GoodCore: **Data-effective** and **Data-efficient** Machine Learning through **Coreset Selection** over **Incomplete Data**

Chengliang Chai
Beijing Institute of Technology

Jiabin Liu
Beijing Institute of Technology

Nan Tang
QCRI, Qatar

Ju Fan
Renmin University of China

Dongjing Miao
Harbin Institute of Technology

Jiayi Wang
Tsinghua University

Yuyu Luo
Tsinghua University
(**Presenter**)

Guoliang Li
Tsinghua University

Data-effective ML

- Data cleaning: obtain high-quality data
  - Inconsistency
  - Outlier
  - Miss Labels
  - Missing Values

ML Model (Acc. : 96%)

Name	Gender	Depart.	Age	Working Years
Nancy	F	Sales	24	2
Amy	M	Finance	42	10
Lei	M	Market	35	2
Lily	F	Market	36	8
James	M	Sales	38	2
Tom	M	Finance	36	6

High-quality Data

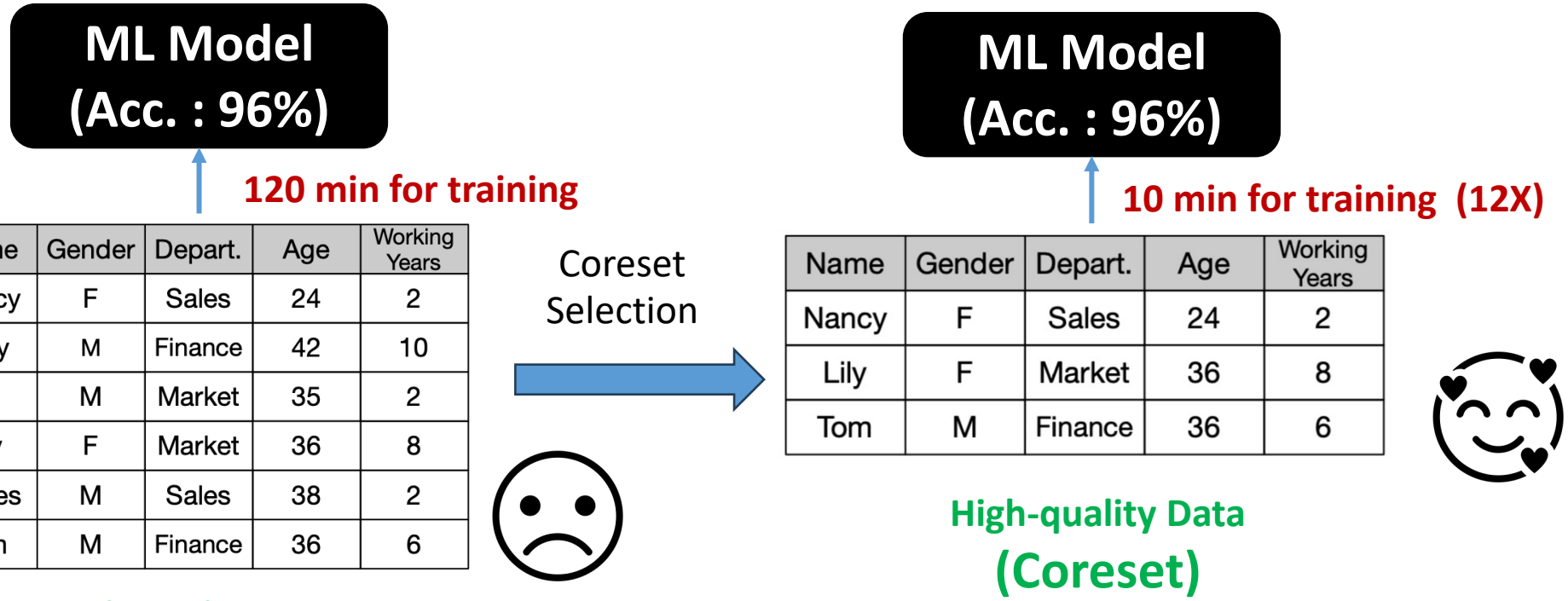
ML Model (Acc. : 87%)

Name	Gender	Depart.	Age	Working Years
Nancy	F	Sales	24	2
Amy		Finance	42	10
Lei	M		35	
Lily	F	Market		8
James	M	Sales	38	2
Tom	M		36	6

Low-quality Data (Missing Value)

Data-efficient ML

- **Algorithm**:
  - Stochastic Gradient Decent (SGD)
- **Data**:
  - **Coreset**: Given a large train set *D*, Coreset is a core subset of *D*, denoted by *C(D)*, which is selected to represent *D* such that M(*C(D)*)≈M(*D*).



Our Goal

Our goal is to achieve both *data-efficienct* and *data-effective* ML !

Name	Gender	Depart.	Age	Working Years
Nancy	F	Sales	24	2
Amy		Finance	42	10
Lei	M		35	
Lily	F	Market		8
James	M	Sales	38	2
Tom	M		36	6

Low-quality Data (Missing Value)

Coreset Selection Over Incomplete Data

ML Model (Acc. : 96%)

10 min for training (12X)

Name	Gender	Depart.	Age	Working Years
Nancy	F	Sales	24	2
Lily	F	Market	36	8
Tom	M	Finance	36	6

High-quality Data (Coreset)

Introduction to coreset selection

- *How to select a coreset (a small subset) that has the same performance theoretically with the full?*

- Gradient Descent

$$\theta^* = \arg\min_{\theta \in \vartheta} f(\theta), f(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta, t_i) \tag{1}$$

