2D-OOB: Attributing Data Contribution through Joint Valuation Framework

Anonymous Author(s)
Affiliation
Address
email

Abstract

Data valuation has emerged as a powerful framework to quantify the contribution of each datum to the training of a particular machine learning model. However, it is crucial to recognize that the quality of various *cells* within a single data point can vary greatly in practice. For example, even in the case of an abnormal data point, not all cells are necessarily noisy. The single scalar valuation assigned by existing methods blurs the distinction between noisy and clean cells of a data point, thereby compromising the interpretability of the valuation. In this paper, we propose `2D-OOB`, an out-of-bag estimation framework for jointly determining helpful (or detrimental) samples, as well as the particular cells that drive them. Our comprehensive experiments demonstrate that `2D-OOB` achieves state-of-the-art performance across multiple use cases, while being exponentially faster. `2D-OOB` excels in detecting and rectifying fine-grained outliers at the cell level, as well as localizing backdoor triggers in data poisoning attacks.

1 Introduction

From customer behavior prediction and medical image analysis to autonomous driving and policy making, machine learning (ML) systems process ever increasing amounts of data. In such data-rich regimes, a fraction of the samples is often noisy, incorrect annotations are likely to occur, and uniform data quality standards become difficult to enforce. To address these challenges, data valuation emerges as a research field receiving increasing attention, focusing on properly assessing the contribution of each datum to ML training [12]. These methods have proven useful in identifying low-quality samples that can be detrimental to model performance, as well as selecting subsets of data that are representative of enhanced model performance [23, 48, 27]. Furthermore, they are widely applicable in data marketplace for fair revenue allocation and incentive design [51, 45, 40].

Nevertheless, existing data valuation methods assign a scalar score to each datum, thereby failing to account for the varied roles of individual cells. This leaves the valuation rationale unclear and can be unsatisfactory and sub-optimal in various practical scenarios. Firstly, whenever a score is assigned to a data point by a particular data valuation method, it is crucial to understand the underlying justifications to ensure transparency and reliability, especially in high-stakes decision making [39]. Secondly, it is important to recognize the fact that even if a data point is of low quality, it is rarely the case that all the cells within this data point are noisy [37, 26, 43]. The absence of detailed insights into how individual cells contribute to ML training inevitably leads to discarding entire data points. This can result in substantial data waste, particularly when only a few cells are noisy and data acquisition is expensive. Finally, in data markets, different cells within a data point may originate from different data sellers [3, 10]. Consequently, a singular valuation for the entire point fails to offer equitable compensation to all contributing parties.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

Age	Income	Experience	Education
25	50000	3	4
34	62000	10	6
45	-1	20	8
29	35000	5	5
40	80000	15	100

(a) Data valuation

Age	Income	Experience	Education
25	50000	3	4
34	62000	10	6
45	$\boxed{-1}$	20	8
29	35000	5	5
40	80000	15	$\boxed{100}$

low valuation scores

(b) Joint valuation

Figure 1: **Comparison of data valuation and joint valuation.** (a) Data valuation evaluates the quality of individual data points, whereas (b) joint valuation evaluates the quality of individual cells. Both panels illustrate the same hypothetical dataset, and the darker colors overlaid represent the higher quality or importance. Joint valuation provides a finer level of attributions than data valuation and aims to describe how features affect data values. As panel (b) illustrates, the joint valuation framework can identify outlier cells highlighted with blue boxes (*i.e.*, -1 in "Income" and 100 in "Education") and provide quantitative interpretations of data values.

Our contributions In this paper, we propose `2D-OOB`, a powerful and efficient joint valuation framework that can attribute a data point's value to its individual features. `2D-OOB` quantifies the importance of each cell in a dataset, as illustrated in Figure 1, providing interpretable insights into which cells are associated with influential data points. Our method is computationally efficient as well as theoretically supported by its connections with `Data-OOB` [23]. Moreover, our extensive empirical experiments demonstrate the practical effectiveness of `2D-OOB` in various use cases. `2D-OOB` accurately identifies cell outliers and pinpoints which cells to fix to improve model performance. `2D-OOB` enables inspection of data poisoning attacks by precisely localizing the backdoor trigger, an artifact inserted into a training sample to induce malicious model behavior [13, 5]. `2D-OOB` is on average 200 times faster than state-of-the-art methods across all datasets examined.

2 Preliminaries

Notations Throughout this paper, we focus on supervised learning settings. For $d \in \mathbb{N}$, we denote an input space and an output space by $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}$, respectively. We denote a training dataset with n data points by $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where (x_i, y_i) is the i-th pair of the input covariates $x_i \in \mathcal{X}$ and its output label $y_i \in \mathcal{Y}$. For an event A, an indicator function $\mathbb{1}(A)$ is 1 if A is true, otherwise 0. For $j \in \mathbb{N}$, we set $[j] := \{1, \ldots, j\}$. For a set S, we denote its power set by 2^S and its cardinality by $|S|$.

DataShapley The primary goal of data valuation is to quantify the contribution of individual data points to a model's performance. Leveraging the Shapley value in cooperative game theory [38], `DataShapley` [12] measures the average change in a utility function $U : 2^{\mathcal{D}} \to \mathbb{R}$ when a data point is removed. For $i \in [n]$, `DataShapley` of i-th datum is defined as follows.

$$\phi_i^{\text{Shap}} := \frac{1}{n} \sum_{k=1}^{n} \frac{1}{\binom{n-1}{k-1}} \sum_{S \subset \mathcal{D}_k^{(i)}} [U(S \cup \{(x_i, y_i)\}) - U(S)] \tag{1}$$

where $\mathcal{D}_k^{(i)} := \{S \subseteq \mathcal{D} | (x_i, y_i) \notin S, |S| = k - 1\}$. `DataShapley` ϕ_i^{Shap} in (1) considers every set $S \in \mathcal{D}_k^{(i)}$ and computes the average difference in utility $U(S \cup \{(x_i, y_i)\}) - U(S)$. It characterizes the impact of a data point, but its computation requires evaluating U for all possible subsets of $\mathcal{D}$, rendering precise calculations infeasible. Many efficient computation algorithms have been studied [15, 25, 50], and in these studies, Shapley-based methods have demonstrated better effectiveness in detecting low-quality samples than standard attribution approaches, such as leave-one-out and influence function methods [20, 9].

Data-OOB As an alternative efficient data valuation method, Kwon and Zou [23] propose `Data-OOB`, which leverages a bagging model and measures the similarity between a nominal label and weak learners' predictions. To be more specific, we suppose a bagging model consists of B weak learners, where for $b \in [B]$, the b-th weak learner $\hat{h}_b$ is given as a minimizer of the weighted empirical risk,

$$\hat{h}_b := \mathrm{argmin}_h \sum_{i=1}^{n} w_{bi} \ell(y_i, h(x_i)),$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function and $w_{bi} \in \mathbb{N}$ is the number of times the i-th datum (x_i, y_i) is selected by the b-th bootstrap dataset. Let $\mathbf{w}_b$ be a weight vector $\mathbf{w}_b := (w_{b1}, \ldots, w_{bn})$ for all $b \in [B]$. For $i \in [n]$ and $\{(\mathbf{w}_b, \hat{h}_b)\}_{b=1}^{B}$, `Data-OOB` of the i-th datum is defined as follows.

$$\phi_i^{\mathrm{OOB}} := \frac{\sum_{b=1}^{B} \mathbb{1}(w_{bi} = 0) T(y_i, \hat{h}_b(x_i))}{\sum_{b=1}^{B} \mathbb{1}(w_{bi} = 0)}, \tag{2}$$

where $T(y_i, \hat{h}_b(x_i))$ is a score function evaluated at (x_i, y_i). We assume that the higher T, the better the prediction. In classification settings, a common choice for T is $\mathbb{1}(y_i = \hat{h}_b(x_i))$, and in this case, `Data-OOB` ϕ_i^{OOB} measures the average similarity between a nominal label y_i and weak learners' predictions $\hat{h}_b(x_i)$ when a datum (x_i, y_i) is *not* sampled in a bootstrap dataset. It intuitively captures the quality of a data point. For instance, when (x_i, y_i) is a mislabeled sample or an outlier, the label y_i is likely to differ from $\hat{h}_b(x_i)$, resulting in ϕ_i^{OOB} being close to zero.

It is noteworthy that `Data-OOB` in (2) can be computed by training a single bagging model, making it computationally efficient. Kwon and Zou [23] show that `Data-OOB` can easily scale to millions of data points, but for `DataShapley` this is often very impractical. In addition, `Data-OOB` is often comparable to or even more effective than `DataShapley` in detecting mislabeled data points and selecting helpful data points [23, 16].

3 Attributing Data Contribution through Joint Valuation Framework

Data valuation quantifies desiderata of data points, however, it does not describe what features contribute and how much to those specific data values. For instance, in anomaly detection tasks, data valuation methods can be deployed to detect anomalous data points, but they do not explain why they are abnormal, which is not generally desirable in practice. To address this challenge, we consider a joint valuation framework and assess a cell score for each feature of a data point. Here, a cell score is designed to quantify how a feature affects the value of an individual data point, attributing a data value to features.

