Constraint-Data-Value-Maximization: Utilizing Data Attribution for Effective Data Pruning in Low-Data Environments*

Danilo Brajovic[1,2], **David A. Kreplin**[3] and **Marco F. Huber**[1,2]
[1] Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Stuttgart, Germany
[2] Institute of Industrial Manufacturing and Engineering IFF, University of Stuttgart, Germany
[3] Heilbronn University of Applied Sciences, Heilbronn, Germany
danilo.brajovic@ipa.fraunhofer.de, david.kreplin@hs-heilbronn.de, marco.huber@ieee.org

Abstract

Attributing model behavior to training data is an evolving research field. A common benchmark is data removal, which involves eliminating data instances with either low or high values, then assessing a model's performance trained on the modified dataset. Many existing studies leverage Shapley-based data values for this task. In this paper, we demonstrate that these data values are not optimally suited for pruning low-value data when only a limited amount of data remains. To address this limitation, we introduce the Constraint-Data-Value-Maximization (CDVM) approach, which effectively utilizes data attributions for pruning in low-data scenarios. By casting pruning as a constrained optimization that both maximizes total influence and penalizes excessive per-test contributions, CDVM delivers robust performance when only a small fraction of the data is retained. On the OpenDataVal benchmark, CDVM shows strong performance and competitive runtime.

1 Introduction

Machine learning models, especially large language models, have an insatiable demand for data, while the availability of data is stagnating. By attributing the influence of training data on model performance, the required amount of data can be reduced, thereby saving energy and improving model quality. Early works in this direction, such as influence functions [Koh and Liang, 2017], aim to gain insights into model behavior by attributing the influence of individual training instances on test instances, thereby serving as a method for explainable AI. Conversely, methods like data Shapley [Ghorbani and Zou, 2019] have been used to assess the influence of single training instances on model performance, referring to this as data value, and applying this understanding for data removal. Typically, these approaches rely on Shapley-based approximations to compute the value of each data instance.

Sorscher *et al.* [2022] benchmark methods for pruning data on ImageNet, showing that novel algorithms for data pruning can improve scaling laws, thus reducing the resource costs associated with modern deep learning.

The motivation for our work is that current data-pruning methods based on Shapley or Banzhaf values suffer from inherent limitations that prevent them from fully exploiting pruning potential. For Data Shapley in particular, this is aligned with Wang *et al.* [2024], who show that it need not outperform random selection unless the utility function satisfies structural assumptions. We first analyze these shortcomings and then leverage our insights to design a new pruning algorithm. Our method formulates pruning as an optimization over a data attribution matrix and is evaluated on the OpenDataVal benchmark [Jiang *et al.*, 2023], achieving strong performance while remaining computationally competitive against state-of-the-art techniques including those identified by Sorscher *et al.* [2022]. Our findings show that there is still substantial room to improve data pruning, which in turn can lower training costs and reduce energy consumption. Our main contributions are:

- We demonstrate that Shapley and Banzhaf-based attributions allocate smaller marginal contributions to instances in larger data clusters. This imbalance causes large clusters to be pruned too early, producing unbalanced removal patterns and suboptimal pruning performance.
- We show that optimal retention sets are budget-specific: the subset that maximizes accuracy for one retention budget need not fully contain the subset optimized for another budget.
- Based on these insights, we introduce Constraint-Data-Value-Maximization (CDVM), a novel algorithm that treats data pruning as an optimization problem over a data attribution matrix.
- We benchmark CDVM on six datasets from OpenDataVal, showing strong performance.

2 Background, Motivation, and Related Work

We begin with a concise overview of data valuation, then examine pruning and other evaluation benchmarks, outline their limitations, and finally introduce two concepts that motivate our method.

*An extended version with proofs, ablations, and additional implementation details is available at http://arxiv.org/abs/2605.11312.

2.1 Data Valuation and Relationship to Pruning

Data valuation assesses the overall impact of individual training instances on the model performance, effectively answering the question, "How much did a training instance contribute to the model's performance?" The value assigned to each training instance i is represented as a scalar. Consequently, the valuation scores for a dataset are expressed as a vector $v \in \mathbb{R}^n$, where n is the number of training instances.

Estimating Data Values

We now introduce the basic notation and the main estimation methods for data values used in this paper. For a comprehensive survey, see Hammoudeh and Lowd [2024] and Sim *et al.* [2022].

- $D = \{(x_i, y_i)\}_{i=1}^n$ is a labeled dataset with inputs x_i, labels y_i and $d_i = (x_i, y_i)$.
- f_D is the model trained on the dataset D.
- θ_D are the corresponding model parameters.
- $f_{D\cup\{d_j\}}$ denotes a model trained on the union of D and the data instance $d_j = (x_j, y_j)$.
- $\mathcal{U}$ represents a utility function, such as accuracy in a classification setting.

Leave-One-Out The simplest approach to estimating the influence of a training instance is the leave-one-out (LOO) method, which involves excluding a particular data instance during training and comparing the model performance or test predictions with and without this instance. This method can be approximated by influence functions [Koh and Liang, 2017] without the need for re-training. The main limitation is that the effect of omitting a single data instance can often be obscured by the remaining data and the inherent noise in the training process [K and Søgaard, 2021]. As a result, many data instances may appear to have a negligible value. Empirical evidence also suggests that LOO is not effective for benchmarks in data valuation [Jiang *et al.*, 2023]. Formally, the LOO value of data instance d_i can be expressed as $V(d_i) = \mathcal{U}(f_D) - \mathcal{U}(f_{D\setminus\{d_i\}})$.