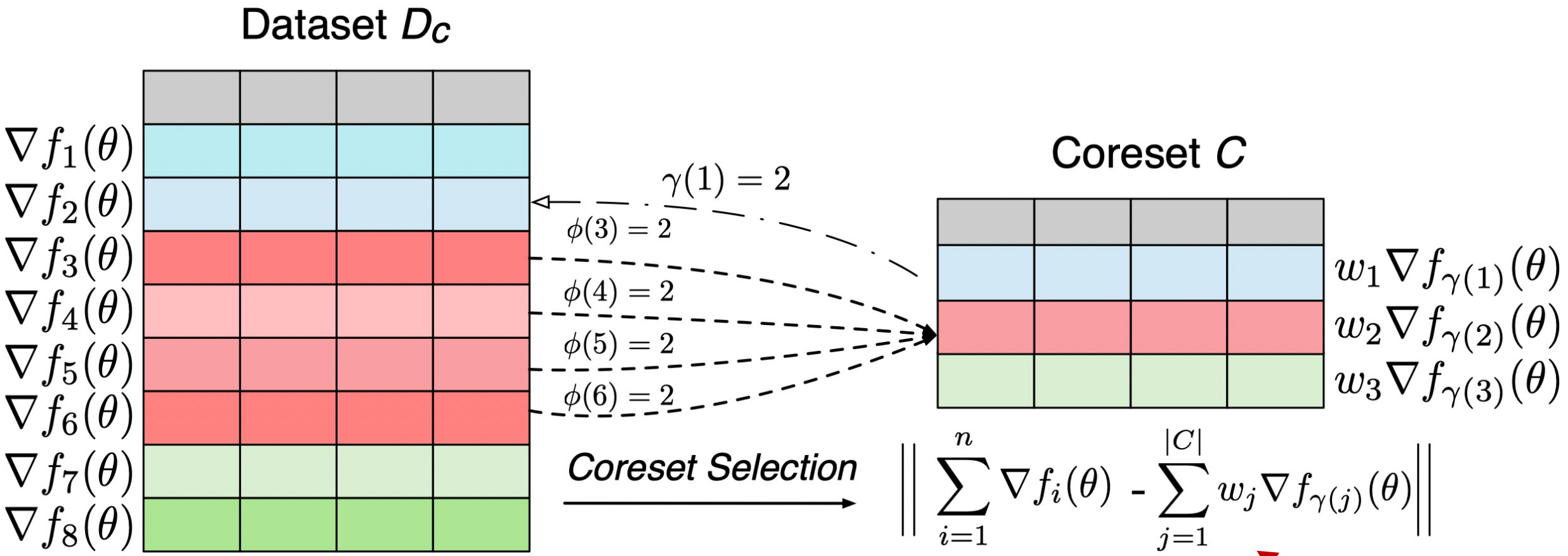
- Gradient-based coreset selection

$$C^* = \arg\min_{C \subseteq D_c, w_j \geq 0} \max_{\theta \in \vartheta} \| \underbrace{\sum_{i=1}^{n} \nabla f_i(\theta)}_{\text{full gradient}} - \underbrace{\sum_{j=1}^{|C|} w_j \nabla f_{\gamma(j)}(\theta)}_{\text{coreset gradient}} \|, \tag{2}$$

$$\textbf{gradient approximation error}$$

$$s.t.\ |C| \leq K$$

Example of coreset selection



$\phi(i) = j$ indicates the mapping from D_C to C

w_j is the weight of each tuple in the coreset.

Gradient-based coreset selection

- Minimize the gradient approximation error $\|\sum_{i=1}^{n} \nabla f_i(\theta) - \sum_{j=1}^{|C|} w_j \nabla f_{\gamma(j)}(\theta)\|$
- Triangle equation (Upper bound minimization of error)

$$\|\sum_{i=1}^{n} \nabla f_i(\theta) - \sum_{j=1}^{|C|} w_j \nabla f_{\gamma(j)}(\theta)\| \leq \sum_{i=1}^{n} \|\nabla f_i(\theta) - \nabla f_{\gamma(\phi_\theta(i))}(\theta)\|$$

- Assign every tuple to the tuple in *C* with most gradient similarity

$$\|\sum_{i=1}^{n} \nabla f_i(\theta) - \sum_{j=1}^{|C|} w_j \nabla f_{\gamma(j)}(\theta)\| \leq \sum_{i=1}^{n} \min_{c_j \in C} \|\nabla f_i(\theta) - \nabla f_{\gamma(j)}(\theta)\|$$

- For convex problem, the bound of gradiet difference of a pair of tuples.

$$\forall i, j, \max_{\theta \in \vartheta} \|\nabla f_i(\theta) - \nabla f_j(\theta)\| \leq \max_{\theta \in \vartheta} O(\|\theta\|) \cdot \|\mathbf{x}_i - \mathbf{x}_j\|$$