To the best of the author's knowledge, Liu et al. [29] first consider a concept of the joint valuation in literature and introduce `2D-Shapley` to quantitatively interpret `DataShapley`. To this end, we denote a 2D utility function by $u : [n] \times [d] \to \mathbb{R}$, which takes as input a subset of data points $S \subseteq [n]$ and a subset of features $F \subseteq [d]$, and measure the utility of a fragment of the given dataset consisting of cells $\{(i,j)\}_{i \in S, j \in F}$, where a tuple (i,j) denotes a cell at the i-th datum and the j-th column. Then, `2D-Shapley` is defined as

$$\psi_{ij}^{\mathrm{2D-Shap}} := \frac{1}{nd} \sum_{k=1}^{n} \sum_{l=1}^{d} \frac{1}{\binom{n-1}{k-1}\binom{d-1}{l-1}} \sum_{(S,F) \subset \mathcal{D}_{k,l}^{(i,j)}} M_u^{i,j}(S,F) \tag{3}$$

where $\mathcal{D}_{k,l}^{(i,j)} := \{(S,F) | S \subseteq [n] \backslash \{i\}, F \subseteq [d] \backslash \{j\}, |S| = k-1, |F| = l-1\}$ and

$$M_u^{i,j}(S,F) = u(S \cup \{i\}, F \cup \{j\}) + u(S,F) - u(S \cup \{i\}, F) - u(S, F \cup \{j\}).$$

The function $M_u^{i,j}$ allows us to quantify how much removing a specific cell at (i,j) from a given set $(S \cup \{i\}, F \cup \{j\})$ affects the overall utility, and `2D-Shapley` $\psi_{ij}^{\mathrm{2D-Shap}}$ evaluates the average $M_u^{i,j}$ across all possible data fragments $(S,F) \subset \mathcal{D}_{k,l}^{(i,j)}$.

Similar to `DataShapley`, the permutation of all rows and columns required for exact `2D-Shapley` calculations presents significant computational challenges. To address this, Liu et al. [29] develop

2D-KNN, which utilizes k-nearest-neighbors models as surrogates to approximate 2D-Shapley values. However, the approximation methods can compromise the accuracy of valuations [23, 16]. Additionally, 2D-KNN still faces challenges scaling to large-scale datasets and high-dimensional settings.

We propose 2D-OOB, an *efficient* and *model-agnostic* joint valuation framework that leverages out-of-bag estimation to attribute data contribution. We further illustrate how 2D-OOB is connected to Data-OOB, thereby facilitating sample-wise interpretation for data valuation in Section 3.2.

3.1 2D-OOB: an efficient joint valuation framework

Our idea builds upon the subset bagging model [14], which is well recognized as an earlier version of Breiman's random forest model [4]. A key distinction from a standard bagging model is that a weak learner in a subset bagging model is trained on a randomly selected subset of features. For $b \in [B]$, we denote the b-th random feature subset by $S_b \subseteq [d]$. Then, the b-th weak learner of a subset bagging model is given as follows.

$$\hat{f}_b := \mathrm{argmin}_f \sum_{i=1}^{n} w_{bi} \ell(y_i, f(x_{i,S_b})),$$

where x_{i,S_b} is a subvector of x_i that only takes elements in a subset S_b. This difference enables us to assess the impact of which features are more influential: if S_b includes a helpful (or detrimental) feature, we can expect the out-of-bag prediction $\hat{f}(x_{i,S_b})$ to be good (or poor). We formalize this intuition and propose 2D-OOB. For $i \in [n]$, $j \in [d]$ and $\{(w_b, S_b, \hat{f}_b)\}_{b=1}^{B}$, the 2D-OOB for the j-th cell of the i-th data point is defined as follows,

$$\psi_{ij}^{\mathrm{2D-OOB}} := \frac{\sum_{b=1}^{B} \mathbb{1}(w_{bi} = 0, j \in S_b) T(y_i, \hat{f}_b(x_{i,S_b}))}{\sum_{b=1}^{B} \mathbb{1}(w_{bi} = 0, j \in S_b)}, \tag{4}$$

where $T : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a utility function that scores the performance of the weak learner $\hat{f}_b(x_{i,S_b})$ on the i-th datum (x_i, y_i). Specifically, for binary or multi-class classification problems, we can adopt $T(y_i, \hat{f}_b(x_{i,S_b})) = \mathbb{1}(y_i = \hat{f}_b(x_{i,S_b}))$. In this case, 2D-OOB measures the average accuracy score of out-of-bag predictions (specifically, when the i-th data point is out-of-bag) if a cell j is used in training $\hat{f}_b$. For regression problems, we can use the negative squared error loss function, defined as $T(y_i, \hat{f}_b(x_{i,S_b})) = -(y_i - \hat{f}_b(x_{i,S_b}))^2$. In practice, $\mathcal{X}$ could also be incorporated into T to suit the specific use case.

While Data-OOB in (2) aims to assess the impact of the i-th datum, 2D-OOB in (4) provides interpretable insights by evaluating the data point with various combinations of features, revealing which cells are influential to model performance. Leveraging subset bagging scheme, 2D-OOB requires a single training of the bagging model, and thus it is computational efficiency.

3.2 Connection to Data-OOB

We now present interpretable expressions of how 2D-OOB connects to Data-OOB in the following proposition. To begin with, we denote a set of subsets of $[d]$ by $\mathcal{S} := \{S \subseteq [d]\}$. With $\{(\mathbf{w}_b, \hat{f}_b)\}_{b=1}^{B}$, we define the i-th Data-OOB when a particular subset S is used as follows and denote it by $\phi_i^{\mathrm{OOB}}(S)$.

$$\phi_i^{\mathrm{OOB}}(S) := \frac{\sum_{b=1}^{B} \mathbb{1}(w_{bi} = 0) T(y_i, \hat{f}_b(x_{i,S}))}{\sum_{b=1}^{B} \mathbb{1}(w_{bi} = 0)}.$$

Proposition 3.1. *For all $i \in [n]$ and $j \in [d]$, $\psi_{ij}^{\mathrm{2D-OOB}}$ can be expressed as follows.*

$$\psi_{ij}^{\mathrm{2D-OOB}} = \mathbb{E}_{\hat{F}_S}[\phi_i^{\mathrm{OOB}}(S) \mid j \in S],$$

where $\hat{F}_S$ is an empirical distribution with respect to S induced by the sampling process.

A proof is given in the Appendix C. Proposition 3.1 shows that 2D-OOB $\psi_{ij}^{\mathrm{2D-OOB}}$ can be expressed as a conditional empirical expectation of Data-OOB provided that the j-th feature is used in Data-OOB computation. It provides intuitive interpretations: for a fixed i and $j \neq k$, $\psi_{ij}^{\mathrm{2D-OOB}} > \psi_{ik}^{\mathrm{2D-OOB}}$ implies that the cell x_{ij} is more helpful to achieve the high OOB score, which serves as an indicator of model performance, than the cell x_{ik}.

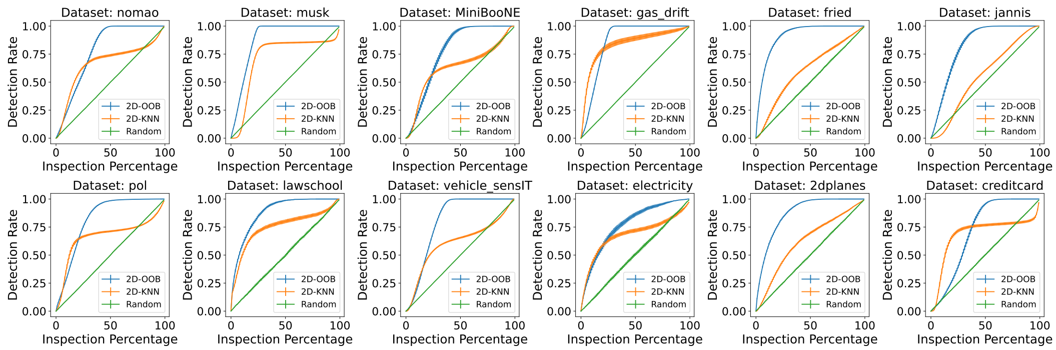

Figure 2: **Cell-level outlier detection rate curves for** `2D-OOB`**,** `2D-KNN`**, and** `Random`**.** The x-axis represents the percentage of inspected cells. The y-axis represents the detection rate, defined as the ratio of the number of detected outlier cells to the total number of outlier cells present in a dataset. The error bars show a 95% confidence interval based on 30 independent experiments. We examine the cells in ascending order, starting from those with the lowest values, and thus a curve closer to the left-top corner indicates better performance. `2D-OOB` efficiently detects the majority of outlier cells by examining only a small fraction of the total cells, while `2D-KNN` and `Random` require scanning nearly all the cells.

4 Experiments

In this section, we empirically show the effectiveness of `2D-OOB` across multiple use cases of the joint valuation: *cell-level outlier detection*, *cell fixation*, and *backdoor trigger detection*. As a summary, `2D-OOB` can precisely identify anomalous cells that should be prioritized for examination and subsequent fixation to improve model performance. In the context of backdoor trigger detection, `2D-OOB` demonstrates its efficacy by accurately identifying different types of triggers within poisoned data, showcasing its proficiency in detecting non-random, targeted anomalies. Our method also exhibits high computational efficiency through run-time comparison.

Throughout all of our experiments, `2D-OOB` uses a subset bagging model with $B = 1000$ decision trees. We randomly select a fixed ratio of features to build each decision tree. Unless otherwise specified, we utilize half of the features for each weak learner and set $T(y_i, \hat{f}(x_{i,S_b})) = \mathbb{1}(y_i = \hat{f}(x_{i,S_b}))$. The run time is measured on a single Intel Xeon Gold 6226 2.9 Ghz CPU processor.

4.1 Cell-level outlier detection

Experimental setting In practical situations, even when dealing with abnormal data points, it is not always the case that all cells are noisy [37, 29, 21]. To simulate more realistic settings, we introduce noise to certain *cells* in the following two-step process: First, we randomly select 20% rows for each dataset. We then select 20% columns uniformly at random, allowing each selected row to have a different set of perturbed cells. We inject noises sampled from the low-probability region into these cells, following Du et al. [8] and Liu et al. [29]. Details on the outlier injection process can be found in Appendix A.3.

We use 12 publicly accessible binary classification datasets from OpenML, encompassing a range of both low and high-dimensional datasets, which have been widely used in the literature [12, 22, 23]. Details on these datasets are presented in Appendix A.1. For each dataset, 1000 and 3000 data points are randomly sampled for training and test datasets, respectively. For the baseline method, we consider `2D-KNN`, a fast and performant variant of `2D-Shapley` [29]. We incorporate a distance regularization term in the utility function T for enhanced performance.

