Localized Data Shapley: Accelerating Valuation for Nearest Neighbor Algorithms

Guangyi Zhang
Shenzhen Technology University
zhangguangyi@sztu.edu.cn

Yanhao Wang*
East China Normal University
yhwang@dase.ecnu.edu.cn

Chengliang Chai
Beijing Institute of Technology
ccl@bit.edu.cn

Qiyu Liu
Southwest University
qyliu.cs@gmail.com

Wei Wang*
HKUST(GZ) and HKUST
weiwcs@ust.hk

Abstract

Data Shapley values provide a principled approach for quantifying the contribution of individual training examples to machine learning models. However, computing these values often requires computational complexity that is exponential in the data size, and this has led researchers to pursue efficient algorithms tailored to specific machine learning models. Building on the prior success of the Shapley valuation for K-nearest neighbor (KNN) models, in this paper, we introduce a localized data Shapley framework that significantly accelerates the valuation of data points. Our approach leverages the distance-based local structure in the data space to decompose the global valuation problem into smaller, localized computations. Our primary contribution is an efficient valuation algorithm for a threshold-based KNN variant and shows that it provides provable speedups over the baseline under mild assumptions. Extensive experiments on real-life datasets demonstrate that our methods achieve a substantial speedup compared to previous approaches.

1 Introduction

Data has emerged as the new oil of the digital economy, driving advances across various fields such as artificial intelligence, healthcare, finance, and beyond. The rapid growth in data collection has created unprecedented opportunities for building powerful machine learning models that can solve complex problems. As organizations increasingly rely on data-driven decision making, the quality and relevance of training data have become a critical factor that determines the success of machine learning applications and consequently impacts different aspects of our daily lives [1]. This paradigm shift has highlighted the fundamental importance of data as a valuable resource that requires careful management, curation, and valuation [2].

Despite the widespread recognition of the importance of data, a significant challenge remains: how to systematically, fairly, and efficiently value individual data points within large datasets. This question is increasingly relevant due to the emergence of data marketplaces, the evolution of privacy regulations, and the development of data-centric AI. Data valuation serves multiple purposes, including compensating data contributors equitably, identifying high-value data for acquisition, and removing harmful or misleading examples [3–5].

The Shapley value (SV), originating from cooperative game theory, is a principled and theoretically grounded approach to addressing the challenge of data valuation [6]. Shapley values provide a

*Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

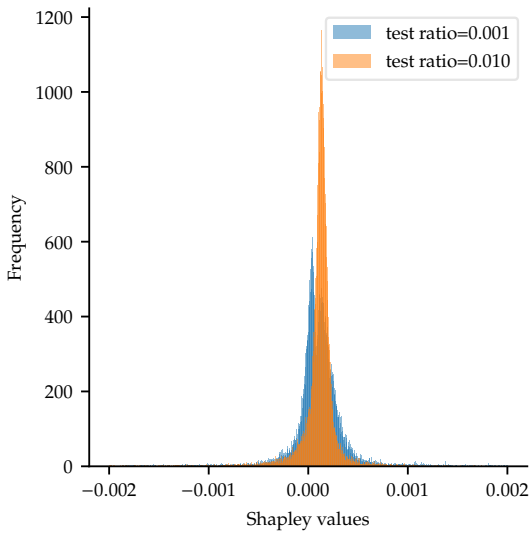

Figure 1: Illustration of the distribution of Shapley values computed for different test data ratios.

mechanism for fairly distributing the collective value of a coalition among its members. When applied to machine learning, data Shapley values [7, 8] measure the contribution of each training example to model performance by considering its marginal value across all possible subsets of the dataset. This approach satisfies important properties such as fairness, additivity, and symmetry, making it a reliable measure for data valuation tasks. Note that it differs from the attribution methods such as SHAP [9], where Shapley values are computed for each feature of a given data point.

However, the computation of data Shapley value is expensive, as it requires enumerating all possible subsets of data points. In fact, it has been shown to be $\#\mathbf{P}$-hard in certain games [10]. A recent breakthrough by Jia et al. [11] exploits the structure of unweighted K-nearest neighbor models (also known as KNN models) to efficiently compute the *exact* data Shapley values. KNN models are a family of classic machine learning models that predict the label of a data point based on the labels of its K nearest neighbors. KNN models can be adapted to modern neural networks by using their learned embeddings as the feature space. The KNN-based Shapley value (KNN-SV) has quickly become one of the leading data valuation techniques [12–15].

Although KNN-SV offers significant speed improvements over the naive approach, it still struggles with large datasets due to its linear dependence on the test data size. In KNN-SV, a value attributed to each test data point is distributed among all training examples based on their contribution to the prediction. Owing to the local nature of KNN models, an average contribution over a sufficient number of test points is required for an accurate valuation. See Fig. 1 for an illustration of the data values computed for different numbers of test points on a real dataset. When the number of test points is small, the Shapley values have a more dispersed distribution, indicating that the values are less stable and more extreme. Hence, it is desirable to use a number of test points that is of the same order of magnitude as the training set size, but this will result in quadratic overall time complexity.

In this paper, we address this key limitation of prior KNN-SV approaches and propose a more efficient algorithm. Our algorithm leverages the distance-based local structure in the data space to decompose the global valuation problem into smaller, localized computations. For a threshold-based KNN variant, our algorithm provides provable speedups over the baseline under mild assumptions. We validate the proposed algorithm through experiments on real-world datasets, demonstrating practical improvements over existing methods.

The remainder of the paper is organized as follows. We first review the preliminaries in Section 2 and present a baseline method that computes KNN-SV by recursion in Section 3. We then present our proposed methods by starting with landmark-based near neighbor search in Section 4 as a building block and introducing a fast method for threshold-based KNN in Section 5. The experimental setup and results are outlined in Section 6. We conclude the whole paper in Section 7.

2 Preliminaries

In this section, we introduce the framework for data valuation based on the Shapley value (SV) and establish our notation for applying this concept to K-nearest neighbor (KNN) models.

2.1 Cooperative Game Theory and Shapley Value

The concept of data valuation can be elegantly formalized through the lens of cooperative game theory. In this framework, we consider a collection of players who can form coalitions to generate collective utility. Formally, a cooperative game consists of a pair (I, v), where $I = \{1, \ldots, n\}$ represents the set of players and $v : 2^I \to \mathbb{R}$ is a utility function that assigns a real value to each possible coalition.

A central question in cooperative game theory concerns fair allocation: How should the total utility be distributed among individual players based on their contributions? The Shapley value, introduced by Lloyd Shapley [6], provides a time-tested solution to this problem. For each player i, the Shapley value $s(i)$ represents the average marginal contribution across all possible coalition formations, i.e.,

$$s(i) = \frac{1}{n} \sum_{S \subseteq I \setminus \{i\}} \binom{n-1}{|S|}^{-1} [v(S \cup \{i\}) - v(S)]\,. \tag{1}$$

The Shapley value *uniquely* satisfies the following four desirable properties, which makes it particularly suitable for data valuation. That is, there does not exist any other value function that can simultaneously satisfy all of them. (1) **Efficiency**: The total utility is completely distributed among all players, i.e., $v(I) = \sum_{i \in I} s(i)$. (2) **Symmetry**: Players with identical marginal contributions receive equal value, i.e., if $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \subseteq I \setminus \{i, j\}$, then $s(i) = s(j)$. (3) **Null Player**: Players who contribute nothing to any coalition receive zero value, i.e., $s(i) = 0$ if $v(S \cup \{i\}) = v(S)$ for all $S \subseteq I \setminus \{i\}$. (4) **Linearity**: Values under multiple utility functions sum to the value under the combined utility, i.e., $s_{v_1}(i) + s_{v_2}(i) = s_{v_1+v_2}(i)$ for all $i \in I$.

