Just Select Twice: Leveraging Low Quality Data to Improve Data Selection

Yifei Zhang[1], Yusen Jiao[2], Jiayi Chen[1], Jieyu Zhang[3*], Frederic Sala[1*]

[1]University of Wisconsin - Madison
[2]University of Waterloo
[3]University of Washington

{yzhang2536, jchen993, fredsala}@wisc.edu
{y29jiao}@uwaterloo.ca
{jieyuz2}@cs.washington.edu

Abstract

Data valuation is crucial for assessing the impact and quality of individual data points, enabling the ranking of data by importance for efficient data collection, storage, and training. Many data valuation methods are sensitive to outliers and require a certain level of noise to effectively distinguish low-quality data from high-quality data, making them particularly useful for data removal tasks. Especially, for instance, optimal transport based method exhibits notable performance in outlier detection but shows only moderate effectiveness in high-quality data selection, attributed to its property of sensitivity to outliers and insensitivity to small variations. To mitigate the issue of insensitivity to high-quality data and facilitate effective data selection, in this paper, we propose a straightforward two-stage approach, JST, that initially performs data valuation as usual, followed by a second-round data selection where the identified low-quality data points are designated as the validation set to perform data valuation again. In this way, high-quality data become outliers with the respect to new validation set and can be naturally popped out. We empirically evaluate our framework instantiated with optimal transport based method for data selection and data pruning on several standard datasets and our framework demonstrates superior performance compared to pure data valuation, especially under the condition with small noise. Additionally, we show the general applicability of our framework to influence function based and reinforcement learning based data valuation methods. The repository is publicly available on Github: `https://github.com/yfeizhang/JST`.

1 Introduction

Access to large, high-quality datasets is essential in machine learning. However, in real-world data collection and curation pipelines, individual data points often inherit different levels of quality and vary importance on the impact of training [1, 2, 3]. Therefore, it is critical to understand and assess such properties of data and to effectively prioritize highly valuable data sources for subset selection. It can assist practitioners in improving model performance efficiently [4, 5] and make strategic, cost-effective decisions in data marketplaces and exchanges [6].

[*] Equal-advising
Corresponding author: Yifei Zhang (yzhang2536@wisc.edu)

2nd Workshop on Attributing Model Behavior at Scale, 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

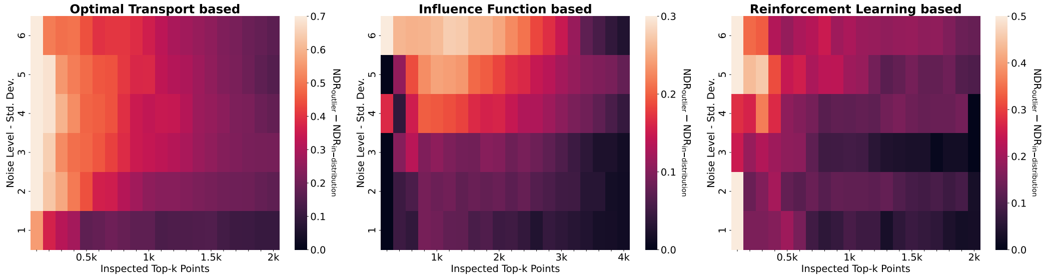


Figure 1: Sensitivity comparison between outliers and in-distribution data. We inject white noise into 50% of CIFAR-10 training set with different noise levels and inspect top k data points for each data valuation method from lowest values for outliers and highest values for in-distribution data, respectively. We evaluate the difference between normalized outlier detection rate ($\mathrm{NDR}_{\mathrm{outlier}}$) and normalized in-distribution data detection rate ($\mathrm{NDR}_{\mathrm{in\text{-}distribution}}$). Outlier data detection rate is greatly larger than in-distribution data detection rate, reflecting higher sensitivity of these data valuation methods to outliers.

Recently, there have been considerable efforts in developing different data valuation methods aimed at evaluating individual data points and assigning a value to each one [5, 7]. It can help to quantify differences between data points and rank them based on their assigned value, establishing a certain order of "quality" or "importance" in training process. For instance, methods based on optimal transport [8] and influence function [9] employ sensitivity analysis to quantify the impact of individual data points on dataset distance and training outcome, respectively. Additionally, importance weight can be obtained through reinforcement learning [10] to evaluate individual data points.

In essence, these data valuation methods should naturally provide a metric for high-quality data subset selection. However, unfortunately, most data valuation methods have predominantly excelled in scenarios with large noise, including intensely corrupted samples [8], randomly flipping labels [8, 9, 10], or domain adaption due to substantial mismatch between training and target domains [9, 10]. Especially, optimal transport based method [8] demonstrates strong performance in outlier detection but exhibits only moderate effectiveness in selecting high-quality data, significantly due to its well-known property of sensitivity to outliers and insensitivity to small variations [11].

As demonstrated in Figure 1, the sensitivity of these data valuation methods to outliers is greatly larger than high-quality data. Therefore, in this paper, we address the following question: *How to improve the sensitivity of data valuation methods to high-quality data and make them better for data selection?*

To this end, we propose JST (Just Select Twice), a data subset selection framework that augments existing data valuation methods by leveraging outliers detected by these methods, as illustrated in Figure 2. Our key insight is that incorporating a second-round subset selection, using detected outliers as the validation set, allows high-quality data to be identified as outliers in the new context. This approach enhances the sensitivity of data valuation methods to high-quality data, establishing a more meaningful order of "importance" or "quality" for subset selection. Even with the inclusion of numerous high-quality data points within the validation set alongside low-quality data in the second-round subset selection due to the non state-of-the-art performance of data valuation methods, we observe that the signals from such a validation set continue to suffice in achieving superior performance in subset selection compared to pure data valuation methods.

2 Preliminaries

Data Selection: We are given a training set $D_{tr} = \{z_1, \ldots, z_n\}$ containing n data points, where each $z_i = (x_i, y_i)$ is drawn from a source distribution $p_{src}(z)$, as well as a validation set $D_v = \{z'_1, \ldots, z'_m\}$ with m data points, where each $z'_i = (x'_i, y'_i)$ is drawn from a target distribution $p_{trg}(z')$ (typically $n > m$). Both sets share the same input-output space $\mathcal{X} \times \mathcal{Y}$ but differ in their underlying distributions, i.e., $p_{src}(z) \neq p_{trg}(z')$.

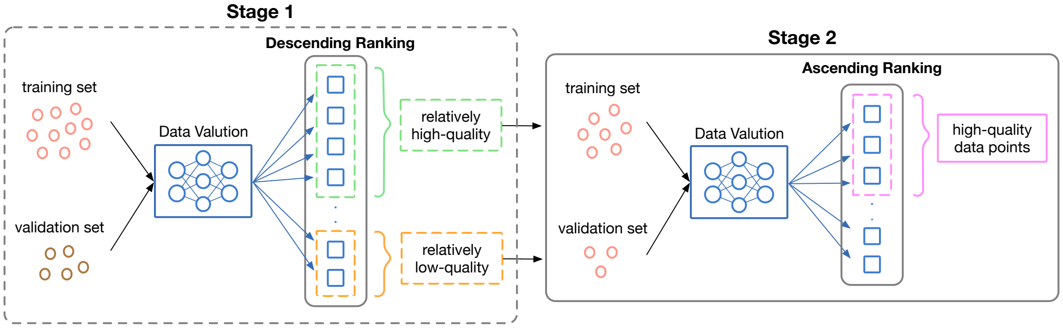


Figure 2: Illustration of JST framework. In stage 1, we perform data valuation as usual and select relatively low-quality data as the validation set. In stage 2, we perform data valuation on the remaining training samples and the new validation set. In this way, high-quality data stands out as outliers with low value scores.

