Differentially-private and plausible counterfactuals

Anonymous authors
Paper under double-blind review

Abstract

Counterfactual explanations are particularly appealing in high-stakes domains such as finance and hiring, as they provide affected users with suggestions on how to alter their profiles to receive a favorable outcome. However, existing methods are characterized by a privacy-quality trade-off. More precisely, as highlighted in recent works, instance-based approaches generate plausible counterfactuals but are vulnerable to privacy attacks, while perturbation-based methods offer better privacy at the cost of lower explanation quality. In this paper, we propose to solve this dilemma by introducing a diverse set of differentially-private mechanisms for generating counterfactuals, providing a high resistance against privacy attacks while maintaining high utility. These mechanisms can be integrated at different stages of the counterfactual generation pipeline (*i.e.*, pre-processing, in-processing or post-processing), thereby offering maximal flexibility during the design for the model provider. We have performed an empirical evaluation of the proposed approaches on a wide range of datasets and models to evaluate their effect on the privacy and utility of the generated counterfactuals. Overall, the results obtained demonstrate that in-processing methods significantly reduce the success rate of privacy attacks while moderately impacting the quality of counterfactuals generated. In contrast, pre-processing and post-processing mechanisms achieve a higher level of privacy but at a greater cost in terms of utility, thus being more suitable for scenarios in which privacy is paramount.

1 Introduction

The use of machine learning models has become widespread in many spheres of our society. However, explaining how these models work is essential for fostering trust in their outcomes Guidotti et al. (2018); Molnar (2020); Vashney (2022). In particular, in this work, we focus on counterfactuals Wachter et al. (2017), which are a form of explanation that suggests modifications to the input profile aiming to alter the model's prediction Wachter et al. (2017). As such, counterfactuals are one of the popular post-hoc techniques to provide explanations to affected users to help them improve their predictions. However, recent works have shown that an adversary can also exploit the information provided by counterfactuals to conduct privacy attacks. For instance, counterfactuals can be leveraged to perform membership inference and model extraction attacks Shokri et al. (2020); Aïvodji et al. (2020). Thus, it is important to investigate the privacy risks associated with counterfactuals as well as to develop countermeasures to mitigate them.

One popular category of methods for generating counterfactuals is perturbation-based approaches Wachter et al. (2017), which generate counterfactuals by perturbing the feature values of an instance towards the decision boundary. However, they have been criticized for their lack of plausibility Laugel et al. (2019). To address these limitations, researchers have proposed instance-based counterfactual methods Laugel et al. (2018); Keane & Smyth (2020), which use samples from the training dataset as a basis for counterfactual generation. A representative method of this family is `NICE` (*Nearest Instance Counterfactual Explanation*) Brughmans et al. (2023), which selects the nearest instance in the training set to generate a counterfactual. However, as shown by Goethals et al. (2023), this type of counterfactual is vulnerable to explanation linkage attacks Dunn (1946) due to their use of training instances. In a nutshell, this attack links the adversary's existing knowledge with counterfactual data to extract sensitive information about profiles from the training set. The authors also proposed a k-anonymization approach to mitigate this vulnerability. However, k-anonymity Samarati

& Sweeney (1998) is known to be vulnerable to homogeneity attacks Machanavajjhala et al. (2007), when all instances in one k-anonymized group share the same sensitive attribute values.

In our setting, the training dataset is considered private and users are assumed to have black-box access to the model (more precisely, label-only), meaning that when they query it, they only receive the model's prediction regarding their query. In addition, they also receive as explanation one counterfactual specific to their query (*e.g.*, if the model's decision is considered undesirable). This generated counterfactual is not considered to be public information and is typically only available to the user who submitted the query.

Summary of contributions. Given that perturbation-based counterfactuals often sufferfrom low plausibility and instance-based counterfactuals are vulnerable to explanation linkage attacks, we propose a hybrid approach leveraging the strengths of both worlds. Specifically, our framework uses instance-based counterfactuals to improve plausibility while incorporating differential privacy (DP) guarantees Dwork et al. (2006) to prevent explanation linkage attacks. In addition, our framework is flexible enough to enable DP at various steps of the counterfactual generation pipeline, as illustrated in Figure 1.

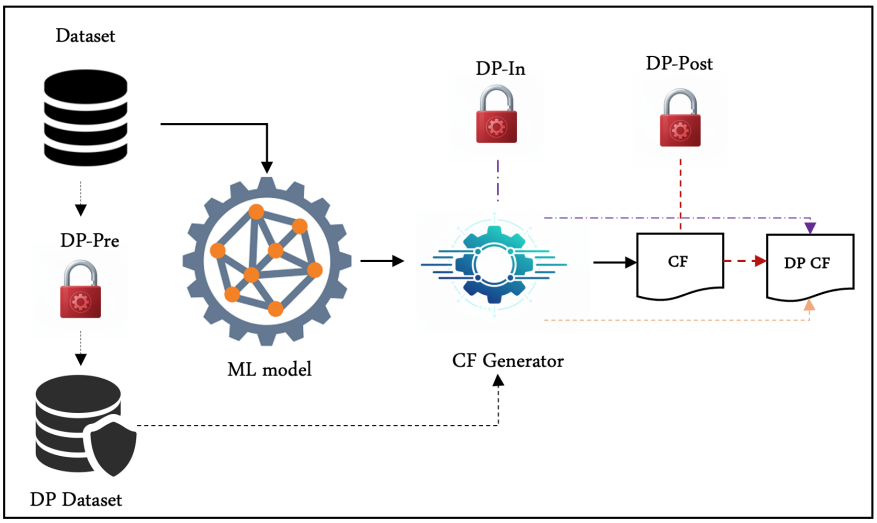


Figure 1: Illustration of the different steps of the counterfactual generation pipeline in which our mechanisms for generating DP counterfactuals can be applied.

More precisely, our main contributions can be summarized as follows:

- **Pre-processed DP-counterfactuals (`DP-Pre`).** In this method, we first generate DP data using the Laplace mechanism Dwork et al. (2014) and randomized response Warner (1965). This DP data is then used as input to the instance-based counterfactual generation mechanism. This approach ensures that the input data itself is differentially-private, mitigating privacy risks at the initial data processing stage without requiring any changes to the rest of the counterfactual generation pipeline.

- **In-processed DP-counterfactuals (`DP-In`).** We have also designed algorithms to implement DP directly in the counterfactual generation mechanism. More precisely, `Inline_DP` is a NICE-based counterfactual approach that works by selecting k nearest neighbours and then applying the Laplace mechanism Dwork et al. (2006) and randomized response Warner (1965) to the feature values to satisfy DP. Afterwards, the counterfactual is generated based on these values. This approach ensures that the features are perturbed directly, thus protecting privacy during the counterfactual generation process. This in-processed mechanism was specifically designed to be integrated into `NICE`, although it can be adapted to other instance-based counterfactual generation methods.

- **Post-processed DP-counterfactuals (`DP-Post`).** In this approach, we apply the randomized response Warner (1965) and the Laplace mechanisms Dwork et al. (2006) as a post-processing[1] step on the counterfactuals generated by the instance-based method. This approach provides a robust privacy-preserving layer after the initial generation of instance-based counterfactuals.

To assess their performance, we have evaluated the DP-counterfactual generation methods across multiple datasets and privacy regimes, using diverse metrics such as counterfactual plausibility and re-identification rate. The results obtained demonstrate their effectiveness in mitigating explanation linkage attacks. Note that we have chosen `NICE` as the instance-based counterfactual generation method because it outperforms other instance-based methods, such as CBR Keane & Smyth (2020) and SEDC Fernández-Loría et al. (2020) and perturbation-based methods like DiCE Mothilal et al. (2020) and CFproto Van Looveren & Klaise (2021) as shown in Brughmans et al. (2023). However, it is possible to directly apply our pre-processed and post-processed techniques to any other instance-based counterfactual approach, while our in-processed mechanisms may need some adaptation regarding how they choose their basic instance and generate counterfactuals.

Outline. The paper is organized as follows. First, in Section 2, we introduce the background notions on counterfactuals and DP necessary to the understanding of our work before describing in Section 3 the existing related work. Afterwards, we present our proposed solutions in Section 4. Finally, in Section 5, we present the experimental findings and analysis to assess the efficiency of each solution in mitigating explanation linkage attacks before concluding with future works in Section 6.

2 Background

In this section, we introduce the necessary background on counterfactuals as well as differential privacy.

2.1 Counterfactuals

As highlighted by Martens & Provost (2014) and Wachter et al. (2017), counterfactual explanations differ from other forms of post-hoc explanation techniques in that they suggest changes to the input profile to alter the model's prediction. More precisely, given an input profile with feature values $x_1^0, \ldots, x_1^d$ and the corresponding model's prediction y_1, a counterfactual explanation method generates a counterfactual with feature values $cf^1, \ldots, cf^n$ satisfying two conditions : (1) the model should assign it a different prediction than the original instance and (2) it should be close to the original instance in terms of a predefined distance, with the Euclidean distance being one of the most commonly used in counterfactuals. This means that counterfactual explanations should be generated by making as few changes as possible to the input profile features, yet still resulting in a different prediction. Various metrics have been defined to measure the quality of counterfactuals Karimi et al. (2022). For instance, `Proximity` measures the distance between the counterfactual and the original instance while `Sparsity` refers to the minimum number of features that need to be changed to achieve the counterfactual. Finally, `Plausibility` measures if the counterfactuals generated are realistic with respect to the training data distribution. For instance, consider a bank customer who applies for a credit limit increase and receives a rejection. She may wish to understand how to modify her financial profile to obtain approval or compare her case with similar applicants who were successful, to understand the reasons for the denial. To meet the aforementioned criteria, the counterfactual profile generated should be relatively similar to the original instance (proximity), involve only minimal changes to the original features (sparsity), while remaining within the bounds of realistic scenarios (plausibility). For example, it would be unrealistic to suggest that the applicant should earn a Ph.D. by the age of eighteen.

The two main families of approaches for counterfactual generation are perturbation-based and instance-based algorithms Karimi et al. (2022). Perturbation-based algorithms adjust feature values towards the decision boundary to achieve the desired model response. While these algorithms minimize the distance between the original instance and counterfactuals, they lack plausibility because they apply the changes towards the decision boundary without considering how these changes affect the generated profile.