In the context of machine learning, we can reinterpret players as individual data points in the training set and the utility function as a performance measure of models trained on different subsets of the data. This naturally leads to a framework for quantifying the contribution of each training data point to the overall model performance.

2.2 KNN-Based Shapley Value (KNN-SV)

In this subsection, we introduce the KNN-based Shapley values with respect to two different utility functions, one for the standard KNN classifier and the other for a threshold-based KNN classifier. These utility functions are specifically targeted for a single test data point, and we conclude with a discussion on how to extend them to multiple test points, which are required for an accurate and balanced valuation.

Given a dataset D of size n, where $z = (x, y) \in D$ with $x \in \mathbb{R}^d$, $y \in \mathcal{Y}$, and $\mathcal{Y}$ is the label space, we want to compute the Shapley value $s(z \mid z_{\text{test}})$ of z with respect to a test point $(x_{\text{test}}, y_{\text{test}})$. Let the weight of z be $w(z \mid z_{\text{test}}) \in \mathbb{R}_+$ that indicates the proximity between x and x_{test} or is simply a constant in the case of an unweighted KNN. Following the formulation in [11], let v be the weighted KNN utility function. We have

$$v(S) = \sum_{i=1}^{\min(|S|,K)} w(z_{\alpha_i(S)} \mid z_{\text{test}})\, \mathbb{1}(y_{\alpha_i(S)} = y_{\text{test}}), \tag{2}$$

where $\alpha_i(S)$ is the index of the i-th closest element of S to x_{test}, and we call i the *rank* of $z_{\alpha_i(S)}$ in S. When $w(z \mid z_{\text{test}})$ is a constant, e.g., $w(z \mid z_{\text{test}}) = 1/K$, the utility function v is derived from the standard unweighted KNN classifier. On the other hand, if we write the distance as $\mathsf{d}(z, z') = \|x - x'\|$, and let the weight of z be the Gaussian kernel as

$$w(z \mid z_{\text{test}}) = \mathcal{K}(\mathsf{d}(z, z_{\text{test}})) = \exp(-\mathsf{d}(z, z_{\text{test}})^2 / 2\sigma^2),$$

where σ measures the *width* of the Gaussian kernel, the utility function v is derived from the weighted KNN classifier instead. Note that one is free to use other metric distances other than the Euclidean distance.

We further consider a similar utility function $\bar{v}$ for a variant of the standard KNN classifier, where the utility of a subset S takes into account only the nearest K points to z_{test} that are within a ball of radius τ centered at z_{test}. Formally, the utility function is defined as

$$\bar{v}(S) = \sum_{i=1}^{\min(|S_\tau|,K)} w(z_{\alpha_i(S_\tau)} \mid z_{\text{test}})\, \mathbb{1}(y_{\alpha_i(S_\tau)} = y_{\text{test}}), \tag{3}$$

where $\alpha_i(S)$ is the index of the i-th closest element of S to z_{test}, $S_\tau(z_{\text{test}}) = \{z \in S \mid \mathsf{d}(z, z_{\text{test}}) \leq \tau\}$, and we write $S_\tau = S_\tau(z_{\text{test}})$ for short. This variant is derived from a more robust threshold-based KNN classifier (TKNN) [14, 16, 17], where a point that is too far away from z_{test} always has zero influence on z_{test}. It has been shown that this variant possesses additional desirable privacy-friendly properties [14].

With the utility functions defined above, the definition of the Shapley value of a data point $z \in D$ is straightforward, following Eq. (1). Formally, the Shapley value of a data point $z \in D$ with respect to a given test point z_{test} is defined as

$$s(z \mid z_{\text{test}}) = \frac{1}{n} \sum_{S \subseteq D-z} \binom{n-1}{|S|}^{-1} \left[v(S+z) - v(S)\right], \tag{4}$$

where we write $D - z = D \setminus \{z\}$ and $S + z = S \cup \{z\}$ for convenience.

In practice, accurately measuring data values requires multiple test points, which typically increases proportionally with the size of the dataset, n. Suppose that there are n_{test} test points in D_{test}, and the data Shapley value of a data point z can be naturally extended as the average over all test points, i.e.,

$$s(z) = \sum_{z_{\text{test}} \in D_{\text{test}}} s(z \mid z_{\text{test}})/n_{\text{test}}. \tag{5}$$

3 Baseline: KNN-SV by Recursion

In this section, we introduce the analytical solution to the data Shapley values for KNN models, introduced in [11], and extend it to the utility functions in Eqs. (2) and (3). This results in a dramatic improvement in time complexity, from $O(2^n)$ to $O(n \log n)$ for a single test point, over the naive approach that enumerates all possible subsets S of the dataset D.

We consider a fixed test point z_{test} throughout this section. Given a subset $S \subseteq D$, recall that $\alpha_i(S)$ is the index of the i-th closest element of S to z_{test}. When the context is clear, for simplicity, we write $z_{\alpha_i(D)}$ as z_i and $w(z_{\alpha_i(D)} \mid z_{\text{test}})$ as w_i. Similarly, we denote by s_i the Shapley value $s(z_i \mid z_{\text{test}})$ for the data point z_i.

We first restate a known result about the pairwise difference of the KNN Shapley values in the following lemma.

Lemma 1 (Jia et al. [11])**.** *Fixing a test point, for any i, j, we have*

$$s_i - s_j = \frac{1}{n-1} \sum_{S \subseteq D - z_i - z_j} \frac{v(S + z_i) - v(S + z_j)}{\binom{n-2}{|S|}}.$$

Based on Lemma 1, we can develop a recursive formula for s_i in the following theorem.

Theorem 2. *Fixing a test point, for any $i < n$, we have*

$$s_i = s_{i+1} + \min(K, i) \frac{w_i \mathbb{1}(y_i = y_{test}) - w_{i+1} \mathbb{1}(y_{i+1} = y_{test})}{i} \quad \textit{and} \quad s_n = \frac{K}{n} w_n \mathbb{1}(y_n = y_{test}).$$

See proof in Appendix A. In summary, in order to compute the Shapley values of all data points in the dataset D, we start with s_n for the farthest data point and then iteratively apply the recursive formula in Theorem 2 to compute the values of $s_{n-1}, \ldots, s_1$, in decreasing order of their distance to the test point, one data point at a time.

It is easy to see that the recursive formula formed by Theorem 2 also works for threshold-based KNN models. The only difference is that the recursion is applied to the set of data points that are within the radius τ of the test point, instead of the entire dataset D.

Time Complexity. We have described a recursive formula for the data Shapley values when there is only a single test point. The calculation requires no more than one sorting of the data points by their distance to the test point, which takes $O(dn + n \log n)$ time. This is a drastic improvement over the naive approach that enumerates all possible subsets S of D, whose time complexity is $O(2^n)$.

However, when considering multiple test points, whose size n_{test} is often in proportion to the size of the dataset n, i.e., $n_{\text{test}} = \Omega(n)$, one sorting for each test point amounts to a time complexity of $O(dn_{\text{test}}n + n_{\text{test}}n \log n)$, which becomes quadratic in the size of the dataset. This is clearly too slow for large-scale applications, especially when n_{test} is large.