Semi-value-based Estimates Semi-value-based techniques quantify the importance of a training instance d_i by its marginal contribution over all subsets $S \subseteq D \setminus \{d_i\}$. For data valuation, three variations were proposed; original Shapley value [Ghorbani and Zou, 2019], Banzhaf [Wang and Jia, 2022], and Beta Shapley [Kwon and Zou, 2022]. Technically, all these methods differ only by the weighting of each subset $w(S)$ and can be expressed as $V(d_i) = \sum_{S\subseteq D\setminus\{d_i\}} w(S)\left[\mathcal{U}(f_S) - \mathcal{U}(f_{S\cup\{d_i\}})\right]$.

These methods generally outperform the LOO estimate in practice. However, their main drawback is their exponential computational complexity. To mitigate this, Monte Carlo or other sampling-based techniques are often used to approximate data values. Notably, data Banzhaf [Wang and Jia, 2022] has proved to be computationally efficient due to the *Maximum Sample Reuse (MSR)* principle. The data value is approximated by sampling subsets $S \subset D$ of the training data with probability p and training a model on each subset. This process is repeated multiple times, and the data value of data instance d_i is computed as the performance difference between subsets where d_i is included versus where it is not.

Out-of-Bag and Memorization Estimates The concept of memorization has been introduced in recent studies, wherein a training instance i is considered "rare" if its exclusion from the training set significantly reduces the probability that i is correctly classified by the same model [Feldman, 2020b; Paul *et al.*, 2021]. A related method used in data valuation is the out-of-bag estimate, known as *DataOob*, where the significance of training instances is assessed using out-of-bag samples [Kwon and Zou, 2023]. In each iteration, the training set is split into in-bag and out-of-bag groups, a model is trained on the in-bag samples, and predictions are made on the out-of-bag samples. The value of a data instance is then determined based on its memorization score during these out-of-bag assessments. Although these techniques are not suited for data attribution (as no test set is involved), they have proven effective in data pruning tasks, even on ImageNet [Sorscher *et al.*, 2022].

Data Pruning and other Benchmarks for Data Valuation

Data-valuation methods are commonly evaluated on three tasks:

1. **Noise Detection:** Identify and remove corrupted or mislabeled examples, which tend to carry large negative value due to their disruptive effect on training [Jiang *et al.*, 2023].
2. **Domain Transfer:** Select a subset of source–domain data that maximizes accuracy on a target–domain test set (e.g., choosing MNIST digits to improve performance on street-number datasets) [Ghorbani and Zou, 2019].
3. **Data Removal:** Measure how model accuracy changes when portions of the training set are removed in order of increasing or decreasing value. Removing high-value instances first should cause a steep accuracy drop, whereas pruning low-value instances should have minimal impact.

In this work, we focus on data removal, since it directly addresses the practical goal of reducing dataset size without sacrificing performance. This is particularly relevant when models must be trained repeatedly under compute or storage constraints. From here on, we use *data pruning* to mean the removal of low-value points. In the literature, authors differ in whether they report results for removing low-value data (pruning) or for removing high-value data first. For instance, the original Data Shapley study [Ghorbani and Zou, 2019] presents low-value pruning curves, while the OpenDataVal framework [Jiang *et al.*, 2023] emphasizes high-value removal. We are not aware of any formal discussion explaining this discrepancy. Empirically, memorization-based or out-of-bag-based methods tend to excel at low-value pruning, whereas Shapley-based techniques often show stronger effects when high-value data is removed first.

Limitations of Data Values for Data Pruning

After reviewing the main approaches to data valuation, we now highlight their shortcomings in the context of data pruning, using a simple clustered setting that we also analyze theoretically in the extended version. Consider the dataset in

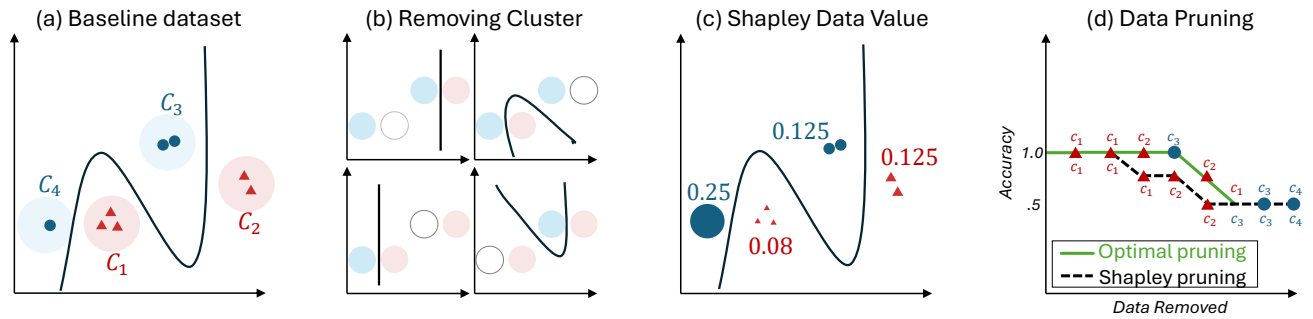

Figure 1: (a) Baseline synthetic dataset comprising 8 points from 4 clusters. (b) Illustrates the changed decision boundary after removing an entire cluster. In each scenario, the decision boundary undergoes significant alterations. (c) Displays the Shapley data value. (d) Test accuracy as we iteratively remove instances (x-axis: removal step 1–8; y-axis: accuracy). In the optimal (green) removal order, clusters are pruned as $c_1, c_1, c_2, c_3, c_2, c_1, c_3, c_4$, whereas the Shapley-based (black) order is $c_1, c_1, c_1, c_2, c_2, c_1, c_3, c_4$ and c_i belongs to the i-th cluster.