Results We calculate the valuations for each cell using our joint valuation framework. Ideally, the outlier cells should receive a low valuation. We then arrange the cell valuations in *ascending* order and inspect those cells with the lowest values first.

Table 1: **Cell-level outlier detection results.** AUC and run-time comparison between `2D-OOB` and `2D-KNN` across the twelve datasets. The average and standard error of the AUC and run-time (in seconds) based on 30 independent experiments are denoted by "average $\pm$ standard error". Bold numbers denote the best method. The AUC value for the `Random` method consistently remains at 0.5 across all datasets. Overall, `2D-OOB` achieves a significantly higher AUC while being orders of magnitude faster than `2D-KNN`.

Dataset	**AUC ↑**		**Run-time ↓**	
	2D-OOB (ours)	2D-KNN	2D-OOB (ours)	2D-KNN
lawschool	**0.88± 0.0027**	0.75± 0.0011	**3.33 ± 0.06**	177.56 ± 1.92
electricity	**0.77± 0.0072**	0.68± 0.0014	**3.39 ± 0.07**	191.38 ± 2.60
fried	**0.91± 0.0015**	0.61± 0.0005	**3.97 ± 0.10**	322.79 ± 2.98
2dplanes	**0.87± 0.0015**	0.62± 0.0005	**3.46 ± 0.05**	295.25 ± 2.37
creditcard	**0.72± 0.0028**	0.69± 0.0011	**4.56 ± 0.10**	662.34 ± 7.12
pol	**0.82± 0.0014**	0.67± 0.0006	**4.34 ± 0.05**	759.33 ± 4.37
MiniBooNE	**0.77± 0.0058**	0.63± 0.0019	**7.46 ± 0.06**	1507.83 ± 14.50
jannis	**0.83± 0.0042**	0.55± 0.0004	**7.98 ± 0.07**	1753.10 ± 12.35
nomao	**0.79± 0.0021**	0.67± 0.0009	**7.69± 0.11**	2564.58 ± 23.11
vehicle_sensIT	**0.81± 0.0014**	0.61± 0.0005	**9.87 ± 0.08**	3113.65 ± 24.54
gas_drift	**0.86± 0.0010**	0.84± 0.0017	**11.28± 0.10**	3878.31 ± 40.72
musk	**0.88± 0.0008**	0.71± 0.0006	**14.09 ± 0.11**	4415.45 ± 22.96
Average	**0.83**	0.67	**6.78**	1636.80

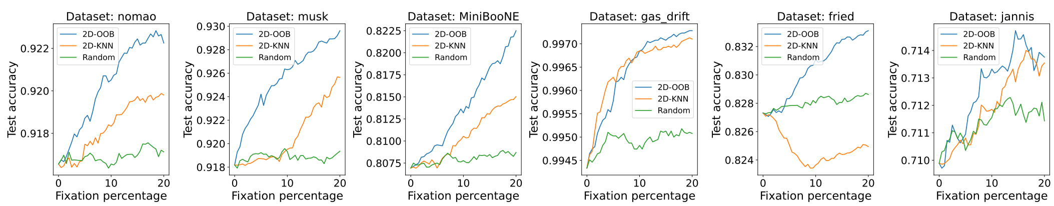

Figure 3: **Cell fixation experiment results (test accuracy curves) for `2D-OOB`, `2D-KNN`, and `Random`.** We replace cells with their corresponding ground-truth annotations, starting with those cells assigned the lowest valuations. The results from 6 datasets are presented, and additional results are provided in Appendix B.2. We conduct 30 independent trials and report the average results. A higher curve indicates better performance. `2D-OOB` demonstrates a superior capability in accurately identifying and rectifying cell-level outliers.

The detection rate curve of inserted outlier is shown in Figure 2. For all datasets, `2D-OOB` successfully identifies over 90% of the outlier cells by inspecting only 30% of the bottom cells. In comparison, `2D-KNN` requires examining nearly 90% of the cells to achieve the same detection level.

We also evaluate the area under the curve (AUC) as a quantitative metric and the run-time. As Table 1 shows, `2D-OOB` achieves an average AUC of 0.83 across 12 datasets, compared to 0.67 for `2D-KNN`, while being significantly faster. For high-dimensional datasets such as the musk dataset, which comprises 166 features, `2D-KNN` would take more than an hour to process, while `2D-OOB` can finish in seconds. Furthermore, we present additional results on **multi-class classification** datasets in Appendix B.1, demonstrating the consistently superior performance and efficiency of `2D-OOB`.

4.2 Cell fixation experiment

Experimental setting A naive strategy to handle cell-level outliers is to eliminate data points that contain outliers. This method, however, risks substantial data loss, particularly when outliers are scattered and data points are costly to collect. We instead consider a cell fixation experiment, where we assume that the ground-truth annotations of outlier cells can be restored with external expert knowledge. At each step, we "fix" a certain number of cells by substituting them with their ground-truth annotations, prioritizing cells that have the lowest valuations. Then we fit a logistic model and evaluate the model's performance with a test set of 3000 samples. It is important to note that correcting normal cells has no effect, whereas fixing outlier cells is expected to enhance the model's performance. We adopt the same datasets and implementations as in Section 4.1.

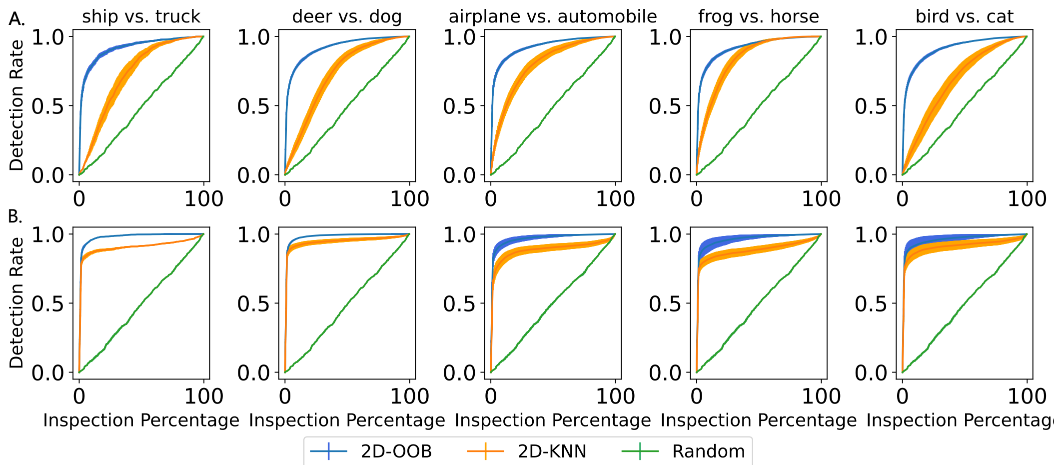

Figure 4: **Backdoor trigger detection rate curve for 2D-OOB, 2D-KNN, and Random.** The panels A. and B. correspond to Trojan square and BadNets square, respectively. We prioritize cells within each poisoned sample, ranking from highest to lowest based on their valuations. The detection rate curve shows the average detection rate across all poisoned samples, and error bars represent a 95% confidence interval based on 15 independent runs. 2D-OOB demonstrates superior performance in detecting the cells implanted with triggers.

Results Figure 3 illustrates the anticipated trend in the performance of 2D-OOB, validating our method's capability to accurately identify and prioritize the most impactful outliers for correction. As cells with the lowest valuations are progressively fixed, 2D-OOB demonstrates a consistent improvement in model accuracy. In contrast, when applying the same procedure with 2D-KNN, such notable performance enhancements are not observed.

Additionally, we investigate a scenario where ground-truth annotations remain unavailable. We adopt the setup from Liu et al. [29], where we replace the outlier cells with the average of other cells in the same feature column. 2D-OOB uniformly demonstrates significant superiority over its counterparts. Results are provided in Appendix B.2.

4.3 Backdoor trigger detection

A common strategy of data poisoning attacks involves inserting a predefined trigger (e.g., a specific pixel pattern in an image) into a few training data [13, 5, 28]. These malicious manipulations can be challenging to detect as they only infect targeted samples. Even when poisoned data are present, it could be difficult to discern the cause of attacks since manually reviewing the images is expensive and time-consuming. In this experiment, we introduce a novel joint valuation task: detecting backdoor triggers in data poisoning attacks. Distinct from random outliers investigated previously, such cell contamination is targeted and deliberate.

We consider two popular backdoor attack algorithms: BadNets [13] and Trojan Attack [28]. The poisoned samples, relabeled as the adversarial target class, are mixed up with the clean data in the training process. As a result, the model is trained to incorrectly treat the trigger as a main feature of the poisoned samples. At the test time, those inputs containing the trigger will be misclassified to the target class. In this context, our goal is to effectively pinpoint the triggers by recognizing them as influential features through our joint valuation framework.

Experimental setting We select 5 pairs of CIFAR-10 classes. For each pair, we designate one as the target attack class and the other as the source class. The training dataset comprises 1000 images. For each attack, we contaminate 15% of the training samples from the source class and relabel them to the target class. Two types of attack triggers are implemented: Trojan square and BadNets square [13, 35, 28]. These triggers are placed in the lower right corner of the original images to minimize occlusion. Details of these attacks are available in the Appendix A.4. In our experiment, the ratio of poisoned cells is approximately 1%. We sample 25% features to build each weak learner.