4 Landmark-based Near Neighbor Search

In this section, we introduce a core building block for our proposed methods, namely landmark-based near neighbor search, which allows to effectively and efficiently shrink the search space of the near neighbors in data Shapley computation. Unlike other near neighbor search methods, our method exhibits several merits: It is inexpensive in indexing, simple to implement, capable of providing a lower bound of the distance to the query point for unvisited points, and, last but not least, amenable to analysis as we will see in the subsequent sections.

The main idea is to pick an arbitrary point z_{mark} as the *landmark* point, and sort all data points $D \cup D_{\text{test}}$ by their distances from z_{mark} in ascending order. Denote by $r_{\text{mark}}(z)$ the rank of a point z in the sorted list and by $B_i(z_{\text{test}}) \subseteq D$ the set of points in D whose differences in rank from that of z_{test} are within i, i.e.,

$$B_i(z_{\text{test}}) = \{z \in D \mid |r_{\text{mark}}(z) - r_{\text{mark}}(z_{\text{test}})| \leq i\}.$$

We call i the *length* of the ball $B_i(z_{\text{test}})$. We also distinguish the left and right halves of the ball, i.e., $B_i^-(z_{\text{test}})$ and $B_i^+(z_{\text{test}})$, where

$$B_i^-(z_{\text{test}}) = \{z \in D \mid r_{\text{mark}}(z_{\text{test}}) - i \leq r_{\text{mark}}(z) < r_{\text{mark}}(z_{\text{test}})\},$$

and

$$B_i^+(z_{\text{test}}) = \{z \in D \mid r_{\text{mark}}(z_{\text{test}}) < r_{\text{mark}}(z) \leq r_{\text{mark}}(z_{\text{test}}) + i\}.$$

The motivation of using landmark points is that for any test point z_{test}, a data point z around z_{test} along the sorted list is likely to be its near neighbor. More importantly, it is possible to derive a lower bound of $\mathsf{d}(z, z_{\text{test}})$ by only considering the distances to the landmark point z_{mark}. That is, by triangle inequality,

$$\mathsf{d}(z, z_{\text{test}}) \geq |\mathsf{d}(z, z_{\text{mark}}) - \mathsf{d}(z_{\text{mark}}, z_{\text{test}})|. \tag{6}$$

This is particularly useful. For example, as the ball $B_i^-(z_{\text{test}})$ expands, the distance $\mathsf{d}(z, z_{\text{mark}})$ is non-increasing, and thus $\mathsf{d}(z, z_{\text{test}})$ is also non-decreasing, which gives us valuable information about $\mathsf{d}(z, z_{\text{test}})$ for any $z \in B_i^-(z_{\text{test}})$ even without actually visiting them. The case for $B_i^+(z_{\text{test}})$ is similar.

4.1 Optimized Landmark Selection

The tightness of the lower bound of $\mathsf{d}(z, z_{\text{test}})$ in Eq. (6) is highly dependent on the distance $\mathsf{d}(z_{\text{test}}, z_{\text{mark}})$. That is, the quality of our bounds can be significantly improved if the test points are close to the chosen landmark point. Therefore, we propose to strategically select multiple landmark points and assign each test point to its nearest landmark point.

More specifically, we propose to select n_L landmark points $D_{\text{mark}} \subseteq D_{\text{test}}$ with a goal of minimizing the maximum distance of any test point to its nearest landmark point, i.e.,

$$\min_{D_{\text{mark}} \subseteq D_{\text{test}}} \max_{z_{\text{test}} \in D_{\text{test}}} \min_{z_{\text{mark}} \in D_{\text{mark}}} \mathsf{d}(z_{\text{test}}, z_{\text{mark}}).$$

The landmark points D_{mark} defined above align exactly with our intention and naturally create a clustering structure through their associated regions of influence, where each region contains points closer to its landmark than to any other landmark. This is also known as the *Voronoi* partition induced by the landmark points. In subsequent sections, we will show that landmark-induced partitions can effectively reveal inherent structures in the data.

The above optimization problem turns out to be the well-studied metric k-center problem, which is known to be $\mathbf{NP}$-hard. Worse still, this problem is impossible to approximate within a factor of 2, unless $\mathbf{P} = \mathbf{NP}$ [18].[2]

Fortunately, there exists a simple greedy algorithm that achieves the best possible approximation ratio of 2 in the worst case. This algorithm is called farthest-first traversal (FFT) [19], which as the name suggests, starts from an arbitrary point and iteratively selects the farthest point from the current set of landmark points until n_L landmark points are selected. In other words, the next chosen landmark point z maximizes the distance against the current set of landmark points, i.e.,

$$\mathsf{d}(z, D_{\text{mark}}) = \min_{z_{\text{mark}} \in D_{\text{mark}}} \mathsf{d}(z, z_{\text{mark}}).$$

A straightforward implementation of the FFT algorithm takes $\mathcal{O}(n_{\text{test}} n_L^2 d)$ time, which may be too slow when n_L is large.

To speed up the FFT algorithm, we first notice that $\mathsf{d}(z, D_{\text{mark}})$ is non-increasing as D_{mark} grows. Thus, we can reduce a factor of n_L in the running time by bookkeeping $\mathsf{d}(z, D_{\text{mark}})$ for each point z and updating it upon the selection of every new landmark point z_{mark}. That is,

$$\mathsf{d}(z, D_{\text{mark}}^{(i)}) = \min\{\mathsf{d}(z, z_{\text{mark}}),\, \mathsf{d}(z, D_{\text{mark}}^{(i)})\},$$

where $D_{\text{mark}}^{(i)}$ is the set of landmark points selected before the i-th iteration and $D_{\text{mark}}^{(i+1)} = D_{\text{mark}}^{(i)} \cup \{z_{\text{mark}}\}$. This avoids scanning the entire set of landmark points when computing $\mathsf{d}(z, D_{\text{mark}})$ for each point z. Based on this observation, the running time is reduced to $\mathcal{O}(n_{\text{test}} n_L d)$.

5 Fast Data Shapley Value Computation for Threshold-based KNN

As discussed previously, given n_{test} test points, the baseline approach in Section 3 computes the data Shapley values for all test points in $\mathcal{O}(dn_{\text{test}}n + n_{\text{test}}n \log n)$ time. In this section, we propose a fast algorithm to compute the data Shapley values for threshold-based KNN in provably less time.

Our main idea is to exploit the truncated structure of the threshold-based KNN classifier. We leverage the landmark-based near neighbor search introduced in Section 4 to shrink the search space of the near neighbors, and it turns out that the size of the search space for any test point can be effectively bounded when the dataset exhibits a stable clustering structure. In the remainder of this section, we first describe the proposed algorithm and then analyze its theoretical properties.

5.1 Algorithm Description

We have introduced the landmark-based near neighbor search in Section 4, and in this section, we show how to utilize it to compute data Shapley values efficiently.

