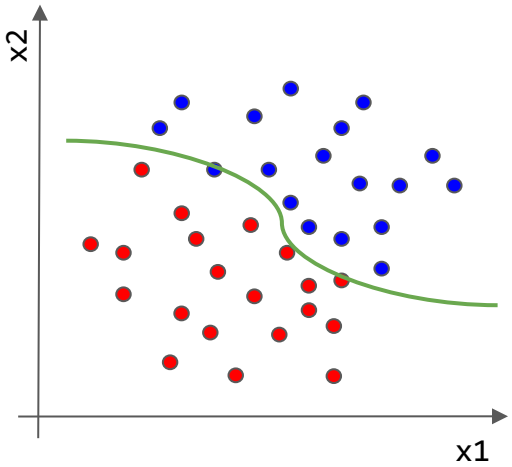
ミスラベルデータの忘却による学習済みモデルの汎化性能の向上手法の提案

DEIM2024 (2024/2/28)

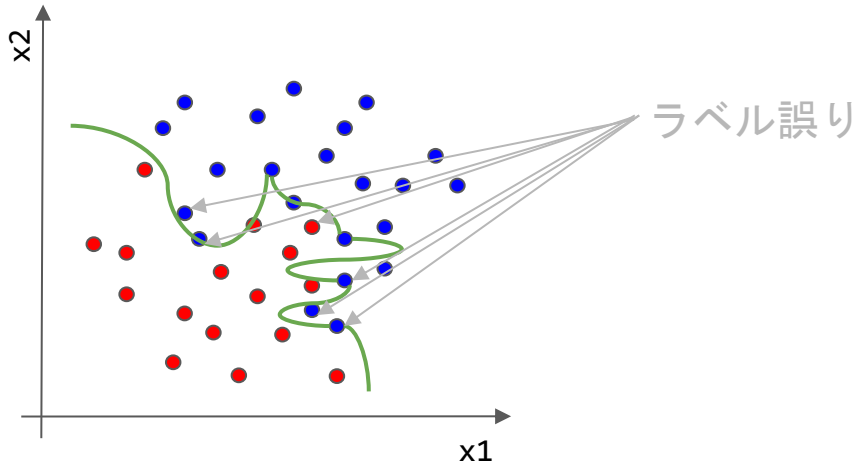
[1]大阪大学 [2]奈良工業高等専門学校
杉浦 一瑳[1] 岡村 真吾[1,2] 山下 恭佑[1] 矢内 直人[1]

背景: 機械学習におけるミスラベルデータの悪影響

ミスラベルデータはモデルの汎化性能を下げる [Krause+, ECCV2016, Arpit+, ICML2017]



ミスラベルデータ無し

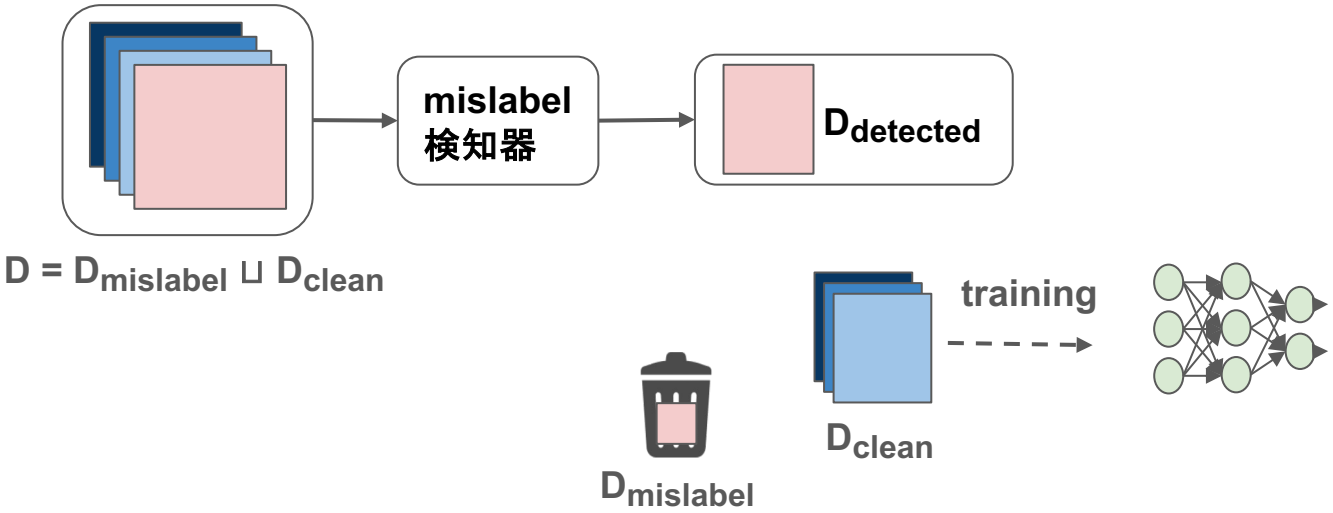


ミスラベルデータ有り

ミスラベルデータに対する既存の対策手法

学習前の処理:

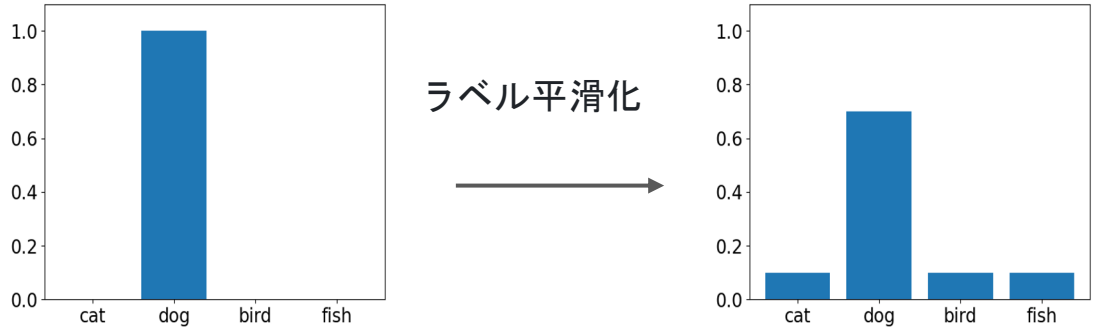
- ミスラベルデータの除去 [Pleiss+, NeurIPS2020, Tanaka+, CVPR 2018]



ミスラベルデータに対する既存の対策手法

学習中の処理:

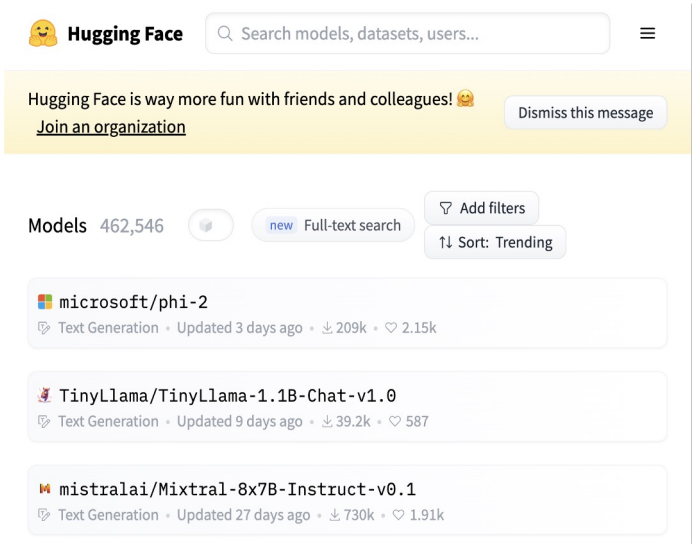
- モデルをラベル誤りにロバストにする [Szegedy+, CVPR2016]



既存手法の課題

- HuggingFace, GitHub等で大規模な学習済みモデルの提供
- 大規模なモデルを再学習して既存手法を適用することは高コスト

→ 事後的に適応可能な対策方法が望まれる



https://huggingface.co/models (accessed 2024/1/11)

解決策: Machine Unlearning による事後的なモデル修正

Machine Unlearning の背景

忘れられる権利 (EUのGDPR 第17条)

- 個人のデータは要求に応じて削除されなければならない [Cloudflare blog]
- 例: グーグルに対するヌード写真の検索結果の削除要請

  [Yahooニュース, "高裁で否定された「忘れられる権利」 新しい権利として認めるべきなのか"]

