Federated Learning 관련 Research Question

Federated Learning에서,

안전하게 데이터 삭제를 할 수 없을까?

IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 18, 2023

FedRecovery: Differentially Private Machine Unlearning for Federated Learning Frameworks

Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou, *Senior Member, IEEE*

PMLC Lab meeting

2025.08.20 (Wed)

목차

1. Machine Unlearning의 배경

1. Machine Unlearning의 배경

'Right to Be Forgotten'
잊혀질 권리

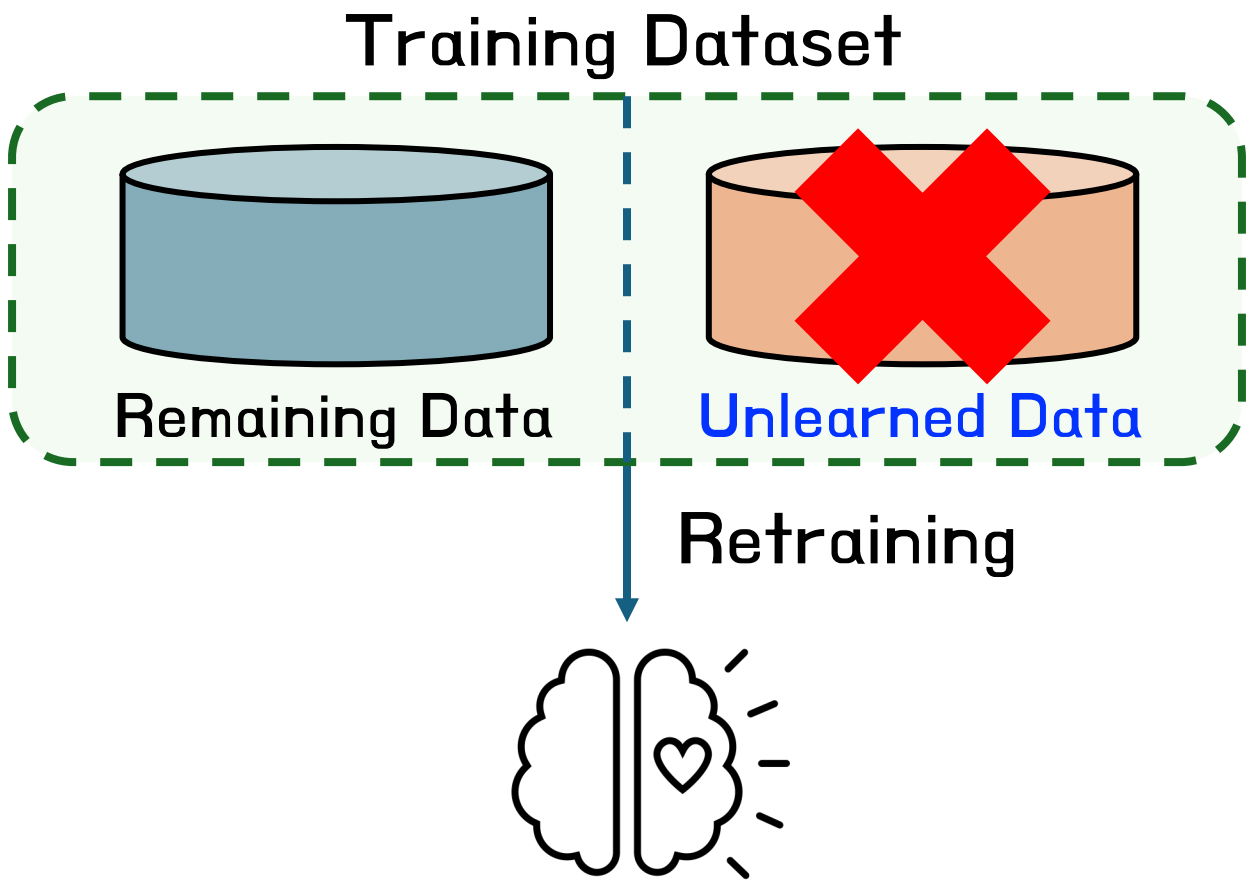
- **최근 10년 이내 전 세계적으로 관련 법률 제정했거나 제정 중**
  - o GDPR (EU), CCPA (USA), APPI (일본), CPPA (캐나다)

개인 및 조직은 자신의 데이터를 삭제 요청할 권리가 있다 !