First of all, we need to slightly adjust the FFT algorithm for our purpose. We denote by $C(z_{\text{mark}})$ the cluster of points that are closer to z_{mark} than any other landmark point. The radius of $C(z_{\text{mark}})$ is the maximum distance from z_{mark} to any point in $C(z_{\text{mark}})$. In addition, we let $\tau_{D_{\text{mark}}}$ be the maximum radius of all clusters induced by the landmark points in D_{mark}. Note that $\tau_{D_{\text{mark}}}$ is non-increasing as the number of landmark points increases. We make two adjustments to the original FFT algorithm. First, we run FFT over $D_{\text{test}} \cup D$ instead of D_{test}. Second, we require that $\tau_{D_{\text{mark}}} \leq \tau$, which can be easily achieved by continuing the iterative process of FFT until the condition is met. This also means that it is not required to specify the number of landmark points in advance.

Since the threshold-based KNN classifier only considers data points within a distance of τ from each test point, an intuitive idea is to explore the search space provided by the landmark-based near neighbor search until we can certify that all remaining unvisited points are beyond a distance of τ from the test point. More specifically, we gradually expand the left half of the ball $B_i(z_{\text{test}})$ centered at each test point z_{test} until the first point z such that

$$|\mathsf{d}(z, z_{\text{mark}}) - \mathsf{d}(z_{\text{mark}}, z_{\text{test}})| > \tau.$$

This immediately implies that the distance $\mathsf{d}(z, z_{\text{test}})$ from any z to z_{test} outside the left half is at least τ. We also apply this process to the right half of the ball. Afterwards, we collect into S all points in

[2] Recall that an algorithm is called γ-approximation if it returns solutions that in the worst case have cost no more than γ times than the cost of the optimum solution.

Algorithm 1: Fast Data Shapley Value Computation for Threshold-based KNN

Input: Integer K, radius τ, datasets D and D_{test}
Output: Data Shapley values $\{s(z)\}_{z \in D}$

Select landmark points D_{mark} from $D \cup D_{\text{test}}$ by FFT such that $\tau_{D_{\text{mark}}} \leq \tau$;
Assign each point in D_{test} to its nearest point in D_{mark};
for $z_{mark} \in D_{mark}$ **do**
    Sort $D \cup D_{\text{test}}$ by their distances to z_{mark} in ascending order;
Initialize $s(z)$ with a default value of 0 for each $z \in D$;
for $z_{test} \in D_{test}$ **do**
    Let z_{mark} be the landmark point associated with z_{test};
    Expand the ball $B^-(z_{\text{test}})$ until the first point z such that $\mathsf{d}(z_{\text{mark}}, z_{\text{test}}) - \mathsf{d}(z, z_{\text{mark}}) > \tau$;
    Expand the ball $B^+(z_{\text{test}})$ until the first point z such that $\mathsf{d}(z, z_{\text{mark}}) - \mathsf{d}(z_{\text{mark}}, z_{\text{test}}) > \tau$;
    $S \leftarrow \{z \in B^-(z_{\text{test}}) \cup B^+(z_{\text{test}}) \mid \mathsf{d}(z, z_{\text{test}}) \leq \tau\}$;
    Let $z_1, \ldots, z_{|S|}$ be the points in S sorted by their distances to z_{test} in ascending order;
    $\phi_{z_{|S|}} \leftarrow \frac{K}{|S|} w(z_{|S|} \mid z_{\text{test}}) \mathbb{1}(y_{z_{|S|}} = y_{z_{\text{test}}})$;
    for $i = |S| - 1, \ldots, 1$ **do**
        $\phi_{z_i} \leftarrow \phi_{z_{i+1}} + \frac{\min(K,i)}{i} \left(w'_i - w'_{i+1}\right)$, where $w'_i = w(z_i \mid z_{\text{test}}) \mathbb{1}(y_{z_i} = y_{z_{\text{test}}})$;
    $s(z) \leftarrow s(z) + \phi_z$ for each $z \in S$;
$s(z) \leftarrow s(z)/n_{\text{test}}$ for each $z \in D$;
return $\{s(z)\}_{z \in D}$;

the ball that are within a distance of τ from z_{test}. Finally, we compute the data Shapley values for all the points in S using the recursive formula in Theorem 2. The detailed procedure is described in Algorithm 1.

To show that Algorithm 1 can be provably faster than the baseline approach, we need to show that for any test point, the total number of points visited in the ball is strictly smaller than n. In the next subsection, we show that this is indeed the case under mild conditions.

5.2 Perturbation Resilience

Before we analyze the theoretical guarantee of Algorithm 1, we introduce a technical notion of perturbation resilience to help us precisely characterize the structure that exists in a dataset.

Worst-case analysis has been criticized for its over-pessimism and conservatism that fail to capture the real performance of many algorithms in practice. In particular, it ignores the structure that exists in real-world datasets. Therefore, in recent years, there is an active trend of *beyond worst-case analysis* in the literature to provide more realistic performance guarantees [20]. One notable example is the notion of *perturbation resilience* proposed by Bilu and Linial [21], which describes the stability of the clustering structure of a dataset under small perturbations.

We first define the notion of perturbation, which distorts the original distance function $\mathsf{d}(\cdot, \cdot)$ by a factor of at most $\xi \geq 1$.

Definition 1 (Perturbation)**.** *Given a clustering instance (D, d), a ξ-perturbation of d is a new distance function d' such that $\mathsf{d}(x, y) \leq \mathsf{d}'(x, y) \leq \xi \mathsf{d}(x, y)$ for all $x, y \in D$.*

Note that the ξ-perturbation of d may no longer be a metric distance. As ξ increases, a larger perturbation is allowed. Then, we say that a clustering instance is perturbation resilient if its optimal clustering remains unchanged up to such a small perturbation of d.

Definition 2 (Perturbation Resilience (PR))**.** *A clustering instance (D, d) is said to be ξ-perturbation resilient if the optimal clustering remains unchanged up to a ξ-perturbation of d.*

Here, the optimal clustering depends on the specific clustering objective. For example, in the metric k-center clustering, the optimal clustering is the one that minimizes the maximum cluster radius for a specific number of clusters k.

Intuitively, a clustering instance is ξ-perturbation resilient if the optimal clustering is stable. Conversely, if the optimal clustering can easily change under a small perturbation, then it is less meaningful to study the clustering structure of the dataset in the first place. Note that perturbation resilience does not mean that the clustering will necessarily become easy. In fact, as indicated in [22], no polynomial-time algorithm can solve the metric k-center clustering problem for ξ-perturbation resilient instances with any $\xi < 2$ unless $\mathbf{RP} = \mathbf{NP}$. Although the worst-case approximation ratio of FFT for metric k-center clustering is at least 2, it turns out that it can recover the optimal clusters in a clustering instance if it is 2-perturbation resilient.

Theorem 3 (Balcan et al. [22])**.** *Let (D, d) be a clustering instance. If (D, d) is 2-perturbation resilient, then* FFT *recovers the optimal clusters of (D, d).*

Furthermore, Theorem 3 can be extended to show that any γ-approximation solution can optimally recover the clusters under any γ-perturbation of d. Theorem 3 opens up a new avenue for analyzing the performance of our landmark-based near neighbor search, and in turn, of Algorithm 1.

5.3 Theoretical Analysis

In this subsection, we show that Algorithm 1 is provably faster than the baseline approach if the dataset exhibits perturbation resilience. We first point out that our landmark-based near neighbor search can be seen as a soft version of the k-center clustering. Then, we show that perturbation resilience provides sufficient separation between clusters to restrict the computation of the data Shapley values within each cluster, which leads to a provable speedup.

