Deep Unlearning via Randomized Conditionally Independent Hessians

Ronak Mehta[*1] ronakrm@cs.wisc.edu
Sourav Pal[*1] spal9@wisc.edu
Vikas Singh[1] vsingh@biostat.wisc.edu
Sathya N. Ravi[2] sathya@uic.edu

[1]University of Wisconsin-Madison [2]University of Illinois at Chicago

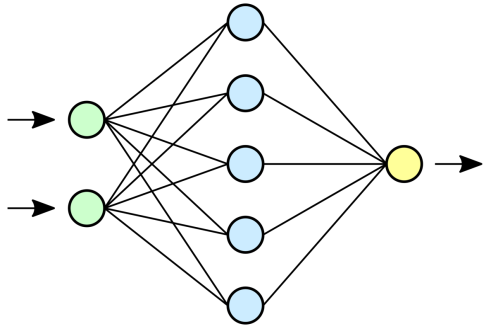
CVPR 2022

Code: https://github.com/vsingh-group/LCODEC-deep-unlearning

Machine Unlearning

$S: \{z_i\}_{\{i=1\}}^{n} \sim D$

Dataset

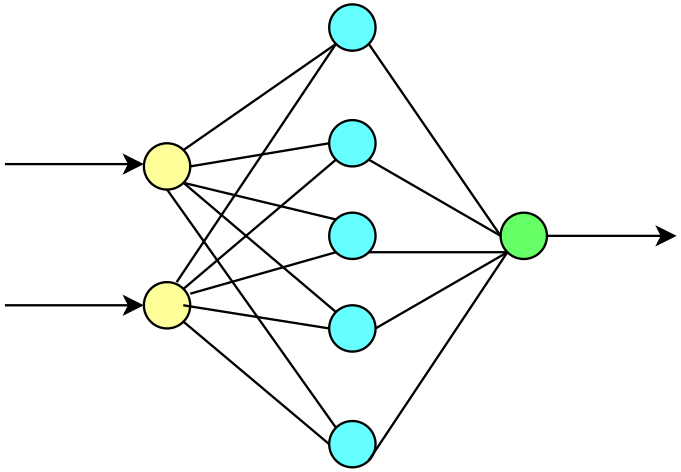


Learning Algorithm

$\hat{w} \in \mathcal{W}$

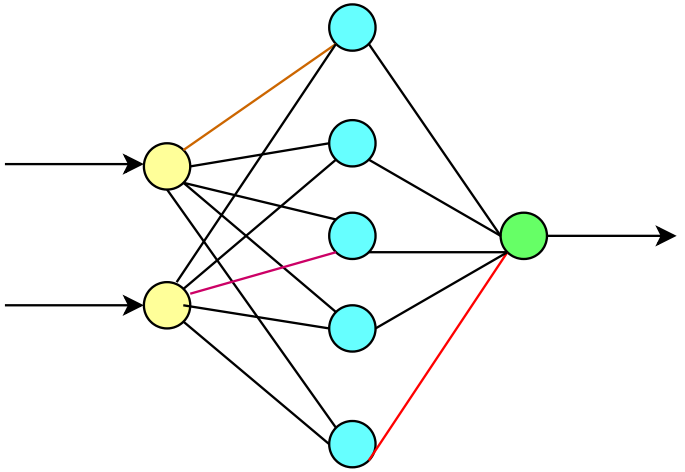
Output Hypothesis

Machine Unlearning



$\widehat{w}$

Remove effect of z'

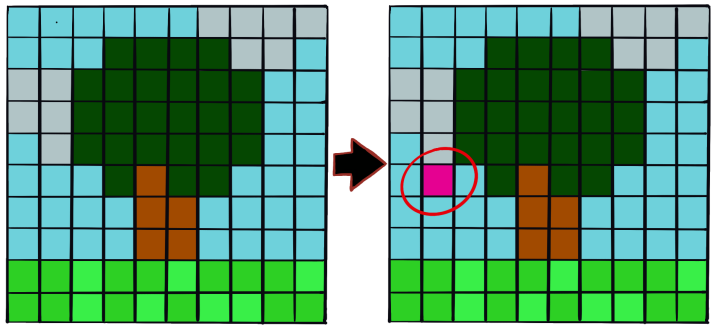


w'

Unlearning Update: $w' = \widehat{w} + g_{\widehat{w}}(z')$

Need for Practical Unlearning

Right To Be Forgotten



Data Poisoning

FTC. California company settles ftc allegations it deceived consumers about use of facial recognition in photo storage app, May 2021.

Is Retraining the Answer?