Given a selection budget k ($k \leq n$), the goal of a data selection procedure is to identify a subset $\hat{D} = \{\hat{z}_1, \ldots, \hat{z}_k\}$ where $\hat{D} \subseteq D_{tr}$, such that the distribution of $\hat{D}$ closely matches the distribution of the validation set D_v, to minimize the impact on the learned model. Therefore, the selected subset should approximate the distribution of the validation set, i.e., $P(\hat{D}) \approx p_{trg}(z')$, where $P(\hat{D})$ is the distribution constructed from $\hat{D}$.

Data Valuation: Similarly, we are given a training set $D_{tr} = \{z_1, \ldots, z_n\}$ containing n data points $z_i = (x_i, y_i)$ and a validation set $D_v = \{z'_1, \ldots, z'_m\}$ with m data points $z'_i = (x'_i, y'_i)$. As before both sets share the same input-output space $\mathcal{X} \times \mathcal{Y}$.

The goal of data valuation is to understand and distribute the validation performance across training data points. To achieve this, we use a function $\mathcal{V}$ computed over the training set D_{tr} to find a score vector $\overline{s} \in \mathbb{R}^n$ that represents the allocation to each data point, described as follows:

$$\overline{s} := \mathcal{V}(D_{tr}, D_v), \ \ where \ \overline{s} \in \mathbb{R}^n. \tag{1}$$

Given a score vector $\overline{s} = [s_1, \ldots, s_n]$ representing scores s_i for each data point $z_i = (x_i, y_i)$ in training set D_{tr}, we can express the process of ranking (in descending order) and indexing the k points starting from index r as follows:

$$D_{sel} := \mathcal{R}(\overline{s})[r : r + k], \ \ where \ |D_{sel}| = k \ and \, D_{sel} \subseteq D_{tr}. \tag{2}$$

3 JST: Just Select Twice

We now present JST, a straightforward two-stage approach to augment existing data valuation methods. In the first stage of the process, we perform data valuation as usual and select data points with lowest value scores as the validation set. Then, in the second stage of the process, we perform data valuation on remaining training data points and select the data points of low value scores as high-quality data.

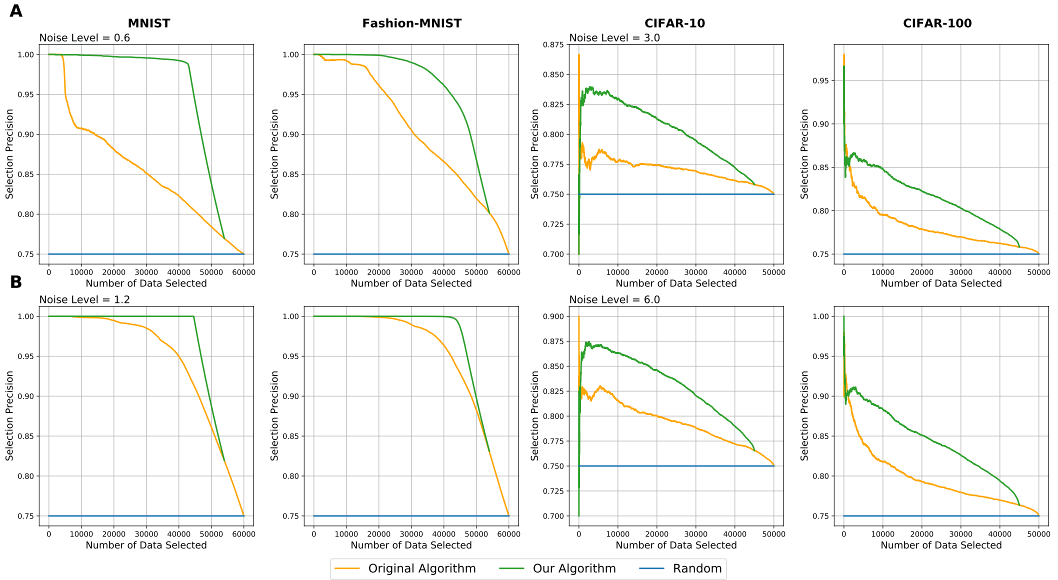


Figure 3: JST framework performance comparison on high-quality data selection. JST framework shows a notable improvement in selection precision compared with pure data valuation method.

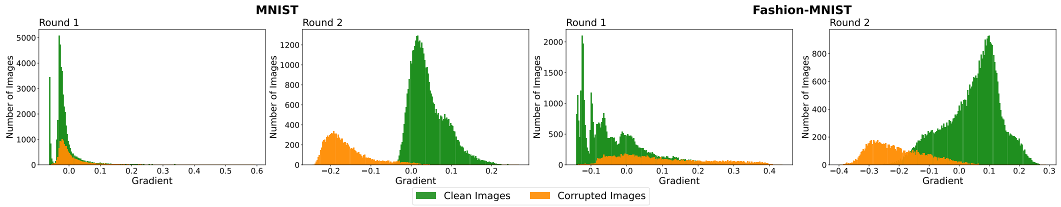


Figure 4: Illustration of the principle of JST framework. In the second-round data selection, high-quality data are popped out as outliers with low value scores, demonstrating near-perfect separation.

Stage 1 (Data Valuation).
Given a training set D_{tr} and a validation set D_v, let D_{sel} of size $|D_v|$ represent the selected data points of lowest value scores obtained by the ranking function $\mathcal{R}(\cdot)[|D_{tr}| - |D_v| : |D_{tr}|]$ of data valuation $\mathcal{V}(D_{tr}, D_v)$.

Stage 2 (Data Selection).
Substitute the original validation set D_v with D_{sel} obtained in the first step and remove data points in D_{sel} from the training set D_{tr}, denoted as $D'_{tr} = D_{tr} \setminus D_{sel}$. Then perform data valuation $\mathcal{V}(\cdot,\cdot)$ again but instead on the changed training set D'_{tr} and validation set D_{sel} and minus the score vector, i.e., $\overline{s'} = -\mathcal{V}(D'_{tr}, D_{sel})$. Finally, the top k high-quality data can be selected by $\mathcal{R}(\overline{s'})[1 : 1 + k]$. We summarize our framework in Algorithm 1.

Algorithm 1 JST Selection

```
Input: training set $D_{tr} = \{(x_i, y_i)\}_{i=1}^{n}$, validation set $D_v = \{(x'_i, y'_i)\}_{i=1}^{m}$, data valuation $\mathcal{V}(\cdot,\cdot)$
Output: data value score vector $\overline{s'}$
Stage one:
Perform data valuation:
    $\overline{s} \leftarrow \mathcal{V}(D_{tr}, D_v)$
Select backward data points:
    $D_{sel} \leftarrow \mathcal{R}(\overline{s})[|D_{tr}| - |D_v| : |D_{tr}|]$
Stage two:
Remove training data points:
    $D'_{tr} \leftarrow D_{tr} \setminus D_{sel}$
Perform data valuation again and minus the score vector:
    $\overline{s'} \leftarrow -\mathcal{V}(D'_{tr}, D_{sel})$
```

Practical Implementation. Practical implementation details of Algorithm 1 and related ablation experiments are deferred at Appendix C.

