A Motivation of Attacking Fair Representations

Attacking mutual information at representation level, as formalized in Section 2, aligns better with fair representation learning (FRL) and is more universal than attacking classifiers. For example, a bank may want to obtain a representation $\boldsymbol{z}$ for each user that can be used to determine their eligibility for *existing* and *upcoming* financial products such as credit cards without concerning about fairness again (Zhao et al., 2019). Here each financial product is associated with a (unique) label, and determining the eligibility entails a classification task. In this case, there are two challenges for delivering a classification-based attack. First, one has to determine and justify which classifier to use and why consider the fairness of this specific classification task. Second, for any upcoming financial product, its label does not exist and one cannot obtain classifier (we need this label to train the classifier), let along attacking it. In contrast, a representation-level attack can overcome the two challenges in a single shot. As discussed in Section 2, for any classifier g acting on $\boldsymbol{z}$, by maximizing the mutual information $I(\boldsymbol{z}, a)$ between $\boldsymbol{z}$ and sensitive feature a, $I(g(\boldsymbol{z}), a)$ will be maximized so long as the fairness concern exists. This implies launching attack on all labels simultaneously, including the ones where classifiers cannot be trained.

B Extension to Multi-class Sensitive Feature

To attack a FRL method trained on multi-class sensitive feature $a \in [K]$, We first define $\tilde{a}_k = \mathbf{1}(a = k)$ to mark whether the sample belongs to the k-th sensitive group. Then we immediately have $I(z, a) \geq I(z, \tilde{a}_k)$ thanks to the data processing inequality. This implies that

$$I(\boldsymbol{z}, a) \geq \frac{1}{K} \sum_{k=1}^{K} I(\boldsymbol{z}, \tilde{a}_k).$$

Note that in RHS each term is the mutual information between $\boldsymbol{z}$ and binarized $\tilde{a}_k$ and is lower bounded by the BCE loss, therefore we can approximate each one with a FLD score readily. This idea is similar to transforming a K-class classification to K one-vs-all binary classifications. We report empirical results in Appendix E.6.

Further simplification is viable by choosing some specific k and attacking $I(z, \tilde{a}_k)$ only. This allows one to launch an attack when K is large. However, it is noteworthy that handling a sensitive feature that involves many groups (large K) is a general challenge for fair machine learning. The difficulty is twofold. First, it involves more complicated constraints, making the problem harder to optimize. Second, by categorizing data into more fine-grained sensitive groups, each group will have fewer samples and the algorithm may suffer from unstable estimation issues (Jiang et al., 2022; Liu et al., 2023). As a result, when the number of sensitive groups is large, fair machine learning methods often bin them into a few larger groups in pre-processing (Zemel et al., 2013; Moyer et al., 2018; Zhao et al., 2019; Reddy et al., 2021).

C Omitted Proof of Theorem 3.4

We start with restating Theorem 3.4.

Theorem C.1. *Suppose that Assumption 3.1, 3.2, and 3.3 hold. Let P and N be the number of poisoning and total training samples, respectively. Set the learning rate to α, and assign the batch size with n. Then, the ratio of poisoning data (i.e., P/N) should satisfy*

$$\frac{P}{N} \geq c + \frac{\alpha C \sigma^2}{2n \|\nabla_\theta U(\theta)\|^2} + \frac{\alpha C}{2}, \tag{7}$$

such that the upper-level loss $U(\theta)$ (i.e., negative FLD score) is asymptotic to an optimal model. Here c is a small constant (e.g., 10^{-4}) for sufficient descent, and θ is a well-trained model on the clean data.

Proof. Without loss of generality, we assume that the victim is trained by mini-batched stochastic gradient descent, i.e., given current parameter θ^{old}, the new value is updated by

$$\theta^{\text{new}} = \theta^{\text{old}} - \frac{\alpha}{n}\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}}),$$

where $L_i(\theta)$ denotes the loss on the i-th training sample. Let p denote the number of poisoning samples selected in the current batch, we have that p follows a Hypergometric distribution. Given $p = n_p$ poisoning samples in the current batch, we collect all randomness in the minibatch gradient as

$$e^{\text{old}} = \sum_{i=1}^{n_p}\epsilon_i + \sum_{i=1}^{n-n_p}\nabla_\theta L_i(\theta^{\text{old}}).$$

According to Assumption 3.3 and 3.2 we have $\mathbb{E}[e^{\text{old}}] = 0$ and

$$\begin{aligned}\mathbb{E}\left[\|e^{\text{old}}\|^2\right] &= \mathbb{E}\left[\left\|\sum_{i=1}^{n_p}\epsilon_i + \sum_{i=1}^{n-n_p}\nabla_\theta L_i(\theta^{\text{old}}\right\|^2\right] \\ &= \sum_{i=1}^{n_p}\mathbb{E}[\|\epsilon_i\|^2] + \sum_{i=1}^{n-n_p}\mathbb{E}[\|\nabla_\theta L_i(\theta^{\text{old}})\|^2] + 0 \\ &\le n\sigma^2,\end{aligned}$$

as all crossing terms have mean zero. Moreover, we can express

$$\begin{aligned}\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}}) &= \underbrace{\sum_{i=1}^{n_p}\nabla_\theta L_i(\theta^{\text{old}})}_{\text{Poisoning samples}} + \underbrace{\sum_{i=1}^{n-n_p}\nabla_\theta L_i(\theta^{\text{old}})}_{\text{Clean samples}} \\ &= \sum_{i=1}^{n_p}\nabla_\theta U(\theta^{\text{old}}) + \sum_{i=1}^{n_p}\epsilon_i + \sum_{i=1}^{n-n_p}\nabla_\theta L_i(\theta^{\text{old}}) \\ &= n_p\nabla_\theta U(\theta^{\text{old}}) + e^{\text{old}}.\end{aligned}$$

This implies that further given Assumption 3.2, the minibatch gradient is a biased estimator of the gradient of the upper-level loss. To see this, by the law of total expectation

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}})\right] &= \mathbb{E}\left[\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}}) \mid p = n_p\right]\right] \\ &= \frac{1}{n}\mathbb{E}\left[\mathbb{E}\left[p\nabla_\theta U(\theta^{\text{old}}) + e^{\text{old}} \mid p = n_p\right]\right] \\ &= \mathbb{E}\left[p\nabla_\theta U(\theta^{\text{old}})\right] \\ &= \frac{1}{n}\nabla_\theta U(\theta^{\text{old}})\mathbb{E}[p] \\ &= \frac{P}{N}\nabla_\theta U(\theta^{\text{old}}).\end{aligned}$$

Under Assumption 3.1, descent lemma gives us

$$\begin{aligned}U(\theta^{\text{new}}) &\le U(\theta^{\text{old}}) + \langle\nabla_\theta U(\theta^{\text{old}}), \theta^{\text{new}} - \theta^{\text{old}}\rangle + \frac{C}{2}\|\theta^{\text{new}} - \theta^{\text{old}}\|^2 \\ &= U(\theta^{\text{old}}) - \alpha\langle\nabla_\theta U(\theta^{\text{old}}), \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}})\rangle + \frac{\alpha^2 C}{2n^2}\|\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}})\|^2.\end{aligned}$$