→ **Privacy 관점으론** 😊**, ML 모델 관점으론** 😭
  - o 데이터셋에서 타겟 데이터만 삭제
  - o 이미 학습된 모델에서 타겟 데이터의 영향 제거

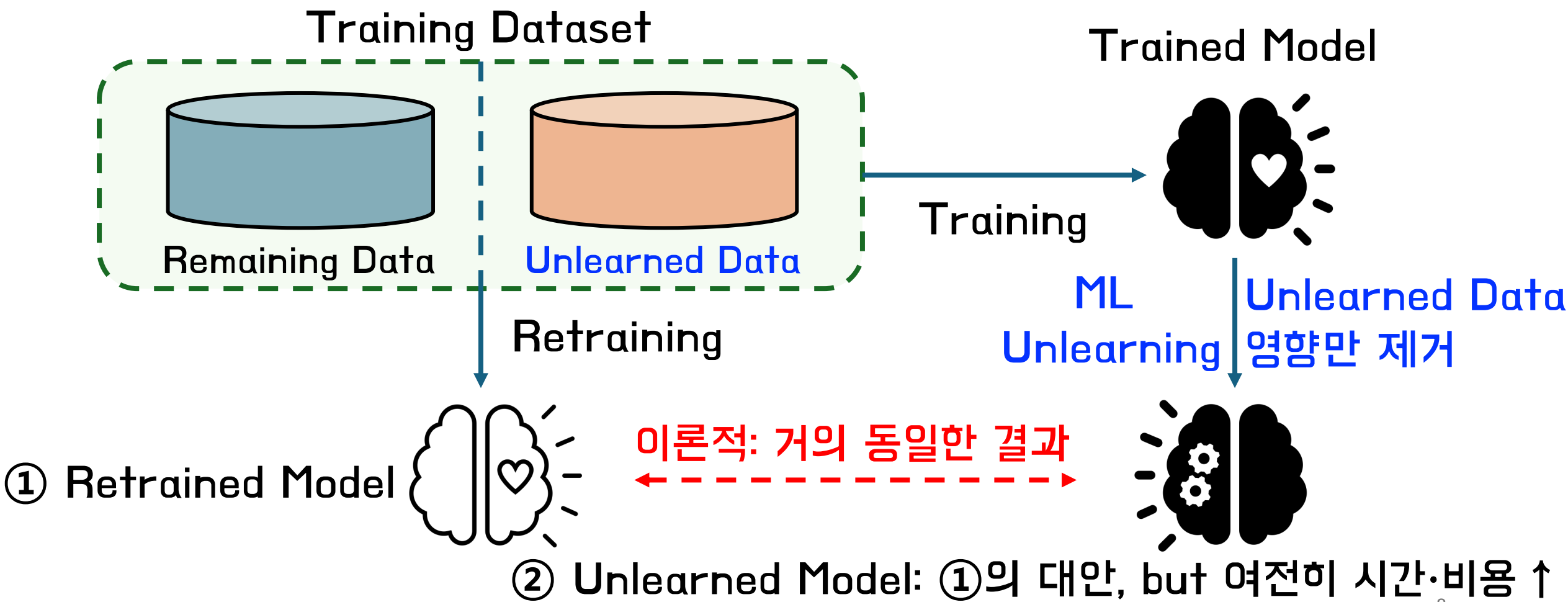
2. Machine Unlearning의 분류

2. Machine Unlearning의 분류 - Naïve Model (1/3)



① Retrained Model: 가장 확실, but 시간·비용 ↑

2. Machine Unlearning의 분류 - Unlearned Model (2/3)



2. Machine Unlearning의 분류 (3/3)

- Strong Unlearning : ① Retrained Model ② Unlearned Model처럼
  삭제 요청된 데이터가 학습에 전혀 사용되지 않은 것과 동일한 상태를 목표로 함
  * 단, 실제 적용 난이도가 높음

[Approximate Machine Unlearning]

- Weak Unlearning : 모델 출력 (예: Accuracy)이 유사하도록 조정하는 것을 목표
  ① 특정 데이터의 영향을 역으로 적용
  ② 파라미터 일부 조정 및 성능 저하 최소화를 위한 튜닝
  * 단, 완전한 데이터 삭제 보장은 불가

3. FedRecovery의 개념

3.1 FedRecovery 핵심 아이디어

1. 학습 알고리즘

: Gradient Descent로 모델 학습

2. Unlearning 알고리즘

: 삭제 요청 클라이언트 영향 제거

→ 두 가지 모두 Gaussian Noise를 활용해 (1) **삭제 요청을 반영한 모델**과 **(2) 처음부터 해당 데이터를 제외하고 학습한 모델**을 통계적으로 구별이 불가능한 상태를 달성