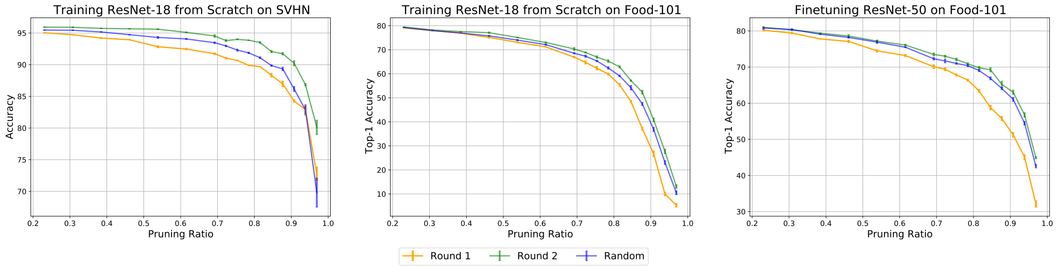


Figure 5: JST framework performance comparison on raw dataset pruning with different pruning ratio. Across each experiment, we perform three individual runs with different random seeds and report the mean performance and the standard deviation as error bars.

4 Experimental Evaluation

4.1 High-quality Data Selection

In real-world scenarios, training data often contain corrupted images [12, 13]. To evaluate the effectiveness of JST framework in selecting high-quality data in the presence of corrupted images, we employ four standard datasets: MNIST, Fashion-MNIST [14], CIFAR-10, and CIFAR-100 [15]. For all four datasets, we inject white noise into 25% of the training set and utilize the clean test set as the validation set. To demonstrate the superior performance of JST framework compared with the pure data valuation method, we add two levels of small noise to the training set as depicted in Figure 3 for noise-level comparative analysis. To achieve the goal of identifying "high-quality" training samples, we rank data points based on their value scores in descending order, selecting a subset with the highest values. For every selection budget, we compute the selection precision, i.e, the percentage of the points that are uncorrupted within the selected points. We compare our framework with pure data valuation method and random selection baseline across these four datasets.

As depicted in Figure 3, our experimental results show a notable enhancement in selection precision across all four datasets, where our JST framework outperforms the pure data valuation method on nearly all selection budgets. Globally, our framework significantly improves the mean rank of uncorrupted data compared to pure data valuation methods. With a very small noise level of standard deviation 0.6 and 3.0 (Figure 3 A), our framework push forward the mean rank by 3,487 for MNIST, 1,798 for Fashion-MNIST, 700 for CIFAR-10, and 861 for CIFAR-100, from initial values of 24,814, 24,269, 22,212, and 22,106, respectively. This underscores the feasibility of employing data selection using our framework under challenging circumstances with small noise. When facing relatively larger noise with a standard deviation of 1.2 and 6.0 (Figure 3 B), our framework still achieves improvements of 785 for MNIST, 607 for Fashion-MNIST, 843 for CIFAR-10, and 1,167 for CIFAR-100, from initial values of 23,064, 23,072, 21,869, and 21,937, respectively. Remarkably, as illustrated in Figure 4, with a noise level of standard deviation 0.6, the data valuation method after being augmented by our framework achieves near-perfect separation between uncorrupted and corrupted data in the simple datasets MNIST and Fashion-MNIST. Furthermore, Figure 4 clearly confirms the feasibility of the principle of our framework, wherein high-quality data are assigned low value scores and popped out as outliers in the second-round data selection, vice versa to the first-round data valuation.

4.2 Raw Dataset Pruning

With the help of commercial search engines, several web-crawled large datasets have been curated [16, 17]. These datasets typically contain heterogeneous noise, such as label noise and out-of-distribution samples, and individual training data points usually vary in quality. To further demonstrate the effectiveness of our JST framework in real-world data collection scenarios, we conduct experiments on two web-crawled datasets, SVHN [18] and Food-101 [19], focusing on the task of raw dataset pruning with being provided a small size of manually cleaned validation set.

More specifically, SVHN is a digit classification dataset with 10 categories (from 0 to 9), cropped from pictures of real-world house number plates obtained from Google Street View images. It contains

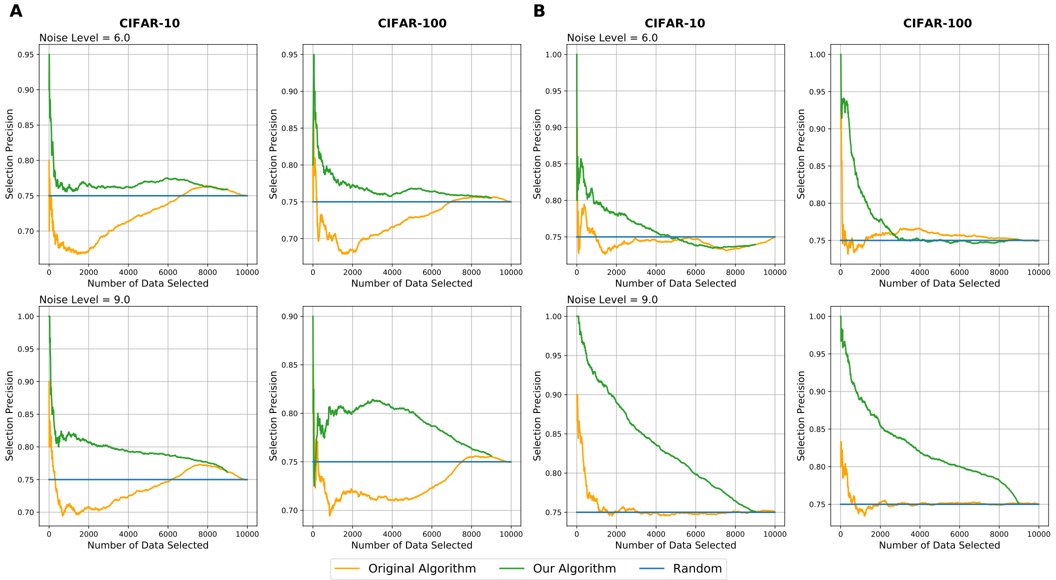


Figure 6: Applying JST framework to different data valuation methods (**A:** influence function based, **B:** reinforcement learning based) on CIFAR-10 and CIFAR-100. Both methods augmented by JST framework improve selection precision to varying extents.

73,257 digits for training and 26,032 digits for testing. Similarly, but with greater complexity, Food-101 dataset, which was crawled online, consists of 101 food categories with 750 training images and 250 test images per category, totaling 101,000 images. The labels for test images have been manually cleaned, while the training set contains some noise, primarily in the form of intense colors, incorrect labels and out-of-distribution samples. We randomly sampled 5,000 digits from the test set of SVHN and 7,500 images from the test set of Food-101 to create their respective validation sets, while leaving the remaining samples to be used as the test sets.

Likewise to Section 4.1, we rank training data points based on their value scores in descending order to identify "high-quality" samples. However, to mitigate the issue of class imbalance, which negatively impacts the trained model performance [20], we simply rearrange the data points sequentially according to their class one by one after each round of ranking. Then, to evaluate the efficacy of our JST framework, we train a ResNet-18 model from scratch on the training set of SVHN and Food-101 with varying pruning ratios, respectively. Additionally, to explore the effectiveness of our framework on pretrained models, we finetune a ResNet-50 model [21], initially pretrained on ImageNet1K, with different pruning ratios. We test all trained neural networks on the reconstructed testing set, excluding samples from the validation set to prevent prior knowledge and maintain a fair accuracy comparison. Still, we compare our JST framework with the pure data valuation method and random selection baseline.