Take expectation on both side with respect to the mini-batch we get

$$\begin{aligned}\mathbb{E}[U(\theta^{\text{new}})] &\leq U(\theta^{\text{old}}) - \alpha\langle\nabla_\theta U(\theta^{\text{old}}), \frac{P}{N}\nabla_\theta U(\theta^{\text{old}})\rangle + \frac{\alpha^2 C}{2n^2}\mathbb{E}\|\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}})\|^2 \\ &= U(\theta^{\text{old}}) - \frac{\alpha P}{N}\|\nabla_\theta U(\theta^{\text{old}})\|^2 + \frac{\alpha^2 C}{2n^2}\mathbb{E}\|\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}})\|^2, \end{aligned} \tag{8}$$

and

$$\begin{aligned}\mathbb{E}\|\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}})\|^2 &= \mathbb{E}\left[\mathbb{E}\left[\|\sum_{i=1}^{n}\nabla_\theta L_i(\theta^{\text{old}})\|^2 \mid p = n_p\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\|p\nabla_\theta U(\theta^{\text{old}}) + e^{\text{old}}\|^2 \mid p = n_p\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[p^2\|\nabla_\theta U(\theta^{\text{old}})\|^2 + \|e^{\text{old}}\|^2 + 2p\langle\nabla_\theta U(\theta^{\text{old}}), e^{\text{old}}\rangle \mid p = n_p\right]\right] \\ &= \mathbb{E}[p^2]\|\nabla_\theta U(\theta^{\text{old}})\|^2 + \mathbb{E}[\|e^{\text{old}}\|^2] + 0 \\ &\leq n^2\|\nabla_\theta U(\theta^{\text{old}})\|^2 + n\sigma^2. \end{aligned}$$

Plugging in back to equation (8) we have

$$\begin{aligned}\mathbb{E}[U(\theta^{\text{new}})] &\leq U(\theta^{\text{old}}) - \frac{\alpha P}{N}\|\nabla_\theta U(\theta^{\text{old}})\|^2 + \frac{\alpha^2 C}{2n^2}(n^2\|\nabla_\theta U(\theta^{\text{old}})\|^2 + n\sigma^2) \\ &= U(\theta^{\text{old}}) - (\frac{\alpha P}{N} - \frac{\alpha^2 C}{2})\|\nabla_\theta U(\theta^{\text{old}})\|^2 + \frac{\alpha^2 C\sigma^2}{2n}. \end{aligned}$$

Therefore, a sufficient descent such that

$$\mathbb{E}[U(\theta^{\text{new}})] \leq U(\theta^{\text{old}}) - c\alpha\|\nabla_\theta U(\theta^{\text{old}})\|^2,$$

for some $c \geq 0$ can be guaranteed by

$$(\frac{\alpha P}{N} - \frac{\alpha^2 C}{2})\|\nabla_\theta U(\theta^{\text{old}})\|^2 - \frac{\alpha^2 C\sigma^2}{2n} \geq c\alpha\|\nabla_\theta U(\theta^{\text{old}})\|^2$$

Rearrange

$$\frac{P}{N} \geq c + \frac{\alpha C\sigma^2}{2n\|\nabla_\theta U(\theta^{\text{old}})\|^2} + \frac{\alpha C}{2}$$

This completes our proof. □

D More Implementation Details

We provide more details about victims' architectures and training. Our model architectures follow official implementations in Moyer et al. (2018); Zhao et al. (2019).

On Adult dataset, we use

- CFAIR:
  - Encoder: linear, representation $\boldsymbol{z} \in \mathbb{R}^{60}$.
  - Discriminators: one hidden layer with width 50, using ReLU activation.
  - Classifier: linear.
  - Training: AdaDelta optimizer with learning rate 0.1, batchsize 512, epochs 50.
- CFAIR-EO:
  - Encoder: linear, representation $\boldsymbol{z} \in \mathbb{R}^{60}$.
  - Discriminators: one hidden layer with width 50, using ReLU activation.

  - Classifier: linear.
  - Training: AdaDelta optimizer with learning rate 0.1, batchsize 512, epochs 50.
- ICVAE-US:
  - Encoder: one hidden layer with width 64, output representation $\boldsymbol{z} \in \mathbb{R}^{30}$, using Tanh activation.
  - Decoder: one hidden layer with width 64, using Tanh activation.
  - Classifier: one hidden layer with width 64. using Tanh activation.
  - Training: Adam optimizer with learning rate 0.001, batchsize 512, epochs 50.
- ICVAE-S:
  - Encoder: one hidden layer with width 64, output representation $\boldsymbol{z} \in \mathbb{R}^{30}$, using Tanh activation.
  - Decoder: one hidden layer with width 64, using Tanh activation.
  - Classifier: one hidden layer with width 64. using Tanh activation.
  - Training: Adam optimizer with learning rate 0.001, batchsize 512, epochs 50.

On German dataset, we use

- CFAIR:
  - Encoder: linear, representation $\boldsymbol{z} \in \mathbb{R}^{60}$.
  - Discriminators: one hidden layer with width 50, using ReLU activation.
  - Classifier: linear.
  - Training: AdaDelta optimizer with learning rate 0.05, batchsize 64, epochs 300.
- CFAIR-EO:
  - Encoder: linear, representation $\boldsymbol{z} \in \mathbb{R}^{60}$.
  - Discriminators: one hidden layer with width 50, using ReLU activation.
  - Classifier: linear.
  - Training: AdaDelta optimizer with learning rate 0.05, batchsize 64, epochs 300.
- ICVAE-US:
  - Encoder: one hidden layer with width 64, output representation $\boldsymbol{z} \in \mathbb{R}^{30}$, using Tanh activation.
  - Decoder: one hidden layer with width 64, using Tanh activation.
  - Classifier: one hidden layer with width 64. using Tanh activation.
  - Training: Adam optimizer with learning rate 0.001, batchsize 64, epochs 300.
- ICVAE-S:
  - Encoder: one hidden layer with width 64, output representation $\boldsymbol{z} \in \mathbb{R}^{30}$, using Tanh activation.
  - Decoder: one hidden layer with width 64, using Tanh activation.
  - Classifier: one hidden layer with width 64. using Tanh activation.
  - Training: Adam optimizer with learning rate 0.001, batchsize 64, epochs 300.

During training, we followed GradMatch (Geiping et al., 2020) and did not shuffle training data after each epoch. For better comparison, victims were always initialized with random seed 1 to remove randomness during the pre-training procedure. In different replications, we selected different poisoning samples with different random seeds. Experiments that consist of 5 replications used seed 1 to 5 respectively.

E More Experiment Results

E.1 Practicability of Perturbing Pre-processed Data

Many poisoning attacks on images classifiers perturb raw data (namely pixels, (Huang et al., 2020; Geiping et al., 2020)), in this work we perturb pre-processed data. However, this does not necessarily undermine the practical significance of our work. To see why attacking the pre-processed data is practical, we take a view from the scope of data security. Many FRL methods are applied in high-stakes domains, such as loan application or job market screening. Due to the privacy concern, Data anonymization has been used by more and more data providers to protect their data privacy and is often done as a part of the data pre-processing as discussed in Iyengar (2002); Ram Mohan Rao et al. (2018). In such cases, a malicious data provider can release a poisoned pre-processed (anonymized) dataset and launch the attack on victim models trained with it.