Huge costs for large scale industrial models

Multiple deletion requests spread across time

Unavailability of original training data

Machine Unlearning

Training Dataset: $\mathcal{S}$

Learning Algorithm: $\mathcal{A}$

Hypothesis Space: $\mathcal{W}$

Unlearning Scheme: $\mathcal{U}$

Deletion Request: $z' \in \mathcal{S}$

An unlearning scheme is (ϵ , δ) forgetting:

$$\mathbb{P}(\mathcal{U}(\mathcal{A}(S), z') \in \mathcal{W}) \leq e^{\epsilon}\mathbb{P}(\mathcal{A}(\mathcal{S} \setminus z') \in \mathcal{W}) + \delta$$

Machine Unlearning

Learning Algorithm, Empirical Risk Minimizer: $\mathcal{A}$

Loss function: f

$$\mathcal{A} : (\mathcal{S}, f) \rightarrow \hat{w}$$

$$\hat{w} = \arg\min F(w) \qquad F(w) = \frac{1}{n} \sum_{i=1}^{n} f(w, z_i)$$

Recall, Unlearning Update: $w' = \hat{w} + g_{\hat{w}}(z')$

Unlearning Scheme

Recall, Unlearning Update: $w' = \hat{w} + g_{\hat{w}}(z')$

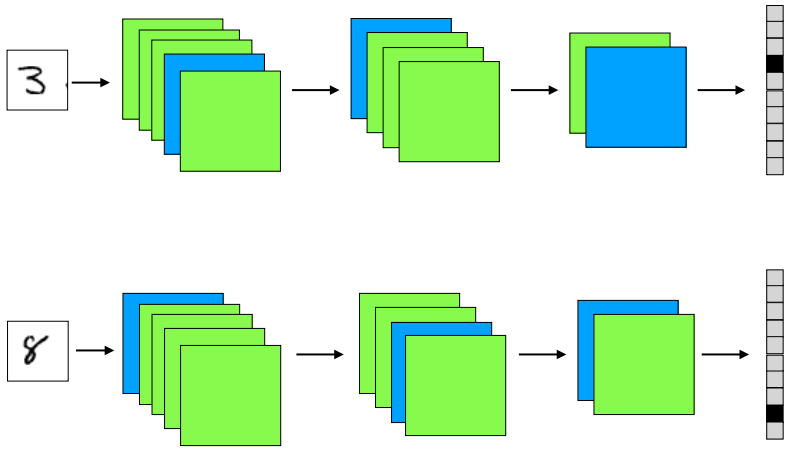
$$g(z') = \frac{1}{n-1} H'^{-1} \nabla f(\hat{w}, z')$$

$$H' = \frac{1}{n-1}\left(n\nabla^2 F(\hat{w}) - \nabla^2 f(\hat{w}, z')\right)$$

The presence of an inverted Hessian makes it necessary to look for approximations that can be instantiated in practical settings with deep neural networks.

Sekhari, A., Acharya, J., Kamath, G., & Suresh, A. T. (2021). Remember what you want to forget: Algorithms for machine unlearning. Advances in Neural Information Processing Systems, 34.

Potential Approximation



Assumption: For all subsets of training samples $S \subset \mathcal{S}$, there exists a subset of trained model parameters $P^* \subset \Theta$ such that

$$f(S) \perp \hat{w}_{\Theta \setminus P^*} | \hat{w}_{P^*}$$

Bottom Line: Find the "*selective*" subset of parameters to update for unlearning a sample.

Potential Approximation

A Naive Approach:

Let $P \subseteq \Theta := \{1, \ldots, d\}$ be the index set of the parameters that are sufficient to update. A direct procedure may be to identify this subset P can be formulated:

$$P = \underset{P \in \mathcal{P}(\Theta)}{\arg\min} \; ||\tilde{w} - \tilde{w}_P||$$

where $\mathcal{P}(\Theta)$ is the power set of elements in Θ i.e. model parameters.

- Exact solution is intractable
- Simple solution based on thresholding still requires full instantiation of the update

A Probabilistic Angle for Selection

Idea: Think about the deep network in $\mathcal{W}$ as a functional on the input space D

Allows us to identify regions in the network that contain most information about the query point (the deletion request)

Finding the subset of parameters to update should satisfy

$$z' \perp w_{\Theta \setminus P} | w_P$$

Digression: CODEC

Let Y be a random variable and $\boldsymbol{X} = (X_1, \ldots, X_p)$ and $\boldsymbol{Z} = (Z_1, \ldots, Z_q)$ be random vectors, with $p \geq 0$ and $q \geq 1$. Then the coefficient denoted as $T(Y, \boldsymbol{Z}|\boldsymbol{X})$ measures the degree of conditional dependence of Y and $\boldsymbol{Z}$ given $\boldsymbol{X}$.