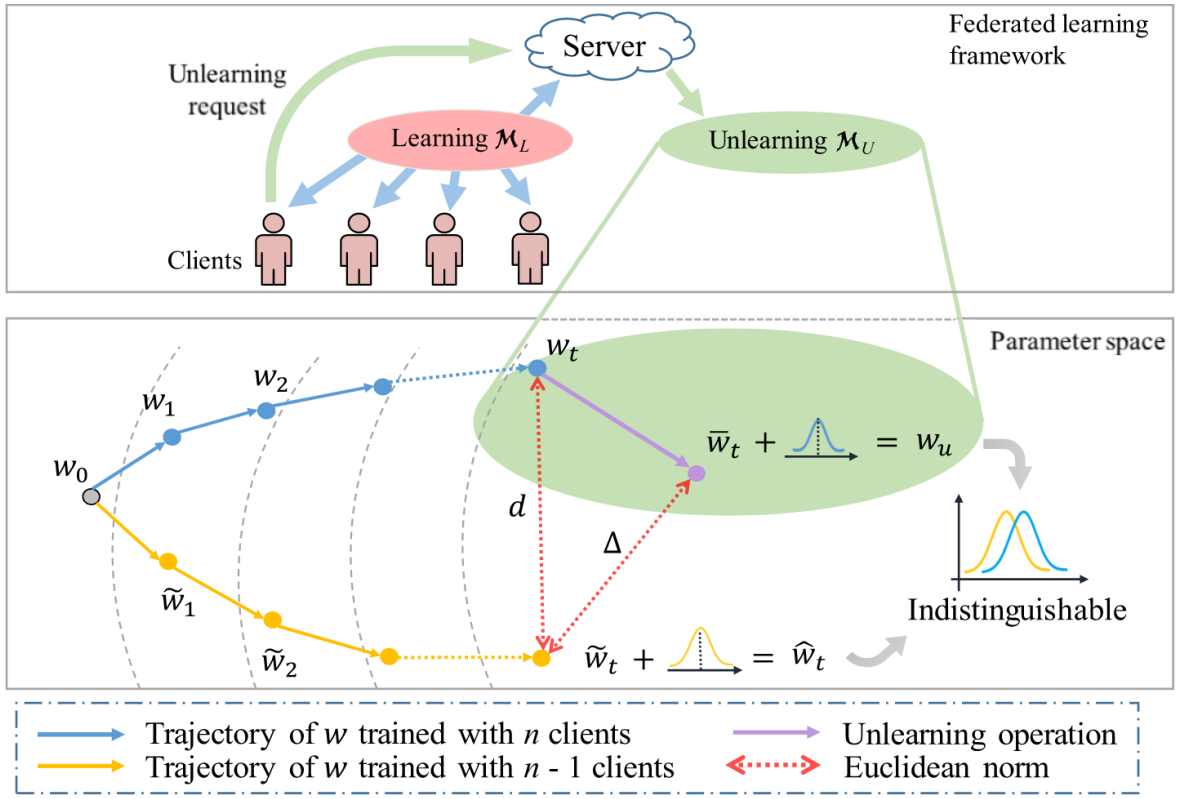


Fig. 2. The framework of the FedRecovery algorithm and its rationale in the parameter space. The learning algorithm $\mathcal{M}_L$ trains the model using the gradient descent method, which corresponds to the blue and yellow trajectories. The unlearning algorithm $\mathcal{M}_U$ removes the influence of the client who wishes to be unlearned, as shown in green. The Gaussian noise is added to guarantee the unlearned model and retrained model are indistinguishable.