E.2 FLD score approximates BCE Loss

We evaluated how well the optimal BCE loss of logistic regression can be approximated by three FLD scores used in our experiments: FLD, sFLD, and EUC. To this end, we train each victim for 50 epochs and compute the empirical optimal BCE loss of a logistic regression to predict a from representation $\boldsymbol{z}$ after each epoch. Then we compare the trend of BCE loss versus FLD, sFLD, and EUC scores.

Figure 4 visualizes the negative value of FLD, sFLD, and EUC and associated BCE losses of four victims trained on Adult and German dataset with and without poisoning samples crafted by corresponding ENG attack after each epoch. In all cases, the three scores approximate how BCE loss changes very well.

E.3 Effects on DP Violations

Here we report increase of DP violations of victims attacked by three ENG-based attacks and AA-based baselines in Figure 5. As analyzed in Section 2, our attack successfully increased the DP violation significantly on most setting, clearly establishing their effectiveness.

E.4 Effects on Accuracy

Here we report change of accuracy of predicting y from representation $\boldsymbol{z}$ learned by victims attacked by three ENG-based attacks and AA-based baselines in Figure 6. In general, all attacks had relatively subtle influence on prediction accuracy, and our proposed ENG-based attacks often changed accuracy less. These results imply that poisoned representations are still of high utility, and it will be difficult for the victim trainer to identify the attack by checking the prediction performance alone.

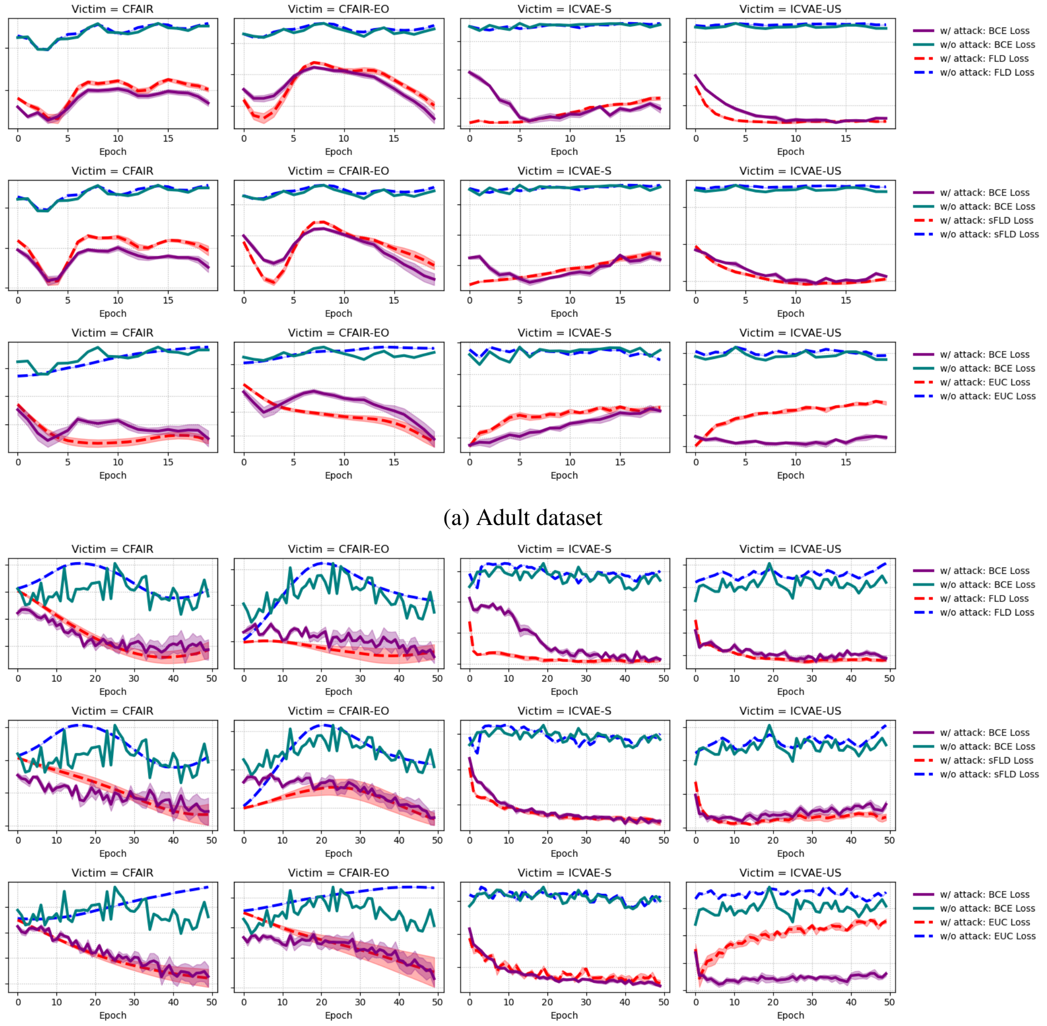
E.5 Performance of Elastic-Net Penalty

Here we report more experimental results on evaluating the performance of elastic-net penalty under varying hyper-parameters. Figure 7 and 9 reports how elastic-net penalty affects the attack performance, corresponding L_1 and L_2 norms of learned perturbations are reported in Figure 10 and 11, respectively.

We further compare elastic-net penalty versus L_1 norm penalty for regulating the attack. Figure 8 shows corresponding decrease of BCE loss and Figure 12 shows corresponding L_1 and L_2 norms. Compared with elastic-net penalty, penalizing L_1 norm only usually resulted in larger L_1 and L_2 norms while the attack performance was hardly improved, especially when small- to intermediate-level coefficient(s) (λ_1 and λ_2) of the regularizer is used.

Given the advantages of utilizing elastic-net penalty, one may ask if it can be used in other attacks such as AA (Mehrabi et al., 2021) as well. Here we delve into the difficulty of such an application, highlighting the suitability of elastic-net penalty for our attack by nature.