[1]Note that we refer to "post-processing" as the application of DP mechanisms to counterfactuals after their generation, rather than the broader definition of post-processing in the context of DP.

Instead, instance-based algorithms select one instance from the training dataset's counterfactual class and use it to generate a counterfactual. The exact process to generate counterfactuals from this instance varies among existing algorithms. In this work, we rely on `NICE` Brughmans et al. (2023), which is one of the state-of-the-art instance-based algorithms that improve the proximity and plausibility in comparison to other instance-based counterfactuals. In a nutshell, `NICE` first identifies the nearest neighbour of the original instance for which the model makes a different prediction. Then, through an iterative process, the feature values of the considered instance are replaced with the values of the nearest neighbour until the model changes its prediction. The selection of the feature values is based on a reward function, which integrates several criteria related to the quality of counterfactual, namely proximity, sparsity and plausibility.

2.2 Differential Privacy

Differential privacy (DP) is a privacy model that provides strong guarantees by ensuring that the contribution of a particular profile will have a limited impact on the output of a computation. More precisely, DP is formally defined as follows Dwork et al. (2014).

Definition 1 (Differential privacy)**.** *A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is ϵ-differentially private if for all $S \subseteq \mathcal{R}ange(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$.* $\Pr[\mathcal{M}(x) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(y) \in S]$.

Two datasets x and y are called adjacent *(or* neighboring*) if they differ in the data of exactly one individual, i.e., $\|x - y\|_1 \leq 1$.*

In this paper, we will use the following mechanisms for building DP counterfactuals. One way to satisfy DP is through the *Laplace mechanism*, which adds Laplacian noise to the output of the function. This Laplacian noise is randomly sampled from a Laplace distribution defined based on the parameters related to the privacy budget and sensitivity of the function Dwork et al. (2006). Specifically, the sensitivity s represents the maximum change to the function output $f \in \mathcal{R}$ between two adjacent datasets that differ by one row. More formally, to ensure ϵ-DP for a query $f(x)$, the Laplace mechanism is applied as follows: $\mathcal{M}(x) = f(x) + Lap\left(\frac{s}{\epsilon}\right)$.

The *exponential mechanism* McSherry & Talwar (2007) enforces DP by selecting a response not solely based on accuracy, but by randomly choosing from a range of possible answers. The selection process is guided by probabilities assigned to each potential response, which are determined according to how well each response aligns with the function's objective. More formally, given a set $\mathcal{R}$ of possible outputs (solutions) and a scoring (utility) function $u : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$ with sensitivity Δu, the exponential mechanism returns $r \in \mathcal{R}$ with a probability proportional to: $e^{\left(\frac{\epsilon u(x,r)}{2\Delta u}\right)}$.

Report noisy max Dwork et al. (2014) is another mechanism able to select the best response to a query in a differentially-private manner. Considering for instance the example of returning the maximum count of some items, Report Noisy Max works by adding independently generated Laplacian noise $Lap(\frac{1}{\epsilon})$ to all counts and then selecting the maximum count among these noisy values.

Finally, *randomized response* (RR) was introduced by Warner (1965) as a surveying technique guaranteing the privacy of individuals when responding to a sensitive query. More precisely, RR flips the user's response based on some probability, while otherwise reporting the original answer. For instance, when reporting the value of a bit, its true value is reported with probability $p = \frac{e^{\epsilon}}{e^{\epsilon}+1}$ while being flipped with probability $1 - p$ Kairouz et al. (2016). RR is the basis for the implementation of local DP Dwork et al. (2014).

Composition theorem: Let $M_1 : \mathbb{N}^{|X|} \to \mathcal{R}_1$ be an ε_1-differentially private algorithm, and let $M_2 : \mathbb{N}^{|X|} \to \mathcal{R}_2$ be an ε_2-differentially private algorithm. Then their combination, defined to be $M_{1,2} : \mathbb{N}^{|X|} \to \mathcal{R}_1 \times \mathcal{R}_2$ by the mapping:

$$M_{1,2}(x) = (M_1(x), M_2(x))$$

is $(\varepsilon_1 + \varepsilon_2)$-differentially private Dwork et al. (2014).

3 Related Work

Privacy attacks leveraging counterfactuals. While counterfactuals, like other explanation techniques, help to gain users' trust by improving transparency, adversaries could exploit them to perform privacy attacks. In particular, model extraction attacks leveraging counterfactuals have been designed Aïvodji et al. (2020); Kuppa & Le-Khac (2021); Wang et al. (2022). More precisely, these previous works have shown that these attacks become more accurate when leveraging the counterfactual explanations compared to standard attacks that do not utilize them. Counterfactuals have also been used to perform membership inference attacks Shokri et al. (2021); Kuppa & Le-Khac (2021). Finally, more recently, Goethals et al. (2023) have introduced a new category of attacks called explanation linkage attacks, which work against instance-based counterfactual explanations. The attack assumes that the adversary will query the model for possible counterfactuals. Afterward, by receiving a counterfactual grounded in the training dataset, the adversary will use it to infer sensitive attributes belonging to an actual instance of the training dataset. While this attack is defined as an explanation linkage attack, it can be considered to implement an attribute inference attack He et al. (2006) through linkage attacks leveraging counterfactuals. The same authors proposed a solution to mitigate this vulnerability using k-anonymity guaranteeing that for each combination of feature values that can identify a record, there are at least $k-1$ instances sharing the same combination Sweeney (2002).

While k-anonymity improves the training dataset's privacy, it has also some drawbacks. First, altering the model's prediction is no longer guaranteed when providing k-anonymous counterfactuals. Second, from a privacy perspective, k-anonymity is known to be vulnerable to homogeneity attacks Machanavajjhala et al. (2007). In practice, such attacks are possible when the sensitive attributes of all members in a group of k records share the same value, in which case the adversary can infer this information without explicitly identifying the target record. Furthermore, k-anonymous explanations need to make specific assumptions about the adversarial knowledge and the dataset distribution, which makes them vulnerable to other privacy attacks Cohen (2022) like predicate singling-out (PSO) attacks (Altman et al., 2021).

Differentially private counterfactuals. To mitigate the vulnerability of perturbation-based counterfactuals to membership inference and model extraction attacks, researchers have suggested adding DP to the generation of counterfactuals. For instance, Nelson (2022); Pentyala et al. (2023) suggested using DP mechanisms to generate synthetic data that can be used to produce counterfactuals.

Nelson (2022) trained a black-box model on real data. Then, using various differentially private techniques (DP-GAN, WDP-GAN, and DP-CTGAN), they generated DP-synthetic datasets, which Dice (the CF generation algorithm they applied) uses as the training set to generate counterfactuals. They used this technique to mitigate vulnerability to membership inference attacks. Pentyala et al. (2023) proposed a differentially private recourse path (based on the FACE algorithm). They suggested an end-to-end DP pipeline to generate a recourse path. First, they trained a differentially private black-box model on the real dataset using PATE (Papernot et al., 2018) or DPSGD (Abadi et al., 2016). Then, a DP clustering technique was applied to the training dataset, achieving cluster centers as DP representatives of the clusters. To count the number of instances in each cluster, Laplacian noise is added to the real counts, and these DP-values are used to evaluate the density while generating a recourse path.

Other works train DP models and then generate counterfactuals for those models to protect against model extraction attacks. Mochaourab et al. (2022) trained a SVM classifier, then proposed adding a DP demonstrator to the model to be used for CF generation. Since this DP version of the model loses accuracy and misclassifies some instances, only robust counterfactuals that are classified in counterfactual class by both models is returned to the user. Huang et al. (2023) suggested two methodologies to generate DP counterfactuals: first, training a DP-logistic regression model on real data and generate DP counterfactuals using this mode, and second, applying post-processing DP techniques, where laplacien noise is applied to the model's predictions, and a noisy counterfactual is generated that its DP-prediction puts it in the counterfactual class. Finally, a recent work Yang et al. (2022) relies on functional mechanisms Zhang et al. (2012) to incorporate DP into perturbation-based counterfactual methods to simultaneously prevent membership inference and model extraction. They use an autoencoder to generate differentially private class prototypes, and retuen these prototype profiles as counterfactuals. A summarized overview of these DP counterfactuals are presented in Table 1.

	Target	Technique	Objective	Benefits	Limitations
Nelson (2022)	PB	DPS	MI prevention	MI prevention	No plausibility
Pentyala et al. (2023)	RP	DPS	Realistic recourse path	High robustness, MI prevention	Distribution change, No plausibility
Mochaourab et al. (2022)	PB	DPT	Robust DP explanation	Provide explanation	Model dependent
Huang et al. (2023)	PB	DPT, DPC	MI prevention	Accurate CF for big datasets	Model dependent, accuracy loss
Yang et al. (2022)	PB	DPC	MI and ME prevention	Robust counterfactual	No plausibility

Table 1: Review of existing DP mechanisms for counterfactual generation. CF is used as an abbreviation for counterfactuals. MI and ME mean, respectively, membership inference and model extraction. PB stands for perturbation-based and RP stands for recourse path. Other abbreviations: DPS: DP synthetic data, DPT: DP training, DPC: DP counterfactuals.

The main drawback of these differentially private counterfactuals is that they focus on generating DP counterfactual without having plausibility as a main objective in mind, which is one pivotal element of our suggested pipeline. Furthermore, some of them are model-dependent. The main objective of these counterfactuals is protection against membership inference and model extraction attacks. In contrast to these previous works, we propose DP mechanisms for generating counterfactuals to protect the training data against re-identification attacks, which can be considered a form of linkage attacks. In addition, our DP approaches are model agnostic, and our results show that it can generate DP counterfactuals that have high plausibility. For these different settings explained here, comparing the utility and privacy aspects between these models and ours is not straightforward.

4 Differentially-Private Counterfactual Generation

In this section, we present our DP approaches to generate privacy-preserving and plausible counterfactuals. More precisely, our framework is flexible in the sense that we have developed mechanisms for different stages of the counterfactual generation pipeline, which enables to control the privacy-utility trade-off in a fine-grained manner.

