Paper review

- what is your data worth?
- precedence-constrained winter value for effective graph data

John (chenxi) Song

What is your data worth?

Amirata Ghorbani[1] and James Y. Zou

Problems

The importance of Data

- Data is the fuel powering artificial intelligence (such as health-care more depends on data generation).

How to equitably assign value for different data points?

- Solution: **Data Shapley**, a novel framework that assigns value to individual data points in supervised learning setting.

Related Works

Shapley Value:

- applied in voting, resource allocation and bargaining.
- **Applied to feature importance score under predictive models**

Background

Supervised Learning:

- Training set > Learning algorithms > Performance metrics
- Input-output pair data

Shapely Value:

- A solution concept in cooperative game theory
- To each cooperative game it assigns a unique distribution (among the players) of a total surplus generated by the coalition of all players.

Data Shapely

Definition:

- computes the equitable share that each player receives from the cooperation (each source in the train data is a player).

Proposition:

$$\phi_i = C \sum_{S \subseteq D-\{i\}} \frac{V(S \cup \{i\}) - V(S)}{\binom{n-1}{|S|}}$$

1. Data does not change the performance, data value = 0 which is $\Phi_i = 0$.
2. If different data sources have equal contribution for performance, $\Phi 1_i = \Phi 2$
3. All performance score is the sum of separate performance scores $\Phi_i(V + W) = \Phi_i(V) + \Phi_i(W)$

where the sum is over all subsets of D not containing i and C is an arbitrary constant. We call ϕ_i the Data Shapley value of source i

Algorithms for implementation

1. Truncated Monte Carlo Shapley (TMC-Shapley)

   - use Monte Carlo (repeated random sampling) method to estimate Shapley value

   - Truncation: train data↑ performance change little

2. Gradient Shapley (G-Shapley)

   - use stochastic gradient descent (randomly select batches to update)

   - Updating the model on one data point with gradient descent

   - Use hyper-parameter search for learning

Algorithms for implementation

1. Truncated Monte Carlo Shapley (TMC-Shapley)

Algorithm 1 Truncated Monte Carlo Shapley

Input: Train data $D = \{1, \ldots, n\}$, learning algorithm $\mathscr{A}$, performance score V
Output: Shapley value of training points: $\phi_1, \ldots, \phi_n$
Initialize $\phi_i = 0$ for $i = 1, \ldots, n$ and $t = 0$
while Convergence criteria not met **do**
  $t \leftarrow t + 1$
  π^t: Random permutation of train data points
  $v_0^t \leftarrow V(\emptyset, \mathscr{A})$
  for $j \in \{1, \ldots, n\}$ **do**
    if $|V(D) - v_{j-1}^t| <$ Performance Tolerance **then**
      $v_j^t = v_{j-1}^t$
    else
      $v_j^t \leftarrow V(\{\pi^t[1], \ldots, \pi^t[j]\}, \mathscr{A})$
    end if
    $\phi_{\pi^t[j]} \leftarrow \frac{t-1}{t}\phi_{\pi^{t-1}[j]} + \frac{1}{t}(v_j^t - v_{j-1}^t)$
  end for
end while

2. Gradient Shapley (G-Shapley)

Algorithm 2 Gradient Shapley

Input: Parametrized and differentiable loss function $\mathscr{L}(.; \theta)$, train data $D = \{1, \ldots, n\}$, performance score function $V(\theta)$
Output: Shapley value of training points: $\phi_1, \ldots, \phi_n$
Initialize $\phi_i = 0$ for $i = 1, \ldots, n$ and $t = 0$
while Convergence criteria not met **do**
  $t \leftarrow t + 1$
  π^t: Random permutation of train data points
  $\theta_0^t \leftarrow$ Random parameters
  $v_0^t \leftarrow V(\theta_0^t)$
  for $j \in \{1, \ldots, n\}$ **do**
    $\theta_j^t \leftarrow \theta_{j-1}^t - \alpha \nabla_\theta \mathscr{L}(\pi^t[j]; \theta_{j-1})$
    $v_j^t \leftarrow V(\theta_j^t)$
    $\phi_{\pi^t[j]} \leftarrow \frac{t-1}{t}\phi_{\pi^{t-1}[j]} + \frac{1}{t}(v_j^t - v_{j-1}^t)$
  end for
end while

Overview

Potential applications:

- use data shapely as an indicator to measure data sources that relates overall performance.
- Data shapely provides a simple method for domain adaptation.

Experiments:

- Identifying data quality
- Using value to adapt to new data

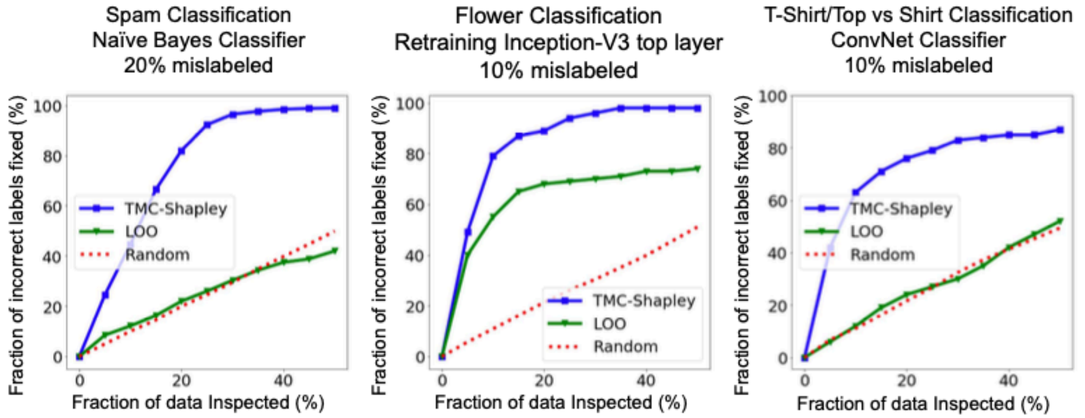
1. Identifying data quality

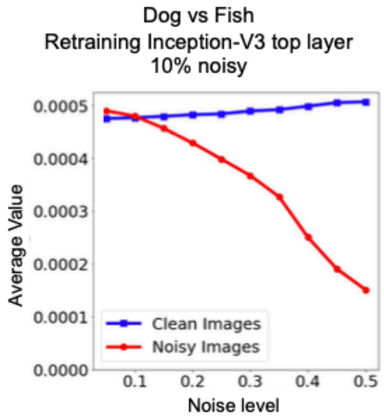
Outline

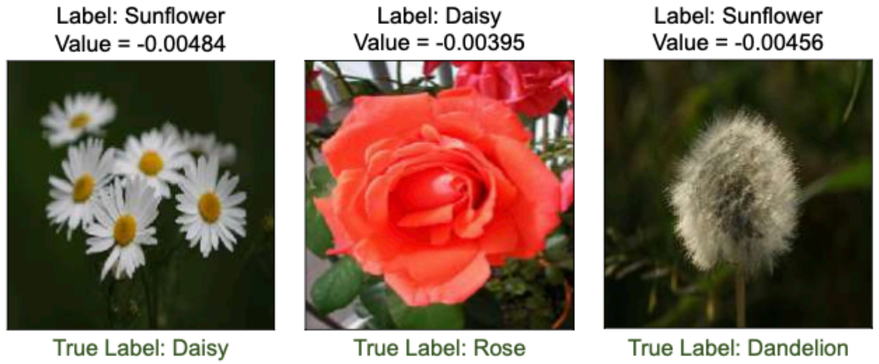
- 1.1 Value of low quality data
- 1.2 All data sources are not created equal