Figure 1(a): the training set is partitioned into four Gaussian clusters

$$D = C_1 \cup C_2 \cup C_3 \cup C_4, \qquad C_r \cap C_s = \emptyset \text{ for } r \neq s,$$

where points in cluster C_k are sampled from a Gaussian distribution $\mathcal{N}(\mu_k, \sigma^2 I)$. Let $n_k = |C_k|$ denote the size of cluster C_k; in this example, $n_1 = 3$, $n_2 = 2$, $n_3 = 2$, and $n_4 = 1$. The black line shows the decision boundary learned by a multi-layer perceptron trained on the full dataset.

The test set is partitioned analogously as

$$\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3 \cup \mathcal{T}_4, \quad |\mathcal{T}_k| = m_k$$

so that $\mathcal{T}_k$ contains the test points associated with cluster C_k (these test points are not shown in the figure).

In this dataset, removing an entire training cluster C_k causes all corresponding points in the test cluster T_k to be misclassified (Figure 1(b)). Hence each cluster C_k contributes a utility $u_k > 0$ if and only if at least one of its points is present in the training subset. The extended version provides further details on this construction.

1. LOO has Redundancy Bias and Attributes Non-zero Value Only to Unique Data We begin with the observation that LOO attributions reward only non-redundant samples. In Figure 1 (a), only the singleton instance in C_4 receives a nonzero value $u_4 > 0$, since its removal introduces a change in the decision boundary (Figure 1 (b)), causing a test error at the respective cluster center. All other instances receive a value of zero due to their redundancy and can therefore be pruned in any order.

2. Shapley and Banzhaf Exhibit Cluster-Size Bias and Cause Imbalanced Pruning In this model, Data Shapley and Data Banzhaf values share a cluster's total utility among all its members: a point in C_k receives Shapley value u_k/n_k (Figure 1 (c)) and Banzhaf value $u_k/2^{n_k-1}$ (see the extended version for the formal derivation).

If all clusters share the same utility $u_k = u$ (e.g., when test clusters T_k are of equal size or when performance is measured via balanced accuracy over clusters), then both Data Shapley and Data Banzhaf assign smaller values to points in larger clusters (extended version), so these points are pruned first. On the synthetic dataset, this leads to initially harmless removal of redundant points from majority clusters, but once the last representative of a cluster is removed, performance drops sharply, in contrast to the optimal strategy that keeps one representative per cluster for as long as possible (Figure 1(d)). Moreover, in this clustered model we can show that, when train and test sets are equally distributed over clusters (i.e., the test set contains m_k instances from cluster C_k with $m_k \propto n_k$), Shapley data values collapse to a constant: every training instance receives the same Shapley value, regardless of its cluster membership; see the extended version for the formal derivation.

This effect can also be observed on real data. In the left plot of Figure 2, we compare DataBanzhaf against random pruning on CIFAR-10, using either 1,000 or 10,000 models to estimate data values. Both DataBanzhaf variants outperform random removal up to about 50% pruning. Beyond that point, the 10,000-model variant plunges significantly below the random baseline, whereas the 1,000-model variant continues to slightly outperform random pruning, even though the larger ensemble should yield more accurate attributions.

3. Pruning Subsets are Budget Specific Finally, we observe that optimal retention sets at different pruning levels are not nested: the subset that maximizes accuracy for one budget s may exclude instances that are essential for another budget $s' \neq s$. In Figure 2 (center), we use our own method to identify the best subsets for retaining 5%, 10% and 15% of the data. We then perform sequential pruning of the remaining instances, always keeping the preselected subset intact and plot test accuracy versus fraction removed. Each accuracy curve peaks exactly at its target retention level (dots), and even a slight deviation from that budget causes a dramatic collapse in performance. A similar pattern appears for Memorization/DataOob (Figure 2, right): removing the highest-value instances first (solid blue curve) initially improves performance before it plummets, whereas retaining those same instances until the very end yields almost state-of-the-art final accuracy. This mirrors the finding of Sorscher *et al.* [2022], namely that the examples most dispensable in data-rich regimes are precisely those that must be kept when data become scarce.

In summary, these observations highlight the need for a pruning strategy that (i) tracks influence at the level of individual test samples and (ii) can flexibly re-optimize for each

pruning budget. To that end, we now review two key building blocks of our approach: data attribution and influence-function–guided pruning.