T is 0 ***iff*** Y and $\boldsymbol{Z}$ are conditionally independent given $\boldsymbol{X}$.

T is 1 ***iff*** Y is almost surely equal to a measurable function of $\boldsymbol{Z}$ given $\boldsymbol{X}$.

- *Chatterjee, Sourav. "A new coefficient of correlation." Journal of the American Statistical Association 116.536 (2021): 2009-2022*
- *Azadkia, Mona, and Sourav Chatterjee. "A simple measure of conditional dependence." The Annals of Statistics 49.6 (2021): 3070-3102*

CODEC: Utility and Limitations

- CODEC for efficient subset selection that is also Sufficient for Predictive Purposes

Issues:

- Cost of tie-breaking for very large discrete values is prohibitive
- The deletion request $z' \in \mathcal{S}$, is not a random variable

Noisy approach: L-CODEC

We introduce a randomized version of CODEC, L-CODEC. For random variables A, B, C:

$$T_L := T\left(\tilde{B}, \tilde{C} | \tilde{A}\right)$$

where $\tilde{B} = B + N(0, \sigma^2)$, and similarly for $\tilde{C}, \tilde{A}$

- Consistent with Randomization Criterion for Conditional Independence

Perturbations are key

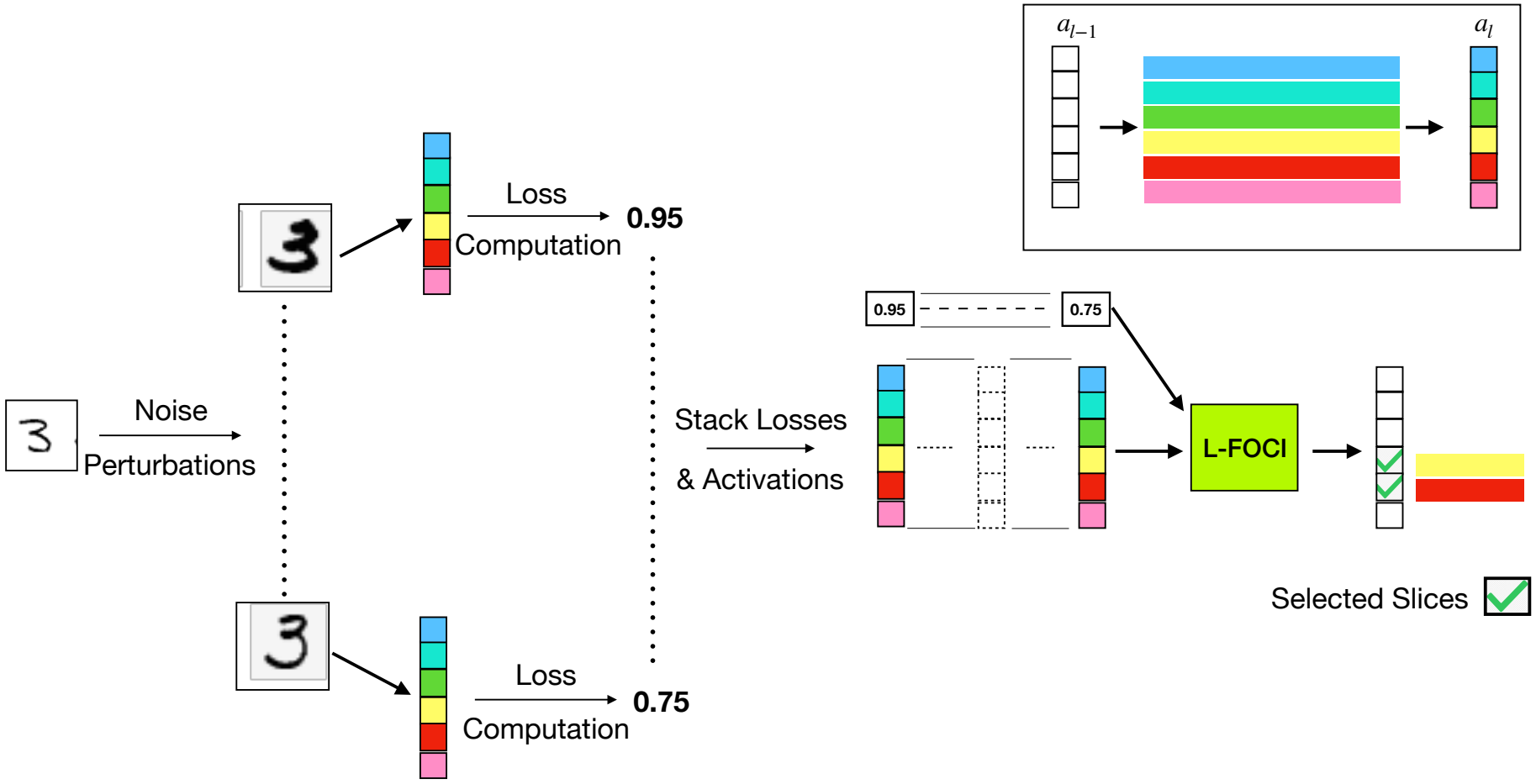
- Neither deletion request nor trained model parameters are random variables
- Perturbation scheme to generate samples for unknown distributions