As illustrated in Figure 5, in both SVHN and Food-101 experiments, our framework achieves much higher accuracy compared to the pure data valuation method. It also outperforms the random selection baseline across most pruning ratios especially in cases with large pruning ratios. By visualization of selected images (deferred at Appendix F), we found that our framework selects the most relevant images to the class, while random selection and the pure data valuation method include monotone image patterns, low-quality data, or out-of-distribution samples, thereby hurting model performance.

4.3 Applying JST Framework to Different Data Valuation Methods

To demonstrate the general applicability of our JST framework, we apply it to other two data valuation methods: influence function based and reinforcement learning based approaches. We use dataset CIFAR-10 and CIFAR-100 to evaluate the effectiveness of JST. The experimental setup is the same as in Section 4.1 with white noise standard deviation of 6.0 and 9.0, except the size of training set and validation set. As these two methods suffer from running time for large dataset, to save computational resources, we randomly sample 10,000 data points from the training set as the training set and 1,000 data points from the test data as the validation set. We use the last layer of ResNet-18 model as their dependent training model. As illustrated in Figure 6, these two methods augmented by our framework, JST, improve selection precision to varying extents, corresponding to higher sensitivity of outliers compared with in-distribution data as shown in Figure 1.

5 Conclusion

In this paper, we introduce JST, a straightforward two-stage framework designed to enhance the sensitivity of existing data valuation methods to high-quality data for subset selection. The JST framework identifies low-quality data points as a validation set, thereby allowing high-quality data to stand out as outliers during the data valuation process. Experiments on multiple datasets demonstrate the effectiveness of JST framework in selecting high-quality data and pruning raw datasets, particularly in scenarios with small noise. We successfully apply JST to different data valuation methods, highlighting the general applicability of our framework, thus making it a valuable tool for enhancing data selection in machine learning.

Acknowledgements

We thank Chris Cai for his contributions to the early-stage survey and coding efforts. We thank Maryam Aliakbarpour and César A. Uribe for their valuable discussions and insights.

References

[1] Weixin Liang and James Zou. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. In *International Conference on Learning Representations*, 2022.

[2] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[3] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[4] Alan F Karr, Ashish P Sanil, and David L Banks. Data quality: A statistical perspective. *Statistical Methodology*, 3(2):137–173, 2006.

[5] Kevin Fu Jiang, Weixin Liang, James Zou, and Yongchan Kwon. Opendataval: a unified benchmark for data valuation. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[6] Sunday Adewale Olaleye and Akwasi Gyamerah Adusei. The new reality of data economy and productization: A conceptual paper. 2024.

[7] Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning:" ingredients", strategies, and open challenges. In *IJCAI*, pages 5607–5614, 2022.

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

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

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

[11] Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

[12] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[13] Junnan Li, Caiming Xiong, and Steven CH Hoi. Mopro: Webly supervised learning with momentum prototypes. *arXiv preprint arXiv:2009.07995*, 2020.

[14] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[16] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[17] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[18] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 7. Granada, Spain, 2011.

[19] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014.

[20] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[23] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020.

[24] Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolas Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[25] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

[26] Dan Feldman. Core-sets: Updated survey. *Sampling techniques for supervised or unsupervised tasks*, pages 23–44, 2020.

[27] Jiawei Huang, Ruomin Huang, Wenjie Liu, Nikolaos Freris, and Hu Ding. A novel sequential coreset method for gradient descent algorithms. In *International Conference on Machine Learning*, pages 4412–4422. PMLR, 2021.

[28] Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 126–134, 2005.

[29] Ivor W Tsang, James T Kwok, Pak-Ming Cheung, and Nello Cristianini. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6(4), 2005.

[30] Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable bayesian logistic regression. *Advances in neural information processing systems*, 29, 2016.

[31] Trevor Campbell and Tamara Broderick. Automated scalable bayesian inference via hilbert coresets. *Journal of Machine Learning Research*, 20(15):1–38, 2019.

[32] Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of deep neural networks against noisy labels. *Advances in Neural Information Processing Systems*, 33:11465–11477, 2020.

[33] Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in neural information processing systems*, 34:14488–14501, 2021.

[34] Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR, 2021.

[35] Murad Tukan, Samson Zhou, Alaa Maalouf, Daniela Rus, Vladimir Braverman, and Dan Feldman. Provable data subset selection for efficient neural networks training. In *International Conference on Machine Learning*, pages 34533–34555. PMLR, 2023.

[36] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2022.

[37] Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[39] Jinkun Lin, Anqi Zhang, Mathias Lécuyer, Jinyang Li, Aurojit Panda, and Siddhartha Sen. Measuring the effect of training data on deep learning predictions via randomized experiments. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 13468–13504. PMLR, 17–23 Jul 2022.

[40] Jiachen T. Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 6388–6421. PMLR, 25–27 Apr 2023.

Appendix

A Background and Related Work

We provide a small amount of background and describe related work.

Data Selection. Active learning and coreset construction are two widely used methods for data subset selection. These methods aim to identify the most representative training data points. Active learning involves iteratively choosing points to label from a large unlabeled dataset based on the model's uncertainty or other heuristics, such as the entropy of predicted class probabilities [22, 23]. Recently proposed data selection methods in continual learning are related to these approaches, determining examples to be stored or labeled for ongoing model training [24, 25].

In contrast, coreset construction begins with the entire dataset and tries to select a small subset that encapsulates the essence of the full dataset. The goal is to have the model trained on the subset performs approximately as well as the one trained on the entire dataset [26, 27]. Many works have tackled this setting to accelerate clustering and learning, proposing coresets in k-means and k-medians clustering [28], SVMs [29], Bayesian logistic regression [30], and Bayesian inference [31]. Recently, newly proposed coreset selection methods have demonstrated efficiency in neural network training [32, 33, 34, 35]. In addition, related coreset techniques have been applied to active learning [22] and continual learning [36].

Data Valuation. Data valuation assesses the contribution of each individual data point to the overall performance of a model, aiming to distribute the validation performance across the training data points [5]. Formally, given a training dataset $D_{tr} = \{z_i\}_{i=1}^N$, a validation dataset D_v, and a model performance metric *PERF* evaluated on the validation set D_v, data valuation methods assign a scalar score to each training sample z_i in the D_{tr} to split *PERF* evaluated on the D_v. For instance, if we define a utility function U over all subsets $S \subseteq D_{tr}$ of the training data as $U(S) :=$ *PERF*$(\mathcal{A}(S))$ evaluated on the validation set D_v with a learning algorithm $\mathcal{A}$, a straightforward method to evaluate the contribution of a training sample z_i is by calculating the leave-one-out (LOO) value, $U(D_{tr}) - U(D_{tr} \setminus \{z_i\})$. It represents the change in model performance when the point is excluded from the training set [8].

In practice, feasible alternatives to LOO exist to estimate the impact of the weight change of a data point on the model performance. Two such approaches are optimal transport based method (LAVA) [8] and influence function based method [9]. The optimal transport based method employs Wasserstein distance between training and validation sets as the performance metric, while the influence function based method utilizes validation loss. Both methods measure the gradient as the scalar value score with respect to the weight change of a data point. In addition, reinforcement learning based method (DVRL) [10] learns a weight function to minimize the weighted empirical risk using policy gradients, thereby obtaining optimal importance weight as the scalar value score.