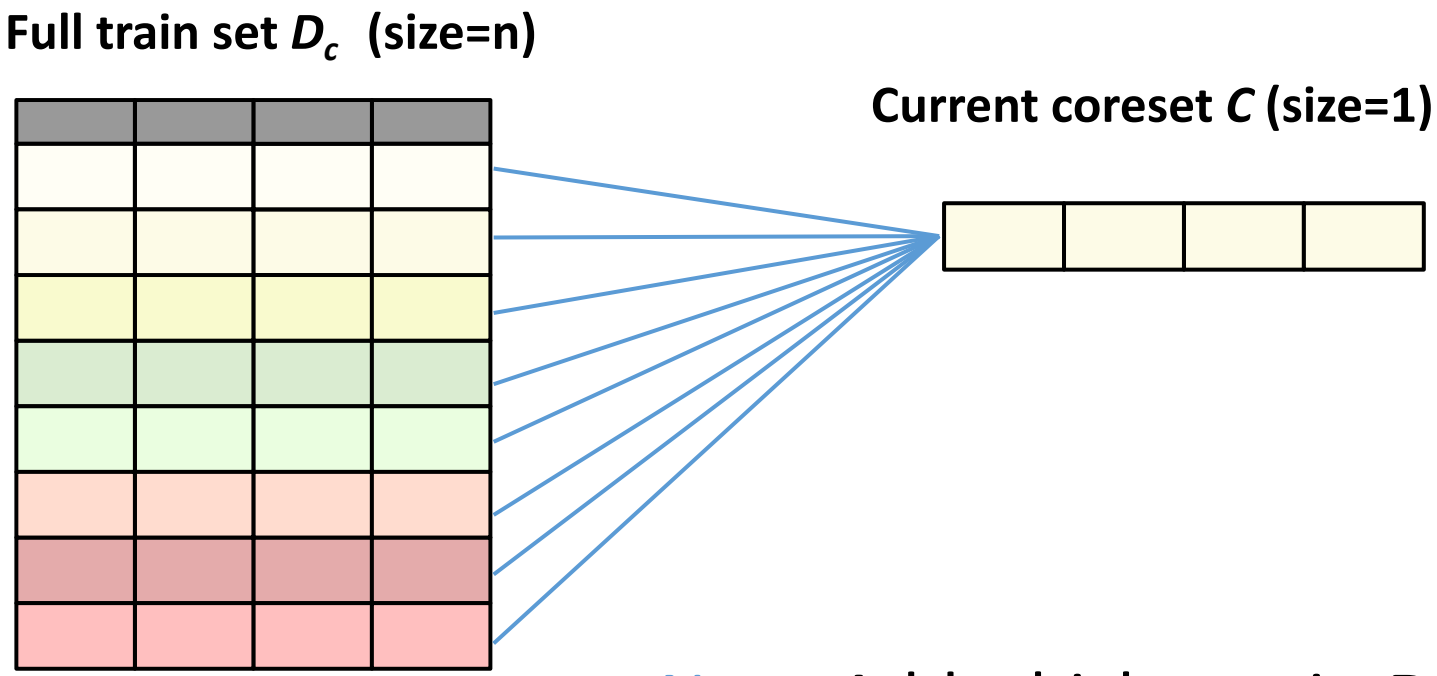
- Problem definition

$$C^* = \arg\min_{C \subseteq D_c} \sum_{i=1}^{n} \min_{c_j \in C} \|\mathbf{x}_i - \mathbf{x}_{\gamma(j)}\|$$

Greedy Solution

$$C^* = \arg\min_{C \subseteq D_c} \sum_{i=1}^{n} \min_{c_j \in C} \|\mathbf{x}_i - \mathbf{x}_{\gamma(j)}\|$$

- Greedy solution with approximate ratio (sub-modular):

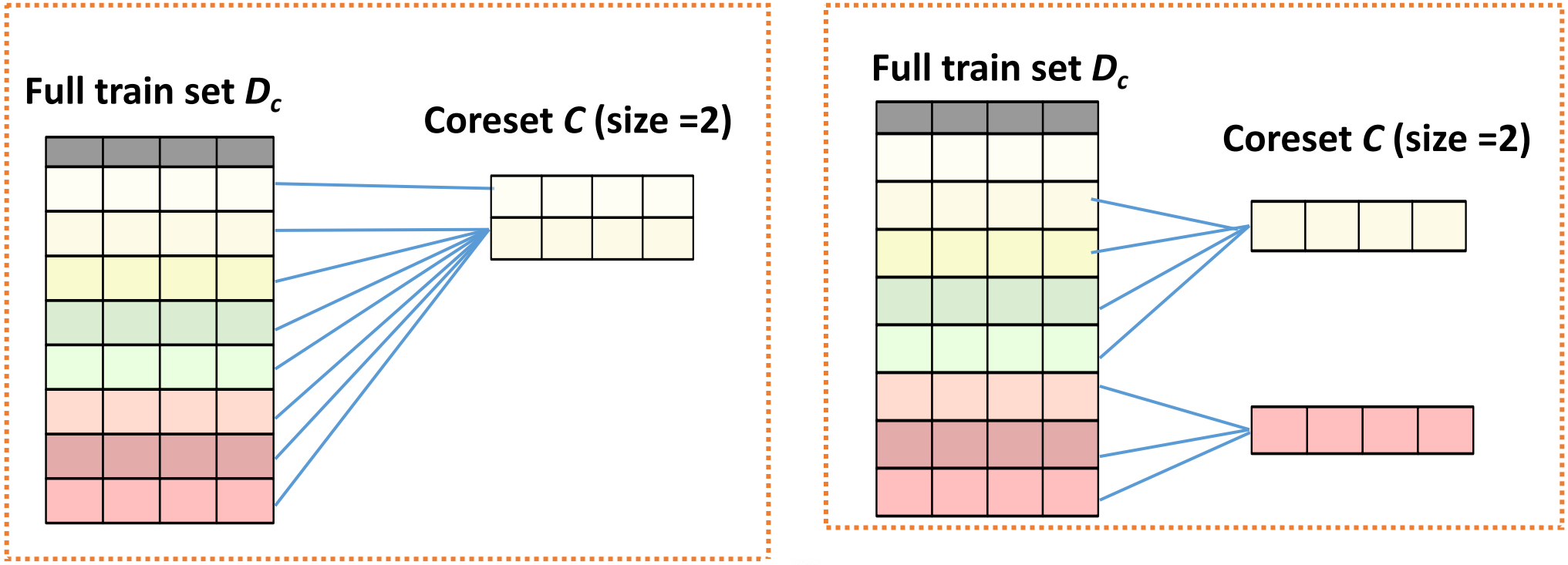


Next: Add which row in $\boldsymbol{D_c}$ to $\boldsymbol{C}$?