Measure effect of model parameters on the sample to be deleted using activations, which formulates the conditional independence test as:

$$f(z') \perp a_{\Theta \setminus P} | a_P$$

where a_P for some subset of parameters P is defined as the activation during model forward pass on the deletion request z'

Perturbations are key



Deep Unlearning Algorithm

Algorithm 1: Unlearning via Conditional Dependence Block Selection

Input: A trained model $\hat{w}$, gradient vectors $\nabla_1 F(\hat{w}), \nabla_2 F(\hat{w})$, sample $z' \in$ to unlearn.

Output: Model w' with z' removed

1. **for** $j \in \{0, \ldots, |J|\}$ *perturbations* **do**
   $\xi^j \sim N(0, \sigma^2)$
   $z'^{,j} = z' + \xi^j$
   $l^j, a^j = f(z'^{,j})$
   end
2. Compute $L^* = \text{L-FOCI}(l^J, a^J)$.
3. Compute $\nabla^2 f(\hat{w}, z')$ via finite differences.
4. Update:

$$H'_{L^*} = \frac{1}{n-1}\left(n\nabla^2_{L^*} F(\hat{w}) - \nabla^2_{L^*} f(\hat{w}, z')\right)$$

$$w'_{L^*} = \hat{w}_{L^*} + \frac{1}{n-1} H'^{-1}_{L^*} \nabla f(\hat{w}, z')_{L^*}$$