1. 1 Value of low quality data

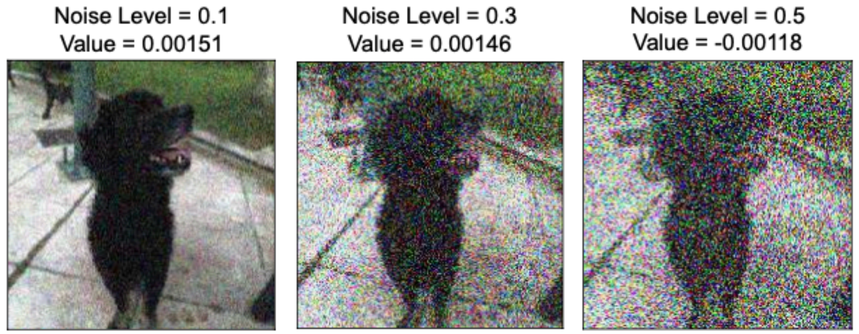
Object: use Data Shapely find the mislabeled data points / noisy data points







(a)



(b)

1. 2 All data sources are not created equal

Object: use Data Shapely find the valuable data in certain context

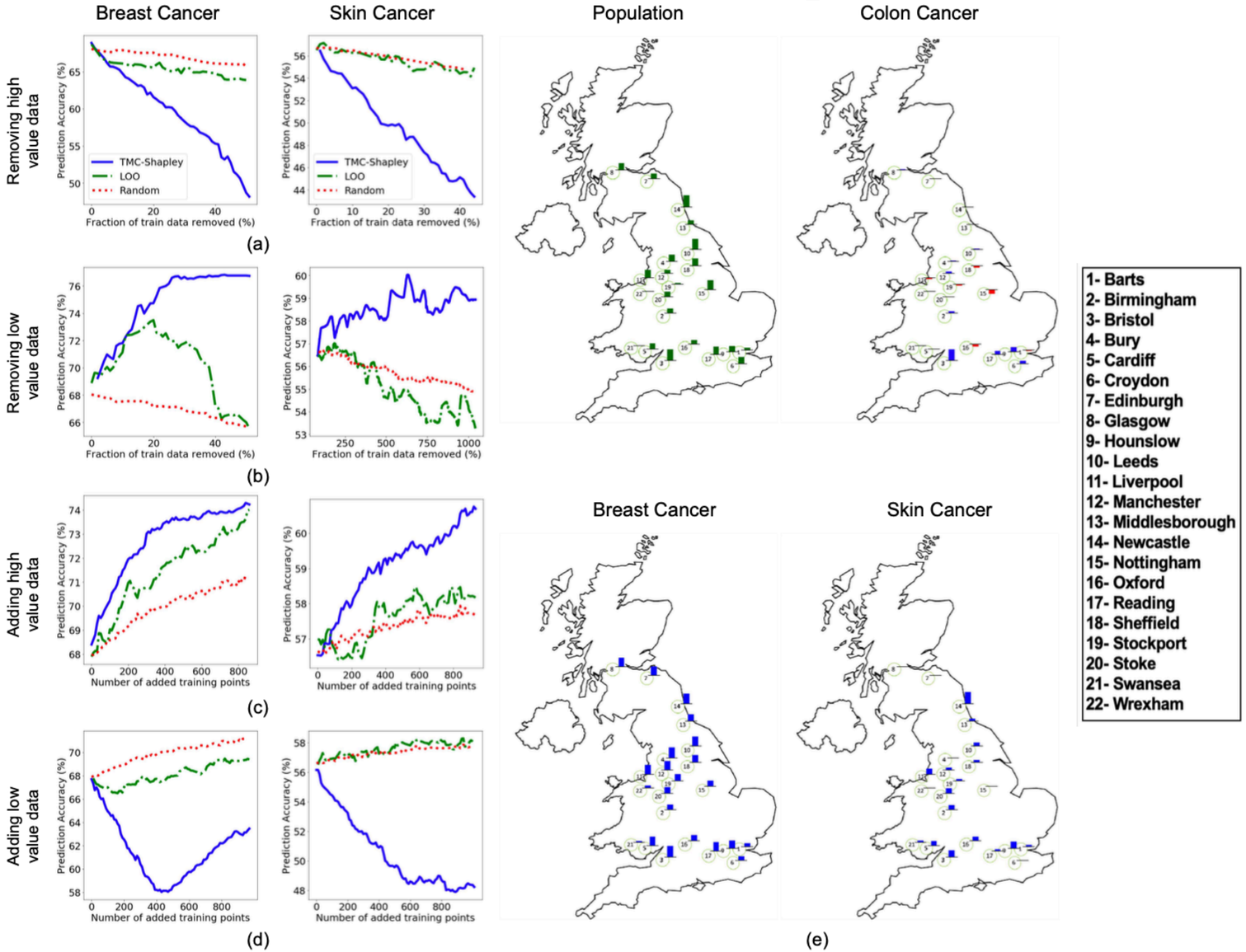
a) removing high value

b) removing low value

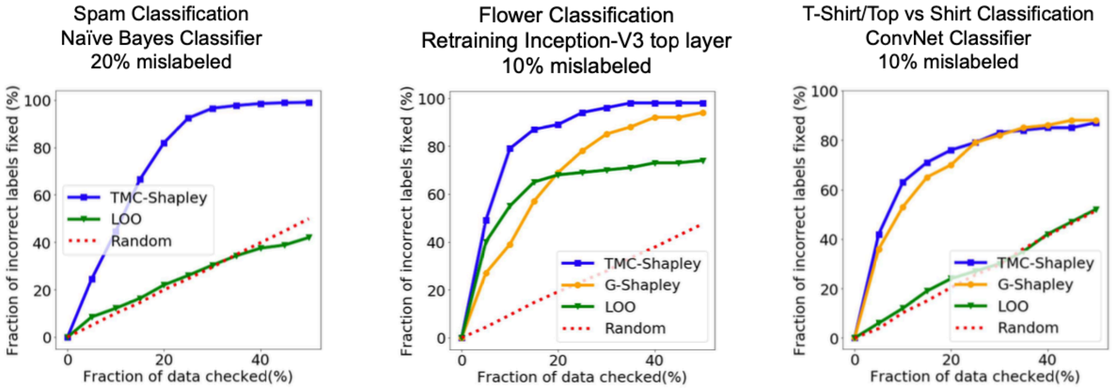
c) adding high value data

d) add low value data

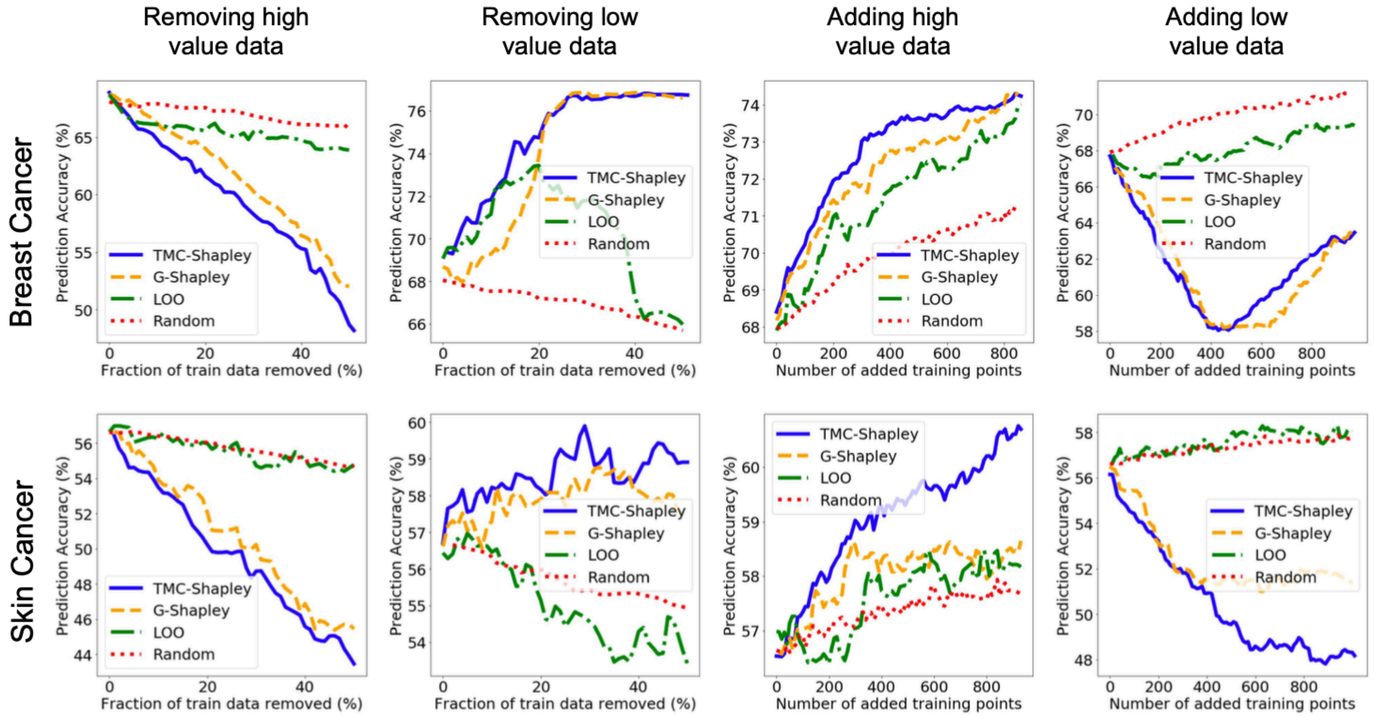
e) evaluate data sources



Observation: removing low shapely value data help with better performance



(a)



(b)

Supplementary Figure 1. G-Shapley algorithm

2. Using value to adapt to new data

Workflow:

Use cheap data to train > computing data shapely > filter training data > retrain

Source to Target	Prediction Task	Trained Model	Original Performance (%)	Adapted Performance (%)
Google to HAM1000	Skin Lesion Classification	Retraining Inception-V3 top layer	29.6	37.8
CSU to PP	Disease Coding	Retraining DeepTag top layer	87.5	90.1
LFW+ to PPB	Gender Detection	Retraining Inception-V3 top layer	84.1	91.5
MNIST to UPS	Digit Recognition	Multinomial Logistic Regression	30.8	39.1
Email to SMS	Spam Detection	Niave Bayes	68.4	86.4

Conclusion

- Under supervised learning setting
- Data Shapley as an equitable framework to quantify the value of individual training sources

References

Articles:

https://en.wikipedia.org/wiki/Shapley_value#:~:text=The%20Shapley%20value%20is%20a,Sciences%20for%20it%20in%202012

https://www.ibm.com/topics/monte-carlo-simulation

Code:

https://github.com/amiratag/DataShapley

Precedence-Constrained Winter Value for Effective Graph Data Valuation

Hongliang Chi, Wei Jin, Charu Aggarwal, Yao Ma

Problems

Many Data valuation for Euclidean data, not for **graph structure data representation**

- 1. Graph data involve both labeled and unlabeled nodes
- 2. Node is interdependent and complex (message-passing) that affect model training
- 3. Computational cost

Solutions:

- **Precedence-Constrained Winter (PC-winter) value** to measure graph data.

Related Works

- Shapley Value (KNN-Shapley, CS-Shapley, Data-OOB…)
- Cooperative games with level structure of cooperation— Winter Value
- Shapley in GNN (GraphSVX, SubgraphX, EdgeSHAPer, GNNShap…)
- Graphs Neural Networks

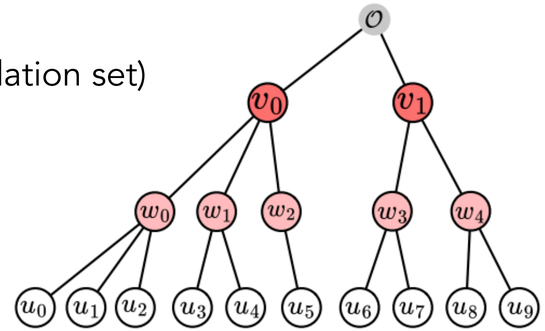
Presupposition

Three Definitions

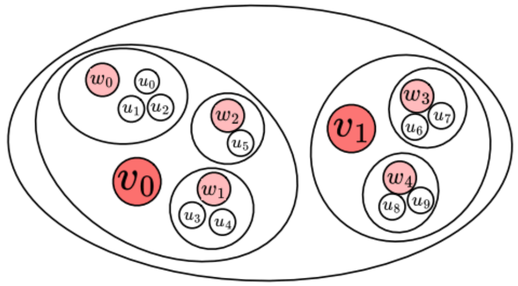
- 1. Computation Tree (K-layer GNN model) : first level of the tree consists of immediate neighbors node and each subsequent level nodes until depth of the tree grows to K.
- 2. Nodes (player set): define as union of nodes in the computation tree of labeled nodes.
- 3. Utility Function:
  - Generate a node-induced graph from computation trees.
  - Use a GNN model to evaluate (held-out validation set)