2.2 Other Related Work

Core-set methods select small subsets that preserve the geometry or distribution of the full dataset [Feldman, 2020a; Jiang *et al.*, 2025]. Recent work combines importance scores with redundancy penalties in discrete optimization formulations [Tan *et al.*, 2025], conceptually related to our CDVM objective. In parallel, Most Influential Subset Selection (MISS) studies subset-level influence and its failure modes [Hu *et al.*, 2024], and several pruning methods leverage training dynamics such as example difficulty or uncertainty [Cho *et al.*, 2025; He *et al.*, 2023]. We view CDVM as complementary to these lines, focusing on attribution-based pruning in low-data regimes.

2.3 Preliminaries: Data Attribution & Influence-Function Pruning

We now introduce two core concepts underlying our method: data attribution and influence-function–guided pruning.

Data Attribution

Data attribution is conceptually related to data valuation, but traces the influence of individual train instances down to specific test samples. The influence of training data on test predictions is quantified using the attribution matrix $\mathbf{T} \in \mathbb{R}^{n \times m}$, where n is the number of train instances and m the number of test instances. A high value of $\mathbf{T}_{i,j}$ indicates that the train instance i significantly impacts the prediction for test instance j. The connection between both is that data values can be estimated by averaging over the columns (test instances) of $\mathbf{T}$, formulated as $v_i = \frac{1}{m}\sum_{k=1}^{m} \mathbf{T}_{i,k}$. This per–test–sample breakdown provides a fine-grained view of dataset contributions, which we leverage directly in our method. Several methodologies have been developed to estimate this influence, with influence functions being one of the pioneering approaches [Koh and Liang, 2017]. TRAK has emerged as a scalable method for data attribution across large datasets [Park *et al.*, 2023].

Influence-Function–Guided Pruning

Yang *et al.* [2023] cast data pruning as a discrete optimization problem over binary removal variables, with the goal of minimizing overall parameter change. Let $r \in \{0,1\}^n$ be the indicator vector specifying which of the n training samples are removed. The parameter change from removing a single instance d_i is given by the influence function $\mathcal{I}(d_i) = \theta_{D\setminus d_i} - \theta_D \approx \frac{1}{n} H_\theta^{-1} \nabla_\theta \mathcal{L}(d_i; \theta_D)$, where H_θ is the Hessian of the total training loss at θ_D. For a subset of removed instances, these influences simply add up. Define the matrix $\mathbf{Z} = \big[\mathcal{I}(d_1), \ldots, \mathcal{I}(d_n)\big]$, so that the total parameter change caused by the selected removals is $\mathbf{Z}\, r$. They then solve

$$\min_{r \in \{0,1\}^n} \big\|\mathbf{Z}\, r\big\|_2 \quad \text{s.t.} \quad \sum_{i=1}^{n} r_i = B\,,$$

where B denotes the removal budget. Although this method achieves strong empirical performance and inspired our approach, it has two major drawbacks. First, it requires (approximate) Hessian inversion for every training instance, which is computationally expensive. Second, because it relies on influence functions, essentially approximating leave-one-out, it inherits LOO's limitations (see Sec. 2.1).

3 Size-Constraint Data-Value-Maximization

Building on the limitations identified in Section 2.1, we now introduce a method that derives data values specifically optimized for pruning. There, we observed that Shapley-based data values can fail because they tend to remove entire clusters first. To overcome this, we leverage the attribution matrix

$$\mathbf{T} \in \mathbb{R}^{n \times m},$$

which describes the influence of each of the n training samples on each of the m test samples. A naive way to derive pruning scores from $\mathbf{T}$ is to average over its columns, but this approach suffers (among other issues) from the cluster-removal limitation noted above. Instead, $\mathbf{T}$ provides fine-grained, per-test influence values that do not suffer from redundancy bias. We leverage this to ensure balanced coverage: at each pruning step, no test sample (and thus no implicit cluster) should have zero total influence. To formalize this, let

$$w \in \{0,1\}^n$$

be the binary indicator vector selecting exactly S out of the n training instances to retain. The induced utility vector for the m test samples is

$$v = \mathbf{T}^\top w \;\in\; \mathbb{R}^m\,, \quad v_j = \sum_{i=1}^{n} \mathbf{T}_{ij}\, w_i\,.$$

A naive pruning objective would be

$$\max_{w} \sum_{j=1}^{m} v_j \quad \text{s.t.} \quad \sum_{i=1}^{n} w_i = S, \quad w_i \in \{0,1\}.$$

This objective maximizes total influence but can still concentrate all value on a few test instances. To ensure balanced coverage, we introduce nonnegative slack variables t_j that caps any excess above a threshold κ. In other words, any amount $\max\{v_j - \kappa,\, 0\}$ is transferred into t_j and subtracted from the objective. We call the resulting formulation Constraint Data-Value Maximization (CDVM):

$$\begin{aligned}
\max_{w,t} \quad & \alpha \sum_{j=1}^{m} v_j \;-\; (1-\alpha) \sum_{j=1}^{m} t_j\,, \\
\text{s.t.} \quad & v = \mathbf{T}^\top w\,, \\
& \sum_{i=1}^{n} w_i = S\,, \\
& t_j \geq 0\,, \quad j = 1, \ldots, m\,, \\
& t_j \geq v_j - \kappa\,, \quad j = 1, \ldots, m\,, \\
& w_i \in \{0,1\}\,, \quad i = 1, \ldots, n\,.
\end{aligned}$$