$$w'_{\setminus L^*} = w'_{\setminus L^*} + N(0, \sigma^2)$$

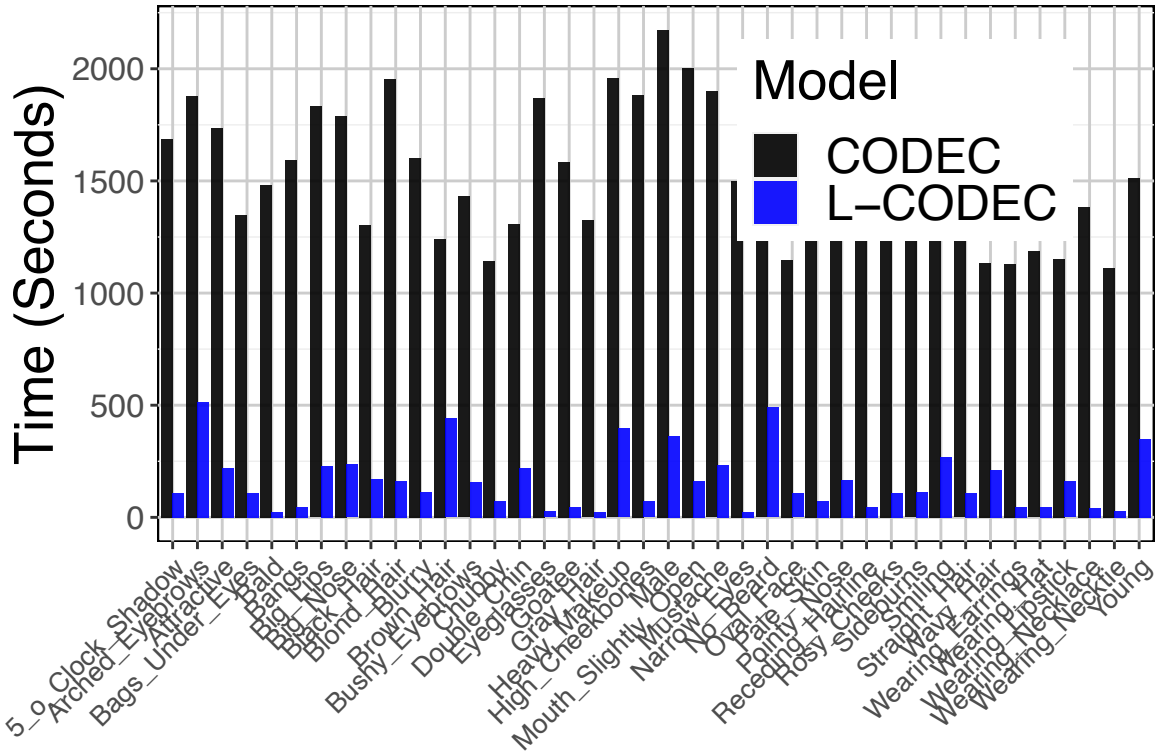
Lemma

Lemma: The gap between the residual gradient norm of the Unlearning update using L-FOCI in Algorithm 1 and a full unlearning update

$$||\nabla \mathcal{L}(w^{'}_{L-FOCI}, D')||_2 - ||\nabla \mathcal{L}(w^{'}_{Full}, D')||_2$$

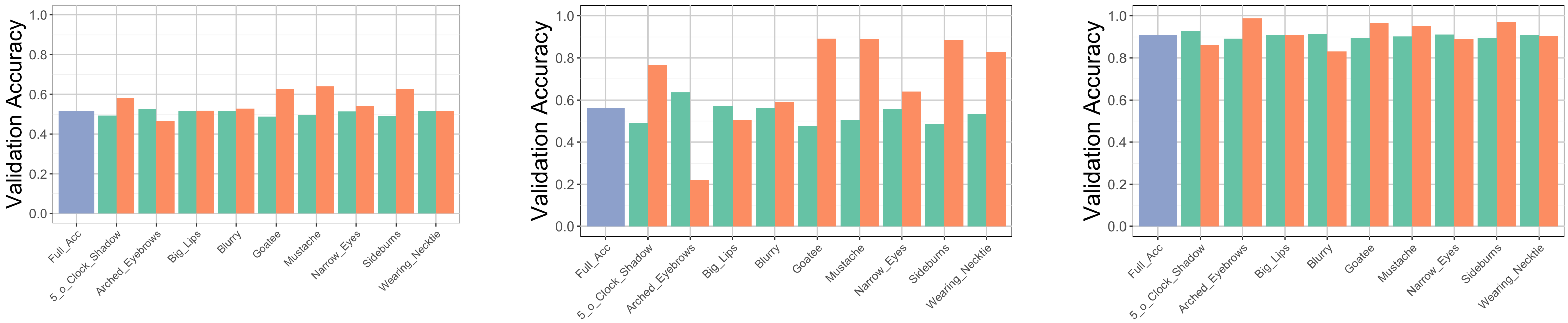
shrinks as $O(1/n^2)$. Here n is size of the original training dataset and D' denotes the residual dataset.