If we can treat landmark points as the selected centers of the k-center clustering, then they virtually partition the dataset into n_L clusters. Given a landmark point z_{mark} and the sorted list of $D \cup D_{\text{test}}$ by their distances to z_{mark} in ascending order, we hope that the points from its cluster $C(z_{\text{mark}})$ are ranked before the points in other clusters.

The next challenge is how to ensure that every point that is within a distance of τ from z_{test} is in the same cluster as z_{test}. Note that this is non-trivial because z_{test} may not be a landmark point (i.e., a center), and there is no guarantee that any two arbitrary points with a distance less than τ will be in the same cluster, no matter what size the cluster radius is. For example, consider two clusters that overlap with each other and two points that lie in the overlapping region. This is crucial for the algorithm to restrict the computation of every test point within the cluster to which it belongs.

We discover that perturbation resilience provides sufficient separation between clusters to address the above challenges. Formally, we prove the following theorem.

Theorem 4. *Let $(D \cup D_{test}, \mathsf{d})$ be a clustering instance with $n_{test} = \mathcal{O}(n)$. If it is 3-perturbation resilient for metric k-center with respect to a cluster number k^* and a maximum cluster radius $\tau^* \geq \tau$, then Algorithm 1 with $n_L = k^*$ returns the exact data Shapley values of each point in D in $\mathcal{O}(n_L n(d + \log n) + n_{test} s(d + \log s))$ time, where s is the size of the largest optimal cluster.*

See proof in Appendix A. When $k^* < n_L$, we will show in Appendix A that the same time complexity as Theorem 4 still holds, albeit with a slightly larger 4-perturbation. The above results show that Algorithm 1 provides a provable speed-up for a wide range of numbers of clusters n_L. For example, let $n_{\text{test}} = \Omega(n)$, and we can expect a running time of $\mathcal{O}(n^{1.5}(d + \log n))$ with $n_L = \sqrt{n}$ when the sizes of the clusters are comparable. This improves over the previous $\mathcal{O}(n^2(d + \log n))$ time.

6 Experiments

In this section, we evaluate the performance of the proposed methods on synthetic and real-world datasets. We aim to answer the following two research questions: (1) What are the effects of the parameters and design choices on the performance of the fast algorithm for threshold-based KNN (Algorithm 1)? (Section 6.1) (2) How much faster do the proposed methods compute the data Shapley values compared to the baseline approach in Section 3? (Section 6.2) Our source code is published for reproducibility.[3]

Datasets. We used both synthetic and real-world datasets in the experiments. The former allows us to experiment freely with a wide range of data characteristics. We select a collection of real-world

[3] `https://github.com/Guangyi-Zhang/tknn-data-shapley`

datasets as listed in Table A1. The dataset size $|D|$ ranges from 10 K to 1 M, and we set $|D_{\text{test}}|$ to be 0.2%-1% of $|D|$. Thus, the total size of $|D| \cdot |D_{\text{test}}|$ is up to the order of 10^{10}.

Experimental Environment. All algorithms were implemented in Python 3.11. All experiments were carried out on a Linux server equipped with 64 CPUs of Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60 GHz and 1511 GB RAM.

6.1 Effects of Parameters and Design Choices

In this subsection, we investigate the effects of the parameters and design choices on the performance of Algorithm 1. The default values for the parameters are $K = 5$, $n_L = 50$, $\tau/d = 0.2$, $\sigma = 0.1$, and FFT for landmark selection. Note that since the data features are normalized to be in the range of $[0, 1]$, the value of τ/d measures the maximum proportion of features of a data point that can deviate arbitrarily from a test point, while the point remains to be considered as a neighbor in threshold-based KNN models.

(a) Effect of n_L

(b) Effect of τ/d

Figure 2: Effect of parameters n_L and τ/d on the performance of Algorithm 1.

Effect of the Number of Landmark Points n_L. We vary the number of landmark points n_L from 1 to 100, and plot the corresponding running time on the 2dplanes dataset in Fig. 2a. As shown, the running time of Fig. 2a first decreases and then increases when more landmark points are selected. This is because every test point can be assigned to a landmark point that is closer to it, and the total number of data points encountered during the ball expansion for a test point is effectively reduced when there are more landmark points. However, the running time increases later when this benefit is outweighed by the high overhead of preprocessing the landmark points. Recall that it is necessary to sort all the data points in $D \cup D_{\text{test}}$ for each landmark point.

Effect of the Ratio of Radius τ and Dimension d. We vary the value of τ/d from 0 to 0.5, and plot the corresponding running time on the 2dplanes dataset in Fig. 2b. It is expected that the running time increases as the radius τ grows, because a larger τ allows more data points to be considered as neighbors of a test point. But fortunately, a small τ is enough in practice due to the localized nature of KNN models.

6.2 Performance Comparison

In this subsection, we compare the performance of the proposed methods with the baseline approach in Section 3 and the state-of-the-art threshold-based approach TNN by Wang et al. [14]. Note that the baseline approach has to sort all the data points in D for each test point, regardless of the radius value τ. We report the average running time and standard deviation over three runs in Table 1, where the fastest running time is highlighted in bold. We also compare their performance for a popular downstream task of mislabel detection; see Appendix C.1 for more details. The default values for the parameters are $K = 5$, $n_L = 50$, and FFT for landmark selection.

Based on the results in Table 1, we observe that our proposed methods consistently outperform the baseline approach across all datasets. The speedup is substantial, with our fastest method running up to 25× faster than the baseline and 11× faster than TNN. The performance advantage of our methods becomes more pronounced as the dataset size increases.

Table 1: Running time (in seconds) comparison of different methods.

Dataset	Baseline	Algorithm 1			TNN
	N/A	$\tau/d = 0.05$	$\tau/d = 0.1$	$\tau/d = 0.2$	$\tau/d = 0.05$
magic	7.05 ± 6.48	2.68 ± 0.02	3.17 ± 0.05	4.70 ± 0.21	**1.29** ± 0.06
2dplanes	85.83 ± 3.00	**8.44** ± 0.15	11.59 ± 0.24	26.75 ± 0.20	27.28 ± 25.13
cifar10	148.60 ± 0.61	75.83 ± 0.50	86.57 ± 2.35	144.05 ± 1.67	**48.16** ± 44.94
dota2	445.44 ± 4.89	249.36 ± 1.74	267.76 ± 7.24	436.65 ± 6.32	**142.52** ± 132.03
skin	2035.00 ± 142.90	**156.88** ± 1.90	159.37 ± 126.42	243.79 ± 207.22	228.59 ± 2.52
covtype	10324.36 ± 131.70	**1918.74** ± 16.94	4135.27 ± 27.07	9107.61 ± 323.44	5009.56 ± 10.33
emnist	6090.47 ± 19.04	**2508.41** ± 73.36	3558.89 ± 130.26	5988.33 ± 384.80	3221.94 ± 7.48
poker	12289.43 ± 41.26	**488.95** ± 4.73	983.97 ± 0.91	3012.10 ± 46.34	5542.69 ± 23.88

Algorithm 1 excels on lower-dimensional datasets (see a controlled experiment in Appendix C.2), such as 2dplanes and poker, where it achieves the fastest runtime when the radius is small. As expected, the performance of Algorithm 1 degrades as the radius increases. However, the running time never exceeds that of the baseline approach, even when τ/d is as large as 0.2. Actually, its worst-case running time is about the same as that of the baseline approach, with the negligible overhead of preprocessing a few landmark points. When the dataset shows well-defined clusters, its running time can be provably better. These results confirm that the proposed algorithm significantly accelerates the computation of data Shapley values, while retaining a robust worst-case running time that is at least as fast as the baseline approach.