This formulation directly remedies the shortcomings identified in Section 2.1 by (1) maximizing each test sample's total influence via $\mathbf{T}$, thereby avoiding redundancy bias; (2)

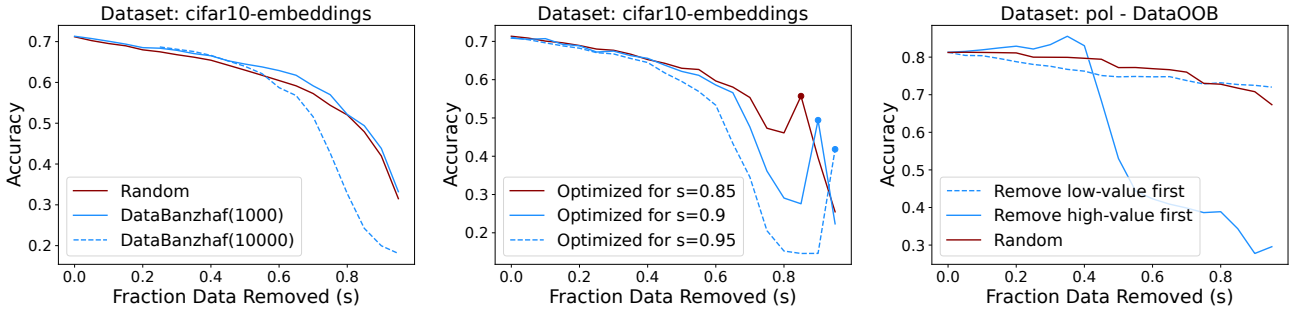

Figure 2: All plots show test accuracy as a function of the fraction of training data removed. **Left:** CIFAR-10 results for DataBanzhaf pruning. We estimate Banzhaf values with 1,000 models (solid blue) and 10,000 models (dashed blue), and compare against random removal (solid red). Both Banzhaf variants outperform random up to 50% pruning, but counterintuitively the 10,000-model variant degrades faster than the 1,000-model version. **Center:** An extreme example of non-nested pruning subsets. Each curve is optimized for exactly 85%, 90%, or 95% removal (i.e., 15%, 10%, 5% retention). Accuracy peaks precisely at the target rate (dots), and removing more or less data causes a steep collapse. **Right:** Memorization/DataOob pruning. The dashed blue curve removes lowest-value instances first; the solid blue curve removes highest-value instances first; and the red curve is random removal. Surprisingly, the very instances whose early removal boosts accuracy (and outperforms random) when data is abundant, must be retained until the end under high removal budgets to again outperform the random baseline.

penalizing any excess above κ, thereby discouraging concentration of influence on a small number of test samples; and (3) enforcing a fixed subset size S to identify the optimal subset for the given budget. Furthermore, because all constraints are linear and some decision variables are integer-valued, the problem can be formulated as a mixed-integer linear program.

In the clustered setting from before, the analysis in the extended version shows that, under the corresponding parameter setting, CDVM preserves at least one representative per cluster whenever $S \geq K$.

3.1 Implementation Details

In our final setup, we relax $w_i \in \{0, 1\}$ to $w_i \in [0, 1]$ and retain the top-S entries of w after solving the linear program. This improves tractability without observable loss; about 10% of entries of w are fractional overall (roughly 6–23% by dataset), with smaller fractions at smaller retention sizes. The algorithm takes as input the attribution matrix $\mathbf{T}$ and two hyperparameters:

- $\alpha \in [0, 1]$: trade-off between total utility and penalty for exceeding κ,
- κ: soft upper bound on the influence per test sample.

Computing $\mathbf{T}$ is the main computational bottleneck, since it requires retraining models on sampled subsets, an expense shared by all semi-value-based methods. Consequently, any parameter used to estimate $\mathbf{T}$ effectively becomes a hyperparameter. Here, we follow the Maximum Sample Reuse (MSR) principle of Wang and Jia [2022]:

1. Sample T subsets $S_t \subseteq D$ by including each training instance with probability p.
2. Train a model on each S_t and record the performance (or indicator of correct classification) on each validation instance.
3. Estimate $\mathbf{T}_{ij}$ as the average difference in that performance for validation instance j when d_i is in versus out of S_t.

The entries $\mathbf{T}_{ij} \in [-1, 1]$ are easily interpretable: -1 means "always causes a mistake" and $+1$ means "always ensures correct prediction." Moreover, $\mathbf{T}$ is sparse, most training instances have zero or negligible influence on most validation instances, which significantly accelerates subsequent optimization.

Once $\mathbf{T}$ is computed, we solve the relaxed CDVM problem using the Disciplined Parametrized Programming framework. This formulation enables caching, so we can quickly resolve the program after it has been solved once. This efficiency allows a lightweight grid search over the two hyperparameters α and κ. We then run the optimization independently for each retained fraction (e.g., 30%, 25%).