Experiment: L-CODEC Speedup



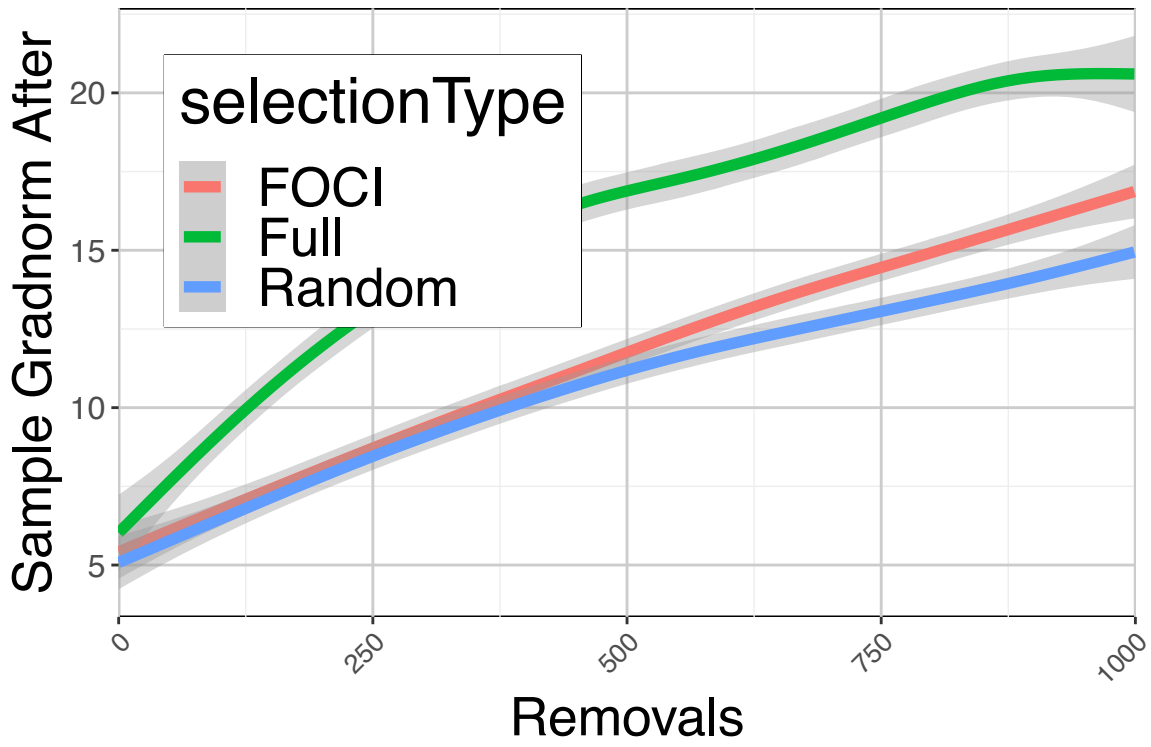
Attributes of CelebA dataset

Experiment: Spurious Feature Regularization CelebA

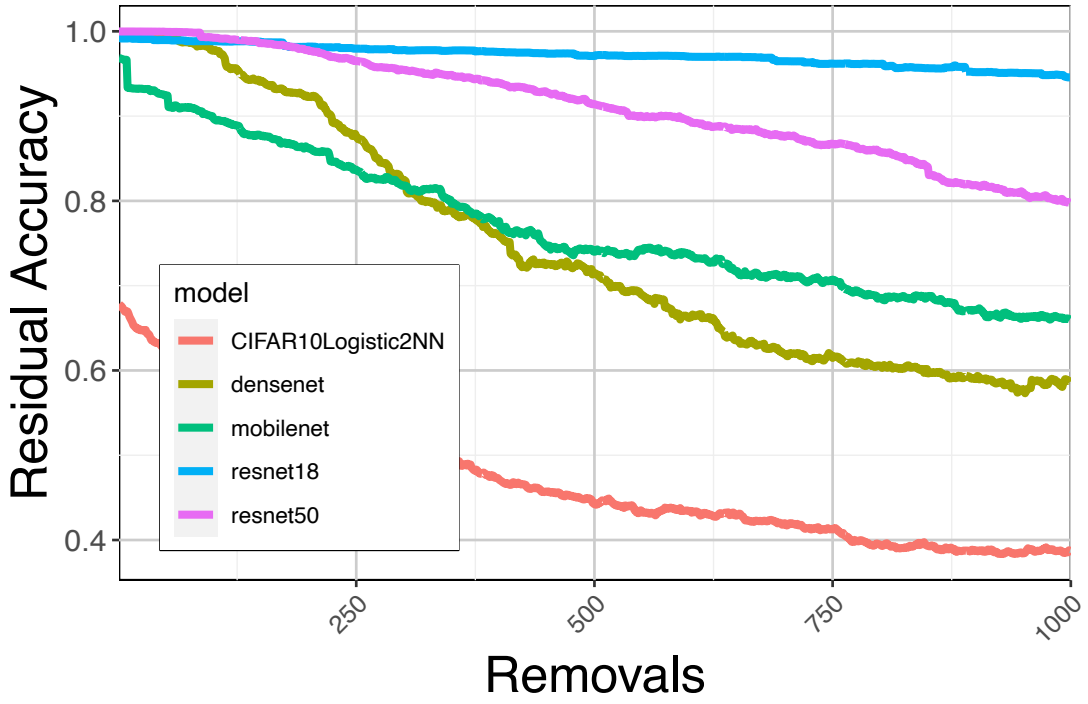


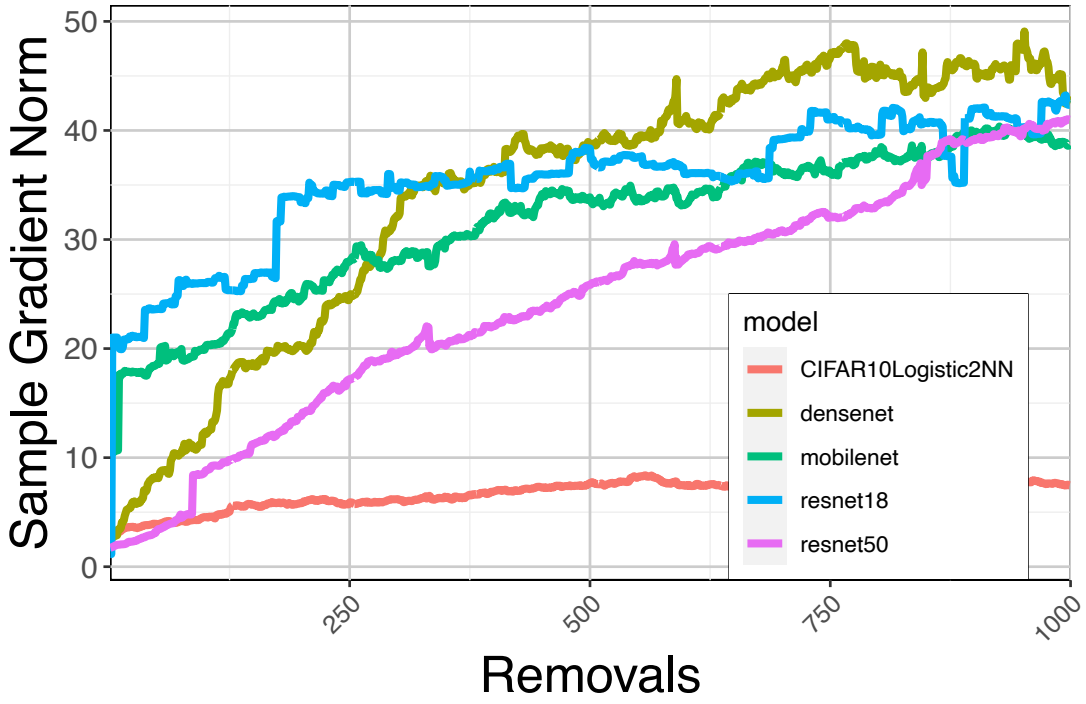
Validation accuracies for a model trained to predict "No Beard" in the CelebA dataset. (L to R) regularization for all