Machine Unlearning under Overparameterization

Jacob L. Block
UT Austin

Aryan Mokhtari
UT Austin and Google Research

Sanjay Shakkottai
UT Austin

NeurIPS 2025, San Diego, USA

The Unlearning Problem

Problem: Modify a model θ^* trained on dataset $\mathcal{D}$ to forget a subset of samples $\mathcal{D}_f$ and produce a new model as if it had been trained only on $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$.

Challenges of Overparameterized Unlearning

- Prior work defines the unlearning solution as the loss minimizer over $\mathcal{D}_r$ [Guo et al.'20; Bae et al. '22; Sekhari et al.'21; . . .]

Challenges of Overparameterized Unlearning

- Prior work defines the unlearning solution as the loss minimizer over $\mathcal{D}_r$ [Guo et al.'20; Bae et al. '22; Sekhari et al.'21; ...]
- Overparameterized setting: Original model $\boldsymbol{\theta}^*$ is already a loss minimizer over $\mathcal{D}_r$

Challenges of Overparameterized Unlearning

- Prior work defines the unlearning solution as the loss minimizer over $\mathcal{D}_r$ [Guo et al.'20; Bae et al. '22; Sekhari et al.'21; ...]
- Overparameterized setting: Original model $\boldsymbol{\theta}^*$ is already a loss minimizer over $\mathcal{D}_r$
- Existing unlearning algorithms perturb $\boldsymbol{\theta}^*$ using loss gradients over $\mathcal{D}_r$ and $\mathcal{D}_f$ [Neel et al.'21; Graves et al. '20; Kurmanji et al. '23; ...]

Challenges of Overparameterized Unlearning

- Prior work defines the unlearning solution as the loss minimizer over $\mathcal{D}_r$ [Guo et al.'20; Bae et al. '22; Sekhari et al.'21; ...]
- Overparameterized setting: Original model $\boldsymbol{\theta}^*$ is already a loss minimizer over $\mathcal{D}_r$
- Existing unlearning algorithms perturb $\boldsymbol{\theta}^*$ using loss gradients over $\mathcal{D}_r$ and $\mathcal{D}_f$ [Neel et al.'21; Graves et al. '20; Kurmanji et al. '23; ...]
- Overparameterized setting: Loss gradients vanish to $\mathbf{0}$ over all of $\mathcal{D}$

Challenges of Overparameterized Unlearning

- Prior work defines the unlearning solution as the loss minimizer over $\mathcal{D}_r$ [Guo et al.'20; Bae et al. '22; Sekhari et al.'21; ...]
- Overparameterized setting: Original model $\boldsymbol{\theta}^*$ is already a loss minimizer over $\mathcal{D}_r$
- Existing unlearning algorithms perturb $\boldsymbol{\theta}^*$ using loss gradients over $\mathcal{D}_r$ and $\mathcal{D}_f$ [Neel et al.'21; Graves et al. '20; Kurmanji et al. '23; ...]
- Overparameterized setting: Loss gradients vanish to $\mathbf{0}$ over all of $\mathcal{D}$

- We introduce:
  - A new definition for the unlearning solution
  - An algorithmic framework to recover it

Defining Unlearning under Overparameterization

- **Goal:** Find the *simplest* model $\boldsymbol{\theta}$ that minimizes the loss over the retain set $\mathcal{J}(\boldsymbol{\theta}; \mathcal{D}_r)$.

- **Definition:** For a model complexity measure $R(\boldsymbol{\theta})$, define the unlearning solution:

$$\boldsymbol{\theta}_r^* \in \operatorname*{argmin}_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) \quad \text{s.t.} \quad \boldsymbol{\theta} \in \operatorname*{argmin}_{\boldsymbol{\theta}'} \mathcal{J}(\boldsymbol{\theta}'; \mathcal{D}_r)$$

- **Key Properties:**
  - $\boldsymbol{\theta}_r^*$ reflects information only from $\mathcal{D}_r$ (for suitable R)
  - Generalizes well

Solving for $\boldsymbol{\theta}_r^*$

- Relax the constraint to its first-order approximation centered at $\boldsymbol{\theta}^*$

$$\min_{\boldsymbol{\Delta}} R(\boldsymbol{\theta}^* + \boldsymbol{\Delta}) \quad \text{s.t.} \quad \boldsymbol{\Delta} \perp \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{x}_i) \quad \forall (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}_r$$