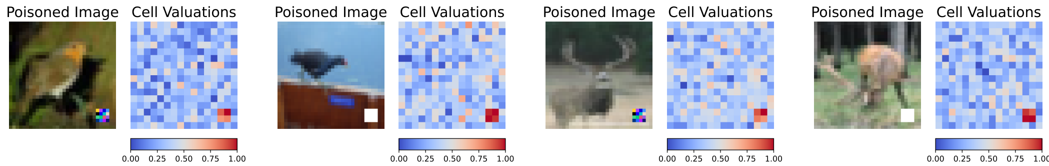

Figure 5: **Qualitative examples for `2D-OOB` in the backdoor trigger detection task.** Each pair of images shows a poisoned image and its cell valuation. The color of the heatmap indicates importance: red cells are more important than blue cells. The first two pairs consider the case the class "bird" is relabeled as "cat", and the latter two pairs consider the case the class "deer" is relabeled as "cat". The heatmaps clearly show that higher cell valuations predominantly concentrate on the regions containing triggers, while areas featuring actual objects receive lower valuations. This pattern suggests that `2D-OOB` effectively captures the triggers as the impactful features responsible for the misclassification of the poisoned samples.

Results We adopt the same detection scheme and baseline methods as in Section 4.1. Ideally, the poisoned cell should receive a high valuation based on the fact that such data point has been relabeled. We plot the detection rate curves of five datasets as shown in Figure 4. `2D-OOB` significantly outperforms `2D-KNN` in detecting both types of triggers. Overall, `2D-OOB` achieves an average AUC of 0.95 across all datasets and attack types, compared to 0.83 for `2D-KNN`.

Qualitative examples Figure 5 displays the heatmaps for poisoned samples based on cell valuations of `2D-OOB`. Areas with higher cell valuations (marked as dark red color) precisely indicate the trigger location in these samples, illustrating the effectiveness of our detection. More examples are included in the Appendix B.3.

4.4 Ablation study

We conduct ablation studies on the cell-level outlier detection task, as outlined in Section 4.1, to examine the impact of the selection and number of weak learners on `2D-OOB` estimations.

Selection of weak learners Although our study primarily employs decision trees as weak learners, it is important to note that `2D-OOB` is **model-agnostic**, enabling the use of any class of machine learning models as weak learners. We compare efficacy of decision trees, logistic regression, a single-layer MLP with 64 dimensions, and a two-layer MLP with 64 and 32 dimensions.

Table 2 presents a comparison of detection AUC across 12 datasets, indicating that `2D-OOB` is not model-free. The selection of weak learners slightly affects the valuation results, with more complex models generally yielding better performance. Nonetheless, all variations of `2D-OOB` outperform `2D-KNN`, highlighting the significant advantages of the `2D-OOB` approach.

The number of weak learners Increasing the number of weak learners allows for a greater number of data-feature subset pairs to be explored, potentially leading to more accurate estimates. However, we empirically observe that beyond a certain threshold, adding extra weak learners does not substantially enhance performance, indicating convergence of the estimation in Appendix B.4. As a summary, we vary the number of weak learners $B \in \{500, 1000, 3000\}$ and compare the cell-level outlier detection performance. Typically, when the number of weak learners is 1000, *i.e.*, $B = 1000$, it is sufficient to achieve converged estimates across different datasets.

Lastly, we present additional ablation study results for other key hyperparameters in Appendix B.4. Apart from the experiments discussed above, we showcase that marginalization of `2D-OOB` can either match or surpass state-of-the-art data valuation methods on standard benchmarks in Appendix D.

5 Related work

Data contribution estimation In addition to the marginal contribution-based methods discussed in Section 2, many other approaches are emerging in the area of data valuation. Just et al. [18] develop a non-conventional class-wise Wasserstein distance between the training and validation sets and use the

Table 2: **Ablation study results of weak learner types.** The average and standard error of the AUC based on 30 independent experiments are denoted by "average ± standard error". Results of 2D-KNN are added for comparison. Different types of weak learners lead to variations in the valuation results, with more complex models generally showing better performance.

Dataset	Decision Tree	Logistic Regression	MLP (single-layer)	MLP (two-layer)	2D-KNN (Baseline)
lawschool	**0.88 ± 0.0027**	0.81 ± 0.0014	0.83 ± 0.0023	0.86 ± 0.0049	0.75 ± 0.0011
electricity	**0.77 ± 0.0072**	0.75 ± 0.0029	0.75 ± 0.0039	0.74 ± 0.0064	0.68 ± 0.0014
fried	**0.91 ± 0.0015**	0.82 ± 0.0023	0.85 ± 0.0020	0.88 ± 0.0027	0.61 ± 0.0005
2dplanes	0.87 ± 0.0015	0.82 ± 0.0026	0.86 ± 0.0026	**0.88 ± 0.0037**	0.62 ± 0.0005
creditcard	0.72 ± 0.0028	**0.74 ± 0.0023**	**0.74 ± 0.0026**	**0.74 ± 0.0071**	0.69 ± 0.0011
pol	0.82 ± 0.0014	0.79 ± 0.0029	0.85 ± 0.0014	**0.86 ± 0.0019**	0.67 ± 0.0006
MiniBooNE	0.77 ± 0.0058	0.77 ± 0.0059	0.80 ± 0.0057	**0.81 ± 0.0119**	0.63 ± 0.0019
jannis	**0.83 ± 0.0042**	0.76 ± 0.0040	0.79 ± 0.0048	0.80 ± 0.0108	0.55 ± 0.0004
nomao	0.79 ± 0.0021	0.82 ± 0.0012	**0.83 ± 0.0010**	**0.83 ± 0.0017**	0.67 ± 0.0009
vehicle-sensIT	0.81 ± 0.0014	0.81 ± 0.0026	0.80 ± 0.0025	**0.82 ± 0.0037**	0.61 ± 0.0005
gas-drift	0.86 ± 0.0010	**0.89 ± 0.0005**	0.88 ± 0.0005	0.88 ± 0.0006	0.84 ± 0.0017
musk	0.88 ± 0.0008	0.87 ± 0.0005	**0.88 ± 0.0005**	**0.88 ± 0.0008**	0.71 ± 0.0006
Average	**0.83**	0.80	0.82	**0.83**	0.67

gradient information to evaluate each data point. Wu et al. [47] extend data valuation to deep neural networks, introducing a training-free data valuation framework based on neural tangent kernel theory. Yoon et al. [48] leverage reinforcement learning techniques to automatically learn data valuation scores by training a regression model. However, all these data valuation methods do not assign importance scores to cells, whereas our method provides additional insights into how individual cells contribute to the data valuations.

Feature attribution Feature attribution is a pivotal research domain in explainable machine learning that primarily aims to provide insights into how individual features influence model predictions. Various effective methods have been proposed, including SHAP-based explanation [30, 31, 24, 7, 6], counterfactual explanation [44, 17, 36, 32, 33], and backpropagation-based explanation [1, 2, 42, 41, 49]. Among these methods, the SHAP-based explanation stands out as the most widely adopted approach, utilizing cooperative game theory principles to compute the Shapley value [38]. While feature attribution offers a potential method to attribute data valuation scores across individual cells, our empirical experiments in Appendix B.1 reveal that this two-stage scheme falls short in efficacy compared to our proposed joint valuation paradigm, which integrates data valuation and feature attribution in a simultaneous process.

6 Conclusion

We propose 2D-OOB, an efficient joint valuation framework that assigns a score to each cell in a dataset, thereby facilitating a finer attribution of data contribution and enabling a deeper understanding of datasets. Through comprehensive experiments, we show that 2D-OOB is computationally efficient and competitive over state-of-the-art methods in both joint valuation tasks.

Limitation and future work While our study primarily explores random forest models applied to tabular datasets and simple image datasets, the potential application of neural network models within the 2D-OOB framework for more complex vision and language tasks presents a promising avenue for future investigation. For instance, in text datasets, tokens or words can be treated as cells. 2D-OOB can be easily integrated into any bagging training scheme that uses language models.

Overall, we believe that our work will inspire further exploration in the field of joint valuation, with the broader goal of improving the transparency and interpretability of machine learning, as well as developing an equitable incentive mechanism for data sharing.

References

[1] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[3] Jens Bleiholder and Felix Naumann. Data fusion. *ACM computing surveys (CSUR)*, 41(1):1–41, 2009.

[4] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[5] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[6] Ian Covert and Su-In Lee. Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 3457–3465. PMLR, 2021.

[7] Ian Covert, Scott M Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33:17212–17223, 2020.

[8] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*, 2022.

[9] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

[10] Raul Castro Fernandez, Pranav Subramaniam, and Michael J Franklin. Data market platforms: Trading data assets to solve data problems. *arXiv preprint arXiv:2002.01047*, 2020.

[11] Matthias Feurer, Jan N Van Rijn, Arlind Kadra, Pieter Gijsbers, Neeratyoy Mallik, Sahithya Ravi, Andreas Müller, Joaquin Vanschoren, and Frank Hutter. Openml-python: an extensible python api for openml. *The Journal of Machine Learning Research*, 22(1):4573–4577, 2021.

[12] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[13] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[14] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[15] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619*, 2019.

[16] Kevin Fu Jiang, Weixin Liang, James Zou, and Yongchan Kwon. Opendataval: a unified benchmark for data valuation. *arXiv preprint arXiv:2306.10577*, 2023.

[17] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615*, 2019.

[18] Hoang Anh Just, Feiyang Kang, Jiachen T Wang, Yi Zeng, Myeongseob Ko, Ming Jin, and Ruoxi Jia. Lava: Data valuation without pre-specified learning algorithms. *arXiv preprint arXiv:2305.00054*, 2023.

[19] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. `https://archive.ics.uci.edu`, 2017. The UCI Machine Learning Repository.

[20] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[21] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13*, pages 831–838. Springer, 2009.

[22] Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *arXiv preprint arXiv:2110.14049*, 2021.

[23] Yongchan Kwon and James Zou. Data-OOB: Out-of-bag estimate as a simple and efficient data value. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 18135–18152. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/kwon23e.html`.

[24] Yongchan Kwon and James Y Zou. Weightedshap: analyzing and improving shapley based feature attributions. *Advances in Neural Information Processing Systems*, 35:34363–34376, 2022.

[25] Yongchan Kwon, Manuel A. Rivas, and James Zou. Efficient computation and analysis of distributional shapley values. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 793–801. PMLR, 13–15 Apr 2021. URL `https://proceedings.mlr.press/v130/kwon21a.html`.