4 Experimental Results

We evaluate CDVM on the six datasets from the OpenDataVal benchmark [Jiang *et al.*, 2023]. The attribution matrix $\mathbf{T}$ is computed on the training–validation split and used to prune training instances. Final performance is then assessed on the held-out test set. Each experiment is run with 25 seeds, and we report the average test accuracy at retention levels from 5% to 30% in steps of 5%. We focus on low retention levels because differences are small at high retention and become pronounced only once important instances start to be pruned. We compare against the following baselines:

- **Random** removal of training samples.
- **DataOob/memorization** showing strong performance in [Kwon and Zou, 2023; Sorscher *et al.*, 2022].
- **DataBanzhaf** [Wang and Jia, 2022], a semi-value–based method grounded in the MSR principle, which we also employ.
- **Influence Optimization** (Yang *et al.* [2023]). We encountered some stability issues with the original code: e.g. optimizing for a 10% final subset occasionally performed better when using the budget for 5%. To avoid

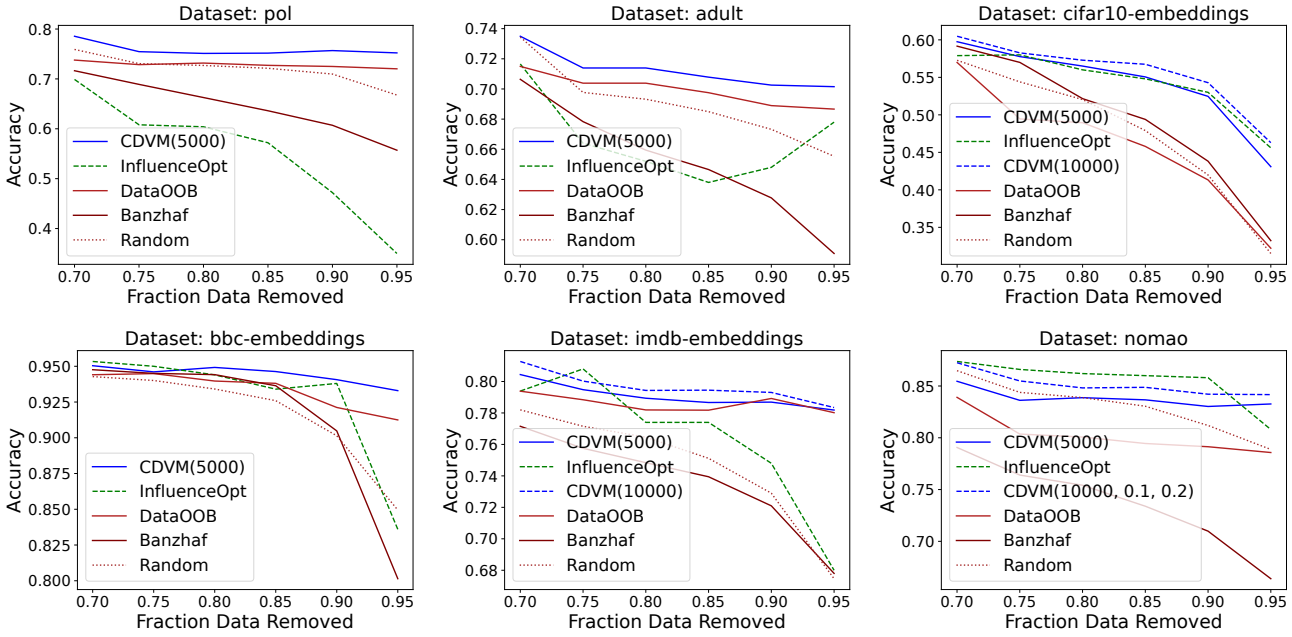

Figure 3: Accuracy on 30%, 25%, 20%, 15%, 10%, and 5% of remaining training data for six datasets in the OpenDataVal benchmark [Jiang *et al.*, 2023]. We utilized a sampling probability of $p = 0.03$ for computing the attribution matrix and automatically optimized parameters for the CDVM method. Out of 36 configurations, CDVM achieved state-of-the-art performance in 28 setups.

underestimating this baseline, at each pruning level we compare against the best accuracy achieved by this method over any budget. We also relaxed the constraint $r_i \in \{0, 1\}$ to a continuous one $r_i \in [0, 1]$, since the original mixed-integer-programming formulation often failed to converge. Consequently, these results should be viewed as an upper bound on the method's performance.

DataBanzhaf serves as a baseline to ensure any performance gains stem from our optimization rather than the attribution algorithm. For CDVM, we fix the sampling probability at $p = 0.03$ and train $T = 5,000$ models (the primary computational bottleneck). In some cases, tuning p or increasing T manually yields gains; we also report those as dashed line when they are significant.

Figure 3 summarizes results over the 36 evaluation instances (6 datasets $\times$ 6 pruning rates), our default CDVM configuration (solid blue) outperforms baselines in 24 cases. Per-dataset tuning (dashed blue) yields some gains and increases the total number of state-of-the-art results to 28. The extended version provides the full tabular breakdown.

Among all baselines, only Yang *et al.* [2023] (green dashed line) outperforms CDVM, mainly on the `nomao` dataset, where it edges out CDVM at 5 of the 6 pruning levels. It also performs competitively on `cifar10` but falls short on `pol` and `adult`, despite being evaluated as an upper bound. On the two text datasets (`bbc` and `imdb`), its gains are occasional and smaller. In contrast, CDVM is the only method that consistently beats random across all six benchmarks, provided hyperparameters are appropriately tuned (see nomao discussion below). DataOob/memorization remains competitive on `imdb` and `bbc` datasets, but never achieves state-of-the-art.