4.1 Pre-processed DP-Counterfactuals

To apply DP at the earliest stage of the pipeline, we propose to generate differentially-private dataset from real data(`DP-Pre`). More precisely, we first use real data to train our models; Then, to generate counterfactuals, instead of using the real training data, we input differentially private datasets to the `NICE` mechanism to generate counterfactuals. In this DP-data generation, Laplace and randomized response mechanisms are applied to the training data, and then the obtained dataset is used to generate counterfactuals. This pre-processed differentially private data generation takes the idea of applying Laplace noise and randomized response from local differential privacy Dwork (2006), while it is applied in the centralized setting and provides instance-level DP guarantee. The idea of using local-DP techniques in centralized setting has previously used to prevent explanation-guided privacy attacks Nguyen et al. (2023). It should be taken into account that the model is trained on the original real data, and the DP dataset is used only during the counterfactual generation mechanism. More precisely, we have created noisy versions of numerical features using the Laplace mechanism while we applied RR to categorical features. The privacy budget assigned to each call of each Lap and RR mechanism is $\frac{\epsilon}{d}$ in which ϵ is the overall privacy budget, and d is the dimensionality (*i.e.*, number of features). Considering the effect of the sequential composition Dwork et al. (2006), this division maintains the privacy budget of ϵ for the solution.

4.2 In-processed DP-Counterfactuals

Another mechanism we have developed to create differentially-private counterfactuals is called `Inline_DP`. The main idea behind this solution is that by increasing the number of participating instances in the counterfactual generation mechanism, new feature values are chosen from several instances instead of using only one instance, as in `NICE`. We select these neighbours following a scoring function, in which the score of each instance of the counterfactual class is computed based on Equation 1.

$$\text{score}_i = \max(\text{distance}) - \text{distance}_i \tag{1}$$

In a nutshell, this formula gives a higher score to instances closer to the original instance. Like `NICE` and our `DP-Pre` algorithm, we use the Heterogeneous Euclidean Overlap Method (HEOM) Wilson & Martinez (1997) with L_1 as our distance. Equation 2 defines this distance for two instances, a and b, each having d features.

$$\text{HEOM}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{d} D_i(a_i, b_i), \tag{2}$$

in which $D_i(a_i, b_i)$ is defined for each attribute i as:

$$D_i(a_i, b_i) = \begin{cases} \frac{|a_i - b_i|}{\text{range}_i}, & \text{if attribute } i \text{ is numerical} \\ 1, & \text{if attribute } i \text{ is categorical and } a_i \neq b_i \\ 0, & \text{if attribute } i \text{ is categorical and } a_i = b_i \end{cases} \tag{3}$$

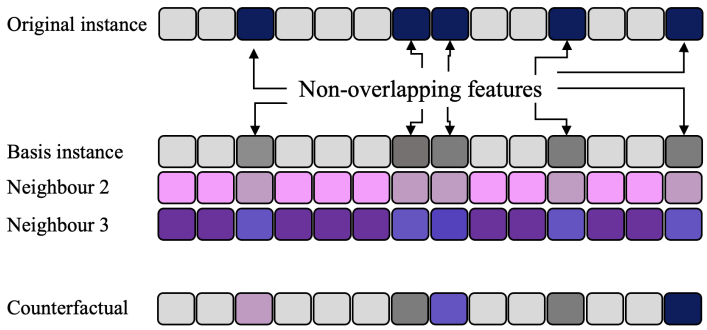


Figure 2: Illustration of `Inline_DP`. In this example, $k = 3$ nearest instances to the original instance are selected from the counterfactual class. Then, one instance is selected as the basis instance and non-overlapping features are identified. Using the Laplacian mechanism and randomized response, non-overlapping values of all instances are privatized. At each iteration, the most rewarding feature value from all possible values among k neighbours is selected and replaced in the original instance until the model changes its prediction.

After choosing the k instances with the highest scores from the training dataset, one of these instances is randomly selected as the basis instance, and non-overlapping features between this instance and the original instance requesting a counterfactual are identified. Then, for the features in the position of these non-overlapping values in all of k neighbours, we apply the Laplace mechanism (for numerical features) and RR (for categorical features). The fraction of the privacy budget assigned for each feature is $\frac{\epsilon}{k \times \#non\_overlapping}$ to maintain the global privacy budget of ϵ. The sensitivity used for the Laplacian noise is 1 for each feature value. After the privatization of feature values, a pool of these private values is created, and at each iteration, the most rewarding value is selected to replace the original instance. After choosing one value, all values for the position of the selected one are removed from the pool. This process continues until a counterfactual is generated or the pool is empty. This approach is illustrated in Figure 2 and Algorithm 1.

In this algorithm, a basis instance is randomly selected to maintain the privacy budget. Another possibility is to use a fraction of the privacy budget to apply a differentially-private mechanism such as the exponential mechanism McSherry & Talwar (2007) to choose the basis instance.

4.3 Post-processed DP-Counterfactuals

With respect to post-processing, we have designed three techniques to achieve DP on counterfactuals after their creation (`DP-Post`). We applied three different differentially private techniques on generated counterfactuals which will described here:

Laplace mechanism and RR. To make the counterfactuals generated by `NICE` differentially private, we applied the Laplace mechanism to numerical features and RR to categorical features. This approach operates *a*

Algorithm 1 In-processed DP-counterfactuals.

Require: x(query), ds(training_set), k(number of neighbours), ϵ (privacy budget)
Ensure: priv_cf (private counterfactual)
priv_cf ← x
Neighbours = knn(x,ds,k)
basis = sample(Neighbours)
ΔF = diff(X,basis) % find non-overlapping features
diffVals ← vals(ΔF,neighbours) % values for non-overlapping features in all neighbours
privVals ← privatize(diffVals) % lines 8-14 Algorithm 2
while priv_cf is not counterfactual **do**
    idx,val = pickBest(privVals) %choose best feature value from list;
    priv_cf$_{idx}$ ← val;
    update diffVals; %remove all values for feature index
end while
return priv_cf

posteriori, meaning that it is applied after generating counterfactuals using `NICE`. Algorithm 2 illustrates the procedure for incorporating RR and the Laplace mechanism into `NICE` counterfactuals to achieve differential privacy. In this implementation, noise is added only to the feature values altered by `NICE`, with an upper bound on the number of modified features set to d. To ensure that the overall privacy budget remains within ϵ, the privacy budget is divided equally among all modified features. According to the sequential composition theorem in DP Dwork et al. (2014), the cumulative privacy loss across multiple independent applications of a DP mechanism is additive. Therefore, for d modified features, the privacy budget allocated to each feature is $\frac{\epsilon}{d}$, ensuring the total budget does not exceed ϵ.

Algorithm 2 Post-Processed DP-counterfactual generation

Require: x(query), CF (nonprivat CF), ϵ, $featVals$(feature values), $catfeats$(categorical features list)
Ensure: priv_cf (private counterfactual)
priv_cf ← CF
diffVals = diff(x,$priv\_c$)
for $i \leftarrow 1$ to $lengh(X)$ **do**
    if $diffVals\_i = 1$ **then**
        if $i \in catfeat$ **then**
            $p = \frac{1}{1+\exp(\epsilon)}$
            with probability(p):
            $priv\_cf_i \leftarrow$ sample($featVals$)
        else
            priv_cf$_i \leftarrow CF_i + lapNoise(scale = s/\epsilon)$ % s: sensitivity
        end if
    end if
end for
return priv_cf;

Report Noisy Max. In the first implementation of post-processing DP techniques, the data distribution was not considered when choosing new feature values for generated counterfactuals. More precisely, when applying Laplacian noise and RR, the only information taken into account about data distribution is the range of values of each feature. Therefore, applying these DP mechanisms to generated counterfactuals results in a loss in plausibility and correctness. This means that changes in the feature values are significant enough to move counterfactuals from the counterfactual to the undesired class. In addition, even when they remain counterfactual, their distance might be quite far from query instances, thus making them less desirable from the user's point of view. To address this, we rely on another DP mechanism, report noisy max, to engage the properties of the training dataset in updating generated counterfactuals. Similarly to other DP mechanisms, the privacy budget is divided by the number of changed features in the counterfactual class, and each of those features is updated using the noisy max value of the feature in the dataset. More precisely, the frequency of all existing values in the training data belonging to the counterfactual class is computed, before the addition of Gaussian noise to this frequency. Afterwards, among these noisy values, the highest one is chosen, and its associated feature value is replaced in the counterfactual.

Feature-based exponential mechanism. The last approach that we have tried is the application of the exponential mechanism in the final stage of the pipeline. This approach also works at the feature level after receiving a non-private counterfactual generated by `NICE`. More precisely, similarly to Report Noisy Max, the frequency of each value repeated in the instances in the counterfactual class is measured, which plays

the role of the utility function for the exponential mechanism. Each value receives a score based on these frequencies and an exponential mechanism is applied to choose a value based on them. The more frequent each value is in the counterfactual class of the training dataset, the higher score it receives as utility. Like other post-processing mechanisms, the privacy budget is divided to the number of features to keep the overall budget lower than ϵ following sequential mechanism principles. By doing so, the statistics of the data distribution in the counterfactual class will help to reduce the utility loss of the generated counterfactuals.

having all techniques explained, we now proceed to the empirical results of each algorithm.

5 Experimental Evaluation

In this section, we evaluate the approaches that we have developed for generating differentially-private counterfactuals. More precisely, we first describe the experimental setting before reporting on the results obtained.

5.1 Experimental Setting

The code and datasets to reproduce our results are available as supplementary materials.

Datasets and models. We have evaluated our DP counterfactual generation methods on six datasets commonly used in literature: `ACS income` Ding et al. (2021) `adult` Asuncion & Newman (2007), `compas` Angwin et al. (2016), `heloc` OpenML (2018), `ACS public coverage` Ruggles et al. (2021) and `default credit` Yeh & Lien (2009). These datasets are all in the category of tabular data, which is the typical use case considered for counterfactual generation. In addition, all these datasets include sensitive information about individuals' financial, social or health status, as well as quasi-identifiers that adversaries can use to perform linking attacks that we detail hereafter.