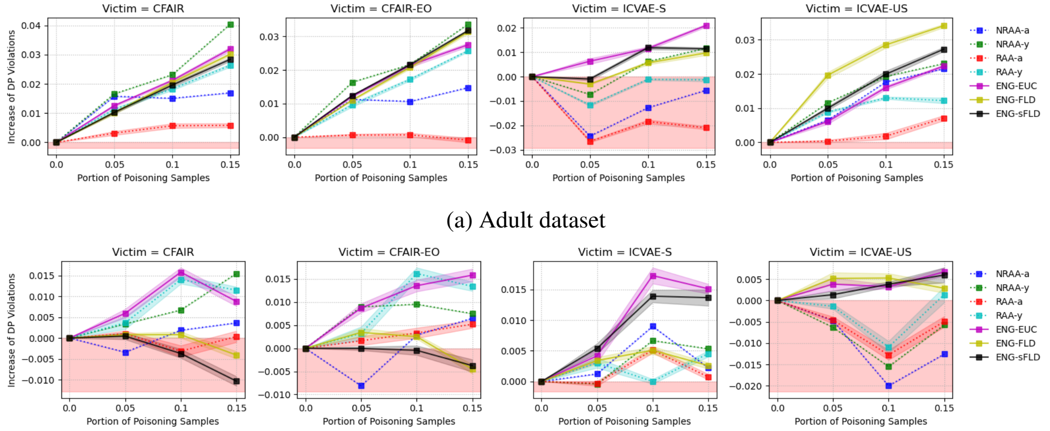
According to Mehrabi et al. (2021), both RAA and NRAA constructed poisoning samples from a chosen anchor point by perturbing its $\boldsymbol{x}$ randomly within a τ-ball and flipping its y or a. The



(a) Adult dataset

(b) German dataset

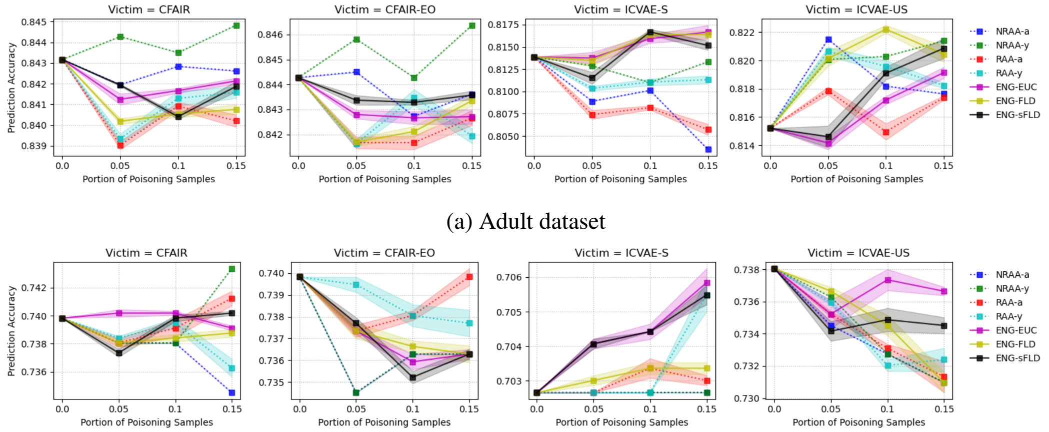
Figure 4: Changes of FLD, sFLD, and EUC loss (the negative score) and corresponding BCE loss. Results are averaged over 5 independent replications and bands show standard errors.



(a) Adult dataset

(b) German dataset

Figure 5: Increase of DP violations from different attackers using 5% - 15% training samples for poisoning, Results are averaged over 5 independent replications and bands show standard errors.



(a) Adult dataset

(b) German dataset

Figure 6: Accuracy of predicting y from $\boldsymbol{z}$ when different attackers using 5% - 15% training samples for poisoning, Results are averaged over 5 independent replications and bands show standard errors.

main difference lies in how they choose these anchor samples: RAA draws them uniformly from the training data. NRAA computes the pairwise distance between all training data and counts how many *neighbors* and chooses the training samples that have the most neighbors with flipped y or a as anchor data. Here two samples are *neighboring* if their y and a are same and the difference in their $\boldsymbol{x}$ is smaller than a pre-specified threshold.

It is viable to define τ-ball based on the elastic-net norm. Nonetheless, the use of τ-ball will make NRAA harder to analyze in general. NRAA is thought stronger (which is empirically verified in both Mehrabi et al. (2021) and our experiment) because the chosen poisoning samples are likely to have higher influence as they have more neighbors. However, sampling from τ-ball will make some poisoning samples have more neighbors while others have less. In fact, the authors of AA used $\tau = 0$ throughout their experiments and we adopted this setting. This difficulty remains unsolved when using elastic-net norm to induce the τ-ball. In conclusion, ENG, as a perturbation-learning-based attack, has a better affinity to the elastic-net penalty than AA baselines.

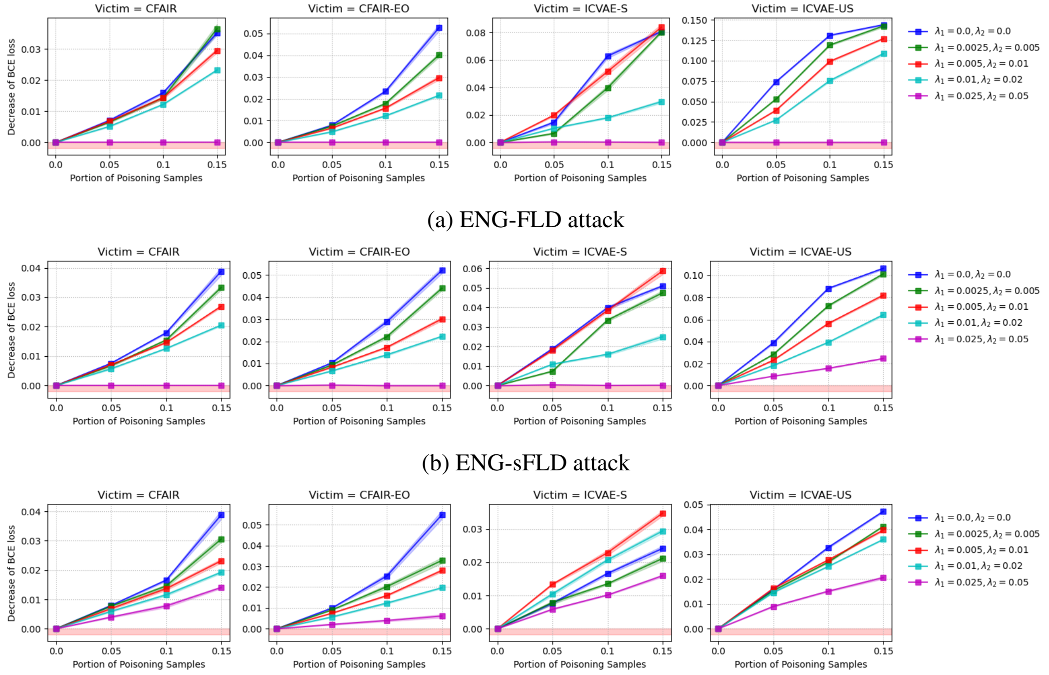
E.6 Attacking Multi-class Sensitive Feature

Here we show empirical results of attacking victims trained on Adult dataset using *race* as the sensitive feature. Due to data imbalance issue we keep *white* and *black* group as they are. All other sensitive groups are re-categorized as *others*. This results in a 3-way sensitive feature a and we measure $I(\boldsymbol{z}, a)$ by the decrease of cross-entropy (CE) loss to predict a from $\boldsymbol{z}$ with a linear Softmax classifier. Note that both the CE loss and the averaged one-vs-all BCE loss (see Appendix B for details) lower bounds $I(\boldsymbol{z}, a)$, but we report CE loss for the sake of better interpretability. Due to the lack of official implementation of CFAIR and CFAIR-EO on multi-class sensitive feature, we only attack ICVAE-US and ICVAE-S. As shown in Figure 13, ENG-based attack outperforms four AA baselines.