7 Conclusion

In this paper, we addressed the challenge of efficiently computing data Shapley values for nearest neighbor algorithms. We leveraged the distance-based local structure in the data space to decompose the global valuation problem into smaller, localized computations. For threshold-based KNN classification, we proposed an algorithm with provable speedups under mild assumptions compared to existing methods. Our comprehensive empirical evaluation on synthetic and real-world datasets verifies the significant speed-ups offered by our proposed methods.

We acknowledge several limitations of our work. The perturbation resilience condition is hard to verify in practice. The landmark-based near neighbor search may suffer from the curse of dimensionality. Potential future research directions include extending our approach to regression tasks, exploring alternative utility functions, and investigating fast algorithms for more general KNN models.

Acknowledgments and Disclosure of Funding

G. Zhang was supported by the Guangdong Provincial College Youth Innovative Talent Project (Grant No. 2025KQNCX075), Natural Science Foundation of Top Talent of SZTU (Grant No. GDRC202520), SZTU University Research Project (No. 20251061020002). Y. Wang was supported by the National Natural Science Foundation of China (Grant No. 62202169). C. Chai was supported by the NSF of China (62472031), the National Key Research and Development Program of China (2024YFC3308200), Beijing Nova Program, CCF-Baidu Open Fund (CCF-Baidu202402). Q. Liu was supported by fundamental research funds for the central universities of Ministry of Education of China (SWU-KR24043). W. Wang was supported by Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No.2023B1212010007, SL2023A03J00934), Guangzhou Municipal Science and Technology Project (No. 2023A03J0003, 2023A03J0013 and 2024A03J0621).

References

[1] Weixin Liang, Girmaw Abebe Tadesse, Daniel Ho, Li Fei-Fei, Matei Zaharia, Ce Zhang, and James Zou. Advances, challenges and opportunities in creating data for trustworthy AI. *Nat. Mach. Intell.*, 4(8):669–677, 2022.

[2] Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning: "ingredients", strategies, and open challenges. In *Proceedings of the Thirty-First*

International Joint Conference on Artificial Intelligence, IJCAI-22, pages 5607–5614. IJCAI Organization, 2022.

[3] Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: a survey. *Mach. Learn.*, 113(5):2351–2403, 2024.

[4] Kevin Fu Jiang, Weixin Liang, James Y. Zou, and Yongchan Kwon. OpenDataVal: a unified benchmark for data valuation. *Advances in Neural Information Processing Systems*, 36:28624–28647, 2023.

[5] Mark Mazumder, Colby R. Banbury, Xiaozhe Yao, et al. DataPerf: Benchmarks for data-centric AI development. *Advances in Neural Information Processing Systems*, 36:5320–5347, 2023.

[6] Lloyd S. Shapley. A value for n-person games. In *Contributions to the Theory of Games, Volume II*, pages 307–318. Princeton University Press, Princeton, NJ, USA, 1953.

[7] Amirata Ghorbani and James Y. Zou. Data Shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[8] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the Shapley value. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019.

[9] Scott Lundberg and Su-in Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774, 2017.

[10] Xiaotie Deng and Christos H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994.

[11] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gürel, Bo Li, Ce Zhang, Costas J. Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *Proc. VLDB Endow.*, 12(11):1610–1623, 2019.

[12] Konstantin D. Pandl, Fabian Feiland, Scott Thiebes, and Ali Sunyaev. Trustworthy machine learning for health care: scalable data valuation with the Shapley value. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 47–57. Association for Computing Machinery, 2021.

[13] Jiachen T. Wang and Ruoxi Jia. A note on "efficient task-specific data valuation for nearest neighbor algorithms". *arXiv:2304.04258*, 2023.

[14] Jiachen T. Wang, Yuqing Zhu, Yu-Xiang Wang, Ruoxi Jia, and Prateek Mittal. A privacy-friendly approach to data valuation. *Advances in Neural Information Processing Systems*, 36: 60429–60467, 2023.

[15] Jiachen T. Wang, Prateek Mittal, and Ruoxi Jia. Efficient data Shapley for weighted nearest neighbor algorithms. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pages 2557–2565. PMLR, 2024.

[16] Jon L. Bentley. A survey of techniques for fixed radius near neighbor searching. Technical Report STAN-CS-75-513, Stanford University, Stanford, CA, USA, 1975.

[17] Yuqing Zhu, Xuandong Zhao, Chuan Guo, and Yu-Xiang Wang. Private prediction strikes back! Private kernelized nearest neighbors with individual Rényi filter. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 2586–2596. PMLR, 2023.

[18] Dorit S. Hochbaum. Approximation algorithms for NP-hard problems. *SIGACT News*, 28(2): 40–52, 1997.

[19] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985.

[20] Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, Cambridge, UK, 2021.

[21] Yonatan Bilu and Nathan Linial. Are stable instances easy? *Comb. Probab. Comput.*, 21(5): 643–660, 2012.

[22] Maria-Florina Balcan, Nika Haghtalab, and Colin White. k-center clustering under perturbation resilience. *ACM Trans. Algorithms*, 16(2):22:1–22:39, 2020.

[23] Chandra Chekuri and Shalmoli Gupta. Perturbation resilient clustering for k-center and related problems via LP relaxations. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, pages 9:1–9:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[24] Alvin E. Roth. *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press, Cambridge, UK, 1988.

[25] S. Shaheen Fatima, Michael J. Wooldridge, and Nicholas R. Jennings. A linear approximation method for the Shapley value. *Artif. Intell.*, 172(14):1673–1699, 2008.

[26] Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the Shapley value based on sampling. *Comput. Oper. Res.*, 36(5):1726–1730, 2009.

[27] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. Bounding the estimation error of sampling-based Shapley value approximation. *arXiv:1306.4265*, 2013.

[28] David Liben-Nowell, Alexa Sharp, Tom Wexler, and Kevin M. Woods. Computing Shapley value in supermodular coalitional games. In *Computing and Combinatorics - 18th Annual International Conference, COCOON 2012, Sydney, Australia, August 20-22, 2012. Proceedings*, pages 568–579. Springer, 2012.

[29] Yongchan Kwon and James Zou. Beta Shapley: a unified and noise-reduced data valuation framework for machine learning. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 8780–8802. PMLR, 2022.

[30] Jiachen T. Wang and Ruoxi Jia. Data Banzhaf: A robust data valuation framework for machine learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 6388–6421. PMLR, 2023.

[31] Tom Yan and Ariel D Procaccia. If you like Shapley then you'll love the core. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):5751–5759, 2021.

[32] 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.

[33] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In *Proceedings of the 39th International Conference on Machine Learning*, pages 9525–9587. PMLR, 2022.

[34] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

[35] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

[36] Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8179–8186, 2022.

[37] R. Dennis Cook and Sanford Weisberg. *Residuals and Influence in Regression*. Chapman and Hall, New York, NY, USA, 1982.

[38] Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. *Inf. Process. Lett.*, 112(1-2):49–54, 2012.

[39] Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. Algorithms for stable and perturbation-resilient problems. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 438–451. Association for Computing Machinery, 2017.

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 main claims made in the abstract and introduction accurately specify the contributions and scope of the paper.