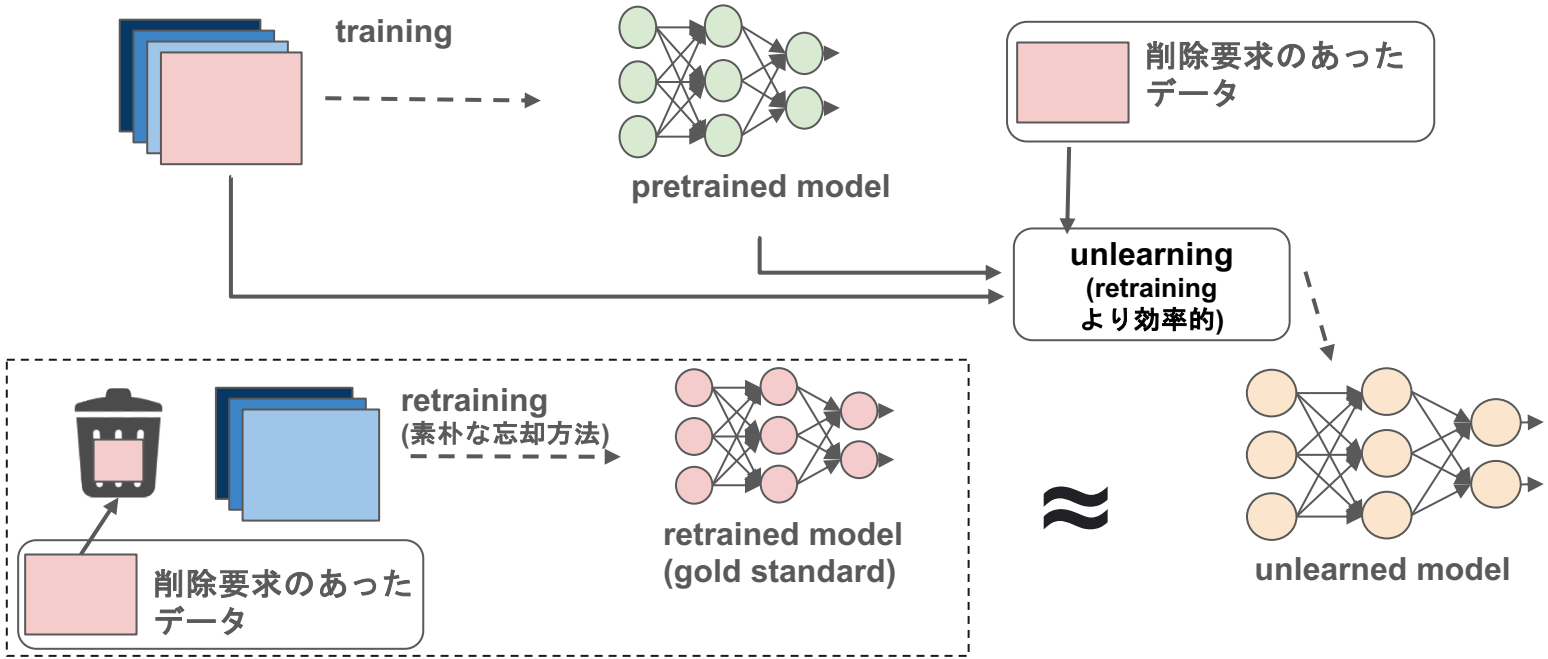
Machine Unlearning の背景

機械学習と忘れられる権利

- 機械学習モデルは訓練データを記憶してしまう
  - データが訓練データに含まれていたか推論する攻撃 [Shokri+ 2017, IEEE S&P]
  - 訓練データを復元する攻撃 [Fredrikson+ 2015, ACM SIGSAC]
  - LLMに訓練データを生成させる攻撃 [Carlini+ 2021, USENIX]
- → 学習に用いたデータに削除要請があったら, データを取り除いて
  モデルを再学習する必要
- 再学習(高コスト)より効率的に忘却したい

Machine Unlearning

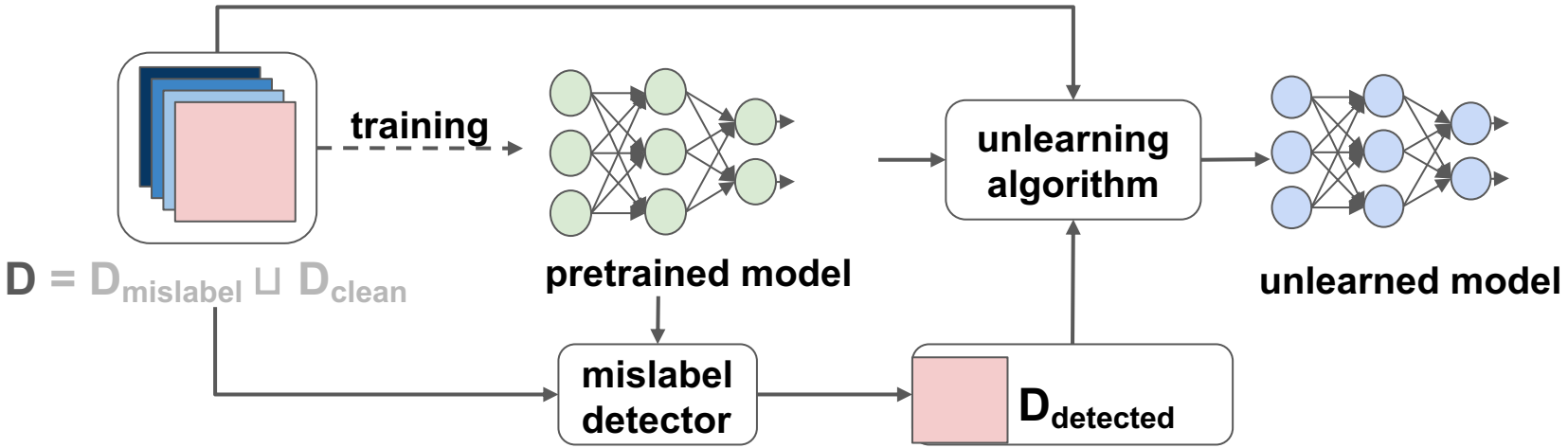
学習済みモデルから訓練データの一部の影響を効率的に取り除く [Nguyen+, 2022]