Given the model-agnostic nature of our DP counterfactual methods, we evaluated their performance across a diverse range of machine learning models: Random Forest (RF), Neural Networks (NN) and Light Gradient Boosting Model (LightGBM). For all these models, we have relied on their implementation on Scikit-Learn Pedregosa et al. (2011). Our experiments were conducted on a cluster of 100 CPU nodes, each with 6GB of memory. Finally, the hyperparameter tuning was conducted via grid search, leveraging Scikit-Learn's model selection utilities to identify the optimal configurations for each combination of model and dataset. The final hyperparameter settings are detailed in Appendix C.

Data preprocessing and training settings. We have followed a unified approach for all datasets and model types to train our models. Each of the datasets was split into a training set that contains 70% of the data, test set (20%) and counterfactual set (10%) to ensure that the instances queried for counterfactuals have not been seen by the model before. Before training the model, we analyze the datasets to extract the required information for our counterfactual mechanisms. In particular, we distinguish between categorical and numerical features and establish the range of their values. The report results are averaged over five executions of each algorithm, with models trained using five different random seeds, evaluated on 1000 instances under the settings described below. To account for the randomness introduced by differential privacy, each differentially-private mechanism was executed 20 times per instance.

The `NICE` library was used as the basis to generate differentially-private counterfactuals. For each model and dataset combination, we have chosen $\epsilon \in \{0.01, 0.1, 1, 5, 10\}$. In addition, to evaluate how the number of neighbours affects the quality of `Inline_DP` counterfactuals, we have run all the experiments of the exponential mechanism using four different values of the number of neighbours: $k \in \{3, 5, 10, 20\}$. The same setting was also used for all datasets, models and values of ϵ and k.

Evaluation metrics. To compare the performance of our methods, we rely on the following metrics.

- The *re-identification rate* quantifies how many exact matches each generated counterfactual has in the training dataset. This metric is classically used in the literature to measure the effectiveness of linkage attacks Herzog et al. (2007); Vidanage et al. (2022). The rationale here is that the generated counterfactual is considered robust against re-identification if it has no exact match in the training

dataset or if there is a lot of matches, thus making it hard for an adversary to identify a specific profile. In contrast, the worst-case situation occurs when only one match exists in the training data, which means that the generated counterfactual discloses sensitive data. We use M_0 (respectively M_1) to denote the proportion of counterfactual explanations having no match (respectively exactly one match) in the training dataset.

$$\mathrm{M}_0 = \frac{1}{|\,\mathrm{X_{CF}}\,|} \sum_{x \in \mathrm{X_{CF}}} \mathbb{I}(x \notin \mathrm{X_{train}}). \tag{4}$$

$$\mathrm{M}_1 = \frac{1}{|\,\mathrm{X_{CF}}\,|} \sum_{x \in \mathrm{X_{CF}}} \mathbb{I}((\sum_{x' \in \mathrm{X_{train}}} \mathbb{I}\{x' = x\}) = 1). \tag{5}$$

- The *recourse cost* Wachter et al. (2017); Karimi et al. (2020; 2022) quantifies the difficulty for a user to change the decision made by the model by computing the distance between the original instance and its counterfactual. More precisely, following NICE, we choose HEOM Wilson & Martinez (1997) using the L_1 norm as our recourse cost. We use $\mathrm{Dist_{rec}}$ to denote the average recourse cost computed over all generated counterfactuals.
- The *plausibility* assesses the realistic aspect of counterfactuals according to training data distribution. While there is a huge literature on possible measures of plausibility, in this work we used the average distance to k-nearest neighbours in the training set as suggested in previous works Dandl et al. (2020; 2024) to assess the generated counterfactuals. One advantage of using this measure is that the effect of potential outliers on the measurements is minimized due to the use of training instances.
- Finally, we define the *validity*, hereafter referred to as `Correctness`, as the proportion of counterfactual explanations that are correct (*i.e.*, whose predicted outcomes differ from the original instance).

5.2 Experimental Results

We conducted experiments to validate our differentially-private counterfactual generation mechanisms across all datasets and models described in the previous section. Notably, similar trends were observed across the other models and datasets, as detailed in supplementary materials.This regular trend has been seen for all datasets, except for the ones with a higher number of features (`default credit`, `heloc`, and `ACS public coverage`) for some settings, which is explained in the supplementary materials. For these datasets, the privacy metrics maintain high values because of the increased number of features, which helps decrease reidentification rates.

Pre-processed DP-counterfactuals. As described in Section 4.1, our initial approach to incorporating privacy into our counterfactuals involved generating a differentially-private dataset and subsequently using this dataset to produce counterfactuals. Table 2 illustrates the privacy-utility trade-offs of using DP-perturbed data via the Laplace mechanism and RR to produce counterfactuals over various privacy budgets. This mechanism achieves a high degree of privacy but significantly decreases the utility. In contrast, increasing the privacy budget lessens the impact on utility. While almost all generated counterfactuals (more than 98.94%) have no match in the training dataset, this high level of privacy comes with a utility loss of respectively 95% and 62% in $\mathrm{Dist_{rec}}$ and $\mathrm{Dist_{Plaus}}$. More precisely, the impact of differential-privacy mechanisms on the training data distributions is detailed in Appendix A. Applying differential privacy techniques on the dataset used to generate counterfatuacle, shifts the distribution, and consequently, shifts generated counterfactuals further from training dataset, and resutls in lower utility.

In-processing DP counterfactuals. To identify the neighbours of the original instance, as described in Section 4.2, we rely on the HEOM distance. The effect of the privacy budget and number of neighbours is evaluated and shown in Table 3. It can be seen that this method significantly improves the privacy of `NICE` in terms of M_0 and M_1. In particular, the results show that M_1 decreases to less than 0.1 for all ϵ values. Also, for all combinations of ϵ and k, M_0 is higher than 99%. More precisely, M_0 and M_1 have been improved by more than 17% and 99% respectively. These results mean that more than 99% of worst cases

Method	ϵ	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.42** ± 0.33	**0.36** ± 0.29	**100.00**	85.38	14.26
`DP-Pre`	0.01	1.05 ± 0.74	0.72 ± 0.5	**100.00**	99.96	0.02
	0.10	1.04 ± 0.74	0.70 ± 0.49	**100.00**	**99.98**	**0.00**
	1.00	0.99 ± 0.73	0.68 ± 0.48	**100.00**	99.94	0.04
	5.00	0.89 ± 0.68	0.62 ± 0.45	**100.00**	**99.98**	**0.00**
	10.00	0.83 ± 0.65	0.59 ± 0.43	**100.00**	99.96	0.02

Table 2: Privacy-utility trade-off for Pre-processed DP-counterfactuals

(M_1) have been mitigated, and more than 99.89% of generated counterfactuals using this mechanism do not re-identify any instance of the training dataset. However, this improvement in privacy comes at a cost in terms of utility. Indeed, while increasing the privacy budget diminishes the privacy, as expected, it reduces the utility cost introduced by this algorithm. For $\epsilon = 10$ and $k = 20$, the lowest utility is achieved, which equals respectively 107%, 61% and 1.66% for Dist_{rec}, $\text{Dist}_{\text{Plaus}}$ and `Correctness`. This high value of Dist_{rec} and $\text{Dist}_{\text{Plaus}}$ is due to the high degree of randomness introduced to the feature values used to generate counterfactuals. In contrast, the highest privacy improvement for different privacy budgets is reached when 5 or 10 neighbours are used to generate counterfactuals, while the utility metrics values are almost the same as those of 20 neighbours. Another trend seen in the results is that increasing the size of neighbour sets used to generate counterfactuals improves utility while still achieving high privacy compared to smaller ones.

Method	ϵ	K	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	1	**0.42** ± 0.33	**0.36** ± 0.29	**100.00**	85.38	14.26
`Inline_DP`	0.01	3	1.09 ± 0.74	0.71 ± 0.49	83.67	99.94	0.04
		5	1.08 ± 0.73	0.69 ± 0.48	98.13	**99.95**	**0.03**
		10	1.03 ± 0.68	0.67 ± 0.46	99.18	99.93	0.05
		20	1.03 ± 0.68	0.66 ± 0.46	99.34	99.93	0.05
	0.10	3	1.09 ± 0.74	0.71 ± 0.48	83.63	99.95	0.04
		5	1.08 ± 0.73	0.69 ± 0.48	98.05	99.94	0.04
		10	1.03 ± 0.68	0.67 ± 0.46	99.25	99.93	0.05
		20	1.02 ± 0.67	0.66 ± 0.46	99.31	99.93	0.05
	1.00	3	1.08 ± 0.74	0.70 ± 0.48	84.21	99.94	0.04
		5	1.06 ± 0.73	0.68 ± 0.47	98.31	99.94	0.04
		10	1.01 ± 0.67	0.66 ± 0.45	99.33	99.93	0.05
		20	1.01 ± 0.67	0.65 ± 0.45	99.42	99.92	0.06
	5.00	3	1.03 ± 0.72	0.67 ± 0.47	86.24	99.94	0.04
		5	1.00 ± 0.7	0.65 ± 0.46	99.14	99.92	0.06
		10	0.95 ± 0.65	0.62 ± 0.44	99.67	99.93	0.05
		20	0.94 ± 0.64	0.62 ± 0.44	99.69	99.93	0.05
	10.00	3	0.97 ± 0.69	0.63 ± 0.44	87.06	99.90	0.07
		5	0.94 ± 0.67	0.61 ± 0.44	99.50	99.92	0.06
		10	0.89 ± 0.62	0.59 ± 0.42	99.82	99.89	0.09
		20	0.88 ± 0.61	0.58 ± 0.42	99.83	99.90	0.07

Table 3: Privacy-utility trade-off for `Inline_DP`

While in almost all settings the `Correctness` is higher than 98%, there are situations in which `DP-In` fails to generate counterfactuals, especially when the number of instances in the neighbours list is equal to 3, where correctness falls to 83.67% for the smallest value of ϵ. Since we are adding noise to the feature values of the selected neighbours in the pool, regardless of how this noise will affect their positions to the decision boundary, these values may not lead to good alternatives for the final counterfactual, which means that probability of this situation increases when the algorithm has a lower number of feature value options to choose from. For this reason, in some cases `Inline_DP` fails to generate valid counterfactuals.