E.7 Defense Against ENG Attacks

Here we present reducing batch size as a defense strategy against ENG attacks. We only defend against successful attacks and focus on Adult dataset where attacks used $\lambda_1 = 0.0025, \lambda_2 = 0.005$. We vary the batch size between 256 to 1024 and keep all other settings same as in Section 4. Resultant decrease of upper-level loss $-s$ and BCE loss are reported in Figure 14. Reducing batch size successfully helped weaken the performance of attacking 3 out of 4 victims in terms of $-s$.

Note that according to Theorem 3.4, it is conceptually viable to increasing learning rate α to defend. However, using a large learning rate in a well-trained victim model may have side effect of making it diverge. Because of this, we consider reducing batch size a better and more practical choice.

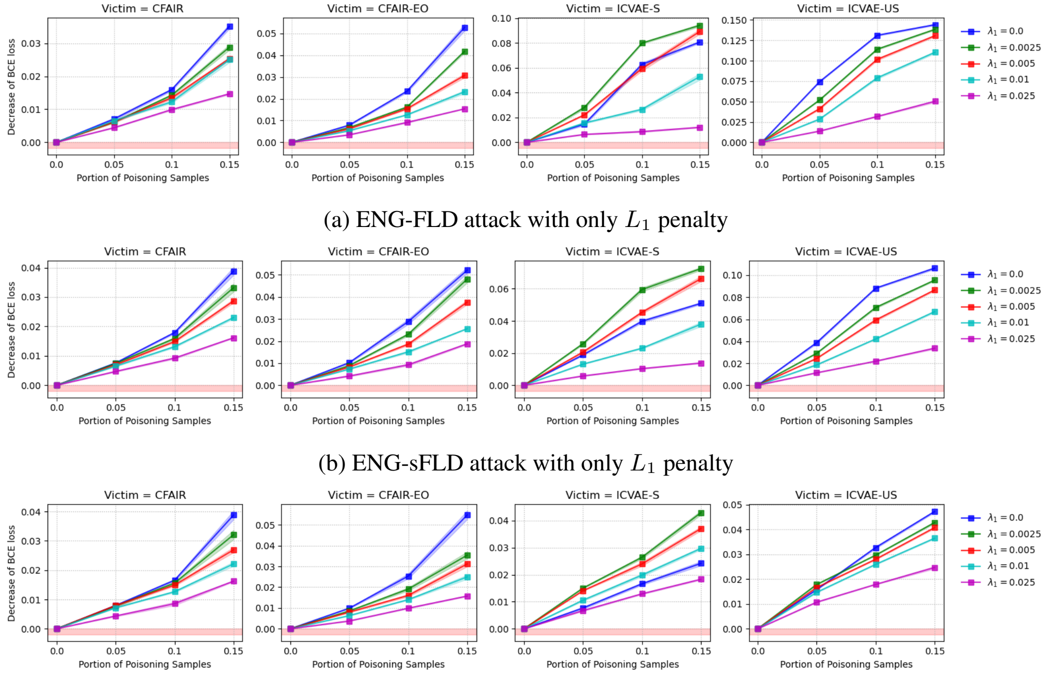


(a) ENG-FLD attack

(b) ENG-sFLD attack

(c) ENG-EUC attack

Figure 7: Decrease of BCE loss achieved by different ENG-based attacks with varying hyper-parameters, victims are trained on Adult dataset. Results are averaged over 5 independent replications and bands show standard errors.

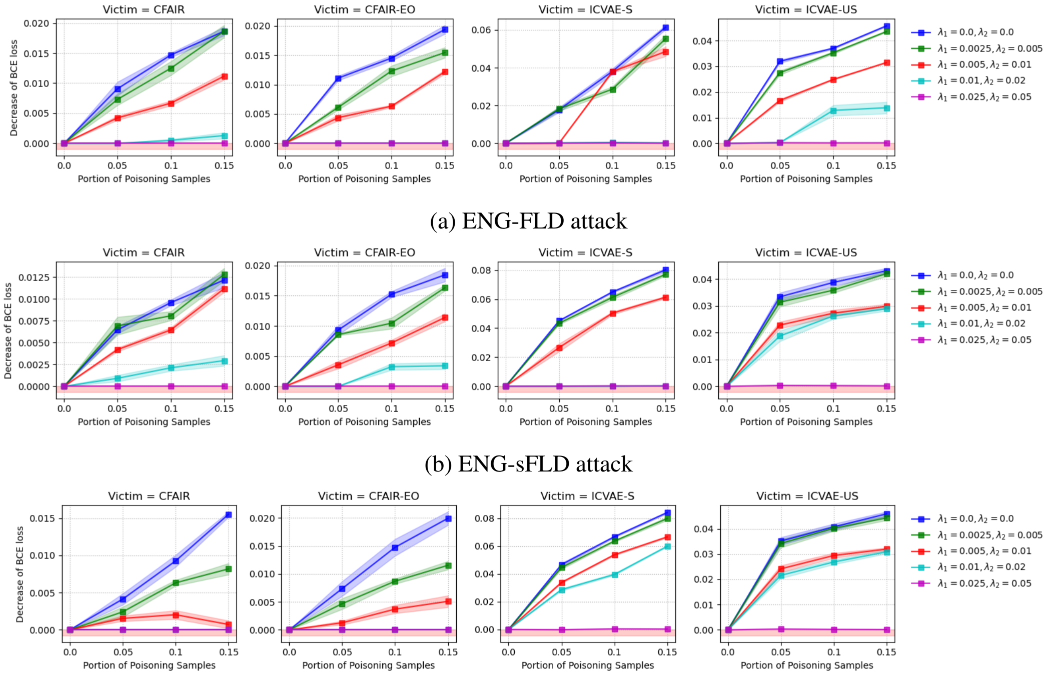


(a) ENG-FLD attack with only L_1 penalty

(b) ENG-sFLD attack with only L_1 penalty

(c) ENG-EUC attack with only L_1 penalty

Figure 8: Decrease of BCE loss achieved by different ENG-based attacks with varying hyper-parameters on L_1 penalty and no L_2 penalty, victims are trained on Adult dataset. Results are averaged over 5 independent replications and bands show standard errors.