Indeed, data valuation is intricately related to data selection methods, but it formalizes the data selection problem by aiming to select a subset of the training set that matches a desired target distribution, as represented by the validation set [8, 10], because data valuation methods incorporate validation performance into individual training samples. This approach is essentially different from active learning and coreset construction, which instead to compress the full training dataset. Therefore, data valuation can be applied to data selection in the scenario of robust learning with a mismatch between training set and target set [10]. However, as demonstrated in Figure 1, data valuation methods are typically more sensitive to outliers rather than high-quality data. Such property makes data valuation ineffective to distinguish high-quality data at a finer-grained scale, especially when there is a small mismatch between training set and target set. To address this issue, motivated by the property, we propose a straightforward two-stage framework, JST, to make data valuation more suitable for data selection. To the best of our knowledge, our work is the first to leverage this property to augment existing data valuation methods.

B Data Valuation Algorithms

This section provides a detailed explanation of three data valuation algorithms applied in our study: optimal transport based method, influence function based method and reinforcement learning based method. Although we have defined notations in the preliminaries section, a thorough set of notations is presented here for clarity. The input space is denoted by $\mathcal{X}$ and the output space by $\mathcal{Y}$. We denote the training set by $D_{tr} = \{z_i\}_{i=1}^{n}$, where each $z_i = (x_i, y_i)$ is drawn from a source distribution $p_{src}(z)$, the validation dataset by $D_v = \{z'_i\}_{i=1}^{m}$, where each $z'_i = (x'_i, y'_i)$ is drawn from a target distribution $p_{trg}(z')$, and a model performance metric *PERF* evaluated on the validation set D_v. Typically, $n > m$ and $p_{src}(z)$ is not required to be the same as $p_{trg}(z')$, i.e., $p_{src}(z) \neq p_{trg}(z')$. The goal of data valuation is to distribute the validation performance across training data points and compute a data score value $s(z_i)$ for each training data point z_i.

LAVA (Optimal Transport based)

LAVA [8] measures the sensitivity of validation performance to changes in the training data. It examines how the optimal transport cost between the training set D_{tr} and the validation set D_v changes when a particular data point in D_{tr} is assigned increased weight. The sensitivity is determined by calculating the gradient of the optimal transport cost with respect to the probability mass.

The optimal transport cost gradient can be calculated as the data value score for the training data point z_i as follows:

$$s(z_i) := t^*[i] - \frac{1}{n-1}\sum_{j\neq i}(t^*[j])$$

where $\{t_i^*\}_{i=1}^{n}$ is the optimal solution of the dual problem, which is expressed as:

$$t^*, u^* := \arg \max_{t,u\in\mathcal{C}^0(\mathcal{X}\times\mathcal{Y})^2} \left\langle t, \frac{1}{n}\delta_{(x_i,y_i)} \right\rangle + \left\langle u, \frac{1}{m}\delta_{(x'_i,y'_i)} \right\rangle$$

where $\mathcal{C}^0(\mathcal{X}\times\mathcal{Y})$ denotes the set of all continuous functions defined on $\mathcal{X}\times\mathcal{Y}$, and $\delta_{(x,y)}$ represents the Delta measure at $(x, y) \in \mathcal{X}\times\mathcal{Y}$. The negative gradient indicates that the loss will decrease when the data point is given more weight, signifying a higher value for the data point, whereas a positive gradient indicates the opposite.

Influence Function

In machine learning, influence function [9] is used to evaluate the impact of a data point on a model's performance by upweighting a specific training point. We denote the influence of a training data point $z_i = (x_i, y_i)$ on the loss $L(z, \theta)$ with respect to the parameters $\theta \in \Theta$ of a validation data point $z'_j = (x'_j, y'_j)$ as $I(z_i, z'_j)$. This can be obtained by:

$$I(z_i, z'_j) := -\nabla_\theta L(z'_j, \hat{\theta})^\top \mathcal{H}_{\hat{\theta}}^{-1} \nabla_\theta L(z_i, \hat{\theta})$$

where $\nabla_\theta L(z'_j, \hat{\theta})$ represents the gradient of the loss $L(z'_j, \hat{\theta})$ with respect to optimized parameters $\hat{\theta}$ evaluated at the validation data point z'_j. Similarly, $\nabla_\theta L(z_i, \hat{\theta})$ represents the gradient of the loss $L(z_i, \hat{\theta})$ with respect to optimized parameters $\hat{\theta}$ evaluated at the training data point z_i. The term $\mathcal{H}_{\hat{\theta}}$ denotes the Hessian matrix of empirical risk with respect to the optimized model parameters $\hat{\theta}$, defined as $\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \hat{\theta})$.

The negative influence predicts a decrease in loss, while the positive influence predicts an increase. Therefore, higher negative influences correspond to more valuable data points, whereas larger positive influences correspond to less valuable points. To evaluate the impact of each training data point z_i on the whole valuation set D_v, we simply sum the influence of the training data point z_i on each validation data point z'_j as the data value score $s(z_i)$, denoted as:

$$s(z_i) := \sum_{j=1}^{m} I(z_i, z'_j)$$

DVRL (Reinforcement Learning based)

DVRL [10] involves using reinforcement learning algorithm to compute the importance weight for each training data point as the data value score. The objective function that DVRL solves in training a model $g : \mathcal{X} \times \mathcal{Y} \to [0, 1]$, which maps data points to their importance weight, is expressed as:

$$\min_{g \in G} \mathbb{E}_{(x',y') \sim p_{trg}(z')}[\mathbb{L}(f_g(x'), y')]$$

$$s.t. \quad f_g = \arg\min_{f \in F} \mathbb{E}_{(x,y) \sim p_{src}(z)}[g(x, y) \cdot \mathbb{L}(f(x), y)]$$

where $G := \{g : \mathcal{X} \times \mathcal{Y} \to [0, 1]\}$, $F := \{f : \mathcal{X} \to \mathcal{Y}\}$ and the loss $\mathbb{L}$ can be MSE or cross entropy. The data value score of a training data point $s(z_i)$ is computed as the importance weight $g(x_i, y_i)$. The objective can be optimized using policy gradient methods. A large importance weight indicates a more crucial data point for the training process, signifying its high value.

C Practical Implementation of JST

In practice, similar to other data valuation and selection methods [2, 5, 8, 37], we begin by using a neural network trained on the validation set D_v to extract features into a low-dimensional space. This allows us to leverage feature relationships effectively and compute data value scores efficiently. Given that manually cleaned validation set D_v is often small, ResNet-18 model [21] is an appropriate choice for feature extraction. However, in the second round, prior knowledge from the clean validation set D_v can erroneously align the new validation set D_{sel} containing low-quality data with the remaining training samples D'_{tr}, negatively impacting performance. Given our limited access to noisy data, we find that leveraging a pretrained ResNet-18 model on ImageNet1K [38] suffices.

Although, to maintain the perfect match between our algorithm and pure data valuation methods, we utilize the same number $|D_v|$ of backward training data D_{sel} in the second round as the validation set, which always mixes up with a non-obvious proportion of high-quality data. However, we find that such mixing up does not affect the superior performance of our algorithm. Even a small portion of signals from low-quality data is sufficient to pop out high-quality data with low value scores. Therefore, in practice, we can reduce the size of the validation set $|D_{sel}|$ to involve more training samples for selection in the second round without degrading performance.

This section we provide the ablation studies of JST framework instantiated by optimal transport based data valuation method on CIFAR-10 dataset with the noise level of standard deviation 6.0.