提案手法

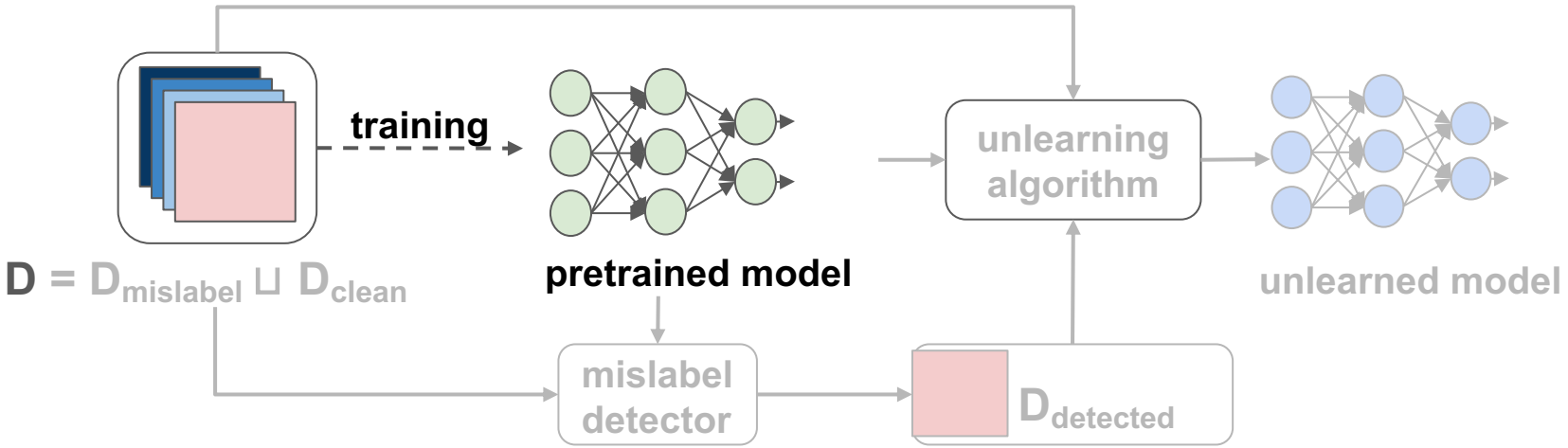
提案手法

学習済みモデルに対して, **ミスラベルデータ**(≠ 削除要求のあったデータ)を検知し, 忘却させることで汎化性能を向上させる



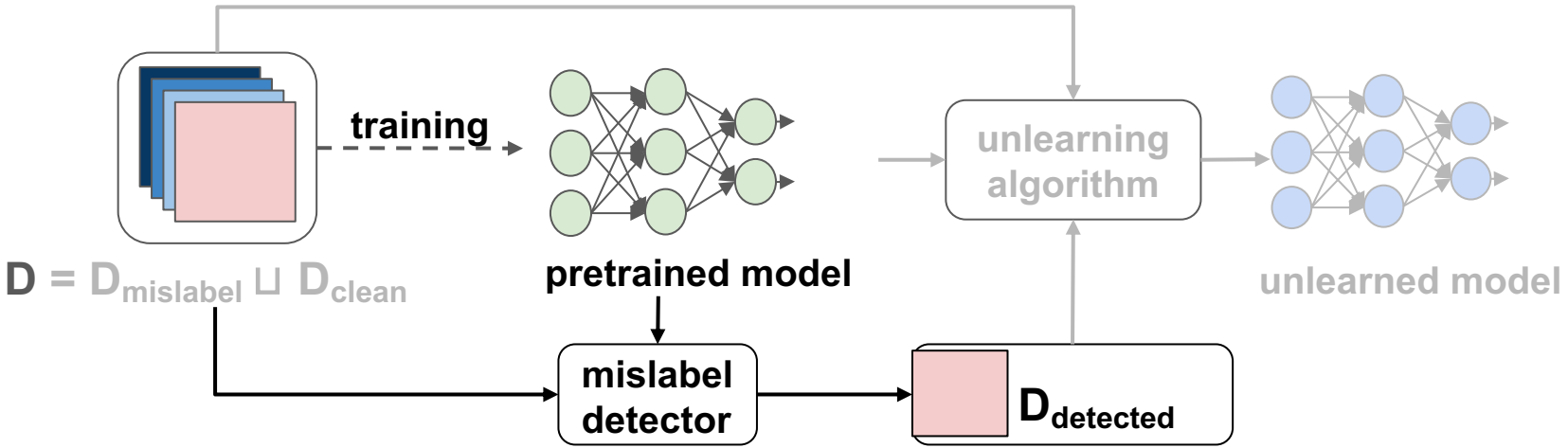
提案手法: 条件設定

ミスラベルデータを含むデータセットD & pretrained modelを持っている



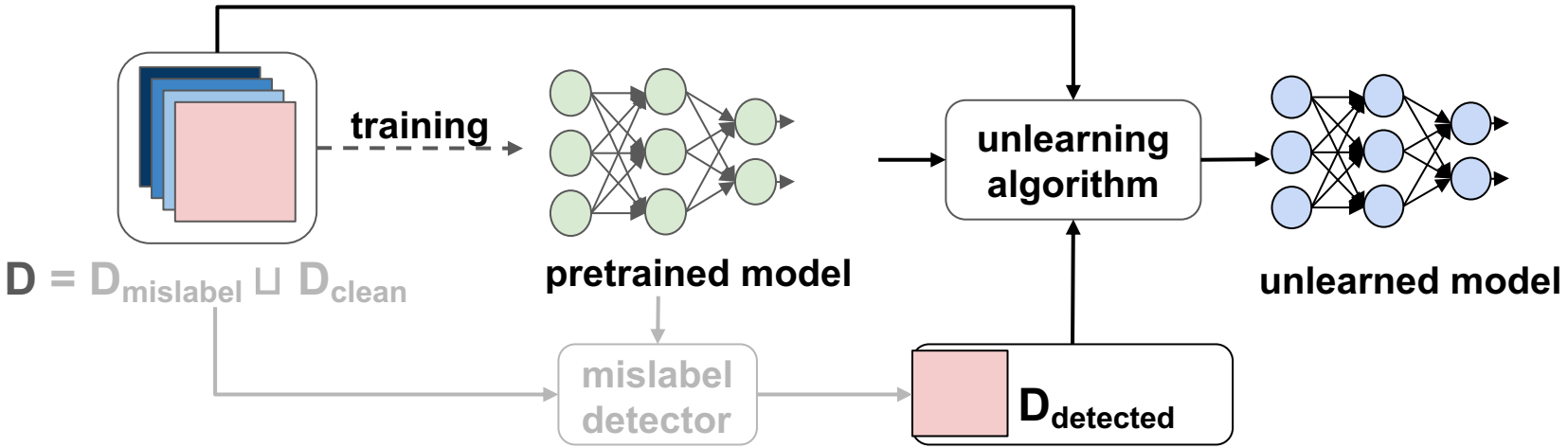
提案手法: 検知ステップ

データセットD, pretrained modelを元に, ミスラベルデータを検知する



提案手法: 忘却ステップ

検知したデータをpretrained modelから忘却させ, 汎化性能を高める

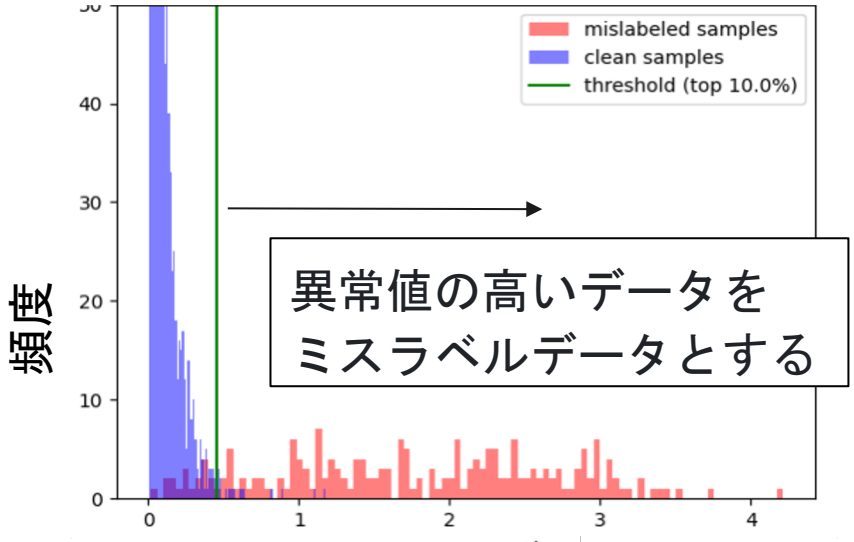


検知手法

ミスラベルデータとクリーンデータを異常値(eg. 損失)で分離する

学習中のクリーンデータと
ミスラベルデータの損失の
挙動は異なる [Pleiss+, NeurIPS2020]

各サンプルの異常値のヒストグラム



異常値 例: $L(z, \hat{\theta})$

z = (x, y),