3.2 FedRecovery 시나리오 (1/3)

1. Federated Learning (파란색 선)

: 각 클라이언트는 자기 데이터로 gradient를 계산하고, 서버는 이를 받아 글로벌 모델 파라미터 w를 반복적으로 업데이트

2. 클라이언트 C_n의 데이터 삭제 요청

: 서버는 C_n의 영향을 제거하기 위해 갖고 있는 정보 활용 필요

→ (1) 일련의 모델 파라미터 $\{w_t\}_{i=0}^{t}$와

(2) gradient $\{\nabla f_i(w_j)\}, i \in [t], j \in [n]$

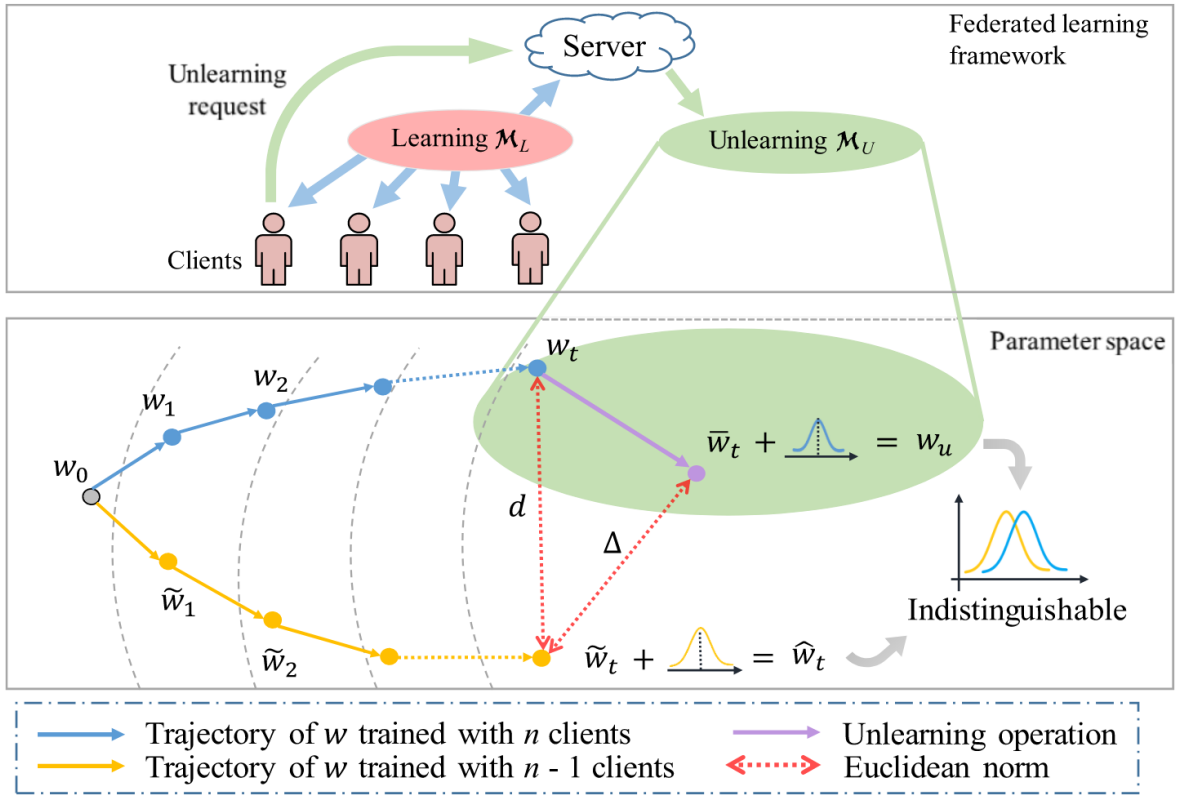


Fig. 2. The framework of the FedRecovery algorithm and its rationale in the parameter space. The learning algorithm $\mathcal{M}_L$ trains the model using the gradient descent method, which corresponds to the blue and yellow trajectories. The unlearning algorithm $\mathcal{M}_U$ removes the influence of the client who wishes to be unlearned, as shown in green. The Gaussian noise is added to guarantee the unlearned model and retrained model are indistinguishable.

3.2 FedRecovery 시나리오 (2/3)

3. 클라이언트 C_n이 없었을 경우 (노란색 선)