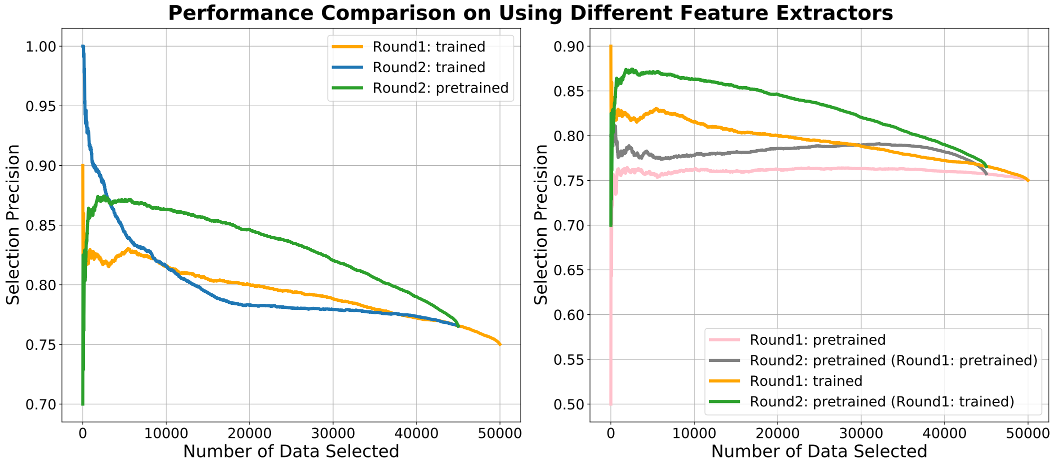


Figure 7: Performance comparison on using different feature extractors in each round. **A:** In the second-round data selection, we have ResNet-18 model trained on the validation set D_v to extract features compared with a pretrained ResNet-18 model on ImageNet1K. The feature extractor of ResNet-18 model trained on the validation set D_v hurts the performance in the second-round data selection. **B:** In both rounds, we have pretrained ResNet-18 model on ImageNet1K to extract features. Our JST framework improves selection precision, though the improvement is less pronounced in the second-round, compared with employing ResNet-18 model trained on the validation set D_v in the first-round data valuation. This confirms that the improvement in our JST framework is not due to switching to a pretrained ResNet-18 model on ImageNet1K for the second-round data selection.

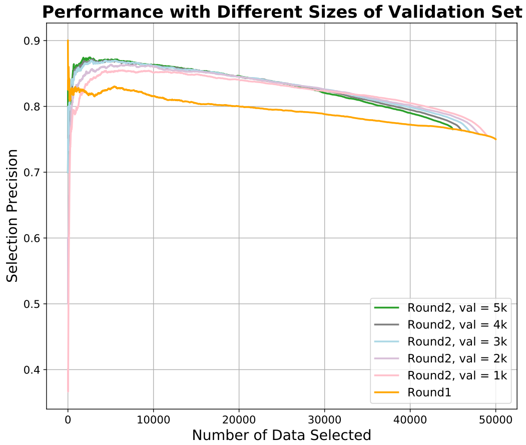


Figure 8: Performance comparison on using different sizes of validation set in the second-round data selection. In the second-round, we ablate different sizes of validation set $|D_{sel}|$ to use, including 5,000, 4,000, 3,000, 2,000, 1,000. The selection precision lines closely align together, indicating the robustness of our JST framework, not requiring many signals from low-quality data to pop out high-quality data.

D JST Framework under Large Distribution Shifts

Although our original goal is to augment existing data valuation with a focus on small noise conditions, we emphasize in this section that our JST framework remains effective under large distribution shifts. In fact, large shifts naturally provide more favorable conditions for both data valuation methods and our framework, making it easier to distinguish between in-distribution data and outliers. Here, instead of contaminating clean images with white noise, we replace 25% of the training data with samples from another dataset as corrupted samples (e.g., mixing 75% CIFAR-10 with 25% MNIST). To eliminate spurious labeling correlations between the clean and corrupted data, we randomly map the labels of the corrupted data to the label space of the clean data. As shown in Figure 9, our JST framework remains effective, and in cases where data valuation methods already perform very well, we at least do not degrade performance.

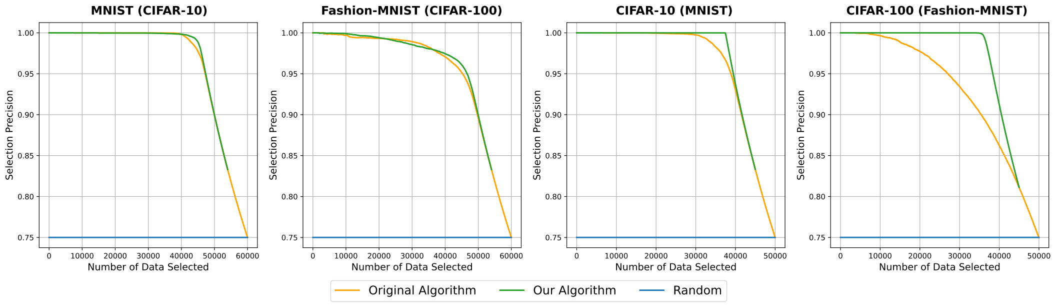


Figure 9: JST framework instantiated by optimal transport based data valuation method under large distribution shifts. Each subfigure title means target dataset (corrupted dataset).

E Where Does the JST Framework Underperform?

In this section, we discuss scenarios where our JST framework underperforms, showing a similar selection precision to the pure data valuation method. Specifically, we observe that marginal contribution based data valuation methods yield unsatisfactory results, as these methods fail to demonstrate a higher detection rate for outliers compared to in-distribution data. We test our framework on three marginal contribution based methods: LOO [5], AME [39], and Data Banzhaf [40]. The experimental setup is the same as in Section 4.3 on CIFAR-10 with white noise standard deviation of 10.0, except the reduced size of training set (5,000) and validation set (500) for LOO to save runtime. As shown in Figure 10 A, these three methods augmented by our framework do not improve selection precision. This is because these methods lack the property of higher sensitivity to outliers (Figure 10 B). Based on this underperformance analysis, we propose a simple test framework to validate the expected behavior of our approach: before applying it to real-world applications, we can assess whether the data valuation method exhibits higher sensitivity to outliers on a specific type of data.