$\hat{\theta}$: 学習後のパラメータ

忘却手法: 近似的な忘却手法

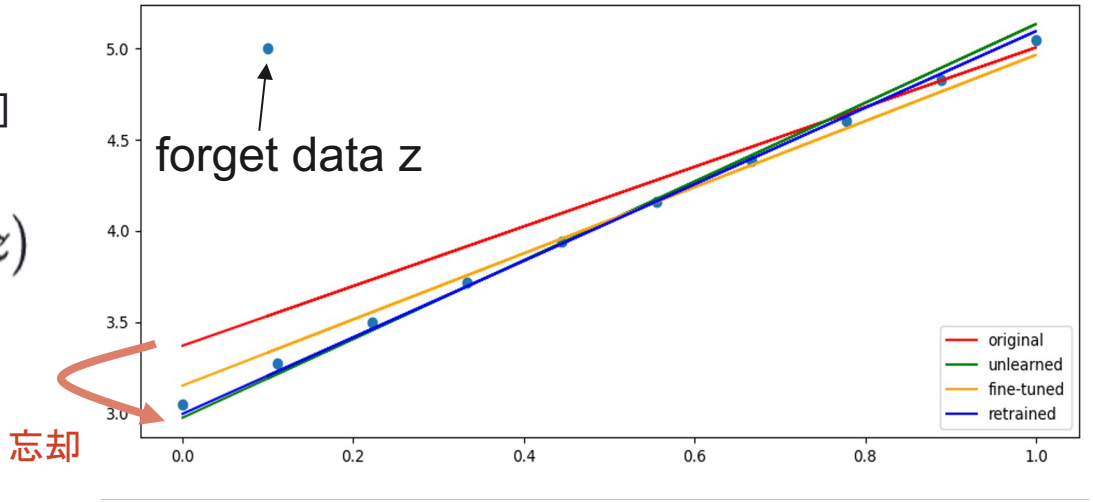
パラメータ更新で忘却
例: 影響関数 [Koh+, ICML2017]

$$\hat{\theta}^{\backslash z} \approx \hat{\theta} + \frac{1}{n} I_{up,params}(z)$$

z 忘却後の
パラメータ

学習済み
パラメータ

更新量



実験

実験設定

- データセット: 手書き数字のMNISTデータセットの(0, 1)の2値分類
- モデル: 2層のマルチパーセプトロン(MLP)
- ミスラベル混入割合r = {1, 5, 10, 20, 30, 40, 50}%
  a. 混入割合は検知性能, 忘却性能に大きく影響する
  b. (x_i, y_i) in D に対して以下のようにラベルを改変する

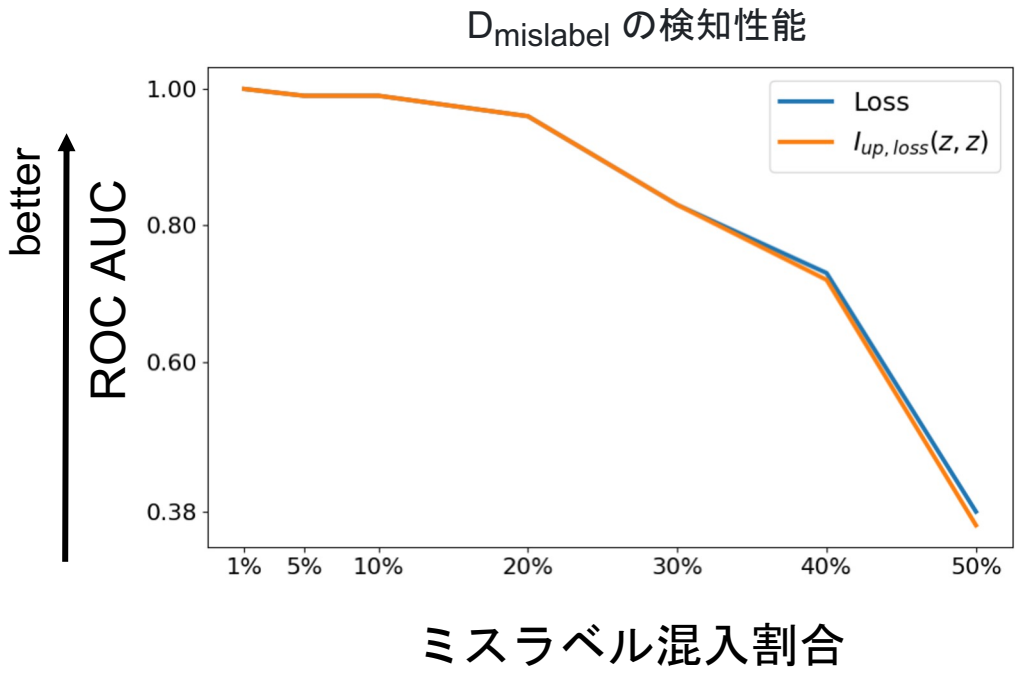
$$y_i = \begin{cases} y_i & \text{with probability } 1 - r, \\ 1 - y_i & \text{with probability } r. \end{cases}$$