Post-processed DP-counterfactuals. Table 4 shows the results of implementing the Laplace mechanism and randomized response on counterfactuals generated by `NICE`. The results obtained demonstrate that while post-processing DP highly improves the privacy of counterfactuals (by more than 17% in M_1, which achieves higher than 99.78% M_0, and more than 95% improvement in M_0 achieving less than 0.18% M_1), it does not perform well in terms of utility. Indeed, both proximity and plausibility experience a significant decrease. For instance, the Dist_{rec} is increased by more than 38% when using Laplacian noise and RR and the $\text{Dist}_{\text{Plaus}}$ by more than 32% for all epsilon budgets. Furthermore, this differentially-private post-processing mechanism

fails to achieve counterfactuals for more than 63% of experiments. This occurs due to the random nature of DP, which is introduced to the final results.

Method	ϵ	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.42** ± 0.33	**0.36** ± 0.29	**100.00**	85.38	14.26
Laplace_Noise_DP	0.01	1.05 ± 0.71	0.74 ± 0.47	30.91	**99.94**	**0.03**
	0.10	1.04 ± 0.71	0.74 ± 0.47	31.03	99.92	0.04
	1.00	0.95 ± 0.7	0.68 ± 0.45	31.95	99.89	0.07
	5.00	0.72 ± 0.61	0.55 ± 0.4	34.05	99.83	0.12
	10.00	0.59 ± 0.52	0.48 ± 0.36	36.36	99.78	0.18

Table 4: Privacy-utility trade-off for *Laplace_Noise_DP*

The results in Table 5 show how report noisy max affects the privacy and utility of generated counterfactuals. It shows that compared to Laplacian noise and randomized response, report noisy max achieves less private results with lower utility loss in terms of Dist_{rec}, $\text{Dist}_{\text{Plaus}}$ and `Correctness`. This means that for datasets with lower privacy considerations, report noisy max is a better option when the objective is to generate more practical model agnostic private counterfactuals. While generated counterfactuals with the lowest privacy budget, compared to `NICE` bring an increase of 162% in Dist_{rec} and 42% on `Correctness`, this cost is less than 14% in terms of plausibility, which decreases to less than 4% with the highest privacy budget, yet providing more than 60% improvement in terms of privacy measures.

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.42** ± 0.33	0.36 ± 0.29	**100.00**	85.38	14.26
`Noisy Max`	0.01	0.61 ± 0.5	0.44 ± 0.32	43.46	**99.57**	**0.33**
	0.10	0.61 ± 0.51	0.36 ± 0.3	61.73	99.01	0.82
	1.00	0.59 ± 0.46	0.32 ± 0.27	67.84	98.91	0.93
	5.00	0.66 ± 0.47	0.32 ± 0.28	85.51	98.01	1.75
	10.00	0.70 ± 0.49	**0.31** ± 0.28	89.98	97.58	2.12

Table 5: Privacy-utility trade-off for `Noisy Max`

The feature-based exponential mechanism provides a higher level of privacy, combined with a higher utility cost in terms of counterfactuals' correctness. For instance, the feature-based exponential mechanism leads to more than 50% loss in `Correctness` and 85% loss in Dist_{rec} compared to `NICE` for the highest ϵ which is 10. The noisy max generated counterfactuals are more plausible (as low as 4% loss in $\text{Dist}_{\text{Plaus}}$ and 18% in terms of `Correctness`), it provides the least private counterfactuals among all post-processed techniques. Thus with respect to privacy, the feature-based exponential mechanism is, on average, more successful than other `DP-Post` mechanisms.

Method	ϵ	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	0.42 ± 0.33	**0.36** ± 0.29	**100.00**	85.38	14.26
`Feature based exponential mechanism`	0.01	0.64 ± 0.46	0.50 ± 0.34	18.02	**99.93**	**0.05**
	0.10	0.64 ± 0.46	0.50 ± 0.34	17.96	99.90	0.08
	1.00	0.62 ± 0.46	0.49 ± 0.34	19.30	99.86	0.11
	5.00	0.48 ± 0.46	0.43 ± 0.33	28.38	99.81	0.15
	10.00	**0.42** ± 0.44	0.40 ± 0.31	37.73	99.82	0.14

Table 6: Privacy-utility trade-off for `Feature based exponential mechanism`

These trends are observed roughly for all datasets except for Adult, which shows different behavior in terms of `Correctness`, where increasing ϵ results in lower correctness, which is the opposite of other datasets and is not expected within the context of differential privacy. This may be related to the properties of the adult dataset. This affects the `NICE` generated counterfactuals to the training dataset and their position relative to the decision boundary. The direction of Laplacian noise and Randomized response is not limited in this work, and analyzing it is out of the scope of this work. It could be a direction for future work to analyze whether it is possible to conduct this noise to move counterfactuals toward the desired class while maintaining privacy, and as a result, keep the correctness of the generated counterfactuals.

Summary of results. After analyzing various techniques for generating privacy-preserving counterfactuals, our study found that the pre-processing and in-processing mechanisms provide the best trade-off among all. While the Dist_{rec} and $\text{Dist}_{\text{Plaus}}$ are a bit high, these techniques are almost always capable of generating valid counterfactuals with the highest values for privacy measures. According to differential privacy standards and best practices, including NIST SP 800-26 Near et al. (2023), it is often impossible to meet all privacy requirements and keep the best utility at the same time. Hence, model providers and users have to decide based on their privacy and utility requirements which method can best address their needs. Here is a overview of the performance of implemented techniques regarding each privacy and utility measures:

- Proximity (Dist_{rec}). While `NICE` leads to the best proximity overall, the lowest proximity cost of suggested differentially-private counterfactuals belongs to `Feature based exponential mechanism` techniques, which is one of our `DP-Post` mechanisms(Section 4.3). Like other postprocessing techniques, this technique exhibits lower `Correctness` compared to `DP-In` and `DP-Pre`.
- Plausibility ($\text{Dist}_{\text{Plaus}}$). According to our experiments, the most plausible differentially-private counterfactuals are generated using the `Noisy Max` technique, followed by `Feature based exponential mechanism`. `Noisy Max` also works better than the two other postprocessing techniques in terms of `Correctness`, but it subperforms other mechanisms like `DP-In` and `DP-Pre` regarding `Correctness` and privacy measures. This can be explained by the nature of this technique, which gives a higher chance of achieving the more common feature values in the dataset, increasing the probability of matching with existing instances in the training dataset. It should be mentioned that even the lowest privacy rates achieved by this technique using the highest privacy budget are still high (for the `ACS income` dataset, it is higher than 97% for $\epsilon = 10$), which still makes them a reliable choice.
- Correctness (`Correctness`). While `NICE` is always successful in generating counterfactuals for all instances, this is not the case for all differentially-private counterfactuals. More precisely, `DP-Pre` technique is always capable of generating valid counterfactuals, while for other methods this is not the case. In particular, `DP-In` succeeds to generate valid counterfactuals for more than 98% instances for $K \geq 5$ while for smaller Ks, it generates valid counterfactuals for more than 83% of instances. The only category of mechanisms that falls short in terms of `Correctness` are `DP-Post` techniques. Thus, if `Correctness` is the most important metric for users (which is often the case as this is the primary objective of counterfactual generation), they should decide between `DP-Pre` and `DP-In` techniques.
- Reidentification rate (M_1 and M_0). `DP-In` is the algorithm that provides the highest privacy for all settings among our differentially-private counterfactuals. While this comes with a very small cost of `Correctness`, an acceptable cost of plausibility and a high cost in terms of proximity, it could be considered as the best achievable trade-off between utility and privacy.

6 Conclusion

In this work, we have proposed a novel approach for enhancing the privacy of counterfactuals by designing differentially-private solutions for generating instance-based counterfactuals that can be used at different stages of the counterfactual generation pipeline. Our experiments demonstrate that for various models trained on numerous real-world datasets, our `DP-In` mechanism can improve privacy with the lowest cost in terms of utility. More precisely, the higher the number of neighbours used, the more private the counterfactuals generated, but also the better the utility. Additionally, our experiments have shown that `DP-Post` (*i.e.*, *a posteriori* DP through RR and Laplace mechanisms) did not provide a good utility-privacy trade-off for privacy-preserving counterfactual generation. In contrast, the `DP-Pre` mechanism provides acceptable trade-offs and could be an interesting approach to generate private counterfactuals.

Moreover, our `DP-Post` mechanisms achieve a worse trade-off between utility and privacy compared to the `DP-In` mechanism, making them mostly interesting in the situation in which the user needs a model-agnostic DP solution for which he is willing to pay a high utility cost. In the other words, while `DP-Pre` and `DP-Post`

mechanisms cannot achieve results as good as `DP-In` mechanism, their model agnostic nature makes them good candidates for scenarios in which the CF generation mechanism should not be altered but rather its output is privatized. In contrast, `DP-In` has been designed specifically to generate differentially-private counterfactuals based on `NICE`, which leads to a high performance in terms of utility and privacy but limits its applicability. This opens the avenue for further research to provide more cost-effective model-agnostic counterfactuals in terms of pre-processed and post-processed DP mechanisms using other DP mechanisms resulting in less utility loss. Another direction for future work is to assess the practical effectiveness of our techniques in defending against other privacy attacks, such as membership inference and model extraction attacks. Since Differenctial privacy has been widely used in defending against membership inference attacks, it is possible to explore if our suggested algorithm can

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Ulrich Aïvodji, Alexandre Bolot, and Sébastien Gambs. Model extraction from counterfactual explanations. *arXiv preprint arXiv:2009.01884*, 2020.

Micah Altman, Aloni Cohen, Kobbi Nissim, and Alexandra Wood. What a hybrid legal-technical analysis teaches us about privacy regulation: The case of singling out. *BUJ Sci. & Tech. L.*, 27:1, 2021.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: Risk assessments in criminal sentencing. `https://github.com/propublica/compas-analysis`, 2016. Accessed 2025-05-15.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Dieter Brughmans, Pieter Leyman, and David Martens. Nice: an algorithm for nearest instance counterfactual explanations. *Data Mining and Knowledge Discovery*, pp. 1–39, 2023.

Aloni Cohen. Attacks on deidentification's defenses. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1469–1486, 2022.

Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *International conference on parallel problem solving from nature*, pp. 448–469. Springer, 2020.