Greedy Solution

$$C^* = \arg\min_{C \subseteq D_c} \sum_{i=1}^{n} \min_{c_j \in C} \|\mathbf{x}_i - \mathbf{x}_{\gamma(j)}\|$$

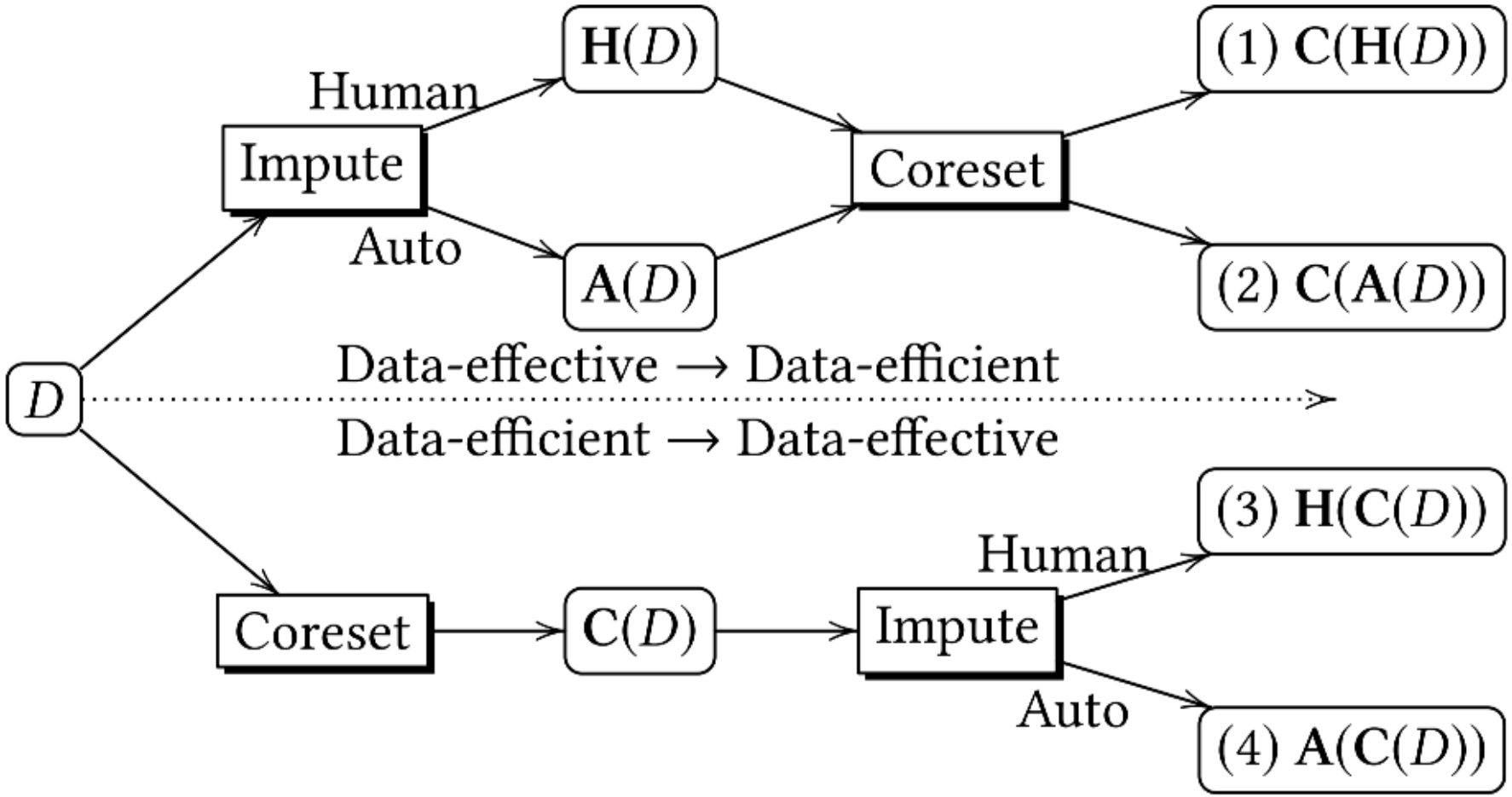
Next: Add which row in D_c to C ?



➢ Compute $\sum_{i=1}^{n} \min_{c_j \in C} \|\mathbf{x}_i - \mathbf{x}_{\gamma(j)}\|$ ✅

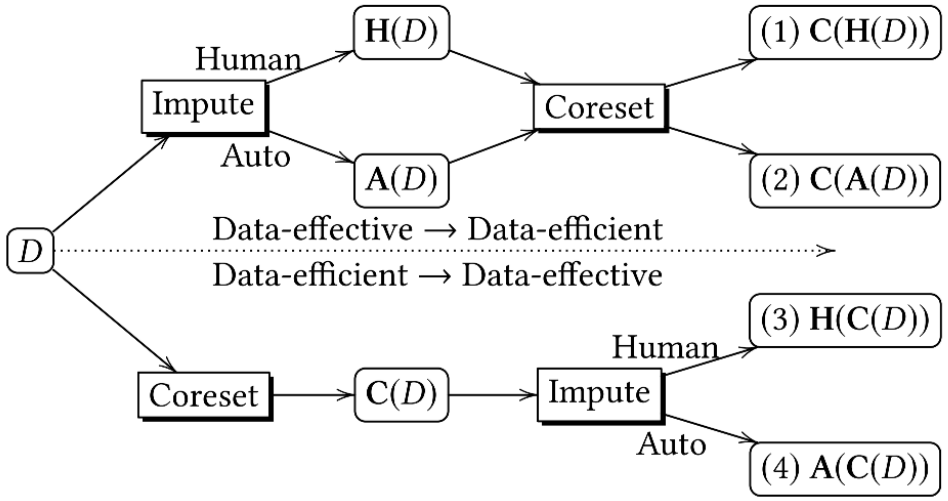
Coreset Selection over Incomplete Data

Intuitive baselines (sequential)



- *First data-effective (impute) then data-efficient (coreset):*
  - (1) Impute-Human: $\mathbf{H}(D) \rightarrow$ Coreset: $\mathbf{C}(\mathbf{H}(D))$
  - (2) Impute-Auto: $\mathbf{A}(D) \rightarrow$ Coreset: $\mathbf{C}(\mathbf{A}(D))$
- *First data-efficient (coreset) then data-effective (impute):*
  - (3) Coreset: $\mathbf{C}(D) \rightarrow$ Impute-Human: $\mathbf{H}(\mathbf{C}(D))$
  - (4) Coreset: $\mathbf{C}(D) \rightarrow$ Auto-Human: $\mathbf{A}(\mathbf{C}(D))$