   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: We discuss the limitations of the work in Section 7.

   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: We provide the full set of assumptions and a complete (and correct) proof for each theoretical result in the paper.

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: We provide all the information needed to reproduce the main experimental results of the paper, including the code and data.

   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: We provide open access to the code, with sufficient instructions to faithfully reproduce the main experimental results. All the data are publicly available.

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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details necessary to understand the results in the paper.

   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: We have reported standard deviation of the key results in Table 1.

   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: We provide sufficient information on the computer resources needed to reproduce the experiments in Section 6.

   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: We conform, in every respect, with the NeurIPS Code of Ethics.

   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: [Yes]

    Justification: The proposed data valuation framework can be used to value data in a more transparent and fair way, which naturally leads to positive societal impacts. Our framework is not tied to any particular application, and there does not exist a direct path to any negative applications to the best of our knowledge.

    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: We do not release any data or models that have a high risk for misuse.

    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: We properly credit and respect the license and terms of use of the assets used in the paper.

    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: [Yes]

    Justification: We provide a detailed description of the code released in the paper.

    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: We do not involve crowdsourcing or 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: We do not involve 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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We do not use LLMs as an important, original, or non-standard component of the core methods in this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Missing Proofs

Theorem 2. *Fixing a test point, for any $i < n$, we have*

$$s_i = s_{i+1} + \min(K, i)\frac{w_i \mathbb{1}(y_i = y_{test}) - w_{i+1}\mathbb{1}(y_{i+1} = y_{test})}{i} \quad \textit{and} \quad s_n = \frac{K}{n} w_n \mathbb{1}(y_n = y_{test}).$$

Proof of Theorem 2. For s_n, i.e., the Shapley value of the farthest data point, it is easy to see that

$$\begin{aligned} s_n &= \frac{1}{n}\sum_{k=0}^{K-1} \frac{1}{\binom{n-1}{k}} \sum_{S \subseteq D - z_n, |S|=k} v(S + z_n) - v(S) \\ &= \frac{1}{n}\sum_{k=0}^{K-1} \frac{1}{\binom{n-1}{k}} \sum_{S \subseteq D - z_n, |S|=k} w_n \mathbb{1}(y_n = y_{\text{test}}) \\ &= \frac{K}{n} w_n \mathbb{1}(y_n = y_{\text{test}}). \end{aligned} \tag{7}$$

Since for any S such that $|S| \geq k$, z_n will not make it into the top-k and thus $v(S + z_n) - v(S) = 0$.

Suppose that we have already computed $s_{i+1}, \ldots, s_n$, and now we want to compute s_i. By Lemma 1, we only need to pay attention to $S \subseteq D - z_i - z_{i+1}$. Divide S into two parts: $S_1 = S \cap \{z_1, \ldots, z_{i-1}\}$ and $S_2 = S \cap \{z_{i+2}, \ldots, z_n\}$. Notice that if $|S_1| \geq K$, $v(S + z_i) = v(S + z_{i+1}) = v(S)$. Therefore, we only need to consider the case when $|S_1| < K$. We have

$$\begin{aligned} s_i - s_{i+1} &= \frac{1}{n-1}\sum_{k=0}^{n-2} \frac{1}{\binom{n-2}{k}} \sum_{|S_1 \cup S_2|=k, |S_1|<K} v(S + z_i) - v(S + z_{i+1}) \\ &= \frac{1}{n-1}\sum_{k=0}^{n-2} \frac{1}{\binom{n-2}{k}} \sum_{|S_1 \cup S_2|=k, |S_1|<K} w'_i - w'_{i+1} \\ &= \frac{w'_i - w'_{i+1}}{n-1} \sum_{k=0}^{n-2} \frac{1}{\binom{n-2}{k}} \sum_{k_1=0}^{\min(K-1,k)} \binom{i-1}{k_1}\binom{n-i-1}{k-k_1} \\ &= \frac{w'_i - w'_{i+1}}{n-1} \frac{\min(K, i)(n-1)}{i} \\ &= \frac{\min(K, i)}{i}(w'_i - w'_{i+1}), \end{aligned}$$

where $w'_i = w_i \mathbb{1}(y_i = y_{\text{test}})$. See [11] for more details on the identity in the second to last step. □

Theorem 4. *Let $(D \cup D_{test}, \mathsf{d})$ be a clustering instance with $n_{test} = \mathcal{O}(n)$. If it is 3-perturbation resilient for metric k-center with respect to a cluster number k^* and a maximum cluster radius $\tau^* \geq \tau$, then Algorithm 1 with $n_L = k^*$ returns the exact data Shapley values of each point in D in $\mathcal{O}(n_L n(d + \log n) + n_{test} s(d + \log s))$ time, where s is the size of the largest optimal cluster.*

Proof of Theorem 4. We first prove that 3-perturbation resilience guarantees a useful technical lemma.

Lemma 5. *Given a center z_{mark} and any point z from a different cluster than that of z_{mark}, we have $\mathsf{d}(z, z_{mark}) > 2\tau^*$.*

Proof. Assume for contradiction that $\mathsf{d}(z, z_{\text{mark}}) \leq 2\tau^*$. We can create a 3-perturbation of d such that a set of centers $D_{\text{mark}} - z_{\text{mark}} + z$ yield a 3-approximation of the k-center clustering. This contradicts the assumption since any 3-approximation solution recovers the optimal clustering, following Theorem 3.

The 3-perturbation is given as follows.

$$\mathsf{d}'(x, y) = \begin{cases} \min\{3\tau^*,\, 3\mathsf{d}(x, y)\} & \text{if } x = z \text{ and } y \in C(z_{\text{mark}}), \\ 3\mathsf{d}(x, y) & \text{otherwise.} \end{cases}$$

It is easy to verify that d' is a 3-perturbation of d. Besides, assigning every point $x \in C(z_{\text{mark}})$ to the new center z respects the inequality

$$\mathsf{d}(x, z) \leq \mathsf{d}(x, z_{\text{mark}}) + \mathsf{d}(z_{\text{mark}}, z) \leq 3\tau^*.$$

The assignments for other data points are either intact or better. Thus, the new centers indeed yield a 3-approximation, completing the proof. □

We also need the help from another lemma.

Lemma 6 (Chekuri and Gupta [23])**.** *Under 2-perturbation resilience, for any two different optimal clusters C_i, C_j, we have*

$$\mathsf{d}(x, y) < \mathsf{d}(x, z) \quad \textit{and} \quad \tau^* < \mathsf{d}(x, z)$$

for any $x, y \in C_i$ and $z \in C_j$.

As a result, when $k^* = n_L$, the clusters induced by D_{mark} recover the optimal clusters by Theorem 3. This implies that the points assigned to the optimal cluster of z_{mark} will be ranked before the points in other clusters, following Lemma 6. What is more, by Lemma 5, the subset S for each test point z_{test} associated with z_{mark} in Algorithm 1 will collect only points from the optimal cluster z_{test} belongs to, because $|\mathsf{d}(z, z_{\text{mark}}) - \mathsf{d}(z_{\text{mark}}, z_{\text{test}})| \geq \tau^* \geq \tau$ for any z from a different cluster. This verifies the stated time complexity.