実験の流れ

- ミスラベルデータの**検知性能**の検証
- **ミスラベルデータの忘却による性能向上**の検証(ミスラベルデータ既知)
- **検知したデータの忘却による性能向上**の検証(提案手法)

ミスラベルデータの検知性能の検証

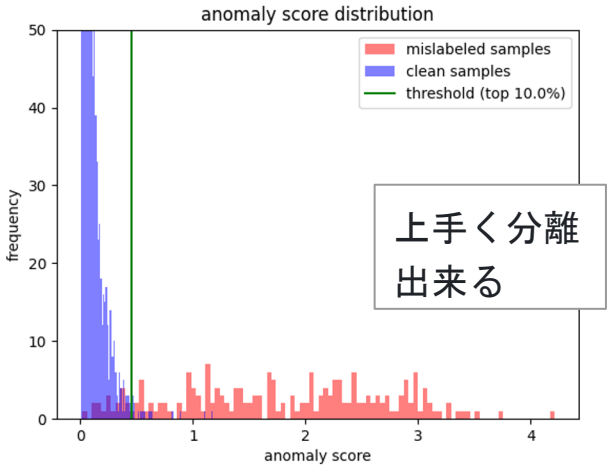
検知性能が混入割合の増加とともに低下した



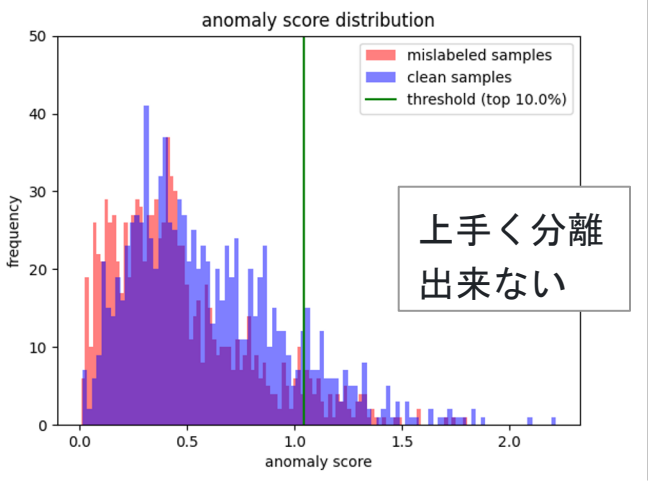
ミスラベルデータの検知性能の検証

ミスラベル混入割合が大きい場合, mislabel sampleとclean sampleが重なってしまい閾値をずらしても検知性能が上がらない

noise ratio = 10%