[26] Andy Leung, Hongyang Zhang, and Ruben Zamar. Robust regression estimation and inference in the presence of cellwise and casewise contamination. *Computational Statistics & Data Analysis*, 99:1–11, 2016.

[27] Weixin Liang, Girmaw Abebe Tadesse, Daniel Ho, L Fei-Fei, Matei Zaharia, Ce Zhang, and James Zou. Advances, challenges and opportunities in creating data for trustworthy ai. *Nature Machine Intelligence*, 4 (8):669–677, 2022.

[28] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and X. Zhang. Trojaning attack on neural networks. In *Network and Distributed System Security Symposium*, 2018. URL `https://api.semanticscholar.org/CorpusID:31806516`.

[29] Zhihong Liu, Hoang Anh Just, Xiangyu Chang, Xi Chen, and Ruoxi Jia. 2d-shapley: A framework for fragmented data valuation. *arXiv preprint arXiv:2306.10473*, 2023.

[30] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[31] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[32] Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*, 2019.

[33] 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*, pages 607–617, 2020.

[34] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018. URL `https://arxiv.org/pdf/1807.01069`.

[35] Ren Pang, Zheng Zhang, Xiangshan Gao, Zhaohan Xi, Shouling Ji, Peng Cheng, and Ting Wang. Trojanzoo: Towards unified, holistic, and practical evaluation of neural backdoors. In *Proceedings of IEEE European Symposium on Security and Privacy (Euro S&P)*, 2022.

[36] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.

[37] Peter J Rousseeuw and Wannes Van Den Bossche. Detecting deviating data cells. *Technometrics*, 60(2): 135–145, 2018.

[38] Lloyd S Shapley et al. A value for n-person games. 1953.

[39] Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning:“ingredients”, strategies, and open challenges. In *Proc. IJCAI*, 2022.

[40] Rachael Hwee Ling Sim, Yehong Zhang, Trong Nghia Hoang, Xinyi Xu, Bryan Kian Hsiang Low, and Patrick Jaillet. Incentives in private collaborative machine learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[41] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[42] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[43] Peng Su, Garth Tarr, and Samuel Muller. Robust variable selection under cellwise contamination. *Journal of Statistical Computation and Simulation*, pages 1–17, 2023.

[44] 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.

[45] Jiachen T Wang, Prateek Mittal, and Ruoxi Jia. Efficient data shapley for weighted nearest neighbor algorithms. *arXiv preprint arXiv:2401.11103*, 2024.

[46] Tianhao Wang and Ruoxi Jia. Data banzhaf: A data valuation framework with maximal robustness to learning stochasticity. *arXiv preprint arXiv:2205.15466*, 2022.

[47] Zhaoxuan Wu, Yao Shu, and Bryan Kian Hsiang Low. Davinz: Data valuation using deep neural networks at initialization. In *International Conference on Machine Learning*, pages 24150–24176. PMLR, 2022.

[48] Jinsung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In *International Conference on Machine Learning*, pages 10842–10851. PMLR, 2020.

[49] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[50] Jiayao Zhang, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. Efficient sampling approaches to shapley value approximation. *Proc. ACM Manag. Data*, 1(1), may 2023. doi: 10.1145/3588728. URL `https://doi.org/10.1145/3588728`.

[51] Boxin Zhao, Boxiang Lyu, Raul Castro Fernandez, and Mladen Kolar. Addressing budget allocation and revenue allocation in data market environments using an adaptive sampling algorithm. *arXiv preprint arXiv:2306.02543*, 2023.

Supplementary Materials

In the supplementary materials, we provide implementation details, additional experimental results, rigorous formalized proofs and data valuation experiment results. Code repository can be found at `https://anonymous.4open.science/r/2d-oob-C4A0/`.

A Implementation details

A.1 Datasets

Tabular datasets We use 12 binary classification datasets obtained from OpenML [11]. A summary of all the datasets is provided in Table 3. These datasets are used in Section 4.1, 4.2, and Appendix D.

For each dataset, we first employ a standard normalization procedure, where each feature is normalized to have zero mean and unit standard deviation. After preprocessing, we randomly partition a subset of the data into two non-overlapping sets: a training dataset and a test dataset, which consists of 1000 and 3000 samples respectively. The training dataset is used to obtain the joint (or marginal) valuation for each cell (or data point). The test dataset is exclusively used for cell fixation (or point removal) experiments when evaluating the test accuracy. Note that for methods that need a validation dataset such as `KNNShapley` and `DataShapley`, we additionally sample a separate validation dataset (disjoint from training dataset and test dataset) to evaluate the utility function. The size of the validation dataset is set to 10% of the training sample size.

Image datasets We create datasets by pairing CIFAR-10 classes, each pair consisting of a target attack class and a source class. The training and test dataset comprises 1000 and 2000 samples respectively. The size of the validation dataset is set to 10% of the training sample size. To manage the computational challenges posed by the baseline method, we employ the super-pixel technique to transform the (32,32,3) image into a 256-dimensional vector. Specifically, we first average the pixel values across three channels for each pixel. Then, we partition these transformed images into equally sized 2×2 grids. In each grid, we use average pooling to reduce the pixel values to a single cell value. These cell values are then arranged into a flattened input vector. We annotate a cell as poisoned if at least 25% of its corresponding grid area contains the trigger.

A.2 Implementation details for different methods

`2D-OOB` `2D-OOB` involves fitting a *subset* random forest model with $B = 1000$ decision trees based on the package "scikit-learn". When constructing each decision tree, we fix the feature subset size ratio as 0.5. Ablations on the hyperparameters can be found in Appendix B.4. For Section 4.3 and Appendix D, we simply adopt $T(y_i, \hat{f}(x_{i,S_b})) = \mathbb{1}(y_i = \hat{f}(x_{i,S_b}))$. For Section 4.1 and 4.2, we further calculate the normalized negative L2 distance between covariates and the class-specific mean in the bootstrap dataset, denoted as d_{norm}. Then we use $T(y_i, \hat{f}(x_{i,S_b})) = \mathbb{1}(y_i = \hat{f}(x_{i,S_b})) + d_{norm}$.

`2D-KNN` `2D-KNN` employs KNN as a surrogate model to approximate `2D-Shapley`. We set the number of nearest neighbors as 10 and the number of permutations as 1000. The hyperparameters are selected based on convergence behavior and we determine the run time until the values converge.

A.3 Implementation details for cell-level outlier generation

Following Du et al. [8] and Liu et al. [29], we replace a given cell with the outlier value. Here, the outlier value is randomly generated from the two-sided "tails" of the Gaussian distribution with the column mean and standard deviation, where the probability of the two-sided tail area is set to be 1%. 4% ($20\% \times 20\%$) of the cells in total are replaced with the corresponding outlier value.

A.4 Implementation details for backdoor trigger generation

Following the prior work [13, 28], we generate the BadNets square and the Trojan square trigger. For BadNets, we adopt the implementation in Nicolae et al. [34]. For Trojan Attack, we use a pretrained ResNet18 model on CIFAR-10 dataset and employ the implementation in Pang et al. [35]. For each attack, we evaluate its effectiveness by training a decision tree model on the poisoned dataset. The accuracy on a clean test set remains nearly unchanged compared to the model trained on an uncontaminated training set, while the attack success rate on a hold-out poisoned test sample set is guaranteed to exceed 75%.

Table 3: **A summary of all the datasets used in 4.1, 4.2, and Appendix D.** These datasets have been commonly used in previous literature [12, 22, 23]

Name	Total sample size	Input dimension	Majority class proportion	OpenML ID
lawschool	20800	6	0.679	43890
electricity	38474	6	0.5	44080
fried	40768	10	0.502	901
2dplanes	40768	10	0.501	727
creditcard	30000	23	0.779	42477
pol	15000	48	0.664	722
MiniBooNE	72998	50	0.5	43974
jannis	57580	54	0.5	43977
nomao	34465	89	0.715	1486
vehicle_sensIT	98528	100	0.50	357
gas_drift	5935	128	0.507	1476
musk	6598	166	0.846	1116

B Additional experimental results

In Section 4.1, we demonstrate that `2D-OOB` shows promising performance in identifying cell-level outliers. This section further shows that our result is not sensitive to the selection of hyperparameters. Furthermore, ours generally performs better than `2D-KNN` in different settings. We also provide additional results for Section 4.2 and 4.3.

B.1 Additional results for cell-level outlier detection

Additional results on multi-class classification datasets We have conducted cell-level outlier detection experiments (as in Section 4.1) on three multi-class classification datasets from the UCI Machine Learning repository [19]. As shown in the table, `2D-OOB` displays superior detection performance and efficiency.

Table 4: **Cell-level outlier experiment results on multi-class classification datasets** The average and standard error of the detection AUC and Elapsed Time (in seconds) based on 30 independent experiments are denoted by “average $\pm$ standard error“.

Dataset	**AUC** $\uparrow$		**Run-time** $\downarrow$	
	`2D-OOB` (ours)	`2D-KNN`	`2D-OOB` (ours)	`2D-KNN`
Covertype	**0.81±0.0156**	0.63±0.0183	**3.98±0.5774**	962.34±1.3383
Dry Bean	**0.88±0.0059**	0.85±0.0192	**3.31±0.4586**	347.80±2.0212
Wine Quality	**0.86±0.0178**	0.57±0.0252	**2.90±0.1240**	269.14±1.1825

Additional baseline: two-stage attribution Once we obtain the data valuation scores, an alternative solution approach to determining cell-level attributions involves leveraging feature attribution methods such as SHAP [30]. We explore an additional baseline method building upon this idea: initially, `Data-OOB` (or any other data valuation method) is computed for the i-th data point, denoted as dv_i. Subsequently, `TreeSHAP` [31] is fitted, using dv_i as the target and the concatenation of x_i and y_i (denoted as $x_i \oplus y_i$) as the predictor. The derived local feature attributions are then interpreted as joint valuation results. We refer to this method as ‘two-stage attribution’.