Susanne Dandl, Kristin Blesch, Timo Freiesleben, Gunnar König, Jan Kapar, Bernd Bischl, and Marvin N Wright. Countarfactuals–generating plausible model-agnostic counterfactual explanations with adversarial random forests. In *World Conference on Explainable Artificial Intelligence*, pp. 85–107. Springer, 2024.

Randall Davis, Andrew W Lo, Sudhanshu Mishra, Arash Nourian, Manish Singh, Nicholas Wu, and Ruixun Zhang. Explainable machine learning models of consumer credit risk. *Available at SSRN*, 4006840, 2022.

Fangchen Ding, Hugo Larochelle, and David Madras. Retiring adult: New datasets for fair machine learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://github.com/zykls/folktables`.

Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

Halbert L Dunn. Record linkage. *American Journal of Public Health and the Nations Health*, 36(12): 1412–1416, 1946.

Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pp. 1–12. Springer, 2006.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Carlos Fernández-Loría, Foster Provost, and Xintian Han. Explaining data-driven decisions made by ai systems: the counterfactual approach. *arXiv preprint arXiv:2001.07417*, 2020.

Sofie Goethals, Kenneth Sörensen, and David Martens. The privacy issue of counterfactual explanations: Explanation linkage attacks. *ACM Trans. Intell. Syst. Technol.*, 14(5), aug 2023. ISSN 2157-6904. doi: 10.1145/3608482.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

Jianming He, Wesley W Chu, and Zhenyu Liu. Inferring privacy information from social networks. In *International Conference on Intelligence and Security Informatics*, pp. 154–165. Springer, 2006.

Thomas N Herzog, Fritz J Scheuren, and William E Winkler. *Data quality and record linkage techniques*, volume 1. Springer, New York, NY, 2007 edition, may 2007.

Catherine Huang, Chelse Swoopes, Christina Xiao, Jiaqi Ma, and Himabindu Lakkaraju. Accurate, explainable, and private models: Providing recourse while minimizing training data leakage. *arXiv preprint arXiv:2308.04341*, 2023.

Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pp. 2436–2444. PMLR, 2016.

Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *Advances in neural information processing systems*, 33:265–277, 2020.

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55(5):1–29, 2022.

Mark T Keane and Barry Smyth. Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai). In *Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020, Proceedings 28*, pp. 163–178. Springer, 2020.

Aditya Kuppa and Nhien-An Le-Khac. Adversarial xai methods in cybersecurity. *IEEE transactions on information forensics and security*, 16:4924–4938, 2021.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based inverse classification for interpretability in machine learning. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations: 17th International Conference, IPMU 2018, Cádiz, Spain, June 11-15, 2018, Proceedings, Part I 17*, pp. 100–111. Springer, 2018.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: unjustified counterfactual explanations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pp. 2801–2807, 2019. ISBN 9780999241141.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *Acm transactions on knowledge discovery from data (tkdd)*, 1 (1):3–es, 2007.

David Martens and Foster Provost. Explaining data-driven document classifications. *MIS quarterly*, 38(1): 73–100, 2014.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103. IEEE, 2007.

Rami Mochaourab, Sugandh Sinha, Stanley Greenstein, and Panagiotis Papapetrou. Demonstrator on counterfactual explanations for differentially private support vector machines. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 662–666. Springer, 2022.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 607–617, 2020.

Joseph P Near, David Darais, Naomi Lefkovitz, Gary Howarth, et al. Guidelines for evaluating differential privacy guarantees. *National Institute of Standards and Technology, Tech. Rep*, pp. 800–226, 2023.

DJ Nelson. Privacy-preserving counterfactual explanations to help humans contest ai-based decisions. Master's thesis, University of Twente, 2022.

Truc Nguyen, Phung Lai, Hai Phan, and My T Thai. Xrand: Differentially private defense against explanation-guided attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11873–11881, 2023.

OpenML. FICO-HELOC-cleaned Dataset. `https://openml.org/d/45554`, 2018. Accessed 2025-05-15.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Sikha Pentyala, Shubham Sharma, Sanjay Kariyappa, Freddy Lecue, and Daniele Magazzeni. Privacy-preserving algorithmic recourse. *arXiv preprint arXiv:2311.14137*, 2023.

Steven Ruggles, Sarah Flood, Ronald Goeken, Josiah Grover, Erin Meyer, Jose Pacas, and Matthew Sobek. Ipums usa: Version 11.0 [dataset]. `https://doi.org/10.18128/D010.V11.0`, 2021. Accessed 2025-05-15.

Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

Reza Shokri, Martin Strobel, and Yair Zick. Exploiting transparency measures for membership inference: a cautionary tale. In *The AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI). AAAI*, volume 13, 2020.

Reza Shokri, Martin Strobel, and Yair Zick. On the privacy risks of model explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 231–241, 2021.

Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 650–665. Springer, 2021.

Kush R Vashney. Trustworthy machine learning. In *Independently published*, 2022.

Anushka Vidanage, Thilina Ranbaduge, Peter Christen, and Rainer Schnell. A taxonomy of attacks on privacy-preserving record linkage. *Journal of Privacy and Confidentiality*, 12(1), 2022.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

Yongjie Wang, Hangwei Qian, and Chunyan Miao. Dualcf: Efficient model extraction attack from counterfactual explanations. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1318–1329, 2022.

Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American statistical association*, 60(309):63–69, 1965.

D Randall Wilson and Tony R Martinez. Improved heterogeneous distance functions. *Journal of artificial intelligence research*, 6:1–34, 1997.

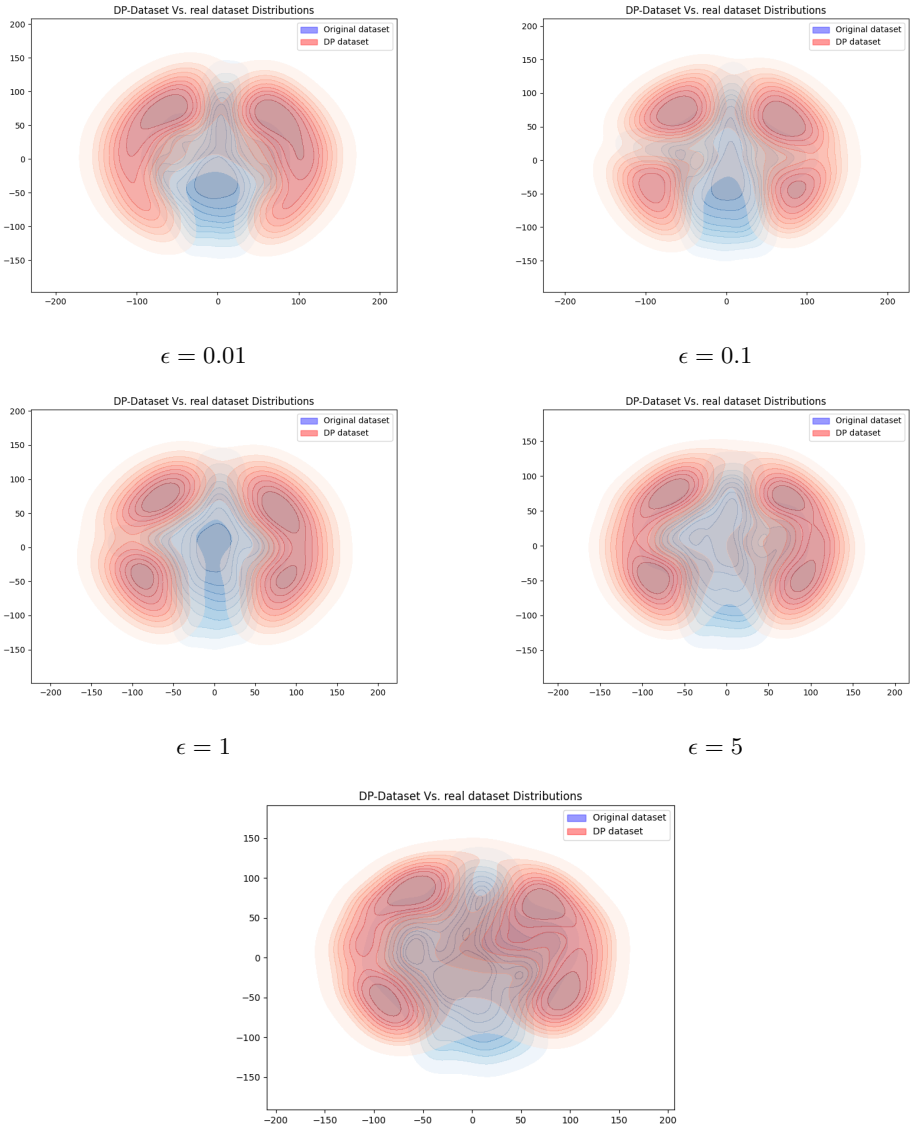
Fan Yang, Qizhang Feng, Kaixiong Zhou, Jiahao Chen, and Xia Hu. Differentially private counterfactuals via functional mechanism. *arXiv preprint arXiv:2208.02878*, 2022.

I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.

Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. *arXiv preprint arXiv:1208.0219*, 2012.

A DP-perturbed distributions

To see the impacts of generating perturbed data over ϵ on the data distribution, Figures 3,to **??** compare the data distribution between the DP (DP dataset obtained using Laplacian noise and RR) and the original data distribution for various datasets. These comparisons demonstrate that increasing the privacy budget leads to the synthetic data distribution become more similar to that of the original dataset. Distribution changes of other datasets under differential privacy follows the same trend and is provided in supplementary materials.



$\epsilon = 0.01$ $\epsilon = 0.1$

$\epsilon = 1$ $\epsilon = 5$

$\epsilon = 10$

Figure 3: Distributions comparison of original and synthetic datasets for different values of ϵ for `ACS income` dataset.

B Datasets and their sensitive attributes

Table 7 summarizes the datasets used to perform empirical study in this work. As shown in the table, all datasets are tabular with various numbers of attributes. Quasi identifiers, sensitive attributes and class attributes are also defined in the table.

