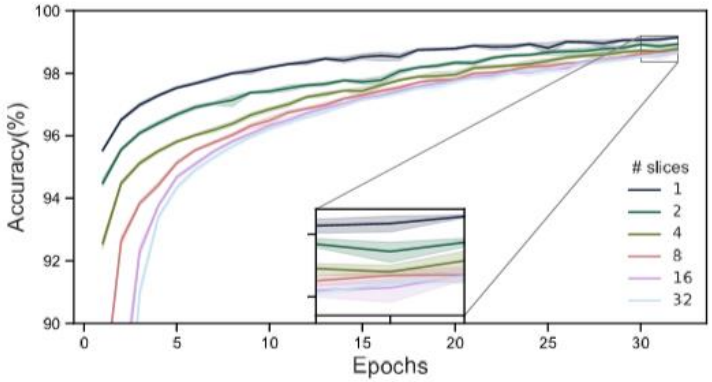
ACCURACY OVER TIME



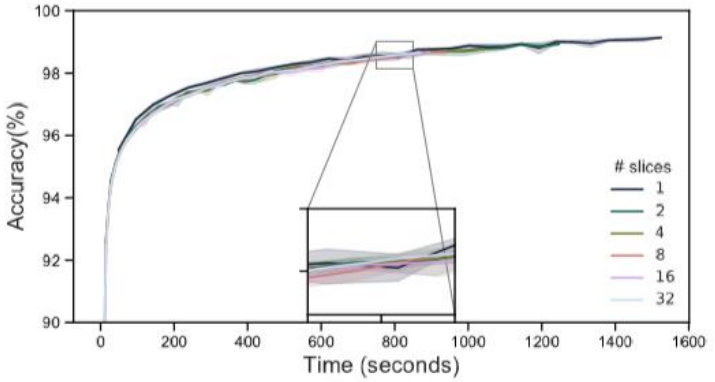
(a) Accuracy vs. # epochs for SVHN dataset.



(b) Accuracy vs. time for SVHN dataset.



(c) Accuracy vs. # epochs for Purchase dataset.



(d) Accuracy vs. time for Purchase dataset.