features, regularization for a random subset, and regularization via FOCI. Green indicates accuracy on the data with that feature, red, without.

Experiment: MNIST Removals



Experiment: CIFAR-10 Removals





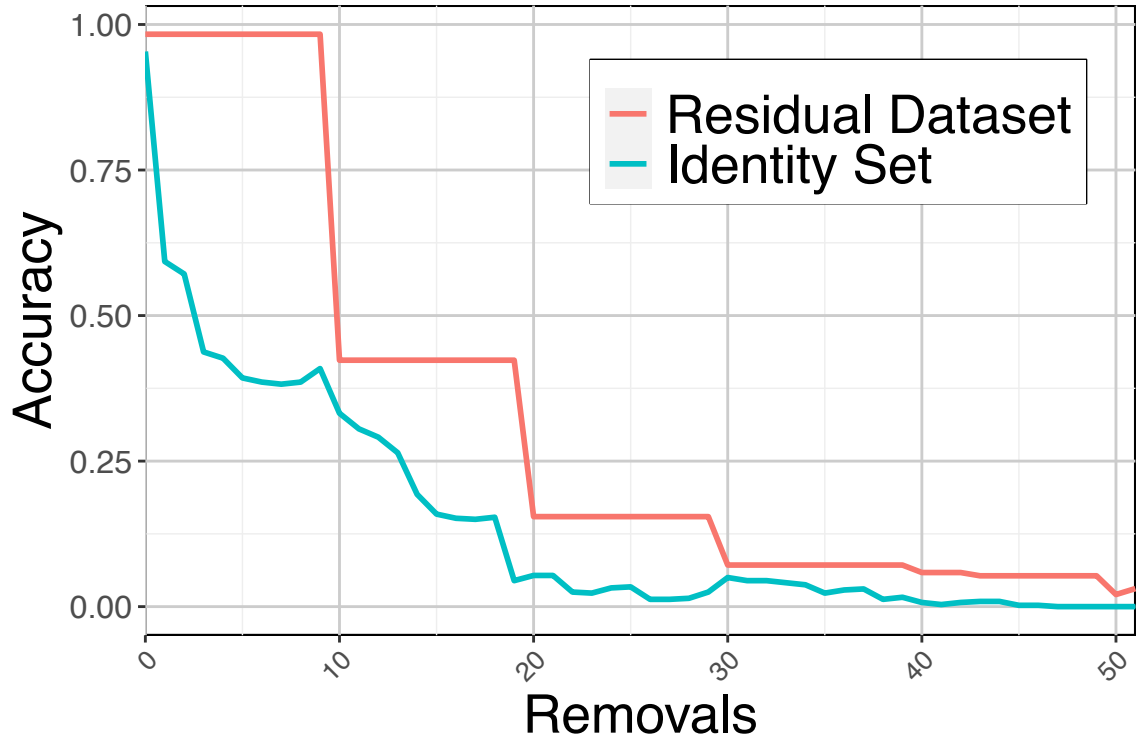
Experiment: Transformers

Dataset: LEDGAR (A multilabel corpus for legal provisions in contracts)
Model: Finetuned DistilBERT