Intuitive baselines (sequential)



Solution	Accuracy	Human Cost	Machine Cost
(1) C(H(*D*))	High	High	Low
(2) C(A(*D*))	Low	None	Low
(3) H(C(*D*))	Low	Low	Low
(4) A(C(*D*))	Low	None	Low

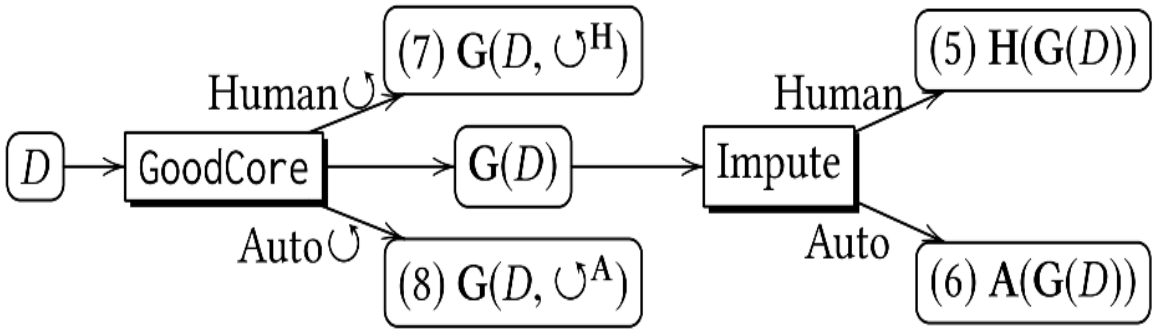
Challenge

- Computing a good coreset from dirty data is to *accurately estimate the ground truth* of each missing value, which has multiple possible repairs.
- The *combinations of all possible repairs* constitute a huge search space.

Key idea of our Solutions (GoodCore)

- Model the combinations of possible repairs as *possible worlds* of the original dirty data
- Selecting *an expected optimal coreset* that can *represent the possible worlds of D* by gradient approximation without training in advance

D ⟶ G(D): Very Time-consuming



Solution	Accuracy	Human Cost	Machine Cost
(1) C(H(*D*))	High	High	Low
(2) C(A(*D*))	Low	None	Low
(3) H(C(*D*))	Low	Low	Low
(4) A(C(*D*))	Low	None	Low
Our goal	High	None or Low	Low
(5) H(G(*D*))	High	Low	High
(6) A(G(*D*))	Medium	None	High
(7) G(*D*, ↻H)	High	Low	Low
(8) G(*D*, ↻A)	Medium	None	Low

An Example: Expected Optimal Coreset

- Problem Definition:

$$C^* = \arg\min_{C \subseteq D} \mathrm{E}[C], \text{ s.t. } |C| \leq K$$

$$\mathrm{E}[C] = \sum_{k=1}^{|\mathcal{I}_W|} p_k S_k$$

Dirty Dataset D

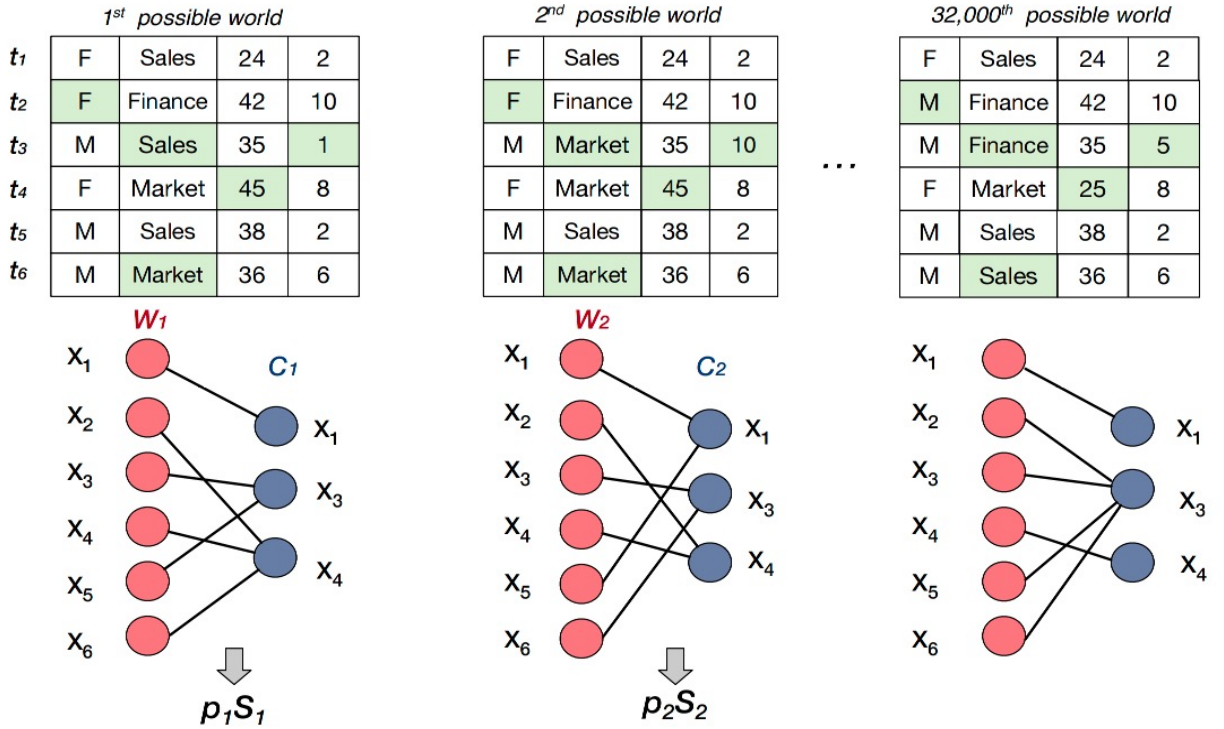
	Name	Gender	Depart.	Age	Working Years
t_1	Nancy	F	Sales	24	2
t_2	Amy		Finance	42	10
t_3	Lei	M		35	
t_4	Lily	F	Market		8
t_5	James	M	Sales	38	2
t_6	Tom	M		36	6