: 나머지 $n-1$개의 클라이언트만으로 생겨난 궤적 $\{\widetilde{w}_i\}_{i=1}^{t}$

4. FedRecovery 목표 : $\bar{w} \in W$ (보라색 선)

과거 클라이언트 정보를 이용해 c_n의 기여도를 제거하고 **새로운 경로 $\bar{w}$ 찾기**

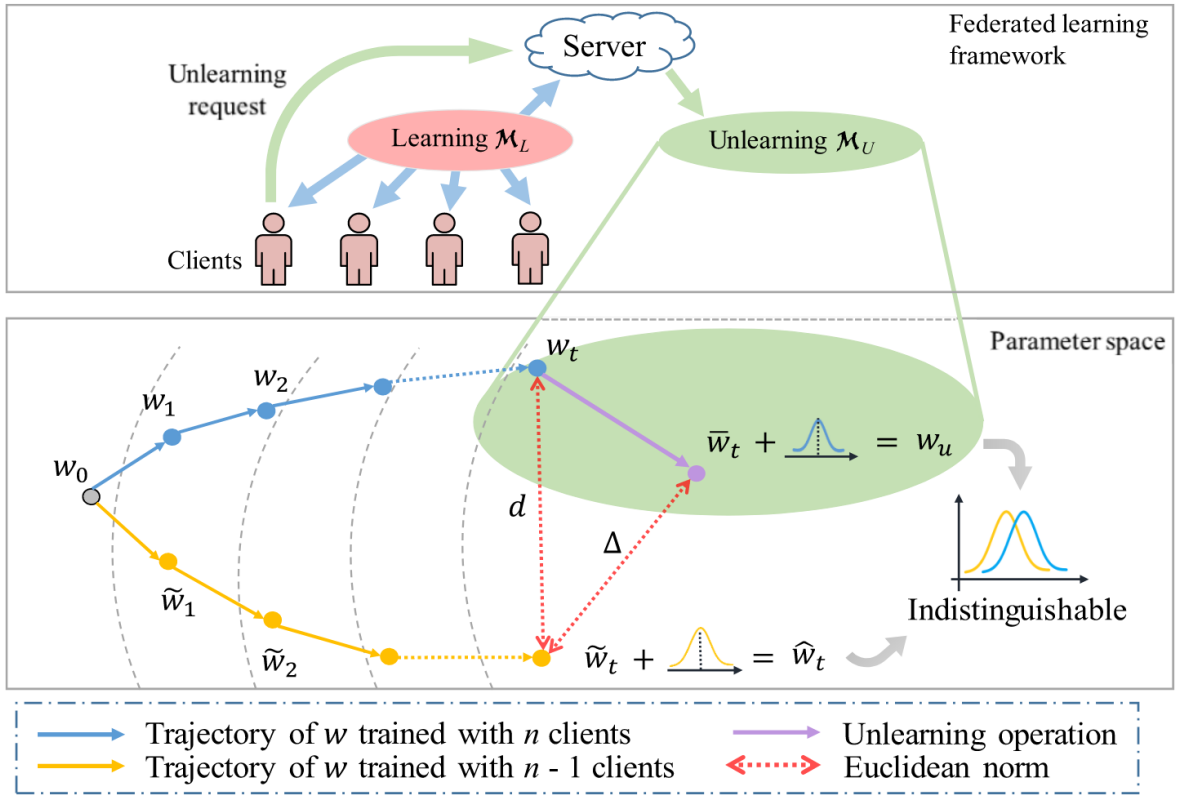


Fig. 2. The framework of the FedRecovery algorithm and its rationale in the parameter space. The learning algorithm $\mathcal{M}_L$ trains the model using the gradient descent method, which corresponds to the blue and yellow trajectories. The unlearning algorithm $\mathcal{M}_U$ removes the influence of the client who wishes to be unlearned, as shown in green. The Gaussian noise is added to guarantee the unlearned model and retrained model are indistinguishable.

3.2 FedRecovery 시나리오 (3/3)

5. FedRecovery는

(1) $\|\bar{w}_t - \tilde{w}_t\| = \Delta, \begin{cases} \bar{w}_t, \text{보정된 모델} \\ \tilde{w}_t, \text{재학습 모델} \end{cases}$ 계산