Three Observations

- 1. Single node (unable and labeled) can impact final representation through feature aggregation
- 2. Level Coalition Structure: a node could serve as “delegate” for other sub-level nodes
- 3. Unilateral Dependency Structure: players contribution is dependent on its ancestors



(a) Contribution Tree



(b) Coalition Structure

Precedence-Constrained Winter value

Constraints:

1. Level Constraint
2. Precedence Constraint

Constraint 1 (Level Constraint)**.** *For any player $p \in \mathcal{P}$, the set of its descendants in the contribution tree is denoted as $\mathcal{D}(p)$. Then, a permutation π aligning with the Level Coalition Structure satisfies the following Level Constraint: $|\pi[i] - \pi[j]| \leq |\mathcal{D}(p)|, \forall i, j \in \mathcal{D}(p) \cup p, \forall p \in \mathcal{P}$, where $\pi[i]$ denotes the positional rank of the i in π.*

Constraint 2 (Precedence Constraint)**.** *A permutation π aligning with the Unilateral Dependency Structure satisfies the following Precedence Constraint: $\pi[p] < \pi[i], \forall i \in \mathcal{D}(p), \forall p \in \mathcal{P}$.*

Precedence-Constrained Winter value

Permissible permutations valuation framework from Shapley value (equation 1 in the paper) within 2 constraints: **PC winter value**

$$\psi_p(\mathcal{P}, U) = \frac{1}{|\Omega|} \sum_{\pi \in \Omega} \left(U\left(\mathcal{P}_p^\pi \cup p\right) - U\left(\mathcal{P}_p^\pi\right) \right), \tag{3}$$

Convergence Criterion. For permutation-based data valuation methods such as `Data Shapley` and `PC-Winter`, we follow convergence criteria similar to the one applied in prior work [13] to determine the number of permutations for approximating data values:

$$\frac{1}{n} \sum_{i=1}^{n} \frac{\left| v_i^t - v_i^{t-20} \right|}{|v_i^t|} < 0.05$$

Precedence-Constrained Winter value

Implementation:

- 1. Generating all permissible permutation (Depth-First Search *preordering*)
- 2. Calculating the PC-Winter value according the equation 3

Challenges:

- 1. All permissible permutations (exponential growth computing resources)
- 2. Retrain Utility function for each permutation
- 3. Feature aggregation computing resource increase as graph’s size bigger

Precedence-Constrained Winter value

Implementation Innovation for efficiency:

- 1. Monte Carlo sampling
  - randomly sample a subset of permissible permutations
- 2. Hierarchical Truncation (DFS for sub trees)
  - The marginal contributions of its late visited child players are insignificant (neighborhood saturation)
  - Insignificant value = 0
- 3. Local Propagation (scalability)
  - Feature aggregation process for the labeled nodes can be done independently within subtree(own computations tree)
  - Perform only necessary sub-trees for labeled node.

Overview

Datasets:

- 6 real-world datasets (Cora, Citeseer, Pubmed, Amazon-Photo, Amazon-Computer, Coauthor-physics.

Modeling:

- node classification task

Datasets

Table 2: Dataset Summary

Dataset	# Node	# Edge	# Class	# Feature	# Train/Val/Test
Cora	2,708	5,429	7	1,433	140 / 500 / 1,000
Citeseer	3,327	4,732	6	3,703	120 / 500 / 1,000
Pubmed	19,717	44,338	3	500	60 / 500 / 1,000
Amazon-Photo	7,650	119,081	8	745	160 / 20% / 20%
Amazon-Computer	13,752	245,861	10	767	200 / 20% / 20%
Coauthor-Physics	34,493	247,962	8	745	100 / 20% / 20%

1.1 Dropping High-value Nodes

Results and Analysis:

- 1. Removal of high-value **unlabeled nodes** results in the significant decline in model performance
- 2. Continue decreasing. (persistent)
- 3. Rebound corresponds to the removal of unlabeled nodes that help improving performance