A Coreset C

	Name	Gender	Depart.	Age	Working Years	
c_1	Nancy	F	Sales	24	2	t_1
c_2	Lei	M		35		t_3
c_3	Lily	F	Market		8	t_4

$\gamma(1) = 1 \quad \gamma(2) = 3 \quad \gamma(3) = 4$

(a) An example of dataset with missing values and a coreset



NP-hard with sub-modular property: can be solved by a greedy algorithm

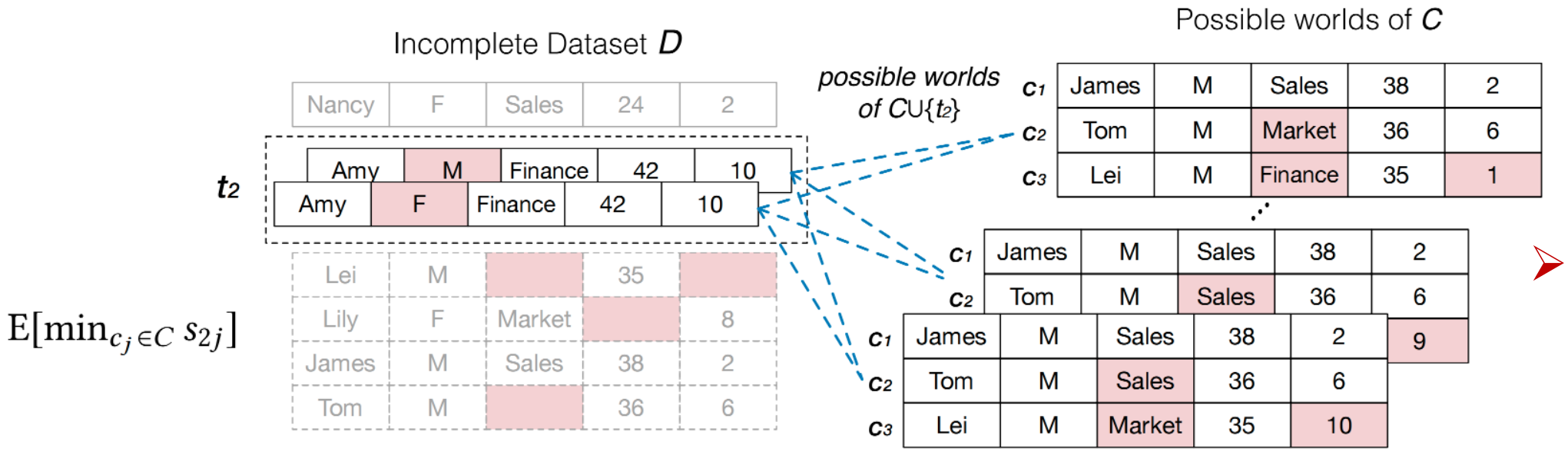
Expectation Computation

- Brute-force method： *prohibitively expensive to compute*

$$\mathrm{E}[C] = \sum_{k=1}^{|\mathcal{I}_W|} p_k \left(\sum_{i=1}^{n} \min_{c_j \in C_k} s_{ij}\right).$$

- Tuple-based expectation computation：

$$\mathrm{E}[C] = \sum_{k=1}^{|\mathcal{I}_W|} p_k \left(\sum_{i=1}^{n} \min_{c_j \in C_k} s_{ij}\right) = \sum_{i=1}^{n} \mathrm{E}\left[\min_{c_j \in C} s_{ij}\right]$$