Dataset	#attributes	Quasi identifiers	Sensitive Attribute	Class attribute
`ACS income`	11	Age, Sex, Race, Relationship, Marital status	Sex	Income
`adult`	11	Age, Sex, Race, Relationship, Marital status	Sex, Race	Income
`compas`	9	Sex, Age, Race	Sex , Race	Low_risk
`default credit`	24	Sex, Education, Mariage, Age	Sex	Default_payment
`heloc`	24	Credit history, utilization, inquiries	Credit history	RiskPerformance
`ACS public coverage`	20	Age, Sex, Race, Relationship, Marital status	Sex, Race, Nativity, DEAR, DEYE, DREM	Target

Table 7: Summary of the characteristics of datasets.

C Hyper-parameters settings for different models and datasets

We used GridSearchCV for hyperparameter tuning for all the models on all datasets to achieve highest accuracy of the models. Table 8 shows the optimum hyperparameters used in model training phase of our experiments.

D Prediction accuracy of models

Table 9 represents the average accuracy of the models trained on our datasets. From each dataset, we excluded features that are either irrelevant for the decision-making process or redundant because they convey information already provided by other features. Obtained accuracy for all datasets is in the normal range of possible accuracy for each dataset. The low accuracy for `compas`and `heloc`datasets is because of the hard task presented by them, and in the literature Dressel & Farid (2018); Davis et al. (2022), most researchers achieved accuracy in this range.

Dataset	NN	RF	LGBM
`ACS income`	activation: tanh alpha: 0.01 hidden_layer_sizes: [50] learning_rate: constant solver: adam	max_depth: 20 min_samples_split: 10 n_estimators: 100	colsample_bytree: 0.8 learning_rate: 0.1 max_depth: -1 min_child_samples: 20 n_estimators: 300 num_leaves: 50 reg_alpha: 1 subsample: 0.8
`adult`	activation: tanh alpha: 0.01 hidden_layer_sizes: [50] learning_rate: constant solver: adam	max_depth: 20 min_samples_split: 10 n_estimators: 100	colsample_bytree: 0.8 learning_rate: 0.1 max_depth: 10 min_child_samples: 10 n_estimators: 100 num_leaves: 31 reg_alpha: 1 subsample: 0.8
Compas	activation: relu alpha: 0.01 hidden_layer_sizes: [50] learning_rate: constant solver: sgd	max_depth: 10 min_samples_split: 2 n_estimators: 100	colsample_bytree: 1 learning_rate: 0.01 max_depth: 5 min_child_samples: 10 n_estimators: 100 num_leaves: 31 reg_alpha: 0 subsample: 0.8
Heloc	activation: tanh alpha: 0.01 hidden_layer_sizes: [50] learning_rate: constant solver: adam	max_depth: 20 min_samples_split: 10 n_estimators: 100	colsample_bytree: 0.8 learning_rate: 0.1 max_depth: 5 min_child_samples: 20 n_estimators: 100 num_leaves: 31 reg_alpha: 1 subsample: 0.8
`default credit`	activation: relu alpha: 0.001 hidden_layer_sizes: [50,50] learning_rate: constant solver: sgd	max_depth: 10 min_samples_split: 2 n_estimators: 100	colsample_bytree: 0.8 learning_rate: 0.01 max_depth: -1 min_child_samples: 10 n_estimators: 300 num_leaves: 100 reg_alpha: 0 subsample: 0.8
`ACS public coverage`	activation: tanh alpha: 0.01 hidden_layer_sizes: [50] learning_rate: constant solver: adam	max_depth: 20 min_samples_split: 10 n_estimators: 100	colsample_bytree: 0.8 learning_rate: 0.1 max_depth: -1 min_child_samples: 10 n_estimators: 300 num_leaves: 50 reg_alpha: 1 subsample: 0.8

Table 8: Parameter Settings for Models Across Datasets

	NN	***RF***	***LGBM***
Adult	0.833	0.835	0.840
ACS income	0.810	0.810	0.829
Compas	0.679	0.677	0.678
def_cre	0.810	0.816	0.812
Heloc	0.720	0.720	0.729
ACS_pub_cov	0.809	0.811	0.813

Table 9: Average model accuracy of different models trained on various datasets.

E Performances across all black-box Models and Datasets

In this section, results for `ACS income`on other models are presented. Since the results for Adult dataset are different from other datasets, results for Adult on RF model are presented here and the difference is explained here. The results for all models on other datasets are presented in the supplementary materials.

E.1 `ACS income`- NN

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.43** ± 0.34	**0.36** ± 0.29	**100.00**	81.72	17.66
`DP-Pre`	0.01	0.82 ± 0.62	0.81 ± 0.55	**100.00**	**99.98**	**0.00**
	0.10	0.81 ± 0.62	0.80 ± 0.56	**100.00**	**99.98**	**0.00**
	1.00	0.76 ± 0.57	0.75 ± 0.53	**100.00**	99.96	**0.00**
	5.00	0.69 ± 0.54	0.68 ± 0.5	**100.00**	99.96	0.04
	10.00	0.64 ± 0.51	0.63 ± 0.47	**100.00**	**99.98**	0.02

Table 10: Privacy-utility trade-off for Pre-processed DP-counterfactuals

Method	Epsilon	K	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	1	**0.43** ± 0.34	**0.36** ± 0.29	**100.00**	81.72	17.66
`Inline_DP`	0.01	3	0.91 ± 0.58	0.82 ± 0.52	93.20	99.97	0.01
		5	0.85 ± 0.55	0.79 ± 0.52	99.33	99.97	0.01
		10	0.82 ± 0.52	0.78 ± 0.51	99.70	99.97	0.01
		20	0.81 ± 0.52	0.77 ± 0.51	99.80	99.96	0.02
	0.10	3	0.91 ± 0.59	0.82 ± 0.52	93.16	**99.98**	0.01
		5	0.85 ± 0.55	0.79 ± 0.52	99.33	99.97	0.01
		10	0.82 ± 0.52	0.78 ± 0.51	99.70	99.97	**0.01**
		20	0.81 ± 0.52	0.77 ± 0.51	99.76	99.97	0.01
	1.00	3	0.90 ± 0.58	0.81 ± 0.52	93.35	99.97	0.02
		5	0.84 ± 0.55	0.78 ± 0.52	99.44	99.97	0.01
		10	0.81 ± 0.52	0.77 ± 0.51	99.72	99.96	0.01
		20	0.80 ± 0.52	0.76 ± 0.51	99.81	99.96	0.02
	5.00	3	0.85 ± 0.57	0.77 ± 0.51	94.03	99.93	0.05
		5	0.79 ± 0.53	0.75 ± 0.51	99.69	99.95	0.03
		10	0.76 ± 0.5	0.73 ± 0.51	99.87	99.95	0.03
		20	0.75 ± 0.5	0.73 ± 0.51	99.89	99.95	0.02
	10.00	3	0.81 ± 0.55	0.72 ± 0.5	94.30	99.92	0.06
		5	0.75 ± 0.51	0.71 ± 0.5	99.84	99.91	0.07
		10	0.71 ± 0.49	0.70 ± 0.5	99.90	99.93	0.05
		20	0.70 ± 0.48	0.70 ± 0.5	99.94	99.92	0.06

Table 11: Privacy-utility trade-off for `Inline_DP`

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.43** ± 0.34	**0.36** ± 0.29	**100.00**	81.72	17.66
Laplace_Noise_DP	0.01	1.30 ± 0.83	0.96 ± 0.58	35.70	99.97	**0.00**
	0.10	1.29 ± 0.83	0.96 ± 0.58	35.67	**99.97**	0.01
	1.00	1.16 ± 0.81	0.86 ± 0.57	36.63	99.93	0.03
	5.00	0.85 ± 0.71	0.65 ± 0.48	38.96	99.84	0.13
	10.00	0.67 ± 0.59	0.54 ± 0.41	41.97	99.80	0.18

Table 12: Privacy-utility trade-off for *Laplace_Noise_DP*

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.43** ± 0.34	0.36 ± 0.29	**100.00**	81.72	17.66
`Noisy Max`	0.01	0.74 ± 0.56	0.49 ± 0.37	47.18	**99.71**	**0.21**
	0.10	0.68 ± 0.55	0.36 ± 0.31	57.04	99.07	0.81
	1.00	0.62 ± 0.49	0.29 ± 0.26	59.90	98.85	1.00
	5.00	0.70 ± 0.49	0.29 ± 0.27	72.58	98.23	1.61
	10.00	0.76 ± 0.52	**0.29** ± 0.27	77.49	97.63	2.22

Table 13: Privacy-utility trade-off for `Noisy Max`

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.43** ± 0.34	**0.36** ± 0.29	**100.00**	81.72	17.66
`Feature based exponential mechanism`	0.01	0.85 ± 0.6	0.64 ± 0.43	26.52	99.91	0.08
	0.10	0.85 ± 0.6	0.64 ± 0.43	26.50	**99.92**	0.08
	1.00	0.82 ± 0.6	0.62 ± 0.43	27.69	99.91	**0.07**
	5.00	0.63 ± 0.58	0.52 ± 0.4	37.15	99.88	0.10
	10.00	0.53 ± 0.53	0.46 ± 0.37	46.64	99.87	0.11

Table 14: Privacy-utility trade-off for `Feature based exponential mechanism`

E.2 `ACS income`- LightGBM

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.34** ± 0.3	**0.37** ± 0.29	**100.00**	73.52	24.68
`DP-Pre`	0.01	0.93 ± 0.69	0.81 ± 0.56	**100.00**	**99.98**	**0.02**
	0.10	0.91 ± 0.69	0.80 ± 0.55	**100.00**	**99.98**	**0.02**
	1.00	0.85 ± 0.67	0.75 ± 0.54	**100.00**	99.96	0.04
	5.00	0.75 ± 0.62	0.67 ± 0.5	**100.00**	99.96	0.04
	10.00	0.65 ± 0.55	0.62 ± 0.47	**100.00**	99.96	0.04