What is left to show is the correctness, i.e., every point that is within a distance of τ from z_{test} must stay in the same optimal cluster as z_{test}, which directly follows from Lemma 6 and $\tau^* \geq \tau$, and they will be collected into the subset S as $|\mathsf{d}(z, z_{\text{mark}}) - \mathsf{d}(z_{\text{mark}}, z_{\text{test}})|$ is a lower bound of $\mathsf{d}(z, z_{\text{test}})$. This completes the proof. □

When $k^* < n_L$, we can show that the same time complexity in Theorem 4 still holds, albeit with a slightly larger 4-perturbation.

Theorem 7. *Let $(D \cup D_{test}, \mathsf{d})$ be a clustering instance with $n_{test} = \mathcal{O}(n)$. If $(D \cup D_{test}, \mathsf{d})$ is 4-perturbation resilient for metric k-center with respect to a cluster number k^* and a maximum cluster radius $\tau^* \geq \tau$, then Algorithm 1 returns the exact data Shapley values of each data point in D in time $\mathcal{O}(n_L n(d + \log n) + n_{test} s(d + \log s))$, where s is the size of the largest optimal cluster.*

Proof. We first show a stronger lemma than Lemma 5 under 4-perturbation. We omit the proof since it is similar to that of Lemma 5.

Lemma 8. *Given a optimal center c and any point z from a different cluster than that of c, we have $\mathsf{d}(z, c) > 3\tau^*$.*

Lemma 8 implies a stronger variant of Lemma 6.

Lemma 9. *If (D, d) is 2-perturbation resilient, then for any two different optimal clusters C_i, C_j, we have $2\tau^* < \mathsf{d}(x, z)$ for any $x \in C_i$ and $z \in C_j$.*

Proof. Let c be the optimal center of C_j. By triangle inequality, we have

$$\mathsf{d}(x, z) \geq |\mathsf{d}(x, c) - \mathsf{d}(c, z)| > 2\tau^*.$$

□

As a result of Lemma 9, we can relax the requirement that every landmark point has to correspond to an optimal center, and let it be any point instead. The statement can be proved by similar arguments as in the proof of Theorem 4. □

B Expanded Related Work

B.1 Shapley Values

Shapley values [6] originated in cooperative game theory as a method for fairly distributing gains among players based on their marginal contributions and have been widely adopted in various domains such as economics [24]. In machine learning, the most well-known application is attribution methods such as SHAP [9], where Shapley values are computed for each feature of a given data point as a form of feature importance scores. Recently, Shapley values have been adapted to quantify the contribution of individual training examples to model performance [7]. Computing exact Shapley values is well-known to be expensive and has been shown to be $\#\mathbf{P}$-hard in certain games [10]. Such a computational challenge has motivated various approximation techniques, including mostly Monte Carlo sampling [25–27] and specialized algorithms for specific games [28]. Our work falls into the latter category.

B.2 Data Valuation

Data valuation aims to assign importance scores to training examples, with the hope of identifying valuable or harmful data points [2–4]. The dominant approaches are based on the concept of leave-one-out (LOO), which measures the marginal contribution of a data point to the utility function when it is removed from the training procedure. In classification settings, a common choice for the utility function is the test accuracy of a model trained on the input. Data Shapley [7] and its variants such as Beta Shapley [29], Data Banzhaf [30], and least core [31], are all based on the LOO principle, but differ in the way marginal contributions are aggregated.

Beyond Shapley values, there exist other approaches, and we discuss some notable ones below. Feldman and Zhang [32] simulate the data values by LOO retraining albeit constrained on a small sample of training data, while DataModels [33] sacrifice the exactness of LOO to achieve better scalability by model predictions. Another line of popular methods are gradient-based. TracIn [34] estimates the importance of a training example by tracing the change in test loss caused by the example during the training process. Variations of influence functions [35, 36] have their roots in robust statistics [37] and offer a gradient-based approximation of LOO values.

B.3 KNN Shapley Values

The KNN model provides a unique opportunity for efficient computation of data Shapley values. Jia et al. [11] are the first to discover an efficient algorithm for computing unweighted KNN Shapley values with a complexity of $\mathcal{O}(dn_{\text{test}}n + n_{\text{test}}n \log n)$. This is a significant improvement over general Shapley computation methods, making it feasible for datasets of moderate size. Wang and Jia [13] provide refinements to the unweighted KNN utility function. Building on this foundation, Wang et al. [15] tackle the weighted KNN case, which turns out to be more challenging due to the normalization factor in the utility function. They propose a dynamic programming algorithm for a hard-label weighted KNN utility function. Furthermore, Wang et al. [14] addressed privacy concerns in computing KNN Shapley values and offered formal privacy guarantees for a threshold-based KNN utility function. Note that in their utility function, all near neighbors within the ball around a test point are equal. Our work focuses on accelerating the computation of KNN Shapley values.

B.4 Clustering and Perturbation Resilience

Clustering is a common technique to exploit the structure of a dataset. Among many clustering methods, the k-center clustering is one of the most popular. It is well-known that the k-center clustering problem is $\mathbf{NP}$-hard, and FFT is proved to be 2-approximate in the worst case. To overcome the over-pessimism and conservatism of worst-case analysis, in recent years, *beyond worst-case analysis* (BWCA) has received increasing attention. One popular BWCA approach for clustering problems is to define a notion of stability, and *perturbation resilience* is one classic stability measure [21]. It has been shown that multiple clustering problems admit polynomial-time algorithms under some degree of perturbation resilience [38, 39]. Balcan et al. [22] analyze the FFT algorithm as a robust solution in BWCA scenarios. Chekuri and Gupta [23] study k-center clustering in the presence of outliers. Our work leverages these existing understandings about perturbation resilience to provably accelerate the computation of KNN Shapley values.

Table A1: Statistics of datasets used in the experiments.

Dataset	$\lvert D\rvert$	$\lvert D_{\text{test}}\rvert$	d	$\lvert D\rvert \cdot \lvert D_{\text{test}}\rvert \cdot d$
magic	15 063	153	10	23 046 390
2dplanes	40 360	408	10	164 668 800
cifar10	49 500	500	512	12 672 000 000
dota2	91 722	927	125	10 628 286 750
skin	194 084	1961	3	1 141 796 172
covtype	578 106	2906	54	90 718 705 944
emnist	696 536	1396	512	497 850 499 072
poker	998 000	2000	10	19 960 000 000

C Additional Experimental Results

C.1 Mislabel Detection

We adopt the popular downstream task of mislabel detection, where 5% of the training points are randomly mislabeled, and we try to detect them using the data points with the lowest data values. We use the F1 score to evaluate the performance of valuation methods. The F1 score of our method is slightly worse than that of the un-thresholded baseline (no more than 1% worse), which is unsurprising given their similar formulations. The F1 score of TNN is 20-50% worse than the others in most datasets. This is most likely due to the fact that TNN does not utilize the ranking signal among neighborhood points. We tune the radius on a validation set sampled from the noisy training set. We use the implementation of TNN by its authors [14].

C.2 Curse of Dimensionality

We conduct a controlled experiment to isolate the "curse of dimensionality" effect. We perform projections by a Gaussian random matrix on the cifar10 dataset to reduce its dimensionality to 10, and use the same ratio of τ/d as before. The baseline takes 116s (148s previously) and our method with $\tau/d = 0.05$ takes 13s (75s previously). The running time of all methods decreases, due to the smaller overhead for distance computation, but the ratio of the running time of the baseline to ours increases. Therefore, it is indeed helpful to reduce the dimensionality by random projections.