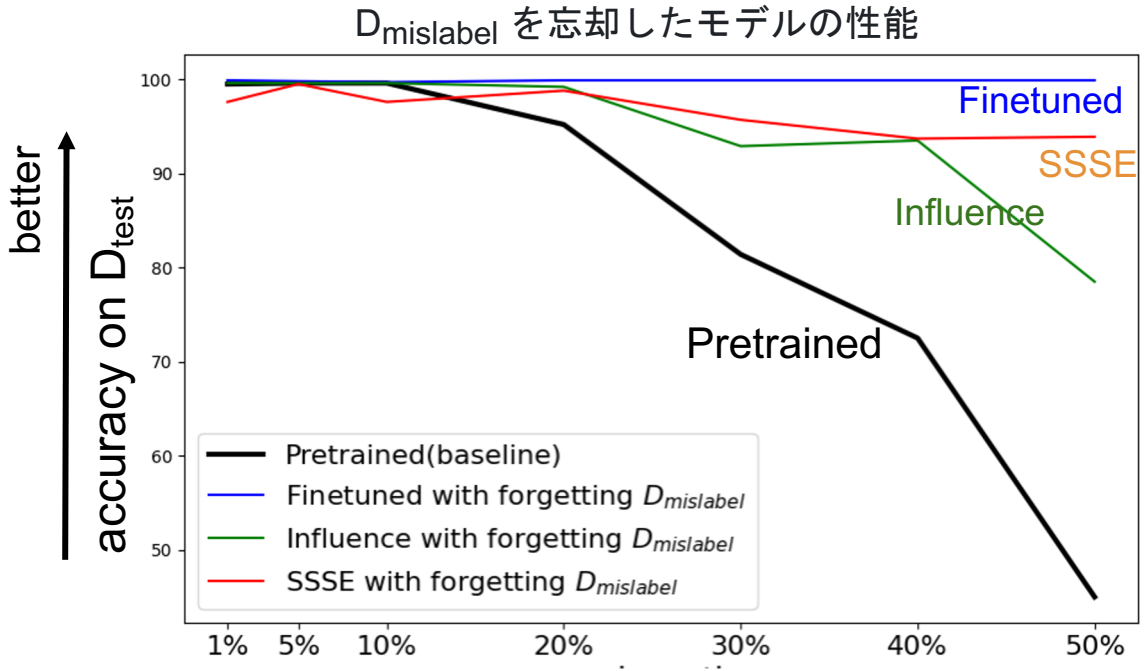


noise ratio = 50%



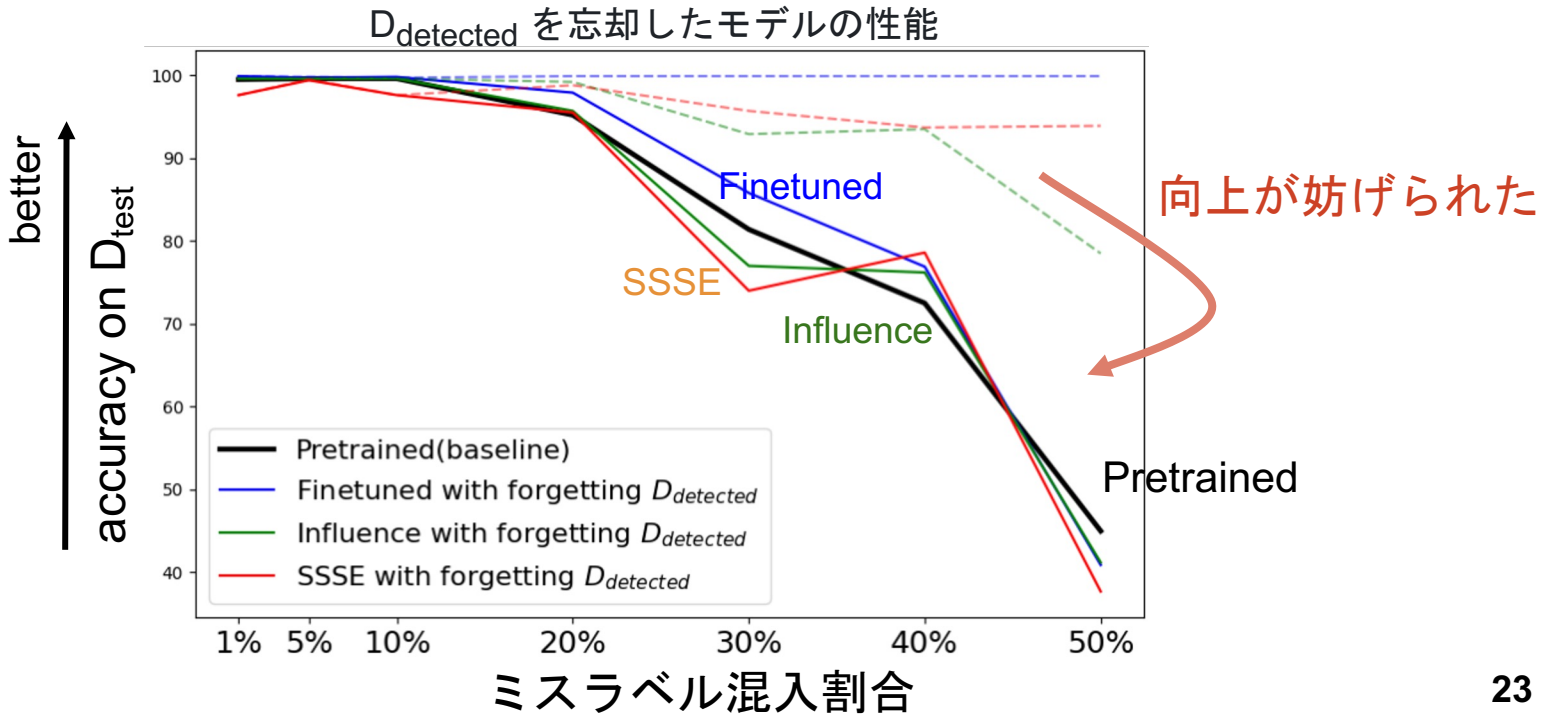
ミスラベルデータの忘却による性能向上の検証

ミスラベルデータ忘却により汎化性能が向上した



検知したデータの忘却による性能向上の検証(提案手法)

ミスラベル混入割合の増加に伴い検知性能が低下し, 性能向上がみられなかった



まとめ

提案手法: ミスラベルデータの忘却による学習済みモデルの汎化性能の向上

実験結果:

- ミスラベル検知: 混入割合が増えるほど検知性能が低下した
- ミスラベルデータの忘却: 汎化性能の向上が確認できた
- 提案手法: 混入割合の増加に伴い検知性能が低下し,