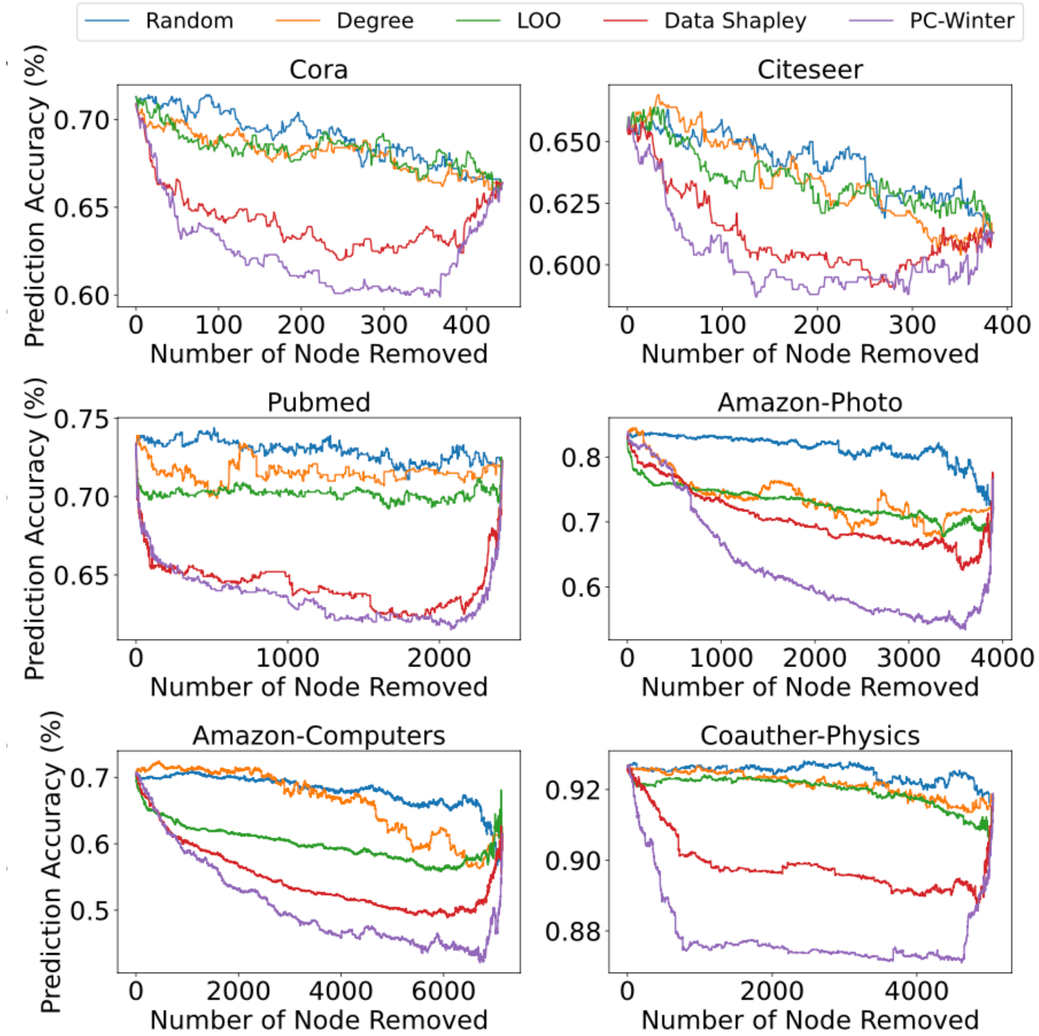


Figure 3: Dropping High-Value Nodes

1.2 Dropping High-value Nodes

Results and Analysis:

- 1. Removal of high-value **mixed labeled and unlabeled nodes.**
- 2. Significant drop reflects removal of labeled node
- 3. Removal of unlabeled nodes has minus impact on performance (little drop or like plateau)

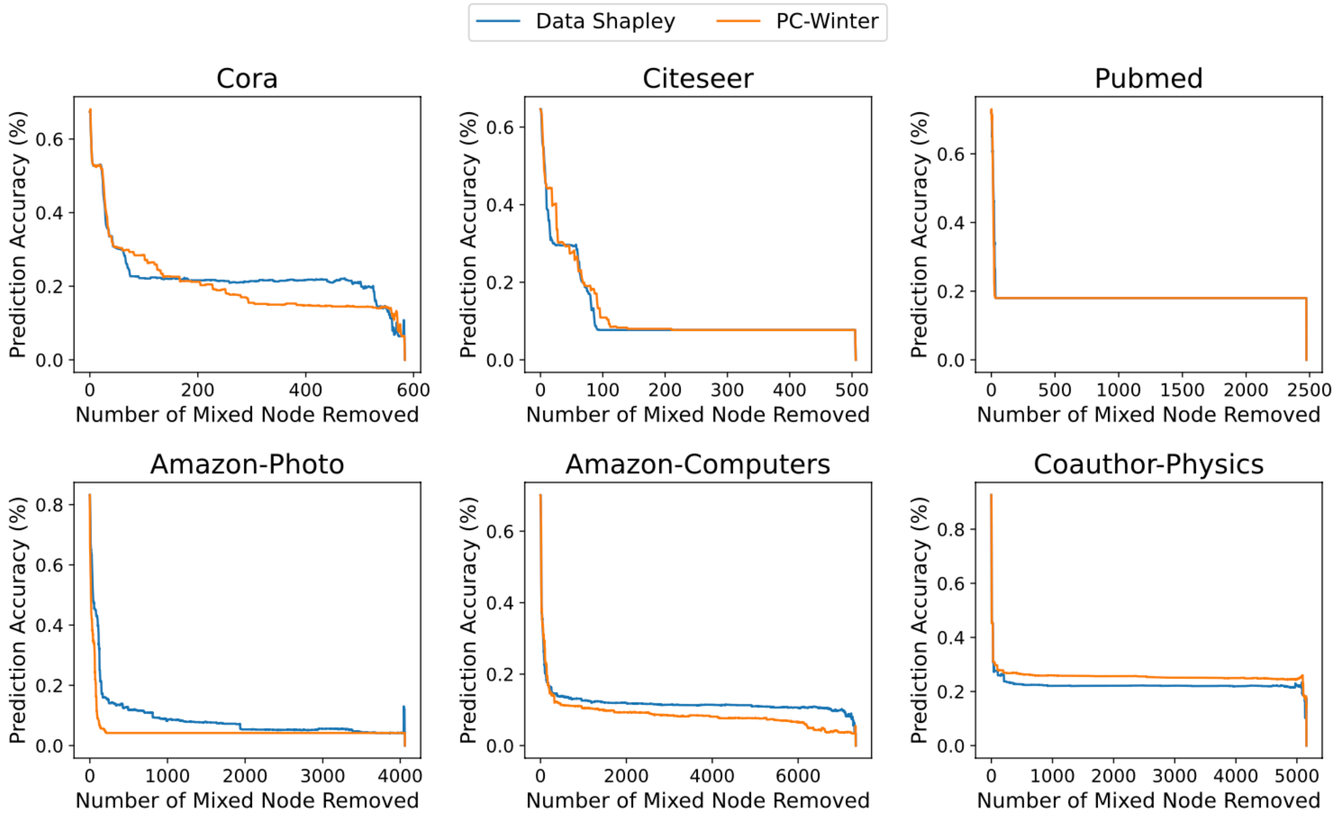


Figure 6: Mixed Node Dropping Experiment

1.3 Dropping High-value Nodes

Results and Analysis:

- 1. Removal of high-value **Labeled nodes**

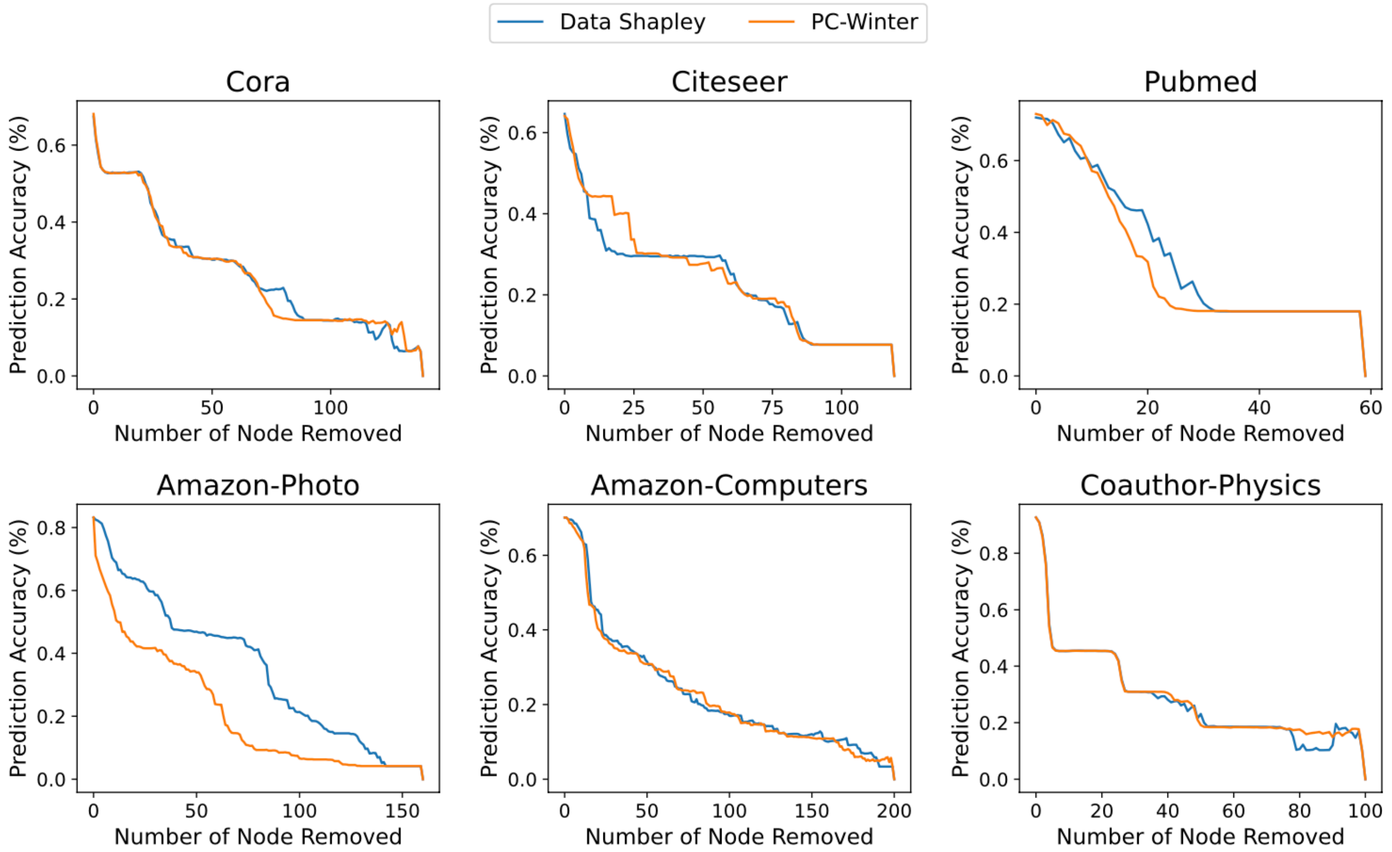


Figure 7: Labeled Node Dropping Experiment

2. Adding High-value Edges

Results and Analysis:

- All baseline model is linear
- PC winter results in a steep performance climb, affirming the PC winter value's efficacy in pinpointing key edges.

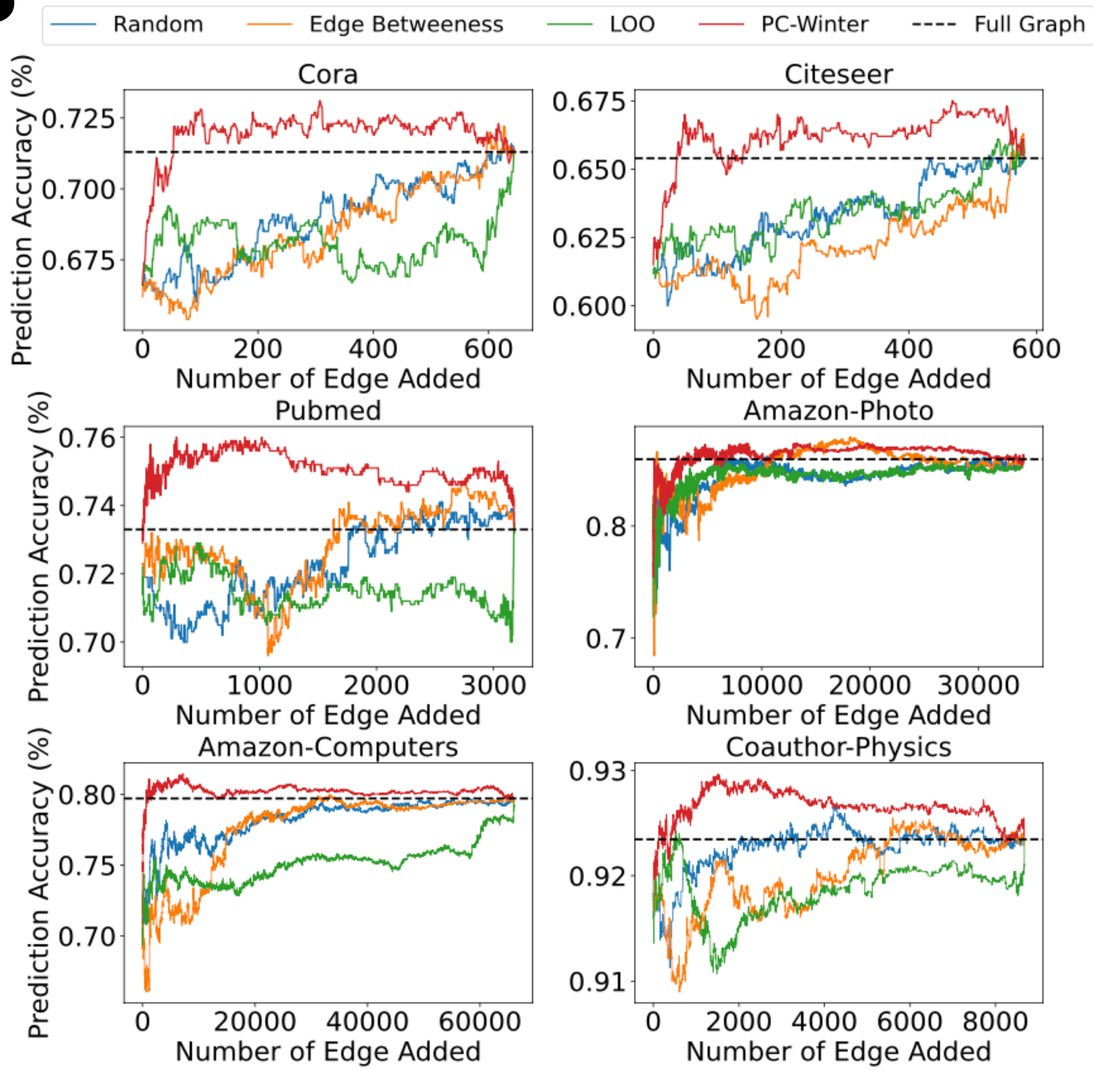


Figure 4: Adding the High-Value Edges

3. Ablation Study

Node-Dropping Experiments

PC-Winter-L:

Satisfying Level Constraint

PC-Winter-P:

Satisfying Precedence Constraint

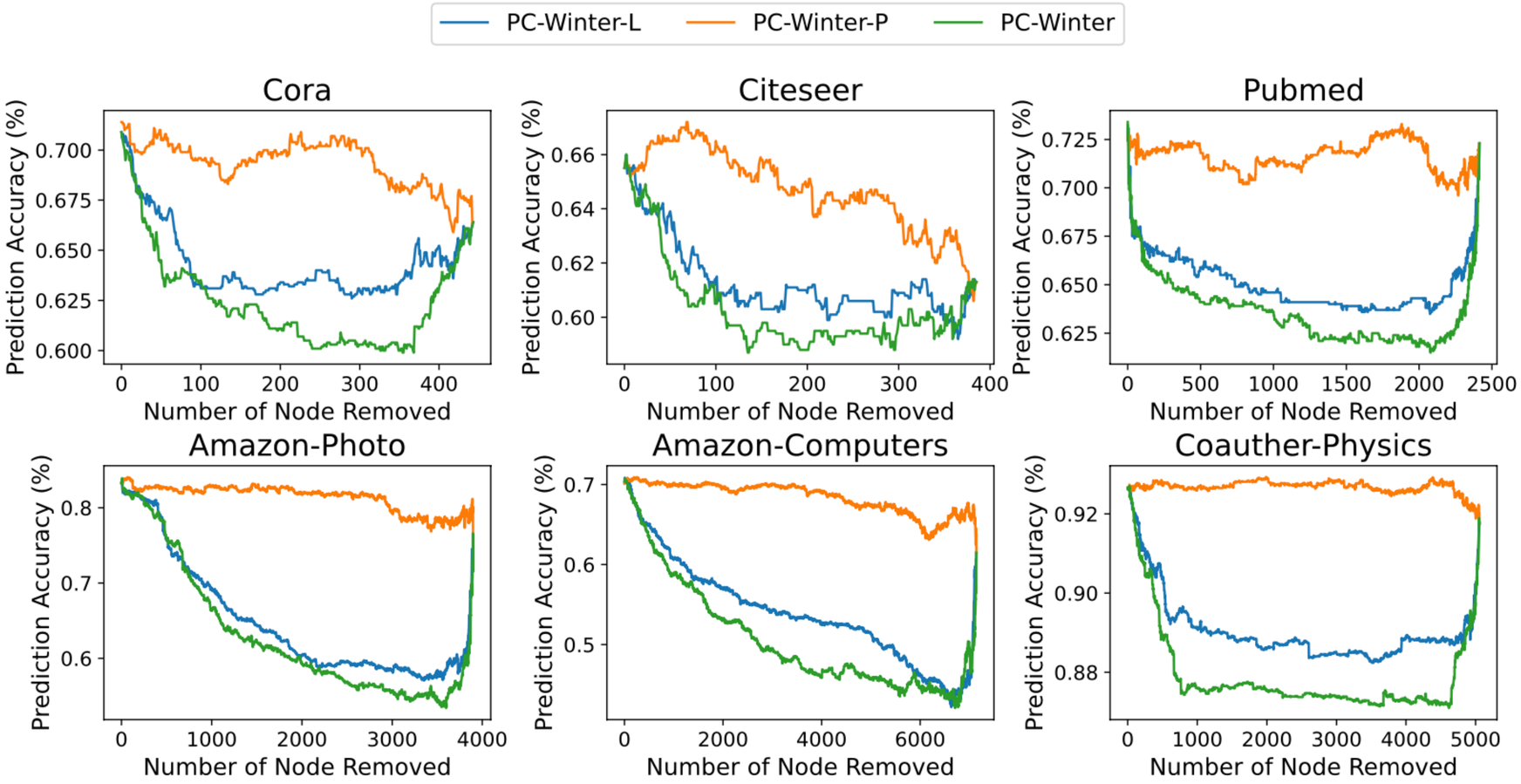


Figure 8: Ablation Study

3. Ablation Study

Node-Dropping Experiments

Increasing the number of permutations generally improves the performance and accuracy of the valuation