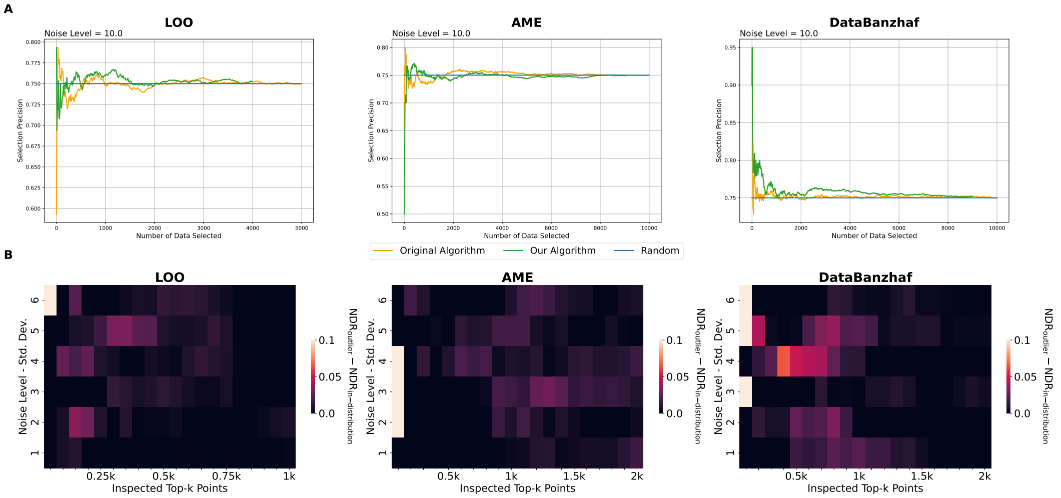
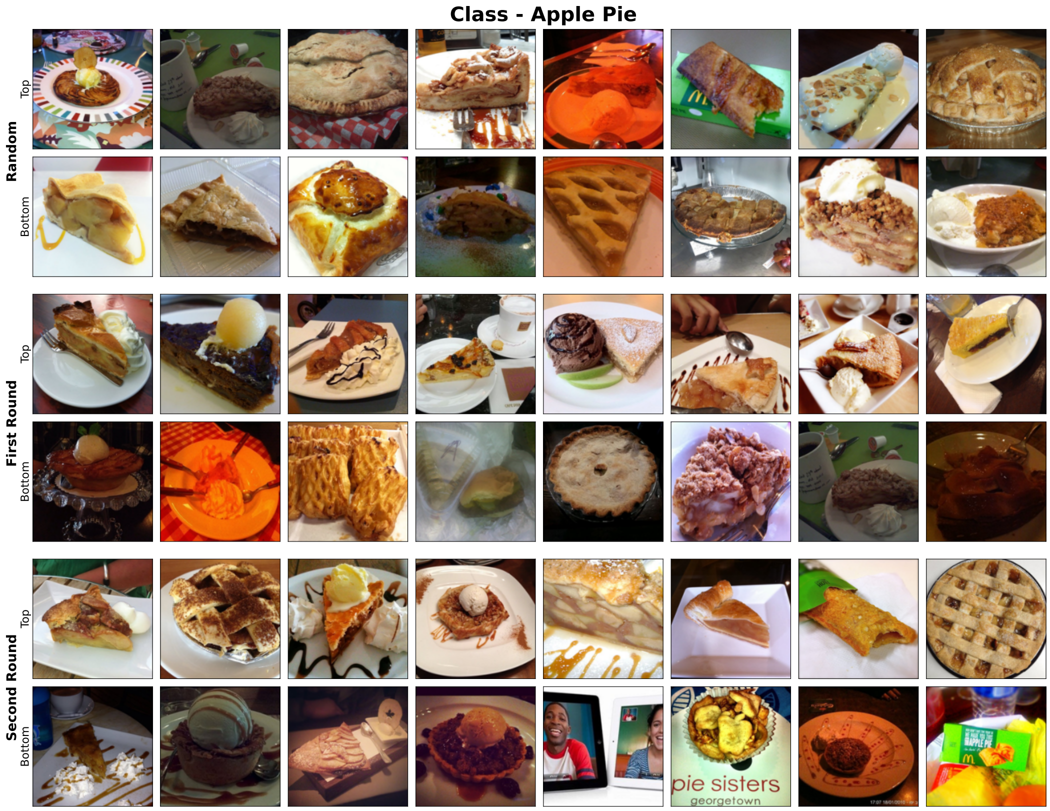
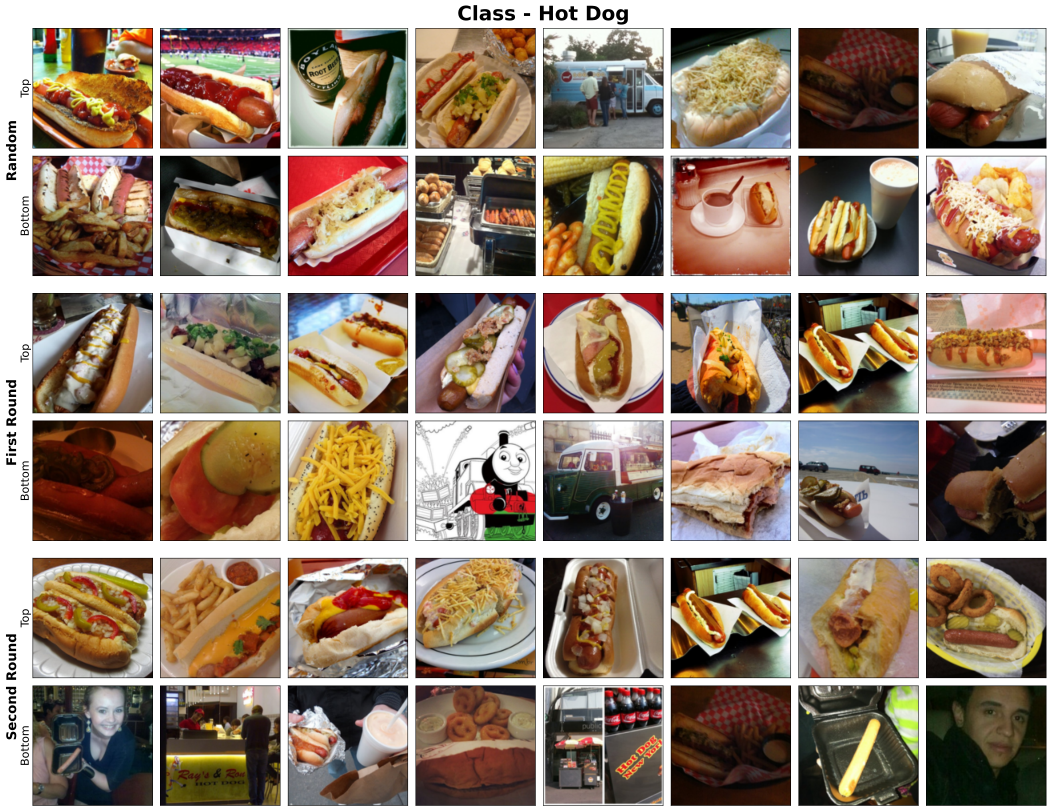


Figure 10: **A:** Evaluating JST framework on marginal contribution based data valuation methods on CIFAR-10. These methods do not improve selection precision compared with the pure data valuation method. **B:** Sensitivity to outliers compared with in-distribution data for marginal contribution based data valuation methods. Following the experimental setup from Figure 1, we evaluate the difference between $\mathrm{NDR}_{\mathrm{outlier}}$ and $\mathrm{NDR}_{\text{in-distribution}}$ on CIFAR-10. The results show that these methods exhibit similar levels of sensitivity to outliers and in-distribution data, which explains the underperformance of our approach when applied to marginal contribution based methods.