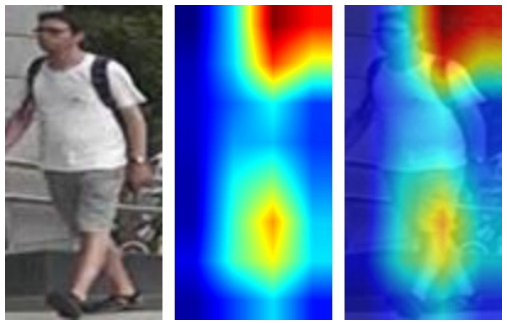
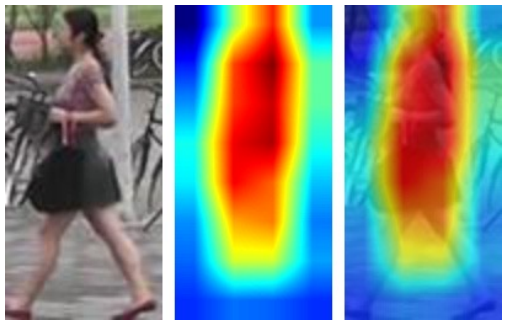
ϵ	# Supported Removals	
	Governing Laws	Terminations
0.1	> 100	> 100
0.01	> 100	> 100
0.001	18	21
0.0005	6	7

Higher ϵ , means lower privacy guarantees, and thus support more removals

Experiment: VGGFace Recognition

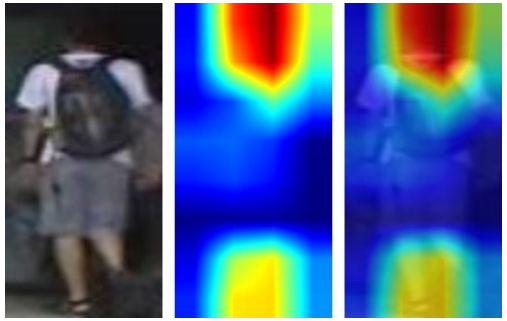
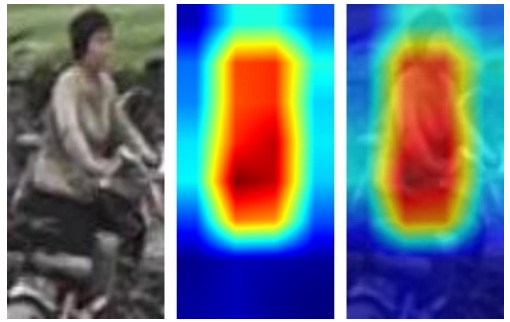


Experiment: Person Re-Identification

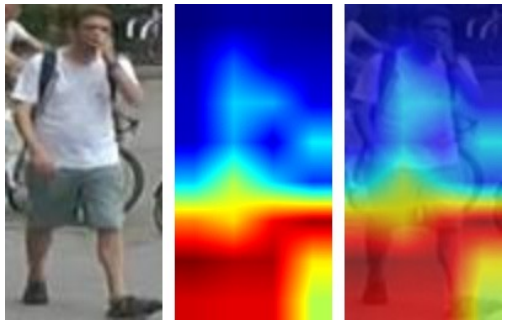
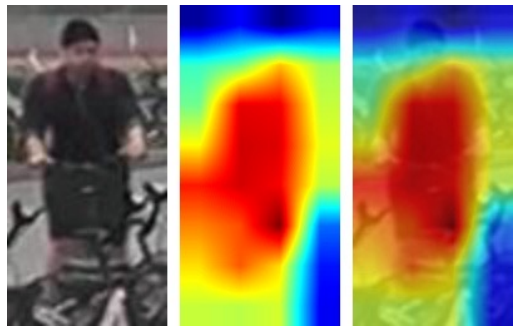



ResNet50

Market1501

MLFN

MobileNet-V2

Scrubbed Sample

Residual Sample

Conclusion and Takeaways

- We propose a selection algorithm to identify sufficient subset of parameters
- Proposed algorithm significantly reduces the computational burden of standard 2^{nd} order unlearning updates
- Unlearning from Deep Neural Networks is now feasible as demonstrated by experiments

Thank you

Code: https://github.com/vsingh-group/LCODEC-deep-unlearning