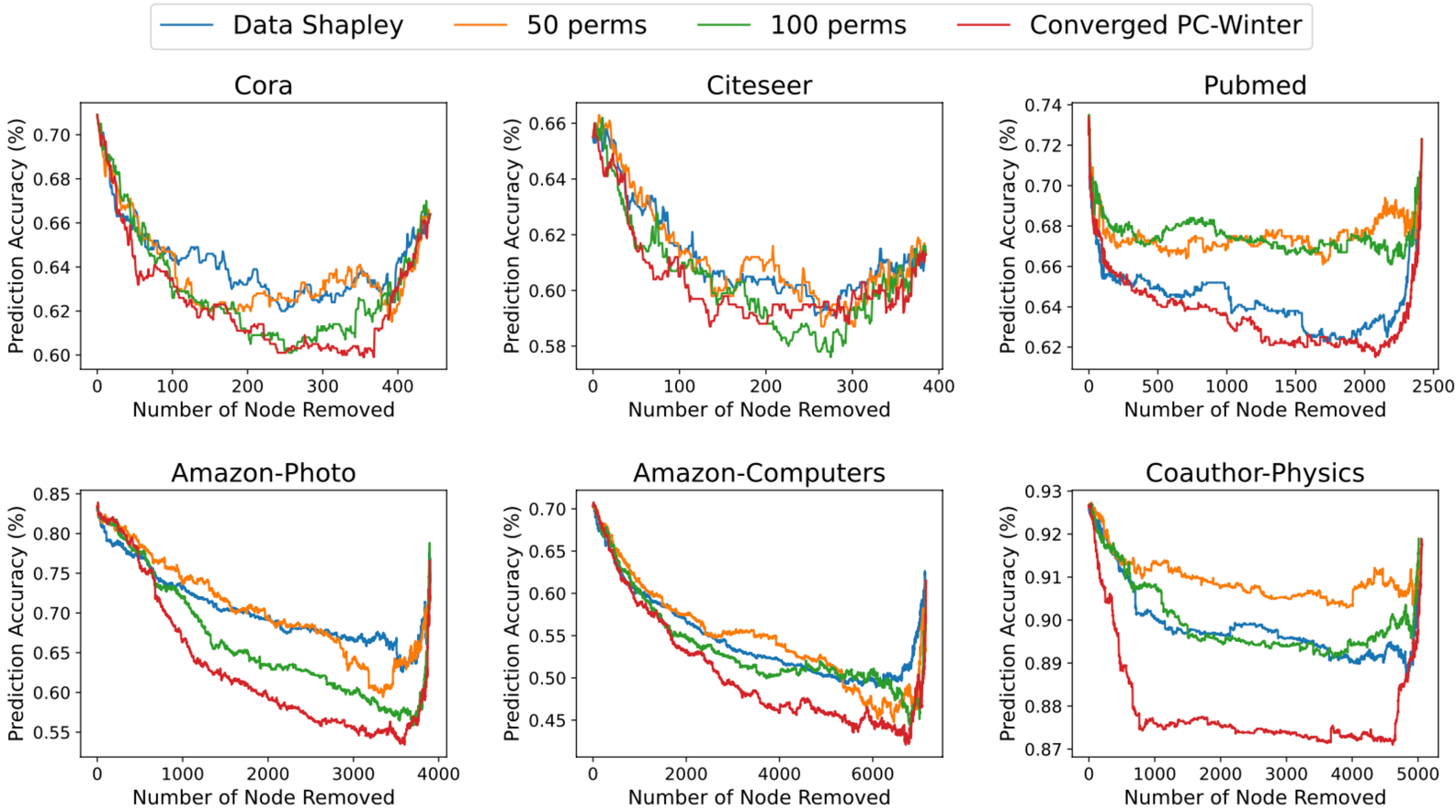


Figure 9: The Impact of Permutation Numbers

3. Ablation Study

Node-Dropping Experiments

- PC winter better than Data Shapley

- large dataset: truncation ratio had a marginal negative effect on performance

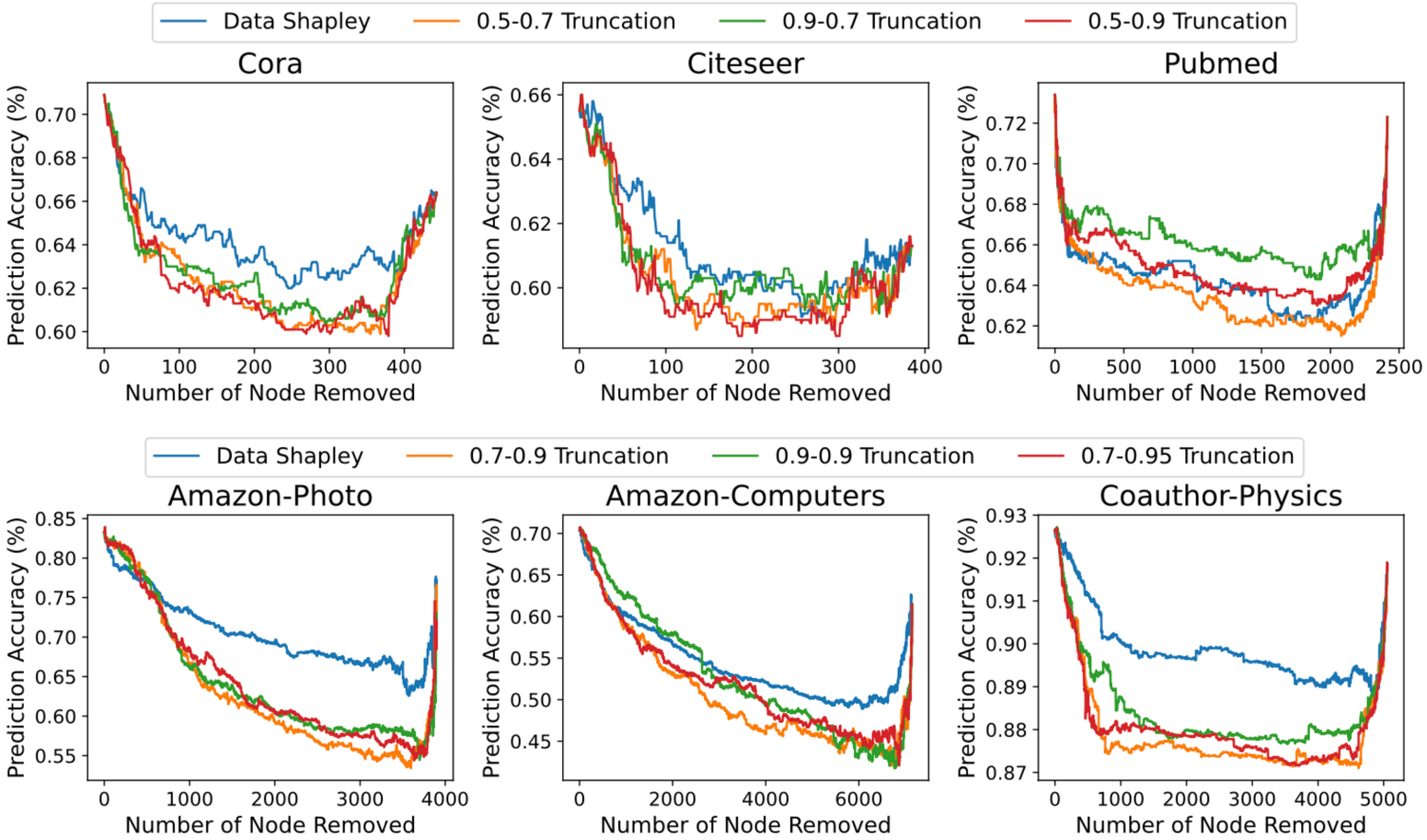


Figure 10: The Impact of Truncation Ratios

Efficiency and Complexity Analysis

PC Winter Algorithms

Table 4: Permutation Number and Time Comparison

Dataset	Truncation	PC-Winter		Data Shapley	
		Perm Number	**Perm Time (hrs)**	**Perm Number**	**Perm Time (hrs)**
Cora	0.5-0.7	325	0.013	327	0.024
Citeseer	0.5-0.7	291	0.018	279	0.037
Pubmed	0.5-0.7	316	0.025	281	0.285
Amazon-Photo	0.7-0.9	418	0.211	109	1.105
Amazon-Computer	0.7-0.9	181	0.662	33	3.566
Coauthor-Physics	0.7-0.9	460	0.119	45	2.642

Complexity:

$$O(L \cdot N_{\text{trun}} \cdot (\frac{N_{\text{trun}}}{2} \cdot F + F^2))$$

Conclusion

PC-Winter is an innovative approach for effective graph data valuation

Introduce a set of strategies for reducing the computational cost, enabling efficient approximation of PC-winter

References

Paper

Chi, H., Jin, W., Aggarwal, C., & Ma, Y. (2024). Precedence-Constrained Winter Value for Effective Graph Data Valuation. *arXiv preprint arXiv:2402.01943*.

Code:

https://anonymous.4open.science/r/graph-data-valuation-B348