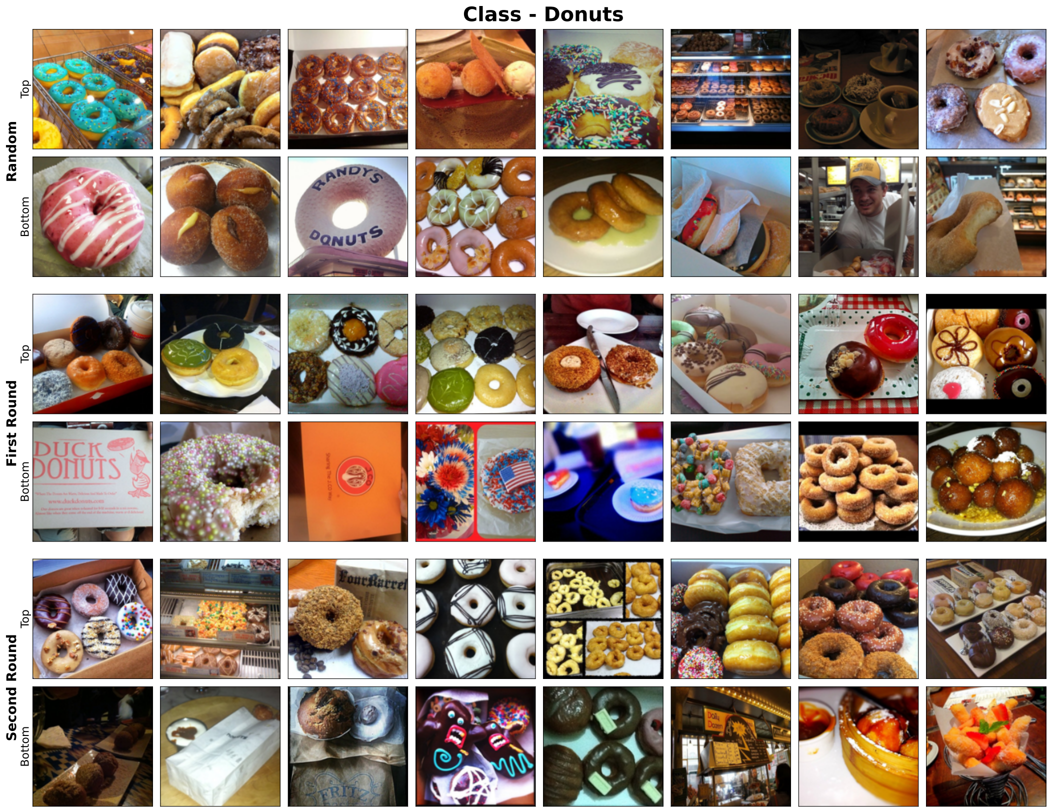
F Extremal Images

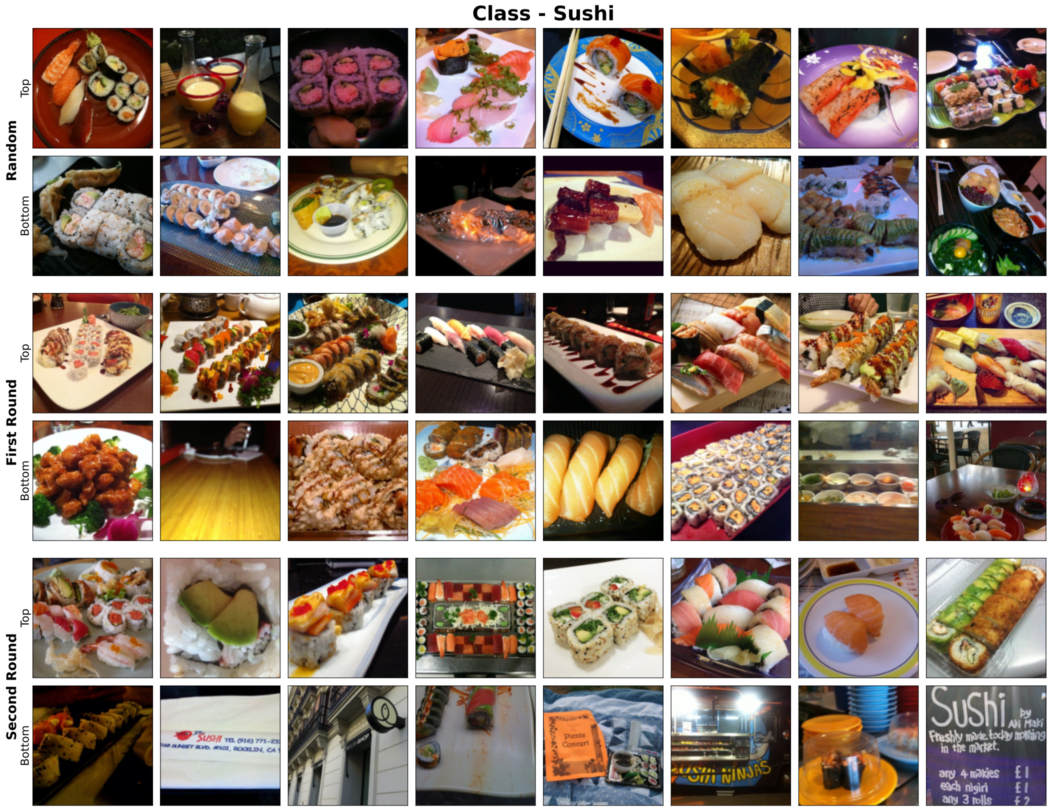


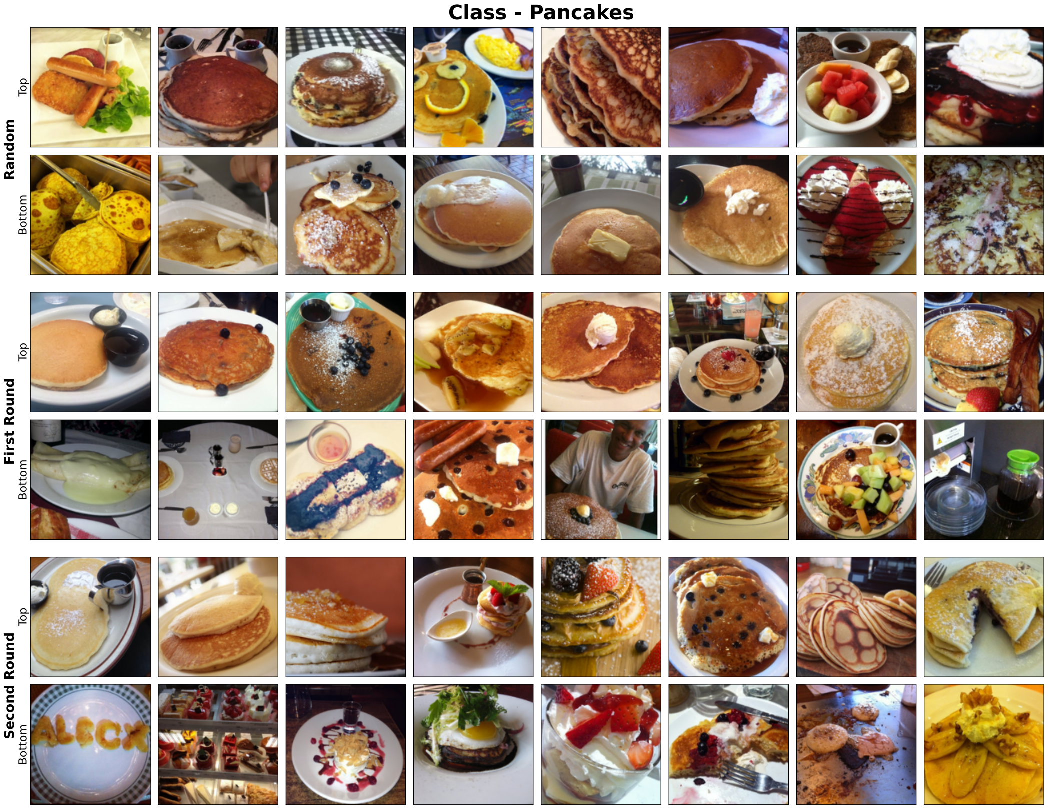
Class - Apple Pie



Class - Hot Dog


Class - Donuts
Random
Top
Bottom
First Round
Top
Bottom
Second Round
Top
Bottom


Class - Sushi
Random
Top
Bottom
First Round
Top
Bottom
Second Round
Top
Bottom


Class - Pancakes
Random
Top
Bottom
First Round
Top
Bottom
Second Round
Top
Bottom