The `nomao` dataset exhibits unusual dynamics for CDVM and Yang *et al.* [2023]'s methods. With default hyperparameters, CDVM initially underperforms random pruning up to an 85% removal rate. We found that manually tuning to $p = 0.1$, $\kappa = 0.2$, $\alpha = 0.1$ restores its advantage. Likewise, Yang *et al.* [2023]'s approach attains its best scores on `nomao` only when its ranked instances are removed first and the remainder are kept at random, i.e., by applying its ranking in reverse. We attribute CDVM's initial failure mode on this dataset to the high proportion of near-zero entries in the attribution matrix $\mathbf{T}$. Increasing the sampling probability p seems to improve this, and using a smaller threshold κ prevents CDVM from assigning too much influence to individual validation instances when overall attributions are small.

4.1 Ablation Study

Runtime Comparison Figure 4(a) plots each method's average runtime against its normalized performance across all 36 settings, where 1 denotes the highest and 0 the lowest accuracy across all methods and budgets (details in the extended version). CDVM achieves the best speed–accuracy trade-off by a wide margin; DataOob outperforms Influence Optimization in efficiency by delivering consistently strong accuracy at lower computational cost. Figure 4(b) further shows that performance improves as the number of times each sample is observed ($p \cdot T$) increases, but increasing T at fixed p yields more stable gains than enlarging subset sizes at fixed T.

Hyperparameter Sensitivity Although CDVM introduces two hyperparameters (α and κ), we find them robust across tasks and can be set without a grid search. We recommend

$$\alpha = 0.5, \quad \kappa = \max_{i,j} \mathbf{T}_{ij} \; + \; S \, \mathrm{mean}_{i,j}(\mathbf{T}_{ij}),$$

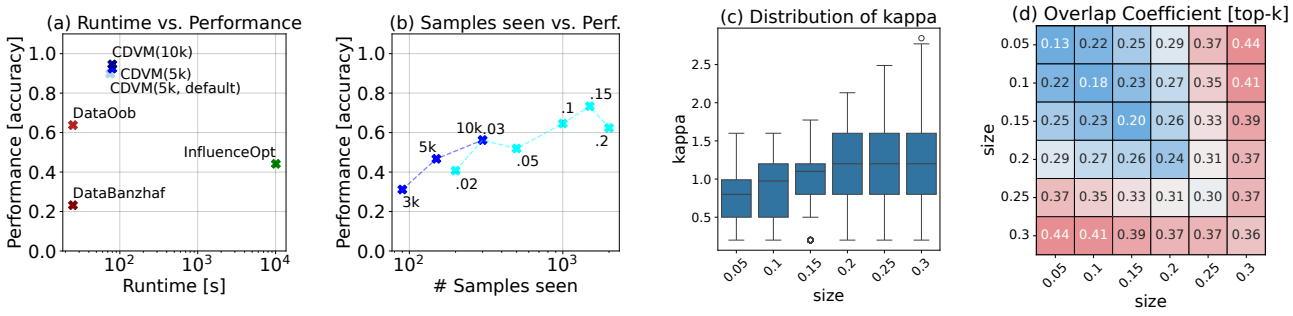

Figure 4: **(a)** Runtime vs. normalized performance for all benchmarked methods, aggregated over six datasets and six pruning levels. **(b)** CDVM performance as a function of how often each sample is seen during training for sampling probability p (cyan) and number of models trained T (blue). **(c)** Distribution of the selected slack threshold κ across datasets and retention fractions. **(d)** Average overlap coefficient $\frac{|A \cap B|}{\min(|A|,|B|)}$ between retained sets at different retention levels, aggregated across datasets and seed pairs. Diagonal values increase with retention size, reflecting greater redundancy at smaller budgets; off-diagonal values exceed diagonal values, indicating that larger retained subsets largely contain the smaller ones.

which adapts κ automatically to the dataset and the retention budget S. The intuition behind this choice is that the effective budget per validation instance should scale with the size of the retained subset: on average, a training example contributes $|S|$ times the mean attribution value. To ensure that highly influential instances are not overly penalized, we additionally add the maximum attribution value. Figure 4(c) confirms this empirically: the κ values selected by grid search increase consistently with the retention size S.

In Figure 4(a) we compare CDVM(5k) with grid-searched hyperparameters against CDVM(5k, default) using the settings above. The performance difference across all datasets and budgets is marginal (≤ 0.05 in normalized performance) while the default configuration incurs zero search overhead, making it a practical choice for most applications.

Nestedness To better understand nestedness, we compute the overlap coefficient between retained sets. Figure 4(d) shows that diagonal values increase with retention size, indicating greater redundancy at smaller budgets. Off the diagonal, larger retained subsets largely contain the smaller ones, showing some nestedness across budgets. This suggests a three-fold structure: instances never selected at any budget, a pool of generally useful instances shared across budgets, and budget-specific instances whose value depends on the retention level. Further analysis is provided in the extended version.

5 Scalability and Computational Cost

By default, each dataset in OpenDataVal is subsampled into smaller splits. To demonstrate that CDVM extends beyond these reduced settings, we include in the extended version a scaling experiment on the Fashion-MNIST dataset (60,000 train, 10,000 test). Thanks to the sparsity of the attribution matrix $\mathbf{T}$, retaining only 5–10% of its entries yields solve times of 10–30 minutes, including hyperparameter grid search, whereas retraining models to estimate $\mathbf{T}$ (a cost shared by all semi-value-based approaches) requires several hours. All experiments were conducted on a single consumer-grade workstation without specialized hardware.