(a) ENG-FLD attack

(b) ENG-sFLD attack

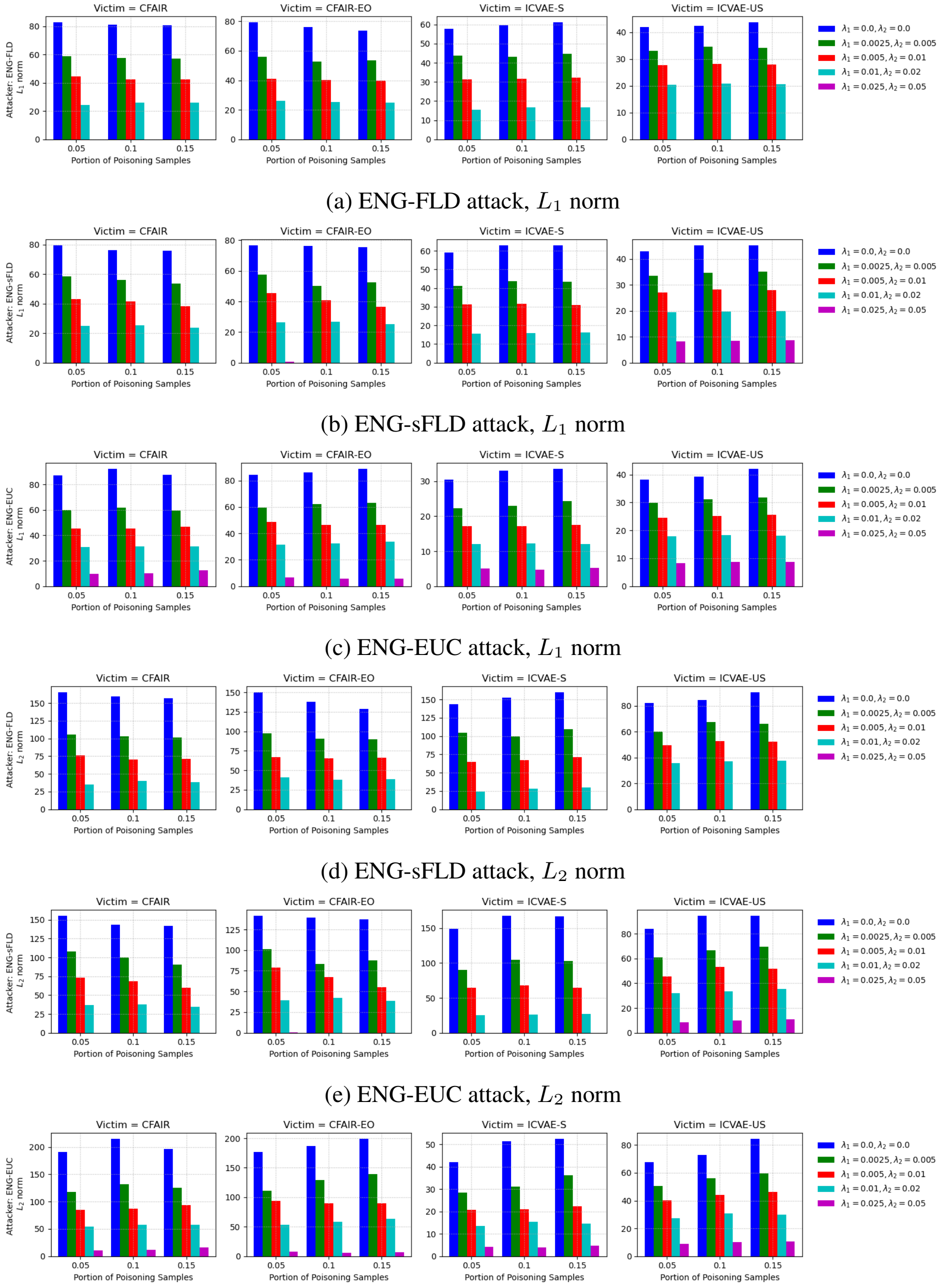
(c) ENG-EUC attack

Figure 9: Decrease of BCE loss achieved by different ENG-based attacks with varying hyperparameters, victims are trained on German dataset. Results are averaged over 5 independent replications and bands show standard errors.

E.8 More Experiment Results on COMPAS and Drug Consumption Datasets

Here we present more experimental results of attacking four FRL methods trained on COMPAS and Drug Consumption datasets. We adopted the same setting for data preprocessing and training-testing splitting pipelines as presented in Section 4 on Adult and German datasets. In specific, we report decrease of BCE losses in Figure 15, increase of DP violations in Figure 16, and accuracy of predicting y from $\boldsymbol{z}$ in Figure 17. Again, our attacks succeeded on all cases, outperforming AA baselines to a large extent.

In terms of model architecture, we adopted the recommended architectures from Zhao et al. (2019) for CFAIR and CFAIR-EO on COMPAS dataset; and the default setting presented in Appendix D on Drug Consumption datasets due to the lack of official implementations. For ICVAE-US and ICVAE-S, we used the default setting from Appendix D on both datasets because of the lack of official implementations.