Table 5 indicates that `2D-OOB` substantially outperforms its two-stage counterpart. We hypothesize that the superiority of our method stems from integrating data valuation and feature attribution into a cohesive framework. Conversely, the two-stage method treats data valuation and feature

Table 5: **Cell-level outlier detection results (AUC) of `2D-OOB` and the two-stage attribution.** Our method shows a better performance than the alternative method by a significant performance margin.

Dataset	AUC ↑	
	`2D-OOB` (ours)	Two-stage attribution
lawschool	**0.88± 0.0027**	0.83± 0.0064
electricity	**0.77± 0.0072**	0.64± 0.0093
fried	**0.91± 0.0015**	0.82± 0.0068
2dplanes	**0.87± 0.0015**	0.80± 0.0058
creditcard	**0.72± 0.0028**	0.67± 0.0051
pol	**0.82± 0.0014**	0.78± 0.0042
MiniBooNE	**0.77± 0.0058**	0.70± 0.0041
jannis	**0.83± 0.0042**	0.62± 0.0043
nomao	**0.79± 0.0021**	0.71± 0.0041
vehicle_sensIT	**0.81± 0.0014**	0.64± 0.0033
gas_drift	**0.86± 0.0010**	0.73± 0.0143
musk	**0.88± 0.0008**	0.68± 0.0028
Average	**0.83**	0.72

Table 6: **Cell-level outlier detection results (AUC) of different joint valuation methods when the row outlier ratio and column outlier ratio are both** 50%**.** Our method consistently outperforms `2D-KNN` even in the presence of significant noise.

Dataset	AUC ↑	
	`2D-OOB` (ours)	`2D-KNN`
lawschool	**0.75± 0.0084**	0.60± 0.0144
electricity	**0.64± 0.0155**	0.60± 0.0106
fried	**0.74± 0.0087**	0.54± 0.0027
2dplanes	**0.74± 0.0063**	0.55± 0.0033
creditcard	**0.63± 0.0055**	0.61± 0.0053
pol	**0.69± 0.0069**	0.60± 0.0042
MiniBooNE	**0.67± 0.0128**	0.60± 0.0048
jannis	**0.70± 0.0113**	0.53± 0.0014
nomao	**0.70± 0.0088**	0.58± 0.0052
vehicle_sensIT	**0.70± 0.0075**	0.55± 0.0031
gas_drift	**0.73± 0.0077**	0.65± 0.0114
musk	**0.77± 0.0063**	0.64± 0.0038
Average	**0.71**	0.59

attribution as separate processes, potentially resulting in sub-optimal outcomes. Furthermore, due to the computational complexity of `TreeSHAP`, the two-stage approach is notably slower compared to our method.

A noisy setting with more outlier cells We consider a more challenging scenario with increased outlier levels, where both the row outlier ratio and column outlier ratio increase from 20% (as in Section 4.1) to 50%. Consequently, this leads to 25% ($50\% \times 50\%$) of the cells being replaced with outlier values. We follow the same outlier generation procedure outlined in Appendix A.3. The findings, presented in Table 6, demonstrate that our method maintains a significantly superior performance over `2D-KNN`, even under such a noisy setting.

B.2 Additional results for cell fixation experiment

Figure 6 presents the results for the cell fixation experiment on 6 additional datasets. `2D-OOB` excels in precisely detecting and correcting relevant cell outliers.

The scenario without ground-truth knowledge Following [29], we examine a situation where external information on the ground-truth annotations of outlier cells is not accessible. In this scenario, we address these outliers by substituting them with the average of other cells in the same feature column. This procedure starts by addressing cells with the lowest valuations, based on the hypothesis

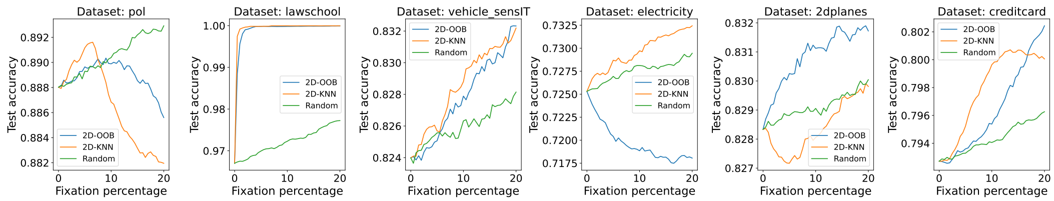

Figure 6: **Cell fixation experiment results (test accuracy curves) for** `2D-OOB`**,** `2D-KNN` **and a random baseline.** We replace cell values with ground-truth values from the cells with the lowest valuation to the highest valuation. The results from 6 datasets are displayed. We conduct 30 independent trials and report the average results. A higher curve indicates better performance. `2D-OOB` sets itself apart by its remarkable precision in detecting and rectifying relevant cell outliers.

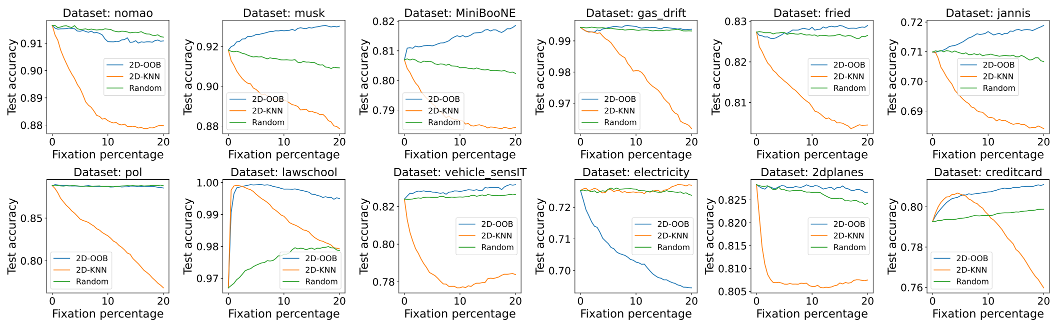

Figure 7: **Cell fixation experiment (without ground-truth knowledge) results (test accuracy curves) for** `2D-OOB`**,** `2D-KNN` **and a random baseline.** We replace cell values with column mean imputations from the cells with the lowest value to the highest value. The results from 6 datasets are displayed. We conduct 30 independent trials and report the average results. A higher curve indicates better performance.

that correcting these cells is likely to maintain or potentially improve the model's performance. As depicted in Figure 7, `2D-OOB` conforms to this expected trend, demonstrating the effectiveness of our method in joint valuation. Conversely, `2D-KNN` fails to show similar performance improvements.

B.3 Additional results for backdoor trigger experiment

We provide additional qualitative examples of backdoor trigger detection experiments in Figure 8.

B.4 Additional results for ablation study

We present the results of ablation study on the number of base learners B and feature subset ratio K/d. Specifically, we examine the AUC of the detection curve in the cell-level outlier detection experiment (refer to Table 1).

The number of base learners B When we increase the number of base learners from 500 to 3000, the detection AUC for each dataset remains unchanged, as shown in Table 7. This indicates that 1000 base learners are sufficient to get an equitable joint valuation.

Feature subset ratio K/d In addition to 0.50, We test two additional feature subset ratios 0.25 and 0.75. The results in Table 8 suggest that in general, the joint valuation capacity of our method is robust to the choice of feature subset ratio.

C Proof of Proposition 3.1

Proof. For simplicity, we denote $\phi_i^{\mathrm{OOB}}(S)$ as $\phi_i(S)$ and $\psi_{ij}^{\mathrm{2D-OOB}}$ as ψ_{ij} in the proof. With the set of subsets $\mathcal{S} := \{S \subseteq [d]\}$ and the definition `Data-OOB` $\phi_i(S)$ for all $i \in [n]$, where S is

Table 7: **Ablation results on the number of base learners B.** The cell-level outlier detection results (AUC) are examined. Increasing the number of base learners from 1000 to 3000 does not yield a notable performance improvement.

Dataset	AUC ↑ $B = 500$	$B = 1000$	$B = 3000$
lawschool	0.86 ± 0.0035	0.88 ± 0.0027	0.88 ± 0.0026
electricity	0.77 ± 0.0062	0.77 ± 0.0072	0.77 ± 0.0070
fried	0.87 ± 0.0022	0.91 ± 0.0015	0.91 ± 0.0014
2dplanes	0.87 ± 0.0016	0.87 ± 0.0015	0.87 ± 0.0015
creditcard	0.72 ± 0.0025	0.72 ± 0.0028	0.72 ± 0.0028
pol	0.78 ± 0.0022	0.82 ± 0.0014	0.82 ± 0.0014
MiniBooNE	0.77 ± 0.0042	0.77 ± 0.0058	0.77 ± 0.0058
jannis	0.78 ± 0.0045	0.83 ± 0.0042	0.83 ± 0.0039
nomao	0.79 ± 0.0018	0.79 ± 0.0021	0.79 ± 0.0020
vehicle_sensIT	0.80 ± 0.0021	0.81 ± 0.0014	0.81 ± 0.0014
gas_drift	0.86 ± 0.0007	0.86 ± 0.0010	0.86 ± 0.0010
musk	0.88 ± 0.0008	0.88 ± 0.0008	0.88 ± 0.0008
Average	0.81	**0.83**	**0.83**

Table 8: **Ablation results on feature subset ratio K/d.** The cell-level outlier detection results (AUC) are examined. Our method's joint valuation capacity remains relatively stable regardless of the selected feature subset ratio.