Extending CDVM to even larger datasets such as ImageNet introduces two main bottlenecks: (i) estimating $\mathbf{T}$ in terms of compute and memory, and (ii) solving the linear program at that scale. For context, prior studies have precomputed influence or memorization estimates on ImageNet by training thousands of ResNet-50 models[1] demonstrating that such training is feasible even on ImageNet, although the resulting attribution matrix is very large (≈ 250 GB for train$\times$test). By thresholding $\mathbf{T}$ to keep only its top 10% nonzeros, this can be reduced to ≈ 25 GB.

On the computational side, $\mathbf{T}$ could be approximated using similarity measures between train and validation samples in feature space, in the spirit of Sorscher *et al.* [2022]. If the resulting linear program is still too large, column generation and warm starts across budgets could further reduce solve time. We leave these extensions to future work.

6 Summary, Limitations & Outlook

In this work, we introduced Constraint-Data-Value-Maximization (CDVM), an optimization-based framework that leverages the data-attribution matrix $\mathbf{T}$ to prune low-value examples in low-data regimes. Across six OpenDataVal tasks, CDVM achieved strong performance while remaining computationally competitive.

The main bottleneck of CDVM, and of other approaches based on semi-values, is the computation and storage of $\mathbf{T}$. On the computational side, future work could explore more efficient attribution estimators. On the storage side, exploiting sparsity or low-rank structure in $\mathbf{T}$ would substantially reduce memory usage and, in turn, speed up optimization. Additional gains in optimization time may be possible by solving the CDVM problem on partitioned submatrices of $\mathbf{T}$ rather than on the full matrix at once.

Finally, because the entries of $\mathbf{T}$ are not additive, CDVM does not explicitly capture higher-order interactions. Incorporating Shapley interaction indices [Muschalik *et al.*, 2024] is an interesting direction to better model these effects.

[1] https://pluskid.github.io/influence-memorization/

Acknowledgements

This paper is funded in part by the German Federal Ministry of Research, Technology and Space under the project "KI-Fogger". To ensure reproducibility, we provide the source code at https://github.com/danilobr94/ijcai2026_cdvm.

With funding from the:

References

[Cho *et al.*, 2025] Yeseul Cho, Baekrok Shin, Changmin Kang, and Chulhee Yun. Lightweight dataset pruning without full training via example difficulty and prediction uncertainty. In *Proceedings of the 42nd International Conference on Machine Learning*, 6 2025.

[Feldman, 2020a] Dan Feldman. Core-sets: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10, 2020.

[Feldman, 2020b] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2020.

[Ghorbani and Zou, 2019] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, pages 4053–4065, 2019.

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

[He *et al.*, 2023] Muyang He, Shuo Yang, Tiejun Huang, and Bo Zhao. Large-scale dataset pruning with dynamic uncertainty. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 7713–7722, 2023.

[Hu *et al.*, 2024] Yuzheng Hu, Pingbang Hu, Han Zhao, and Jiaqi W. Ma. Most influential subset selection: Challenges, promises, and beyond. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc.

[Jiang *et al.*, 2023] Kevin Fu Jiang, Weixin Liang, James Zou, and Yongchan Kwon. Opendataval: A unified benchmark for data valuation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[Jiang *et al.*, 2025] Wenyu Jiang, Zhenlong Liu, Zejian Xie, Songxin Zhang, Bingyi Jing, and Hongxin Wei. Exploring learning complexity for efficient downstream dataset pruning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 67259–67279, 2025.

[K and Søgaard, 2021] Karthikeyan K and Anders Søgaard. Revisiting methods for finding influential examples. *arXiv preprint arXiv:2111.04683*, 11 2021.

[Koh and Liang, 2017] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *34th International Conference on Machine Learning, ICML 2017*, volume 4, pages 2976–2987, 2017.

[Kwon and Zou, 2022] Yongchan Kwon and James Zou. Beta shapley: A unified and noise-reduced data valuation framework for machine learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 8780–8802. PMLR, 28–30 Mar 2022.

[Kwon and Zou, 2023] 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.

[Muschalik *et al.*, 2024] Maximilian Muschalik, Hubert Baniecki, Fabian Fumagalli, Patrick Kolpaczki, Barbara Hammer, and Eyke Hüllermeier. shapiq: Shapley interactions for machine learning. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[Park *et al.*, 2023] Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[Paul *et al.*, 2021] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc.

[Sim *et al.*, 2022] Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning: "ingredients", strategies, and open challenges. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5607–5614. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Survey Track.

[Sorscher *et al.*, 2022] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*,

NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[Tan *et al.*, 2025] Haoru Tan, Sitong Wu, Wei Huang, Shizhen Zhao, and Xiaojuan Qi. Data pruning by information maximization. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 50133–50155, 2025.

[Wang and Jia, 2022] Jiachen T. Wang and R. Jia. Data banzhaf: A robust data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*, 2022.

[Wang *et al.*, 2024] Jiachen T. Wang, Tianji Yang, James Zou, Yongchan Kwon, and Ruoxi Jia. Rethinking data shapley for data selection tasks: Misleads and merits. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[Yang *et al.*, 2023] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *The Eleventh International Conference on Learning Representations*, 2023.