(a) ENG-FLD attack, L_1 norm

(b) ENG-sFLD attack, L_1 norm

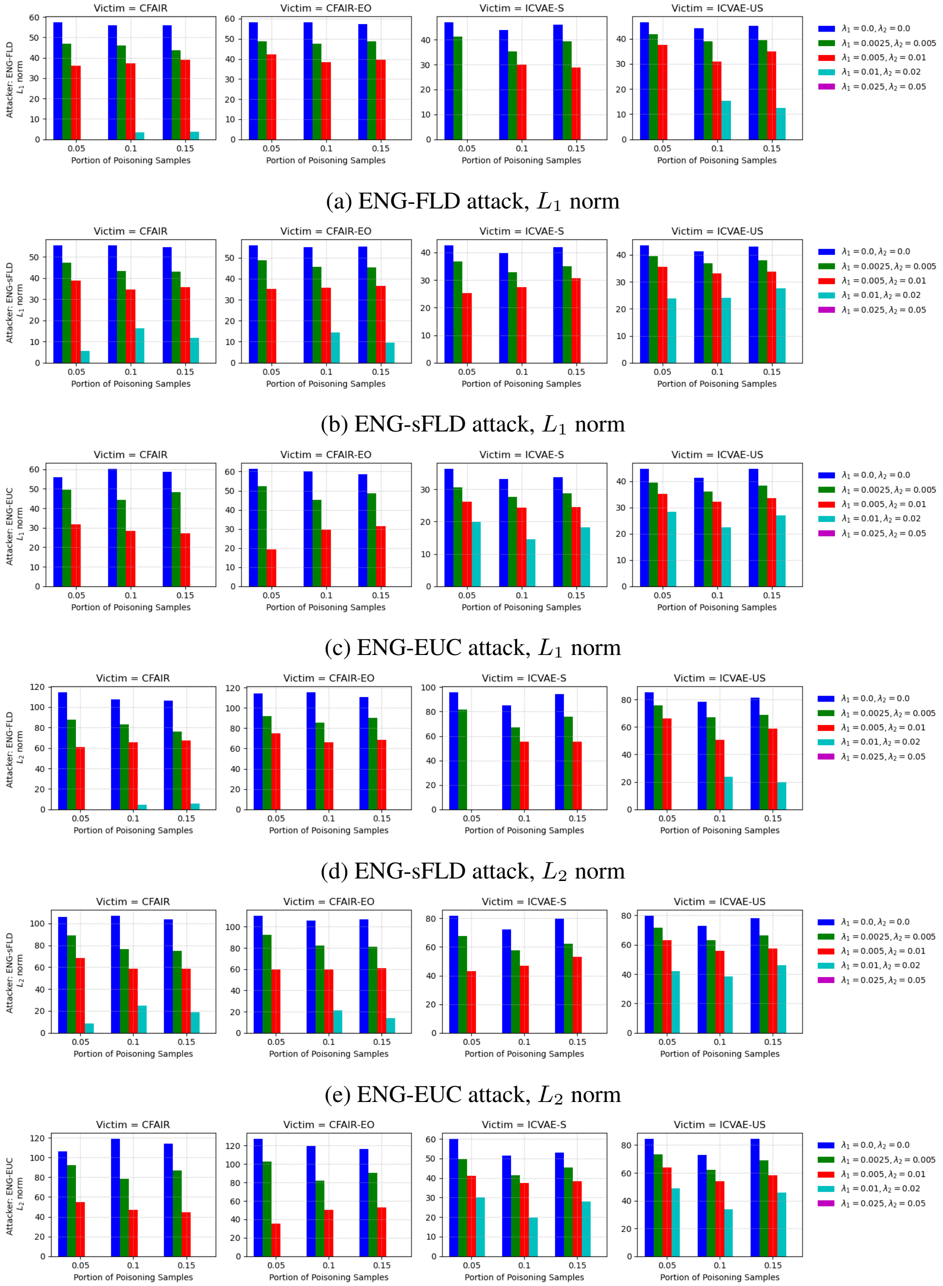
(c) ENG-EUC attack, L_1 norm

(d) ENG-sFLD attack, L_2 norm

(e) ENG-EUC attack, L_2 norm

(f) ENG-sFLD attack, L_2 norm

Figure 10: Averaged L_1 and L_2 norm of perturbations learned by different ENG-based attacks with varying hyper-parameters, victims are trained on Adult dataset.



(a) ENG-FLD attack, L_1 norm

(b) ENG-sFLD attack, L_1 norm

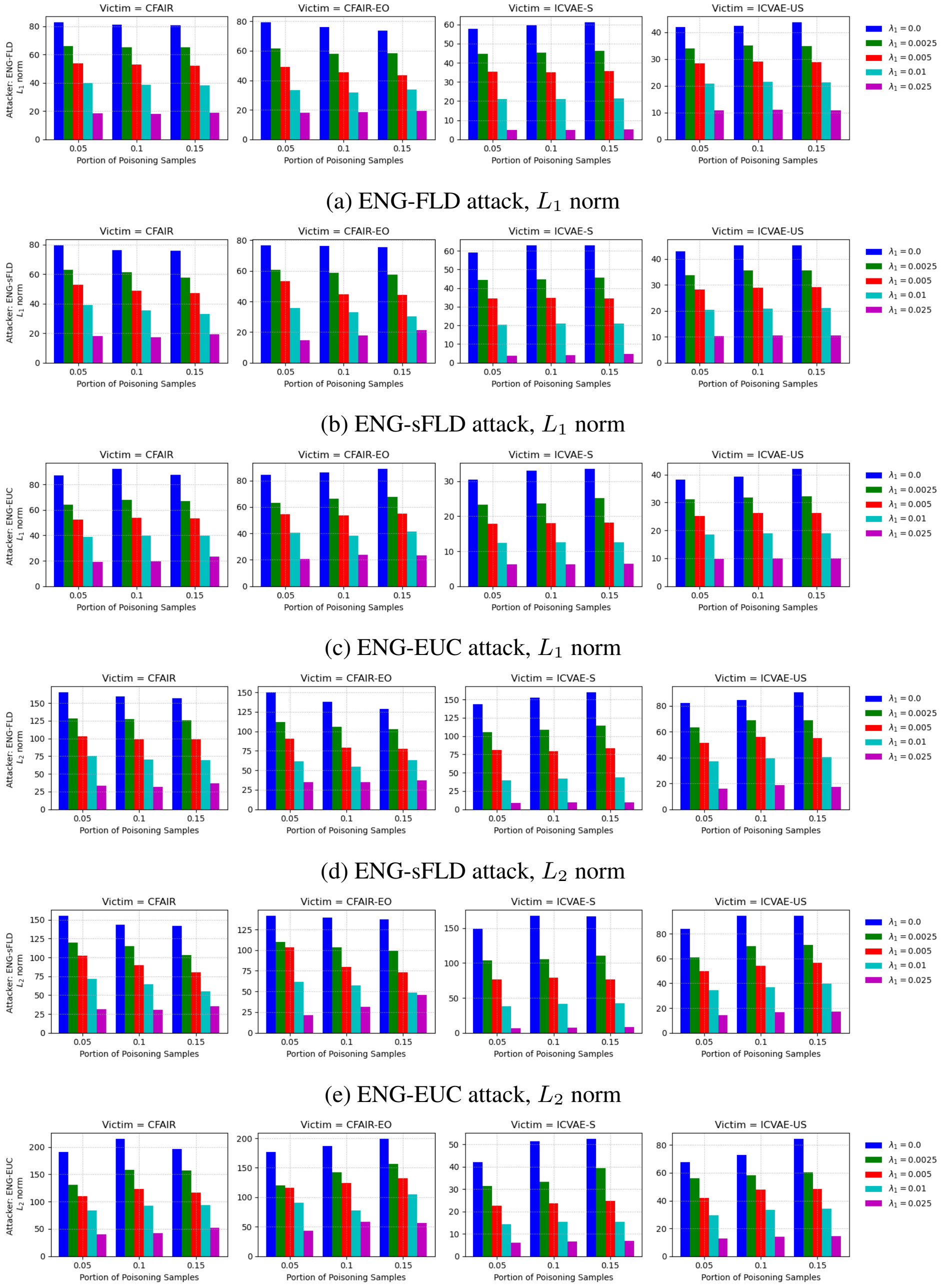
(c) ENG-EUC attack, L_1 norm

(d) ENG-sFLD attack, L_2 norm

(e) ENG-EUC attack, L_2 norm

(f) ENG-sFLD attack, L_2 norm

Figure 11: Averaged L_1 and L_2 norm of perturbations learned by different ENG-based attacks with varying hyper-parameters, victims are trained on German dataset.



(a) ENG-FLD attack, L_1 norm

(b) ENG-sFLD attack, L_1 norm

(c) ENG-EUC attack, L_1 norm

(d) ENG-sFLD attack, L_2 norm

(e) ENG-EUC attack, L_2 norm

(f) ENG-sFLD attack, L_2 norm

Figure 12: Averaged L_1 and L_2 norm of perturbations learned by different ENG-based attacks with varying hyper-parameters on L_1 penalty and no L_2 penalty, victims are trained on Adult dataset.