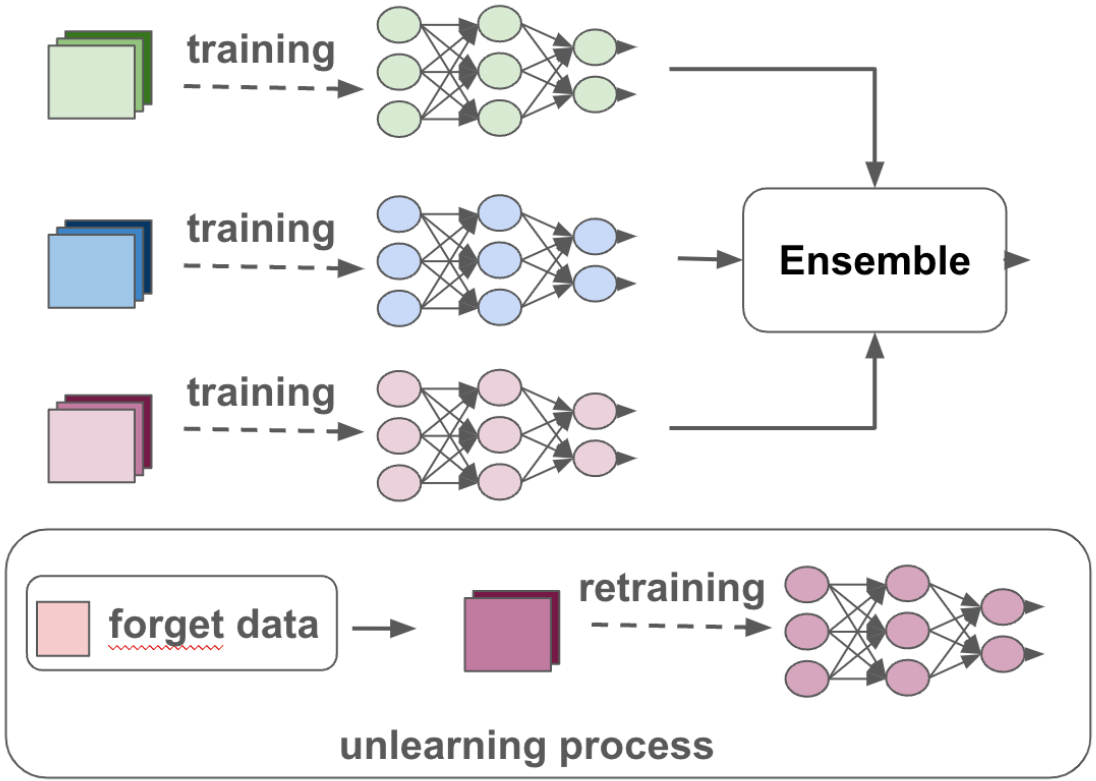
  性能向上がみられなかった

今後の課題: 事後的な検知性能の向上

コード : https://github.com/speed1313/mislabel-unlearning

Appendix

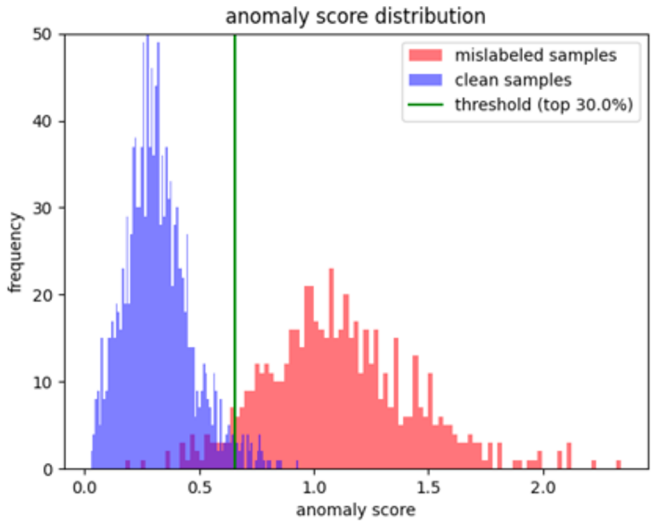
SISA(Sharded, Isolated, Sliced, and Aggregated) training



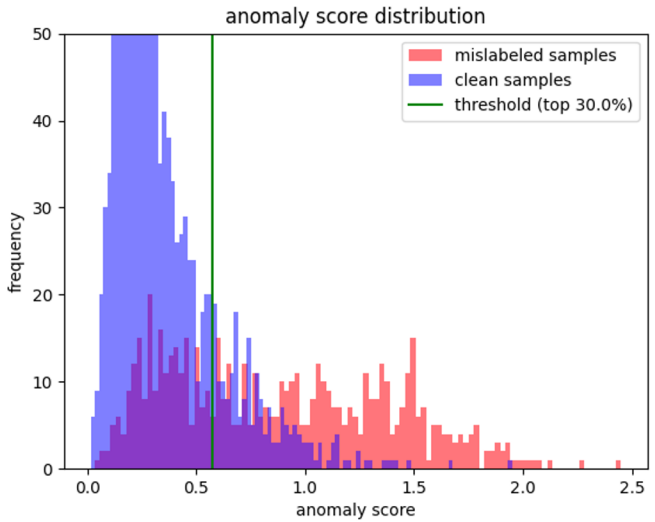
epochによる分離のしやすさの違い

mislabelデータはcleanデータに比べて学習が遅いという性質から
epochが小さいほど分離しやすい

epoch = 10



epoch = 20



Influence Function(影響関数)によるUnlearning[Koh+ 2017, ICML]

ある訓練データサンプルzを除いた時の, パラメータの変化を近似する関数

$$\hat{\theta}^{\backslash z} - \hat{\theta} \approx -\frac{1}{n} I_{up,params}(z)$$

ただし, $I_{up,params}(z) := -H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta})$.

これを用いると,

$$\hat{\theta}^{\backslash z} \approx \hat{\theta} + \frac{1}{n} I_{up,params}(z)$$

と再学習せずに$\hat{\theta}^{\backslash z}$を計算できる.

Machine Unlearning 研究の潮流 [Nguyen+, 2022]

- モチベーション
  - 忘却対象のみ忘れさせ, 他の性能は保ちたい
  - 効率的に忘却させたい

- 評価指標
  - 忘却前後の忘却対象のaccuracy, 残りのaccuracyの変化
  - 忘却前後のMembership Inference Attackの攻撃成功率
  - 忘却に要する時間
  - etc