Dataset	AUC ↑ $K/d = 0.25$	$K/d = 0.50$	$K/d = 0.75$
lawschool	0.86± 0.0026	**0.88± 0.0027**	**0.88± 0.0024**
electricity	**0.79± 0.0070**	0.77± 0.0072	0.73± 0.0070
fried	0.86± 0.0024	**0.91± 0.0015**	0.89± 0.0007
2dplanes	0.82± 0.0015	0.87± 0.0015	**0.88± 0.0014**
creditcard	**0.73± 0.0029**	0.72± 0.0028	0.71± 0.0028
pol	0.66± 0.0031	**0.82± 0.0014**	**0.82± 0.0014**
MiniBooNE	**0.78± 0.0076**	0.77± 0.0058	0.77± 0.0049
jannis	**0.84± 0.0035**	0.83± 0.0042	0.82± 0.0043
nomao	**0.79± 0.0019**	**0.79± 0.0021**	0.78± 0.0021
vehicle_sensIT	**0.81± 0.0014**	**0.81± 0.0014**	0.80± 0.0015
gas_drift	**0.88± 0.0009**	0.86± 0.0010	0.86± 0.0009
musk	**0.89± 0.0008**	0.88± 0.0008	0.88± 0.0008
Average	0.81	**0.83**	0.82

a feature subset, we denote the cardinality of $\mathcal{S}$ as $L := |\mathcal{S}| = 2^d$. Let γ_b be a weight vector $\gamma_b := (\gamma_{b1}, \dots, \gamma_{bL})$ for all $b \in [B]$, where $\gamma_{bl} \in \{0, 1\}$ and $\gamma_{bl} = 1$ indicates the l-th subset is used in the b-th weak learner. With $\{w_b, \gamma_b, \hat{f}_b\}_{b=1}^{B}$, we can also denote the i-th `Data-OOB` on l-th feature subset S_l as

$$\phi_i(S_l) = \frac{\sum_{b=1}^{B} \mathbb{1}(w_{bi} = 0)\mathbb{1}(\gamma_{bl} = 1)T(y_i, \hat{f}_b(x_{i,S_l}))}{\sum_{b=1}^{B} \mathbb{1}(w_{bi} = 0)\mathbb{1}(\gamma_{bl} = 1)}.$$

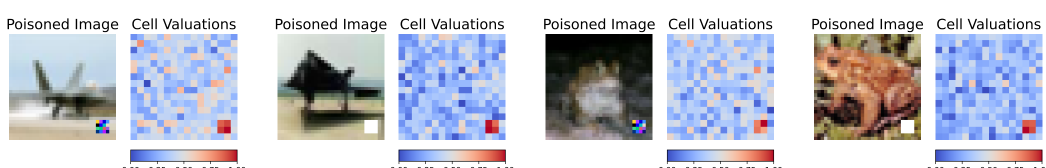

Figure 8: **Qualitative results on more datasets for the backdoor trigger detection experiment.** The first two images originate from the class "airplane" while relabeled as "automobile". The latter two images originate from the class "frog" while relabeled as "horse".

With slight abuse of notation, the formulation of `2D-OOB` in (4) can be expressed as follows.

$$\begin{aligned}
\psi_{ij} &= \frac{\sum_{l=1}^{L}\sum_{b=1}^{B}\mathbb{1}(w_{bi}=0)\mathbb{1}(\gamma_{bl}=1)\mathbb{1}(j\in S_l)T(y_i,\hat{f}_b(x_{i,S_l}))}{\sum_{l=1}^{L}\sum_{b=1}^{B}\mathbb{1}(w_{bi}=0)\mathbb{1}(\gamma_{bl}=1)\mathbb{1}(j\in S_l)} \\
&= \sum_{l=1}^{L}\mathbb{1}(j\in S_l)\frac{\sum_{b=1}^{B}\mathbb{1}(w_{bi}=0)\mathbb{1}(\gamma_{bl}=1)T(y_i,\hat{f}_b(x_{i,S_l}))}{\sum_{l=1}^{L}\sum_{b=1}^{B}\mathbb{1}(w_{bi}=0)\mathbb{1}(\gamma_{bl}=1)\mathbb{1}(j\in S_l)} \\
&= \sum_{l=1}^{L}\mathbb{1}(j\in S_l)\frac{\sum_{b=1}^{B}\mathbb{1}(w_{bi}=0)\mathbb{1}(\gamma_{bl}=1)}{\sum_{l=1}^{L}\sum_{b=1}^{B}\mathbb{1}(w_{bi}=0)\mathbb{1}(\gamma_{bl}=1)\mathbb{1}(j\in S_l)}\frac{\sum_{b=1}^{B}\mathbb{1}(w_{bi}=0)\mathbb{1}(\gamma_{bl}=1)T(y_i,\hat{f}_b(x_{i,S_l}))}{\sum_{b=1}^{B}\mathbb{1}(w_{bi}=0)\mathbb{1}(\gamma_{bl}=1)} \\
&= \sum_{l=1}^{L}\alpha_{i,j,l}\phi_i(S_l),
\end{aligned}$$

where $\alpha_{i,j,l}\propto\mathbb{1}(j\in S_l)\sum_{b=1}^{B}\mathbb{1}(w_{bi}=0)\mathbb{1}(\gamma_{bl}=1),\forall i\in[n],j\in[d],l\in[L]$ and $\sum_{l=1}^{L}\alpha_{i,j,l}=1$. Define $P_i(S_l|j\in S_l,\{w_{bi}\}_{b=1}^{B})=\alpha_{i,j,l}$, which specifies an empirical distribution of the feature subset S, conditioned on $j\in S$, in relation to the bootstrap sampling process. Here, $\mathbb{1}(j\in S_l)$ implies the distribution is conditioned on the presence of the j-th feature within the feature subset S_l. w_{bi} indicates whether the i-th sample is out-of-bag in the b-th bootstrap, and γ_{bl} indicates whether the l-th feature subset is selected in the b-th weak learner. Thus, the point mass is determined by the sampling process.

□

D Data valuation experiment

In this section, we show that `2D-OOB-data`, the marginalization of `2D-OOB`, offers an effective approach to data valuation. This serves as the basis of our enhanced performance in joint valuation.

Marginalization `2D-OOB` aims to attribute data contribution through cells. Consequently, by summing up `2D-OOB` over all columns, we can derive data contribution values. For $i\in[n]$, we define the `2D-OOB-data` ψ_i^{data} as follows.

$$\psi_i^{data} := \frac{1}{d}\sum_{j=1}^{d}\psi_{ij}^{\mathrm{2D-OOB}}, \tag{5}$$

Proof. Based on definition of `2D-OOB-Data`, for $i\in[n]$,

$$\begin{aligned}
\psi_i^{data} := \frac{1}{d}\sum_{j=1}^{d}\psi_{ij}^{\mathrm{2D-OOB}} &= \frac{1}{d}\sum_{j=1}^{d}\sum_{l=1}^{L}\alpha_{i,j,l}\phi_i^{\mathrm{OOB}}(S_l) \\
&= \sum_{l=1}^{L}\left(\frac{1}{d}\sum_{j=1}^{d}\alpha_{i,j,l}\right)\phi_i^{\mathrm{OOB}}(S_l),
\end{aligned}$$

where $\alpha_{i,j,l}$ is defined in Appendix C. We have $\sum_{l=1}^{L}(\frac{1}{d}\sum_{j=1}^{d}\alpha_{i,j,l})=\frac{1}{d}\sum_{j=1}^{d}\sum_{l=1}^{L}\alpha_{i,j,l}=1$. Denote $P_i(S_l|\{w_{bi}\}_{b=1}^{B})=\frac{1}{d}\sum_{j=1}^{d}\alpha_{i,j,l}$, which induces the empirical expectation of `Data-OOB` with respect to S_l. □

Based on discussions in Section 3.2, the marginalizations also connect with `Data-OOB`:

Proposition D.1. *For all $i\in[n]$, the marginalizations ψ_i^{data} can be expressed as follows.*

$$\psi_i^{data} = \mathbb{E}_{\hat{F}_S}[\phi_i^{OOB}(S)],$$

where the notations follow the same definitions as Proposition 3.1.

Proposition D.1 indicates 2D-OOB-data ψ_i^{data} can be expressed as the average Data-OOB value for the i-th data point. As a result, 2D-OOB-data is expected to inherit the advanced ability of Data-OOB in terms of data valuation, as will be empirically examined in the Appendix D.

Experimental setting Following the standard protocol in Kwon and Zou [22, 23] and Jiang et al. [16], we randomly select 10% of the data points and change its label to the other class. For joint valuation methods, we calculate the valuation of each cell and perform the marginalization over features to obtain the data valuation scores. For the baseline methods, we further incorporate several state-of-the-art data valuation methods including DataShapley [12], KNNShapley [15], DataBanzhaf [46], LAVA [18], and Data-OOB [23]. Implementation details are listed below. To guarantee a fair comparison, we also employ the decision tree as the base model in DataShapley and DataBanzhaf. Mislabeled data detection and data removal experiment are examined based on this setting. We adopt the same 12 datasets as outlined in Section 4.1.

Data-OOB Data-OOB involves fitting a random forest model without feature subset sampling, consisting of 1000 decision trees.

DataShapley We use a Monte Carlo-based algorithm. The Gelman-Rubin statistics is computed to determine the termination criteria of the algorithm. Following Jiang et al. [16], We adopt the threshold to be 1.05. To ensure a fair comparison with the proposed method, we employ the decision tree model for the utility evaluation.

KNNShapley We set the number of nearest neighbors to be 10% of the sample size following Jia et al. [15].

LAVA We calculate the class-wise Wasserstein distance following Just et al. [18]. The "OTDD" framework is adopted to complete the optimal transport calculation.

DataBanzhaf We adopt the implementation from Jiang et al. [16]. We employ the decision tree model and set "the number of models to train" to 1000.

D.1 Mislabeled data detection

We calculate the precision-recall curve by comparing the actual annotations, which denote whether data points are mislabeled, against the data valuation scores computed by different methods. Mislabeled data typically have a detrimental impact on model performance. Therefore, data points that receive a lower valuation score are regarded as having a higher chance of being mislabeled. We then determine AUCPR (the AUC of the precision-recall curve) as a quantitative metric to assess the detection efficacy.