Solving for $\boldsymbol{\theta}_r^*$

- Relax the constraint to its first-order approximation centered at $\boldsymbol{\theta}^*$

$$\min_{\boldsymbol{\Delta}} R(\boldsymbol{\theta}^* + \boldsymbol{\Delta}) \quad \text{s.t.} \quad \boldsymbol{\Delta} \perp \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{x}_i) \quad \forall (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}_r$$

- Apply regularization $\hat{R}$ to drift variable $\boldsymbol{\Delta}$

Solving for $\boldsymbol{\theta}_r^*$

- Relax the constraint to its first-order approximation centered at $\boldsymbol{\theta}^*$

$$\min_{\boldsymbol{\Delta}} R(\boldsymbol{\theta}^* + \boldsymbol{\Delta}) \quad \text{s.t.} \quad \boldsymbol{\Delta} \perp \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{x}_i) \quad \forall (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}_r$$

- Apply regularization $\hat{R}$ to drift variable $\boldsymbol{\Delta}$
- Solve relaxed, regularized problem:

$$\tilde{\boldsymbol{\Delta}} = \operatorname*{argmin}_{\boldsymbol{\Delta}} R(\boldsymbol{\theta}^* + \boldsymbol{\Delta}) + \hat{R}(\boldsymbol{\Delta}) \quad \text{s.t.} \quad \boldsymbol{\Delta} \perp \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{x}_i) \quad \forall (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}_r$$

Solving for $\boldsymbol{\theta}_r^*$

- Relax the constraint to its first-order approximation centered at $\boldsymbol{\theta}^*$

$$\min_{\boldsymbol{\Delta}} R(\boldsymbol{\theta}^* + \boldsymbol{\Delta}) \quad \text{s.t.} \quad \boldsymbol{\Delta} \perp \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{x}_i) \quad \forall (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}_r$$

- Apply regularization $\hat{R}$ to drift variable $\boldsymbol{\Delta}$
- Solve relaxed, regularized problem:

$$\tilde{\boldsymbol{\Delta}} = \operatorname*{argmin}_{\boldsymbol{\Delta}} R(\boldsymbol{\theta}^* + \boldsymbol{\Delta}) + \hat{R}(\boldsymbol{\Delta}) \quad \text{s.t.} \quad \boldsymbol{\Delta} \perp \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^*, \boldsymbol{x}_i) \quad \forall (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}_r$$

- Updated model is then $\boldsymbol{\theta}^* + \tilde{\boldsymbol{\Delta}}$

Theoretical Guarantees

Given a **model class** and **complexity measure** $R(\theta)$:

- We construct a specific drift regularizer $\hat{R}(\boldsymbol{\Delta})$
- Show that solving the relaxed problem gives an exact/approximate solution to the exact unlearning problem

Theoretical Guarantees

- **Linear Models:**
  - Any complexity measure R
  - For $\hat{R} = \mathbf{0}$, the solution to the relaxed problem satisfies **exact unlearning**

Theoretical Guarantees

- **Linear Models:**
  - Any complexity measure R
  - For $\hat{R} = \mathbf{0}$, the solution to the relaxed problem satisfies **exact unlearning**
- **Linear Networks:**
  - Complexity measure $R =$ operator norm
  - For a suitable $\hat{R}$, the solution to the relaxed problem again satisfies **exact unlearning**

Theoretical Guarantees

- **Linear Models:**
  - Any complexity measure R
  - For $\hat{R} = \mathbf{0}$, the solution to the relaxed problem satisfies **exact unlearning**
- **Linear Networks:**
  - Complexity measure $R =$ operator norm
  - For a suitable $\hat{R}$, the solution to the relaxed problem again satisfies **exact unlearning**
- **2-Layer Perceptrons:**
  - Complexity measure $R =$ network width
  - For a specific $\hat{R}$, the relaxed solution **interpolates** $\mathcal{D}_r$ with **width** $\leq |\mathcal{D}_r|$

Practical Algorithm

- We instantiate our framework as a practical algorithm MinNorm-OG which
  - Accesses batches of retain set samples
  - Employs an iterative update