(2) Δ를 특정 Gaussian Noise로 보정해서

보정된 모델을 재학습 모델과 **통계적으로 구분 불가능**하게 만든다

* 손실함수가

(1) Convex : 최적점 유일, 거리 상한 쉽게 계산

(2) Non-Convex: Smoothnesss^2로 상한 추정

* 여기서는 $\tilde{w}_t$도 계산 불가하고, Δ 추정만 가능

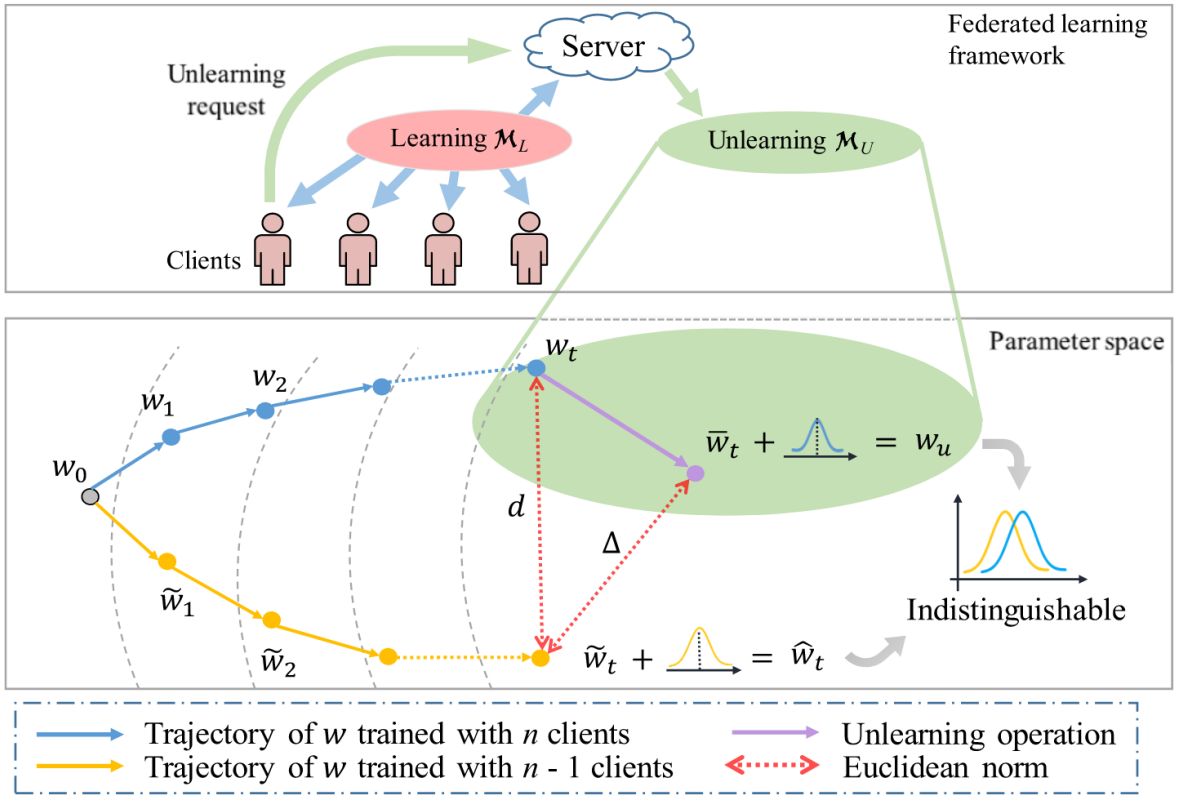


Fig. 2. The framework of the FedRecovery algorithm and its rationale in the parameter space. The learning algorithm $\mathcal{M}_L$ trains the model using the gradient descent method, which corresponds to the blue and yellow trajectories. The unlearning algorithm $\mathcal{M}_U$ removes the influence of the client who wishes to be unlearned, as shown in green. The Gaussian noise is added to guarantee the unlearned model and retrained model are indistinguishable.

3.3 FedRecovery 핵심 제안 (1/2)

1. Gradient Residual δ

: 삭제 요청한 클라이언트가 남긴 영향

→ 그렇다면 단순히 δ 합산해서 영향력을 빼면 되지 않나? **NO.** 각 라운드마다 δ 의 영향력이 다르게 증폭되어 단순합 불가.

한계1. 클라이언트 종속성

- 다른 gradient도 영향 받음

한계2. Stochastic 학습

- $\widetilde{w}_t$는 유일하지 않음

→ 보정된 모델과 재학습 모델이 구분 불가능하게 만들어야 함!