As shown in Table 9, 2D-OOB-data consistently outperforms 2D-KNN-data across all datasets, suggesting its superior ability to detect mislabeled data points. It is worth noting that 2D-OOB-data's results are on par with Data-OOB, while significantly exceeding the performance of other data valuation methods. These results are in line with our theoretical analysis regarding the resemblance between Data-OOB and 2D-OOB-data. However, it is important to highlight that applying Data-OOB to the joint tasks is not feasible as mentioned earlier, underscoring the necessity for the development of 2D-OOB.

D.2 Point removal experiment

Removing low-quality data points has the potential to enhance model performance. Based on this idea, we employ the point removal experiment, a widely used benchmark in data valuation [23, 12, 22]. According to the calculated data valuation scores, we progressively remove data points from the dataset in *ascending* order. Specifically, we begin by removing the data points with the lowest data valuations. Each time we remove a datum, we fit a logistic model and use the held-out test set consisting of 3000 instances to evaluate the model performance. The expected behavior is that the model performance will improve initially as the detrimental data points are gradually eliminated from the training process. Removing an excessive number of data points may result in a drastically altered dataset. Consequently, we opt to remove the bottom 20% data points.

Table 9: **Point-level mislabeled data detection results.** AUCPR of different data valuation and (marginalized) joint valuation methods. The average and standard error of the AUCPR based on 30 independent experiments are denoted by "average ± standard error". Bold numbers denote the best method, for data valuation and joint valuation respectively. The AUCPR value for the `Random` method consistently remains at 0.5 across all datasets. `2D-OOB-data` exhibits performance comparable to `Data-OOB`, while significantly surpassing `2D-KNN-data` (the marginalization of `2D-KNN`) and all other data valuation methods.

Dataset	Data Valuation					Joint Valuation (Marginalized)	
	`KNNShapley`	`LAVA`	`DataBanzhaf`	`DataShapley`	`Data-OOB`	`2D-KNN-data`	`2D-OOB-data` (ours)
lawschool	0.66 ± 0.013	0.13 ± 0.003	0.46 ± 0.008	0.88 ± 0.007	**1.00 ± 0.000**	0.46 ± 0.011	**0.99 ± 0.002**
electricity	0.22 ± 0.008	0.11 ± 0.002	0.18 ± 0.005	0.26 ± 0.007	**0.44 ± 0.007**	0.20 ± 0.006	**0.39 ± 0.007**
fried	0.40 ± 0.014	0.11 ± 0.002	0.22 ± 0.007	0.35 ± 0.009	**0.76 ± 0.007**	0.34 ± 0.010	**0.73 ± 0.008**
2dplanes	0.46 ± 0.016	0.12 ± 0.002	0.32 ± 0.007	0.54 ± 0.009	**0.78 ± 0.008**	0.44 ± 0.011	**0.68 ± 0.010**
creditcard	0.37 ± 0.007	0.11 ± 0.003	0.16 ± 0.004	0.28 ± 0.006	**0.40 ± 0.007**	0.20 ± 0.005	**0.40 ± 0.007**
pol	0.19 ± 0.017	0.11 ± 0.002	0.37 ± 0.010	0.58 ± 0.012	**0.93 ± 0.004**	0.29 ± 0.018	**0.87 ± 0.005**
MiniBooNE	0.41 ± 0.013	0.13 ± 0.006	0.23 ± 0.007	0.41 ± 0.010	**0.78 ± 0.007**	0.36 ± 0.008	**0.78 ± 0.007**
jannis	0.20 ± 0.007	0.11 ± 0.002	0.14 ± 0.003	0.17 ± 0.005	**0.38 ± 0.010**	0.19 ± 0.006	**0.37 ± 0.010**
nomao	0.61 ± 0.012	0.14 ± 0.003	0.33 ± 0.010	0.58 ± 0.009	**0.87 ± 0.006**	0.33 ± 0.011	**0.88 ± 0.005**
vehicle_sensIT	0.22 ± 0.009	0.11 ± 0.002	0.21 ± 0.007	0.33 ± 0.011	**0.56 ± 0.010**	0.14 ± 0.005	**0.56 ± 0.010**
gas_drift	0.87 ± 0.013	0.16 ± 0.006	0.42 ± 0.009	0.75 ± 0.008	**0.98 ± 0.002**	0.88 ± 0.006	**0.98 ± 0.002**
musk	0.33 ± 0.010	0.11 ± 0.003	0.31 ± 0.007	0.47 ± 0.012	**0.85 ± 0.005**	0.21 ± 0.008	**0.85 ± 0.005**
Average	0.41	0.12	0.28	0.47	**0.73**	0.34	**0.71**

Test accuracy curves throughout the data removal process are shown for 12 datasets (Figure 9). A higher curve signifies better performance in terms of data valuation. Overall, `2D-OOB-data` demonstrates similar performance to `Data-OOB`, while significantly outperforming all other data valuation methods and the random baseline. When a few data points with poor quality are removed, the test performance of `2D-OOB-data` exhibits an evident increase. However, such a positive trend does not apply to other popular data valuation methods including `DataShapley` and `LAVA`. These findings highlight the potential of `2D-OOB-data` in selecting a subset of critical data points that can maintain model performance when the dataset is pruned.

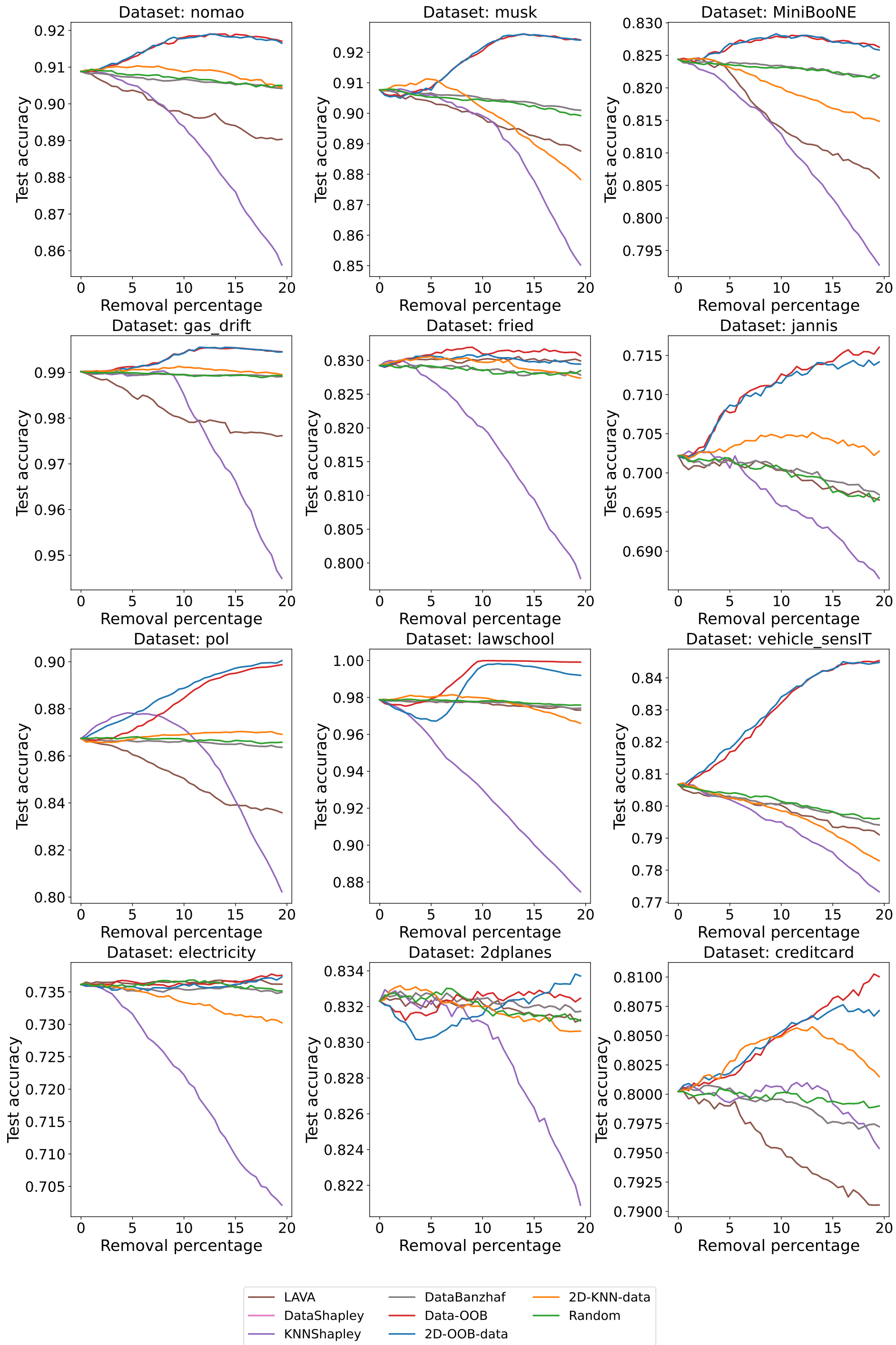

Figure 9: **Point removal experiment results (test accuracy curves) of 7 data valuation methods – `2D-OOB-data`, `2D-KNN-data`, `Data-OOB`, `LAVA`, `DataBanzhaf`, `DataShapley`, `KNNShapley` and a random baseline.** We remove data points from the lowest valuation to the highest valuation. The results from 6 binary classification datasets are displayed. For each dataset, we conduct 30 independent trials and report the average results. A higher curve indicates better performance. `2D-OOB-data` demonstrates superior ability in finding a set of helpful data points.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions have been clearly stated in the abstract and section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper has included discussion about limitations in section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper discusses theoretical interpretation in section 3.2 and the proof has been included in Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All implementation details have been included in section 4 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
    (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
    (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
    (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
    (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper uses open-source datasets, detailed in Appendix A.1, and the code repository is included in the supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Experiment settings have been clearly stated in section 4 and Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports error bars for all experiment results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides information on computer resources in section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper fully adheres to the NeurIPS Code of Ethics as outlined in the provided guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please refer to Appendix A for dataset citations.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.