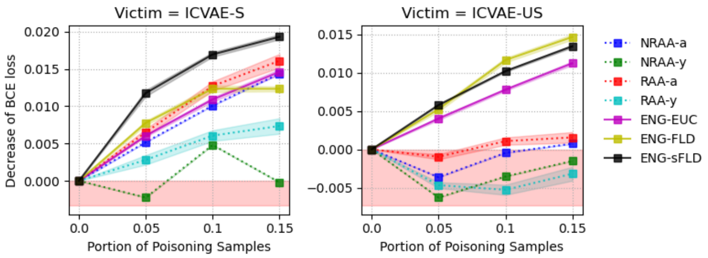
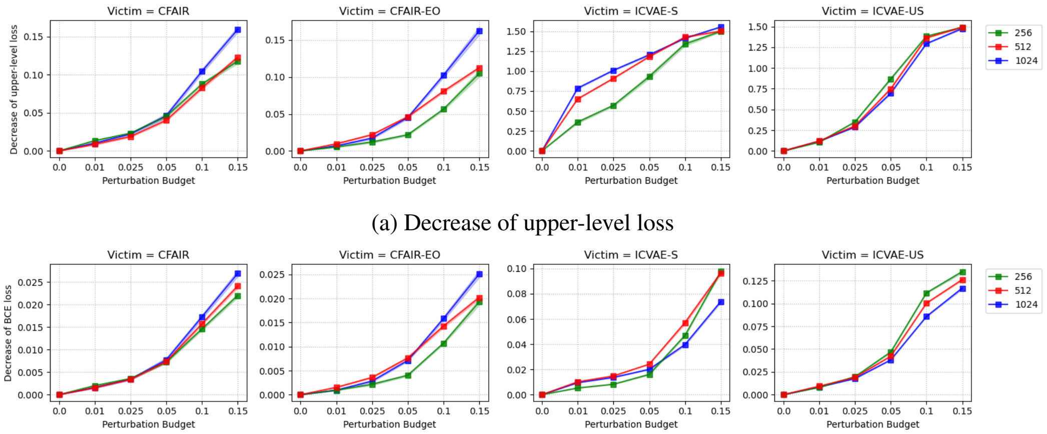


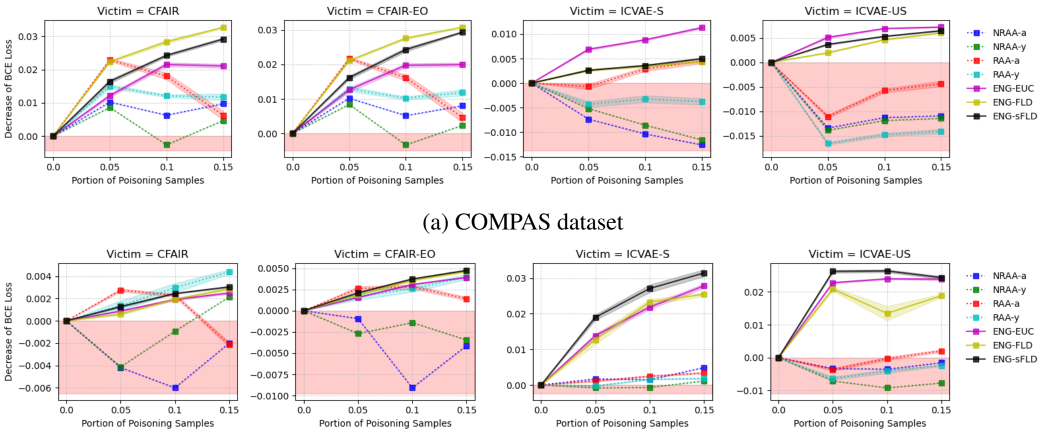
Figure 13: Decrease of CE loss from attacking victims trained on Adult dataset with sensitive feature *race*.



(a) Decrease of upper-level loss

(b) Decrease of BCE loss

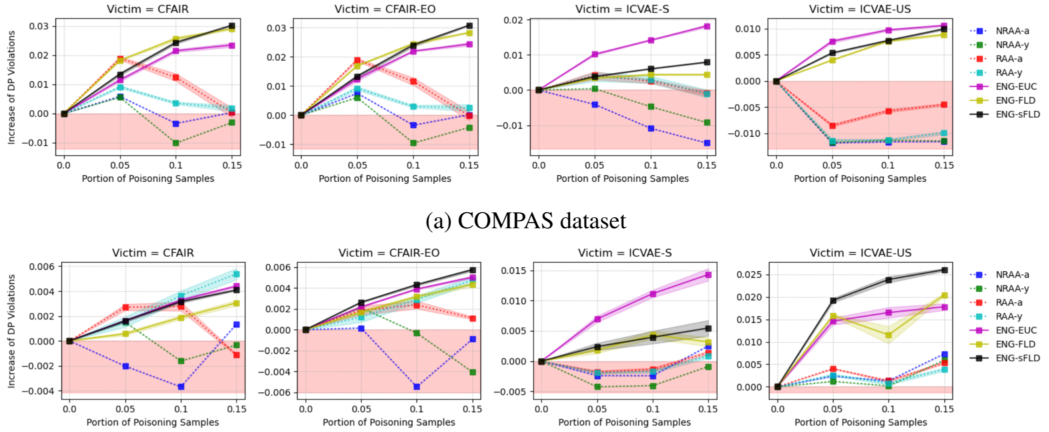
Figure 14: Effectiveness of reducing batch size as a defense against the proposed ENG-based attacks, different attackers using 1% - 15% training samples for poisoning. Results are averaged over 5 independent replications and bands show standard errors.



(a) COMPAS dataset

(b) Drug Consumption dataset

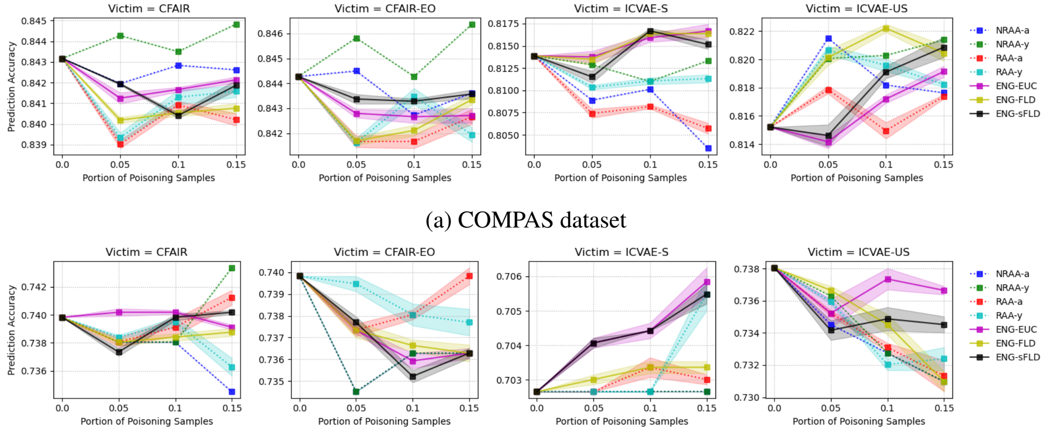
Figure 15: Decrease of BCE loss from different attackers using 5% - 15% training samples for poisoning, Results are averaged over 5 independent replications and bands show standard errors.



(a) COMPAS dataset

(b) Drug Consumption dataset

Figure 16: Increase of DP violations from different attackers using 5% - 15% training samples for poisoning, Results are averaged over 5 independent replications and bands show standard errors.



(a) COMPAS dataset

(b) Drug Consumption dataset

Figure 17: Accuracy of predicting y from z when different attackers using 5% - 15% training samples for poisoning, Results are averaged over 5 independent replications and bands show standard errors.