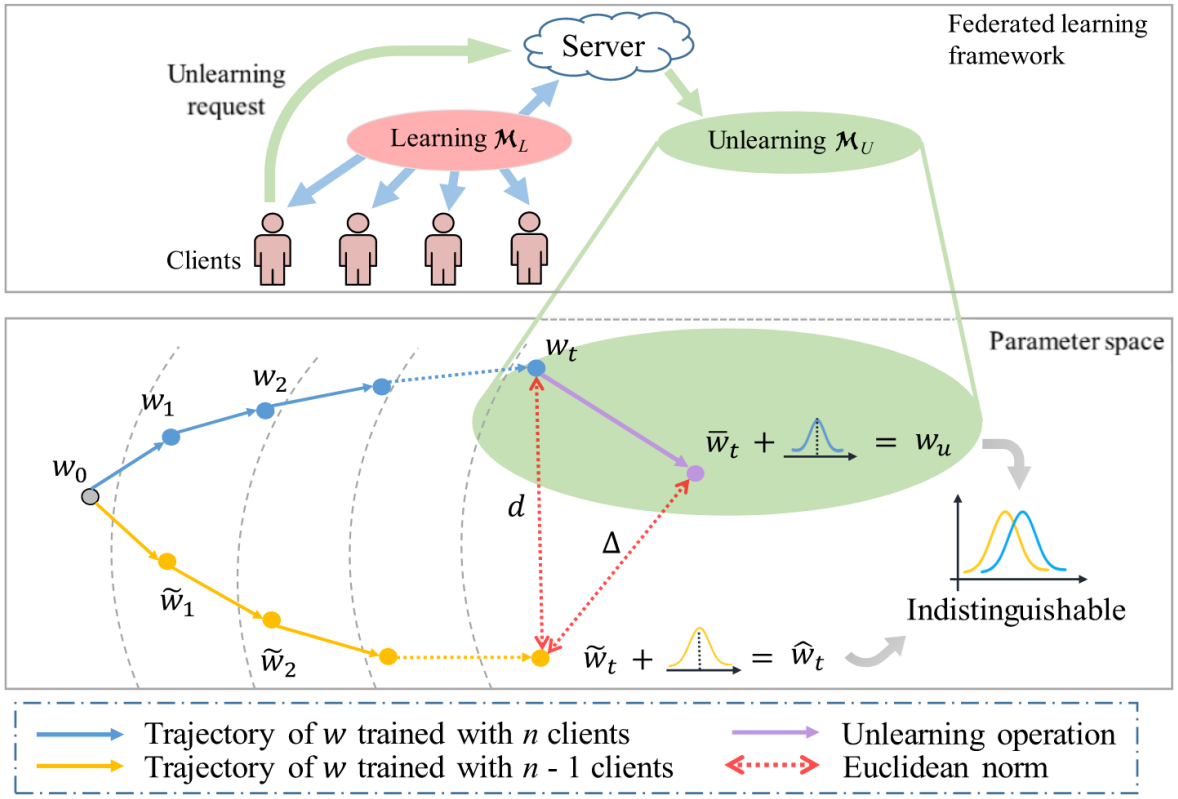


Fig. 2. The framework of the FedRecovery algorithm and its rationale in the parameter space. The learning algorithm $\mathcal{M}_L$ trains the model using the gradient descent method, which corresponds to the blue and yellow trajectories. The unlearning algorithm $\mathcal{M}_U$ removes the influence of the client who wishes to be unlearned, as shown in green. The Gaussian noise is added to guarantee the unlearned model and retrained model are indistinguishable.

3.3 FedRecovery 핵심 제안 (2/2)

2. Indistinguishability[3] (DP 보장)

: 보정된 모델 $\bar{w}_t$와 재학습 모델 $\tilde{w}_t$는 완전히 같을 수 없으므로, Gaussian Noise를 추가

$\rightarrow\ w_u = \bar{w}_t + z,\ z \sim N(0,\ \sigma^2 I_d)$

* Gaussian Noise 성질

: 평균이 다른 두 분포 $(\bar{w}_t, \tilde{w}_t)$여도 충분히 큰 노이즈가 더해지면 분포가 거의 겹치게 되어 통계적으로 두 모델 간 차이를 구분할 수 없게 됨