Practical Algorithm

- We instantiate our framework as a practical algorithm MinNorm-OG which
  - Accesses batches of retain set samples
  - Employs an iterative update
- Set $R(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$ and $\hat{R}(\boldsymbol{\Delta}) = \lambda \|\boldsymbol{\Delta}\|_2^2$ for some $\lambda \geq 0$

Practical Algorithm

- We instantiate our framework as a practical algorithm MinNorm-OG which
  - Accesses batches of retain set samples
  - Employs an iterative update

- Set $R(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$ and $\hat{R}(\boldsymbol{\Delta}) = \lambda \|\boldsymbol{\Delta}\|_2^2$ for some $\lambda \geq 0$

- For batch $\mathcal{B}_r \subseteq \mathcal{D}_r$, alternate between

  - **Unlearning subproblem**: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \tilde{\boldsymbol{\Delta}}$

$$\tilde{\boldsymbol{\Delta}} = \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \|\boldsymbol{\theta} + \boldsymbol{\Delta}\|_2^2 + \lambda \|\boldsymbol{\Delta}\|_2^2 \quad \text{s.t. } \boldsymbol{\Delta} \perp \nabla f(\boldsymbol{\theta}, \boldsymbol{x}_i) \quad \forall \boldsymbol{x}_i \in \mathcal{B}_r$$

Practical Algorithm

- We instantiate our framework as a practical algorithm MinNorm-OG which
  - Accesses batches of retain set samples
  - Employs an iterative update

- Set $R(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$ and $\hat{R}(\boldsymbol{\Delta}) = \lambda \|\boldsymbol{\Delta}\|_2^2$ for some $\lambda \geq 0$

- For batch $\mathcal{B}_r \subseteq \mathcal{D}_r$, alternate between

  - **Unlearning subproblem**: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \tilde{\boldsymbol{\Delta}}$

$$\tilde{\boldsymbol{\Delta}} = \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \|\boldsymbol{\theta} + \boldsymbol{\Delta}\|_2^2 + \lambda \|\boldsymbol{\Delta}\|_2^2 \;\text{ s.t. }\; \boldsymbol{\Delta} \perp \nabla f(\boldsymbol{\theta}, \boldsymbol{x}_i) \quad \forall \boldsymbol{x}_i \in \mathcal{B}_r$$

  - **Loss descent**: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \frac{\eta}{|\mathcal{B}_r|} \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{B}_r} \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}; \mathcal{B}_r)$

Experimental Results

Data Poisoning

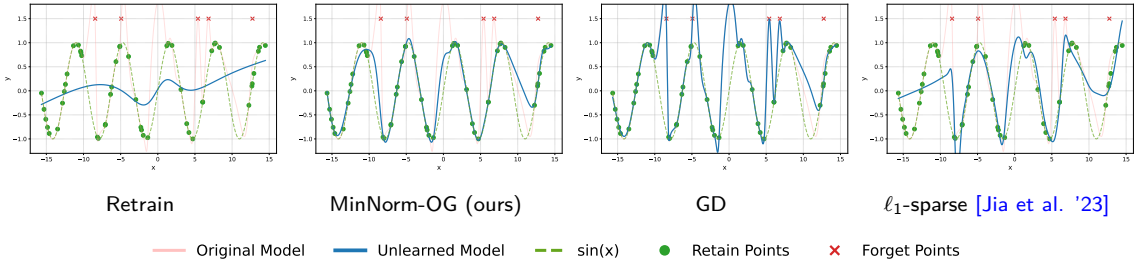

Figure: Example unlearned model fits after 100 unlearning epochs

Experimental Results

- Two additional unlearning experiments:
  - **Class Unlearning:** Remove knowledge of a specific class (colored versions of CIFAR-10 and TinyImageNet)
  - **Feature Unlearning:** Forget samples so the model learns a spurious color correlation on $\mathcal{D}_r$ (colored CIFAR-10)
- Comparisons to additional baselines:
  - SCRUB and NegGrad+ [Kurmanji et al. '23]
  - NPO [Zhang et al. '24]
  - SalUn [Fan et al. '24]
  - ...

Reference

J. L. Block, A. Mokhtari, S. Shakkottai “Machine Unlearning under Overparameterization”, NeurIPS 2025. arXiv: 2505.22601 [cs.LG]

Thank you!