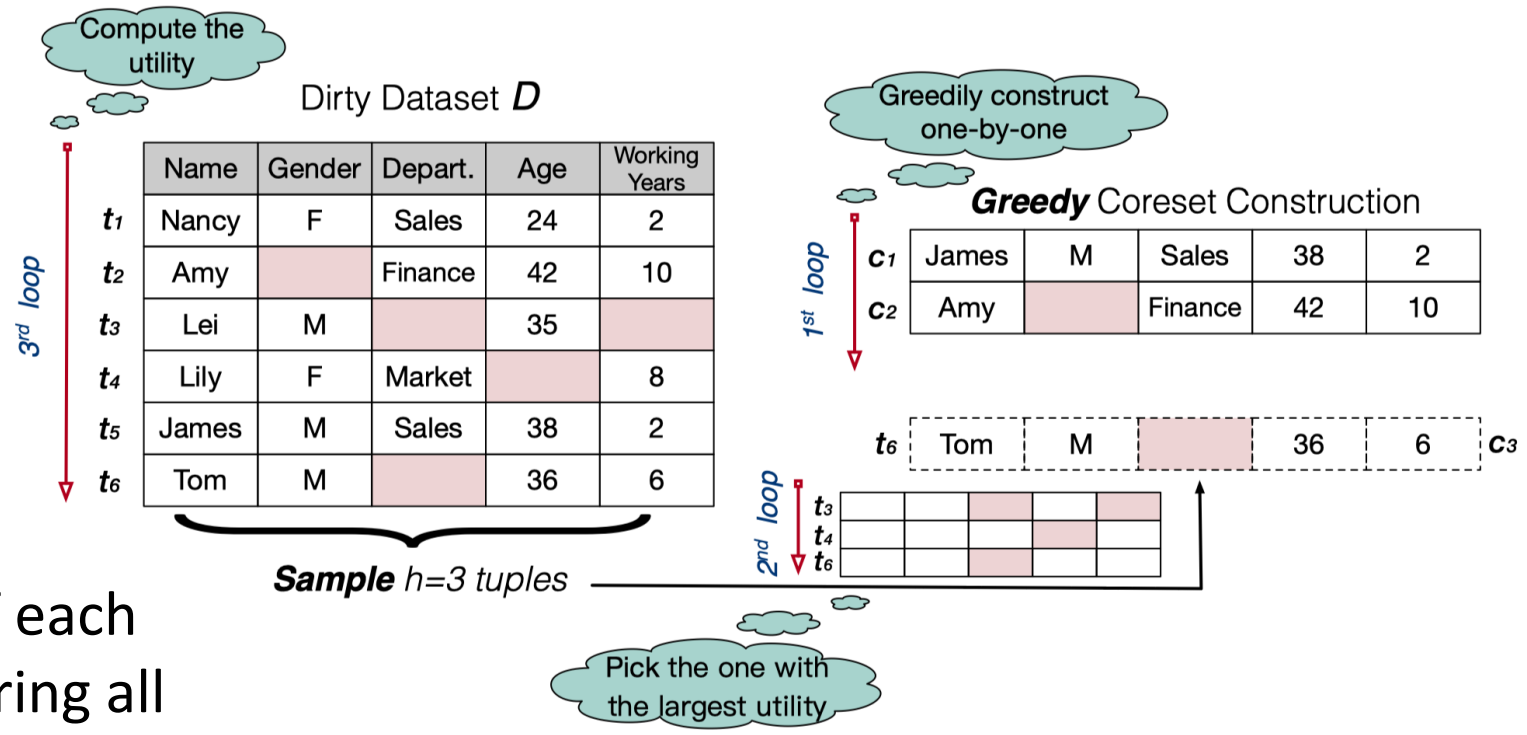


- Time Complexity：

$$O(KhnL^K)$$

A Three-loop Framework

- First loop: Add tuples one-by-one
- Second loop: Sample a bacth of tuples, and select the one with the largest utility to add.
- Third loop: Compute the utility of each tuple in the second loop, considering all the possible worlds.



Imputation-in-the-loop Strategies

- One tuple each iteration：
  - Impute (Auto/Human) the tuple with the highest utility at each iteration.
  - Time complexity $O(KhnL^2)$.
- One batch each iteration with human-in-the-loop:
  - One tuple per iteration by humans is too clostly.
  - Impute a batch of tuples after several iterations.

Algorithm 3: Batch algorithm of GoodCore

Input: D, K, h, batch size b.
Output: A coreset C, weight $\mathbb{W}$.

```
1  C = ∅, cnt = 0;
2  while |C| < K do
3      Sample h tuples as T_sample ⊆ D \ C
4      for each tuple t ∈ T_sample do
5          E[t|C] = ComputeUtility(t, C, D);
6      t* = arg max_{t∈T_sample} E[t|C] ;
7      C = C ∪ {t*};
8      if 𝕀[t*] = 1 then
9          cnt++;
10     if cnt = b then
11         Ask the human to impute the incomplete tuples;
12         cnt = 0;
13 Compute the weight 𝕎.
14 return C, 𝕎;
```

Experiments

Table 1. Statistics of datasets

Dataset	$\|D\|$	M	**# Incomp. Tuples**	**Task**
Nursery	10960	9	3218	Multi-Class.
HR	18287	12	5475	Binary Class.
Adult	32842	14	10752	Binary Class.
Credit	131,000	11	76813	Binary Class.
BikeShare	13300	15	4821	Regression
Air	437,200	18	128,372	Regression