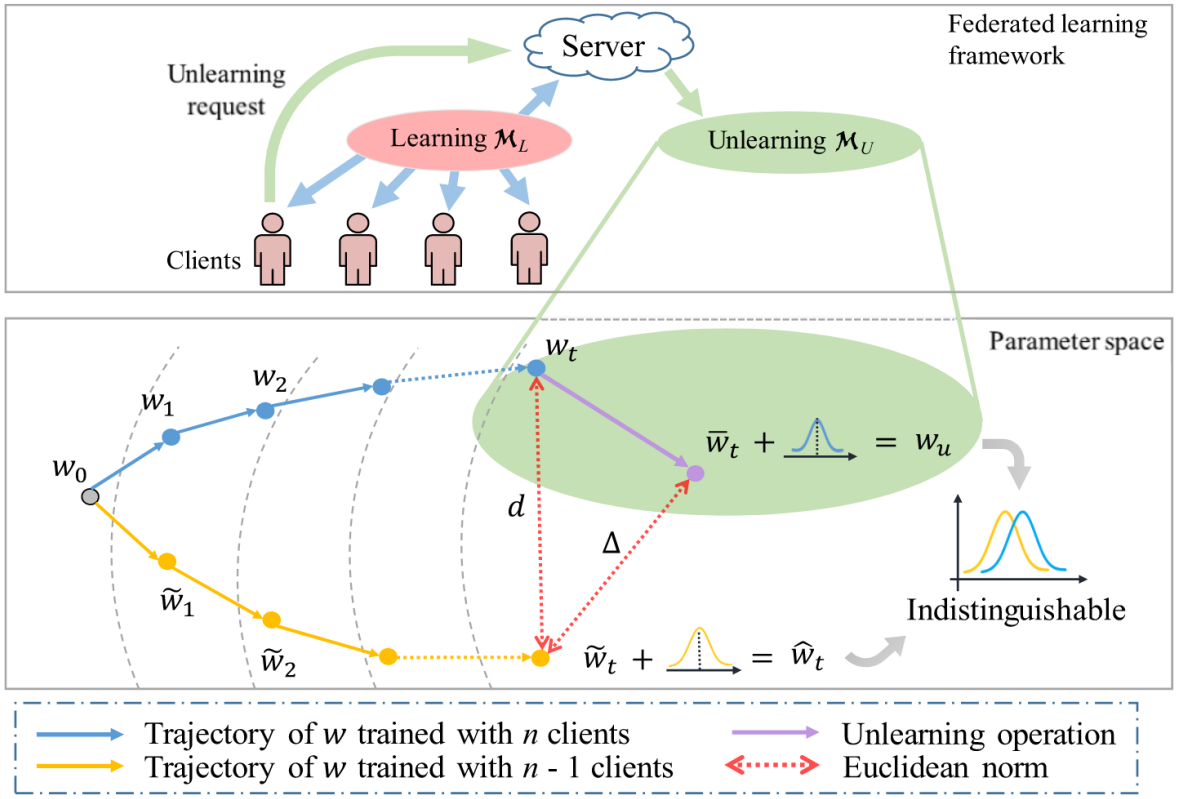


Fig. 2. The framework of the FedRecovery algorithm and its rationale in the parameter space. The learning algorithm $\mathcal{M}_L$ trains the model using the gradient descent method, which corresponds to the blue and yellow trajectories. The unlearning algorithm $\mathcal{M}_U$ removes the influence of the client who wishes to be unlearned, as shown in green. The Gaussian noise is added to guarantee the unlearned model and retrained model are indistinguishable.

4. FedRecovery에 동형암호 적용 가능성

4.1 FedRecovery에 동형암호 적용 가능성

Gradient Disclosure (기울기 노출) 위험성

: Raw Gradients*를 서버에 공유하면 개인정보가 노출될 수 있음

→ 그래서 기존 Federated Learning에서는 암호화 기법 사용해 보호하고자 함

하지만 Machine Unlearning 환경에서의

* 서버는 Unlearning 요청의 정당성을 반드시 검증해야 함 (논문 가정)

(1) DP 기반 FL 경우, 각 클라이언트의 noisy gradient를 그대로 기여도로 보고, FedRecovery에 적용할 수 있음

(2) 동형암호 기반 FL 경우, 서버가 개별 클라이언트의 기여도를 검증하거나 구분 불가하여 Unlearning 수행 불가

* Raw Gradients: 클라이언트가 로컬 데이터로 학습한 뒤, 아무런 보호 (예. 노이즈, 암호화 등) 없이 그대로 서버에 보내는 gradient 값

4.2 FedRecovery에 동형암호 적용 방법 아이디어

- 클라이언트의 적극적인 참여 (단순 아이디어)

: 삭제 요청 클라이언트 k가 자신의 과거 업데이트에 마이너스를 곱한 뒤 암호화해서 서버로 보내면 서버가 해독 없이 종합에서 빼는 방법

· $\widetilde{w}_t = w_t - w_t^k$

- 삭제 전 모델: w_t
- 클라이언트 k의 기여도: w_t^k
- 삭제 후 모델: $\widetilde{w}_t$

5. 참고 자료

5. 참고 자료

1. Lipschitz 연속성 (함수 변화율 제한하는 개념)

: 함수 $f(x)$ 자체가 입력 변화량에 비례해서만 변한다는 제약

$$\|f(x_1) - f(x_2)\| \leq L \cdot \|x_1 - x_2\|$$

2. 손실 함수에서의 L-Smoothness

: 여기서는 gradient 함수 $\nabla f(w)$가 Lipschitz 연속성을 따른다고 가정. 즉,

$$\|\nabla f(w_1) - \nabla f(w_2)\| \leq L \cdot \|w_1 - w_2\|$$

- 의미: 기울기(gradient)가 급격하게 변하지 않는다
- 결과: loss surface가 smooth하다는 성질을 준다
- 장점: gradient descent의 수렴 분석에 유용

5. 참고 자료

3. (ϵ, β) −Indistinguishability

: X와 Y를 확률적으로 구분하기 매우 어렵다는 보장

정의역 R에서 정의된 확률변수 X, Y가 있을 때,

모든 가능한 부분집합 $\boldsymbol{S} \subseteq \boldsymbol{R}$에 대해 다음이 성립하면,

$$\begin{cases} \Pr[X \in S] \leq e^{\epsilon} \cdot Pr[Y \in S] + \beta \\ \Pr[Y \in S] \leq e^{\epsilon} \cdot Pr[X \in S] + \beta \end{cases}$$

X와 Y는 서로 (ϵ, β) −Indistinguishability (구분 불가능)하다고 한다.

5. 참고 자료

4. Sensitivity

: 하나의 데이터가 결과에 미치는 최대 영향력을 수치화한 것

$$\Delta = \max \| q(D) - q(D') \|$$

5. Gaussian Mechanism

: 확률변수 $\boldsymbol{X \sim N(\mu_1, \sigma^2 I_d)}, \boldsymbol{Y \sim N(\mu_2, \sigma^2 I_d)}$ 일 때,

만약 $\|\mu_1 - \mu_2\| \leq \Delta$를 만족한다면,

임의의 $\beta > 0$에 대해 X와 Y는 (ϵ, β) −Indistinguishability 하다.

단, $\epsilon = \frac{\Delta^2}{2\sigma^2} + \frac{\Delta}{\sigma}\sqrt{2\log(1/\beta)}$ 로 정의한다.