Table 15: Privacy-utility trade-off for Pre-processed DP-counterfactuals

Method	Epsilon	K	$Dist_{rec}$ ↓	$Plaus_{prox}$ ↓	Correctness ↑	M_0 ↑	M_1 ↓
`NICE`	-	1	**0.34** ± 0.3	**0.37** ± 0.29	100.00	73.52	24.68
`Inline_DP`	0.01	3	0.99 ± 0.66	0.79 ± 0.52	88.53	99.96	0.03
		5	0.97 ± 0.68	0.77 ± 0.53	97.51	99.95	0.04
		10	0.95 ± 0.68	0.75 ± 0.53	98.76	99.95	0.04
		20	0.94 ± 0.68	0.75 ± 0.53	99.03	99.95	**0.03**
	0.10	3	0.99 ± 0.66	0.78 ± 0.52	88.65	**99.96**	0.03
		5	0.97 ± 0.68	0.77 ± 0.53	97.58	99.95	0.04
		10	0.94 ± 0.68	0.75 ± 0.53	98.79	99.95	0.03
		20	0.94 ± 0.68	0.75 ± 0.53	99.05	99.95	0.04
	1.00	3	0.98 ± 0.66	0.77 ± 0.52	**178.46**	99.95	0.04
		5	0.95 ± 0.68	0.76 ± 0.53	97.89	99.95	0.04
		10	0.93 ± 0.68	0.74 ± 0.52	98.94	99.95	0.03
		20	0.92 ± 0.67	0.74 ± 0.53	99.17	99.94	0.04
	5.00	3	0.92 ± 0.65	0.73 ± 0.51	90.59	99.93	0.06
		5	0.89 ± 0.66	0.72 ± 0.51	98.88	99.93	0.05
		10	0.86 ± 0.65	0.70 ± 0.51	99.48	99.91	0.07
		20	0.85 ± 0.65	0.70 ± 0.51	99.52	99.93	0.05
	10.00	3	0.86 ± 0.63	0.69 ± 0.49	91.00	99.92	0.07
		5	0.82 ± 0.63	0.68 ± 0.49	99.38	99.92	0.06
		10	0.79 ± 0.61	0.66 ± 0.49	99.73	99.92	0.07
		20	0.77 ± 0.61	0.66 ± 0.49	99.78	99.91	0.07

Table 16: Privacy-utility trade-off for `Inline_DP`

Method	Epsilon	$Dist_{rec}$ ↓	$Plaus_{prox}$ ↓	Correctness ↑	M_0 ↑	M_1 ↓
`NICE`	-	**0.34** ± 0.3	**0.37** ± 0.29	**100.00**	73.52	24.68
Laplace_Noise_DP	0.01	1.26 ± 0.81	0.85 ± 0.56	37.28	99.97	0.03
	0.10	1.26 ± 0.81	0.84 ± 0.56	37.10	**99.98**	**0.02**
	1.00	1.15 ± 0.8	0.78 ± 0.54	35.98	99.94	0.05
	5.00	0.83 ± 0.71	0.61 ± 0.46	33.57	99.89	0.09
	10.00	0.63 ± 0.6	0.52 ± 0.4	33.14	99.82	0.15

Table 17: Privacy-utility trade-off for *Laplace_Noise_DP*

Method	Epsilon	$Dist_{rec}$ ↓	$Plaus_{prox}$ ↓	Correctness ↑	M_0 ↑	M_1 ↓
`NICE`	-	**0.34** ± 0.3	0.37 ± 0.29	**100.00**	73.52	24.68
`Noisy Max`	0.01	0.81 ± 0.6	0.48 ± 0.35	41.66	**99.78**	**0.21**
	0.10	0.77 ± 0.59	0.38 ± 0.3	49.92	99.49	0.43
	1.00	0.73 ± 0.54	0.31 ± 0.26	55.72	99.14	0.73
	5.00	0.84 ± 0.52	0.28 ± 0.26	76.40	97.28	2.43
	10.00	0.89 ± 0.55	**0.27** ± 0.26	96.25	96.39	3.29

Table 18: Privacy-utility trade-off for `Noisy Max`

Method	Epsilon	$Dist_{rec}$ ↓	$Plaus_{prox}$ ↓	Correctness ↑	M_0 ↑	M_1 ↓
`NICE`	-	**0.34** ± 0.3	**0.37** ± 0.29	**100.00**	73.52	24.68
`Feature based exponential mechanism`	0.01	0.80 ± 0.58	0.56 ± 0.4	26.98	99.95	0.05
	0.10	0.80 ± 0.59	0.57 ± 0.4	27.28	**99.96**	**0.04**
	1.00	0.79 ± 0.58	0.56 ± 0.39	27.55	99.96	0.04
	5.00	0.68 ± 0.58	0.51 ± 0.38	31.81	99.92	0.08
	10.00	0.48 ± 0.55	0.44 ± 0.35	42.49	99.94	0.06

Table 19: Privacy-utility trade-off for `Feature based exponential mechanism`

E.3 adult- RF

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
NICE	-	**0.13** ± 0.11	**0.21** ± 0.2	**100.00**	47.54	38.14
DP-Pre	0.01	0.38 ± 0.21	0.37 ± 0.28	**100.00**	**97.98**	**1.42**
	0.10	0.38 ± 0.22	0.37 ± 0.28	**100.00**	97.60	1.72
	1.00	0.38 ± 0.25	0.37 ± 0.29	**100.00**	97.00	2.02
	5.00	0.38 ± 0.32	0.37 ± 0.31	**100.00**	96.02	2.70
	10.00	0.38 ± 0.35	0.37 ± 0.33	**100.00**	95.00	3.22

Table 20: Privacy-utility trade-off for Pre-processed DP-counterfactuals

Method	Epsilon	K	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
NICE	-	1	**0.13** ± 0.11	**0.21** ± 0.2	**100.00**	47.54	38.14
Inline_DP	0.01	3	0.46 ± 0.25	0.39 ± 0.27	87.36	98.47	0.98
		5	0.41 ± 0.19	0.36 ± 0.26	90.73	98.77	0.80
		10	0.39 ± 0.17	0.35 ± 0.25	93.16	**98.86**	**0.78**
		20	0.39 ± 0.16	0.35 ± 0.25	94.92	98.82	0.84
	0.10	3	0.46 ± 0.25	0.39 ± 0.27	87.70	98.43	0.96
		5	0.41 ± 0.19	0.36 ± 0.26	90.85	98.70	0.85
		10	0.39 ± 0.17	0.35 ± 0.25	93.06	98.80	0.79
		20	0.38 ± 0.17	0.35 ± 0.25	94.96	98.79	0.83
	1.00	3	0.45 ± 0.25	0.38 ± 0.27	89.80	97.65	1.40
		5	0.40 ± 0.19	0.36 ± 0.26	92.08	98.03	1.21
		10	0.38 ± 0.17	0.35 ± 0.25	93.98	98.11	1.19
		20	0.37 ± 0.17	0.34 ± 0.25	95.65	98.14	1.21
	5.00	3	0.40 ± 0.25	0.36 ± 0.27	94.22	95.54	2.65
		5	0.36 ± 0.19	0.33 ± 0.25	95.49	95.79	2.44
		10	0.34 ± 0.17	0.33 ± 0.25	96.33	95.81	2.55
		20	0.33 ± 0.17	0.32 ± 0.25	97.31	95.86	2.46
	10.00	3	0.36 ± 0.24	0.34 ± 0.26	95.21	93.71	3.60
		5	0.32 ± 0.19	0.32 ± 0.25	97.21	93.97	3.53
		10	0.30 ± 0.17	0.31 ± 0.25	97.58	94.00	3.53
		20	0.30 ± 0.17	0.31 ± 0.25	98.33	94.14	3.58

Table 21: Privacy-utility trade-off for Inline_DP

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
NICE	-	**0.13** ± 0.11	**0.21** ± 0.2	**100.00**	47.54	38.14
Laplace_Noise_DP	0.01	0.74 ± 0.43	0.49 ± 0.33	70.47	**96.56**	**1.91**
	0.10	0.73 ± 0.43	0.49 ± 0.33	70.34	96.48	1.98
	1.00	0.65 ± 0.43	0.45 ± 0.32	66.75	95.55	2.53
	5.00	0.42 ± 0.35	0.35 ± 0.28	59.62	91.83	4.78
	10.00	0.30 ± 0.27	0.29 ± 0.25	57.57	88.65	6.81

Table 22: Privacy-utility trade-off for *Laplace_Noise_DP*

As it can be seen in the results, the trend in correctness for DP-Post, when Laplace noise and randomized response are applied on NICE generated counterfactuals, is different from that of other datasets. Instead of increasing correctness with the privacy budget, higher privacy budgets for this dataset result in lower correctness, which may be related to the properties of this dataset. This affects on the NICE generated counterfactuals to the training dataset and their position relative to the decision boundary. The direction of Laplacien noise and Randomized response are not limited in this work and analyzing it is out of the scope of this work. It could be a direction for future work to analyze if it is possible to conduct this noise to move counterfactuals toward the desired class while maintaining the privacy, and as a result, keep the correctness of the generated counterfactuals.

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.13** ± 0.11	0.21 ± 0.2	**100.00**	47.54	38.14
`Noisy Max`	0.01	0.43 ± 0.33	0.31 ± 0.26	53.24	**87.98**	**6.44**
	0.10	0.32 ± 0.26	0.22 ± 0.23	57.71	73.55	13.10
	1.00	0.31 ± 0.22	**0.20** ± 0.21	74.10	73.84	14.04
	5.00	0.32 ± 0.23	0.21 ± 0.21	80.68	76.67	13.40
	10.00	0.33 ± 0.23	0.21 ± 0.21	80.59	76.77	13.30

Table 23: Privacy-utility trade-off for `Noisy Max`

Method	Epsilon	$Dist_{rec} \downarrow$	$Plaus_{prox} \downarrow$	Correctness ↑	$M_0 \uparrow$	$M_1 \downarrow$
`NICE`	-	**0.13** ± 0.11	**0.21** ± 0.2	**100.00**	47.54	38.14
`Feature based exponential mechanism`	0.01	0.52 ± 0.34	0.36 ± 0.27	50.94	**94.70**	3.09
	0.10	0.51 ± 0.34	0.36 ± 0.27	50.95	94.64	3.13
	1.00	0.49 ± 0.34	0.35 ± 0.27	50.37	94.60	**3.05**
	5.00	0.31 ± 0.31	0.30 ± 0.25	50.84	91.84	4.99
	10.00	0.20 ± 0.22	0.26 ± 0.24	48.78	90.89	5.76

Table 24: Privacy-utility trade-off for `Feature based exponential mechanism`

E.4 other results

All the other results for other datasets and models are provided in the supplementary materials. The source code of the project, which makes the results reproducible, is also provided.