Effectiveness

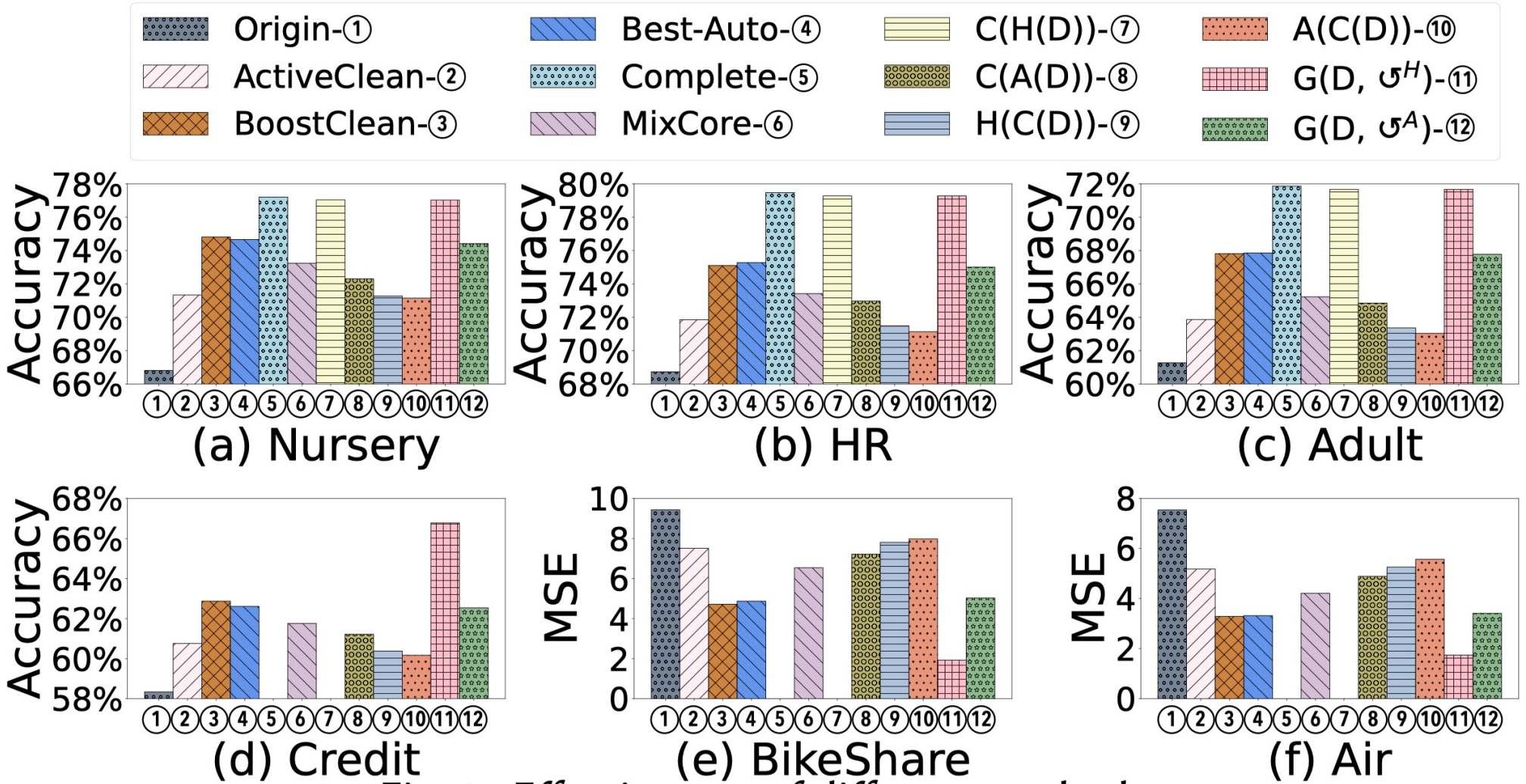


Fig. 8. Effectiveness of different methods.

Efficiency

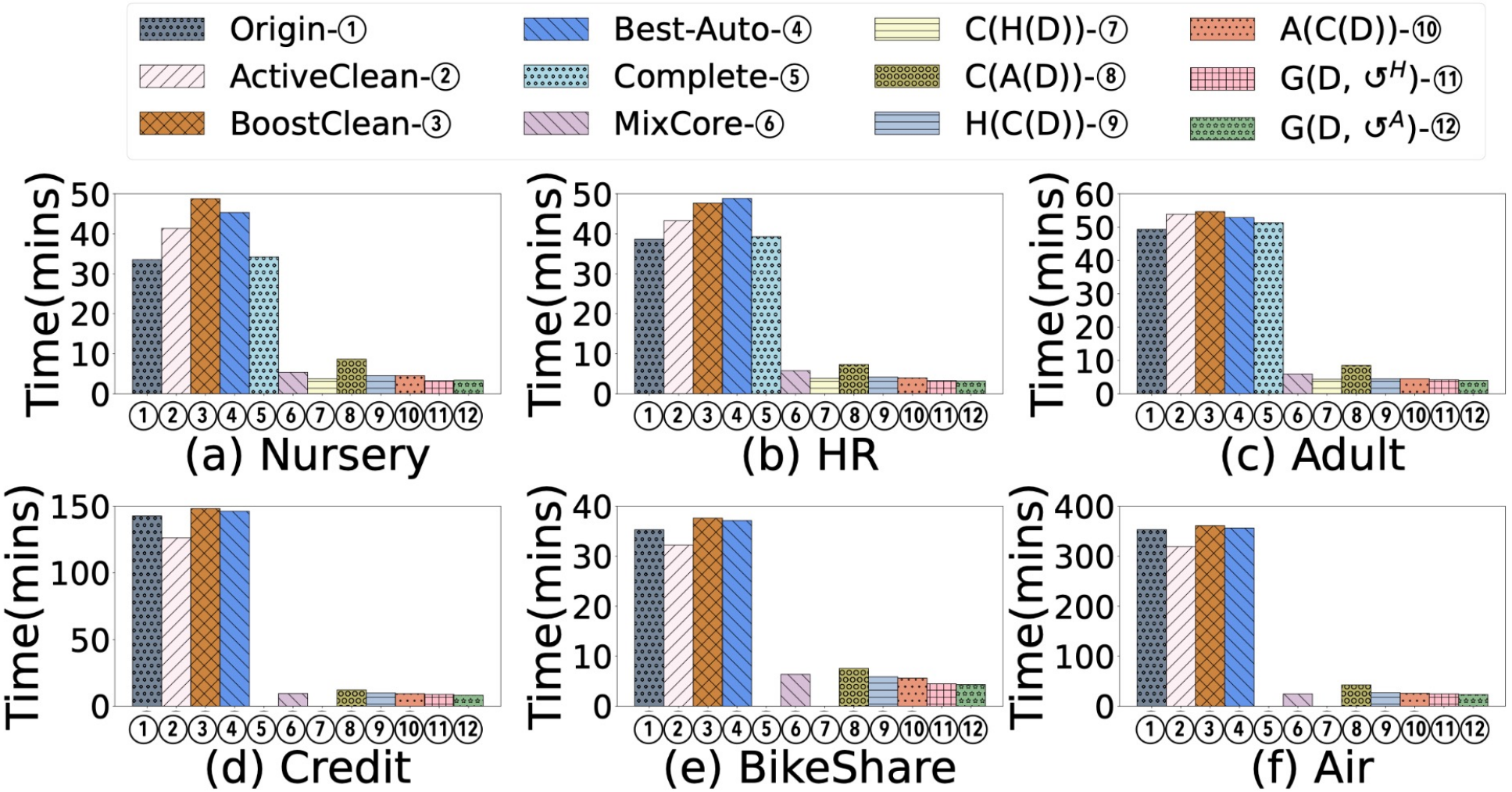


Fig. 9. Efficiency of different methods.

Human cost

Table 2. Human cost of different methods

Datasets	$\mathbf{G}(D, \circlearrowleft^{\mathbf{H}})$	$\mathbf{H}(\mathbf{C}(D))$	$\mathbf{C}(\mathbf{H}(D))$
Nursery	37	22	3278
HR	44	32	5475
Adult	63	81	10752
Credit	52	67	-
BikeShare	38	25	-
Air	98	102	-

Thanks!