Unifying and Optimizing Data Values for Selection via Sequential Decision-Making

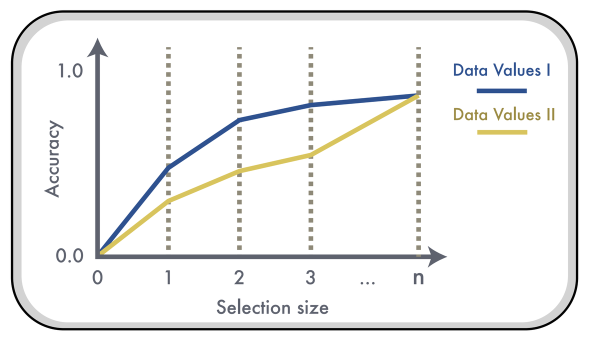
Hongliang Chi[1] Qiong Wu[2] Zhengyi Zhou[2] Jonathan Light[1] Emily Dodwell[2] Yao Ma[1]
[1]Rensselaer Polytechnic Institute [2]AT&T Chief Data Office

Motivation and Contributions

Data valuation usually computes a score for each point, then sorts. But downstream selection is an **ordered prefix problem**: every early subset must be useful.

Not one fixed subset. Training data selection is rarely a one-shot choice at a known budget. Data arrives in batches, earlier picks stay in the set, and LLM fine-tuning makes the data choice a major lever. We therefore optimize a nested sequence of prefixes, not only a single size-k subset.

- Shapley, Banzhaf, LOO, and semi-values aggregate marginal contributions independently of a target selection curve.
- The evaluation objective is the area under the selection curve (AUSC): average utility across all prefixes, not one endpoint.
- High redundancy makes one-step marginal scores unreliable for long-horizon selection.



How to read the curve. Each x-position is a budget; each point is selected-prefix utility. Good rankings rise early and stay high when the final budget is unknown.
Prefix lens: track `score` $\to$ `policy` $\to$ `curve`, not score alone.

Core failure mode. Pointwise scores treat examples independently after valuation. The sequential view keeps the interaction structure visible: selected points can be redundant, complementary, or useful only after other points are chosen.

`state`=selected prefix, `action`=next item, `return`=curve mean.

Data values for selection should encode an optimized selection sequence, not only local marginal rewards.

What changes.

- **Unification.** Cast data selection as a deterministic sequential decision problem whose return is exactly AUSC.
- **Interpretation.** Show common semi-values act as myopic policies under linear utility surrogates.
- **Optimization.** Introduce a bipartite coverage surrogate that preserves submodular structure and scales to classical ML and LLM data selection.

Area under the selection curve (AUSC). For an ordering π, AUSC is the mean utility over all selected prefixes. It rewards rankings that rise early and remain useful across budgets:

$$\mathrm{AUSC}(\pi) \quad = \quad \frac{1}{n}\sum_{k=1}^{n} U(S_k^{\pi})$$

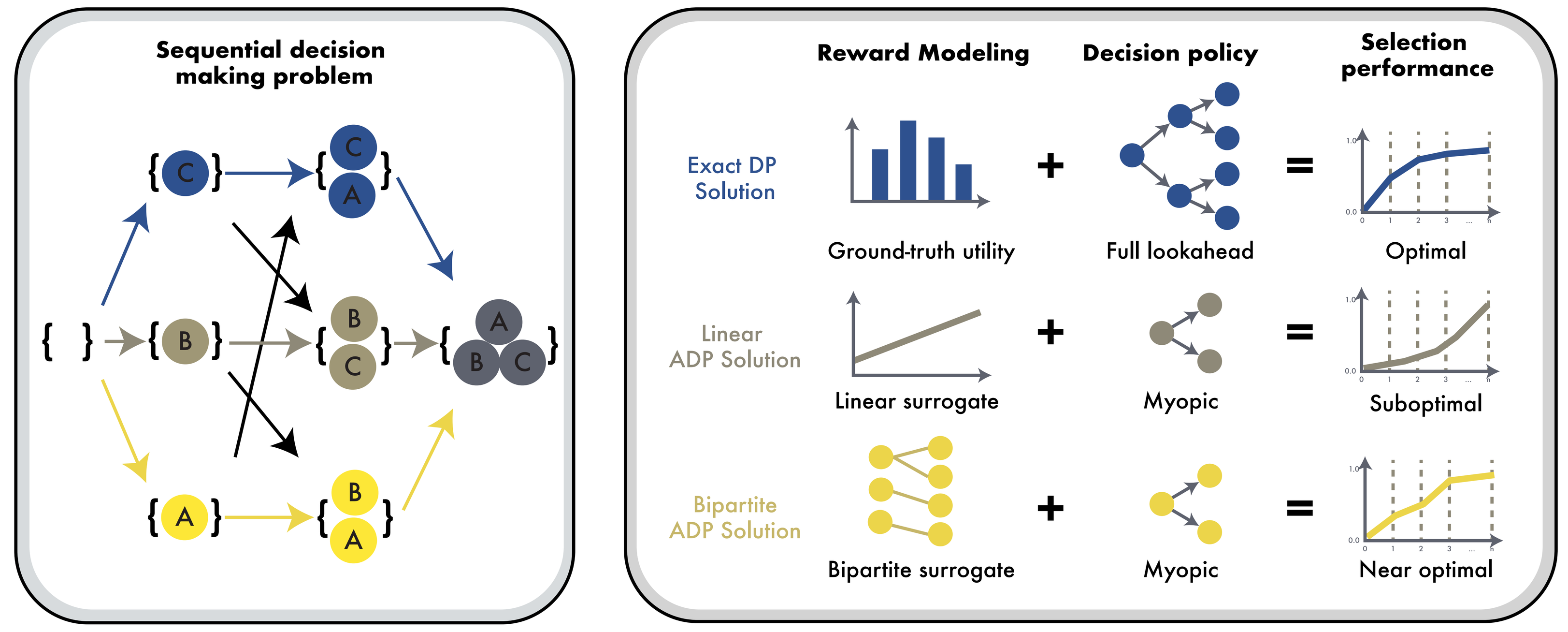
- Rewarding marginal gain alone collapses to the constant $U(\mathcal{D})$.
- Rewarding prefix utility distinguishes rankings by early and medium-budget usefulness.
- Exact DP is feasible for small n and gives a reference optimum for measuring gaps.

RQ1: How close are existing values to optimal DP? Exact DP exposes nontrivial gaps between optimized policies and existing rankings:

Answer. Best existing baselines are **8.39%–23.76% below optimal DP**. The gap is small when utility is close to additive, and larger when selected points substitute for one another. This motivates the paper's shift from scoring examples in isolation to optimizing the whole prefix sequence.

Operational message. Optimize the ranking for curve mean, not one endpoint. When the final budget is uncertain, every prefix should already be useful.

Unified Decision Framework



One framework links subset utility, decision policy, and the observed selection curve.

Sequential objective. For a selection order π, every prefix S_k^{π} contributes to the return:

$$\pi^{\star} \quad = \quad \arg\max_{\pi} \frac{1}{n}\sum_{k=1}^{n} U(S_k^{\pi})$$

$$V(s) = \begin{cases} 0, & |s| = n, \\ \max_{a \in \mathcal{D}\setminus s} \{U(s \cup \{a\}) + V(s \cup \{a\})\}, & \text{otherwise.} \end{cases}$$

DP view. State is the selected prefix, action is the next candidate, and return is the accumulated selection curve. Exact DP gives the reference optimum but has $O(n2^n)$ state–action scale.

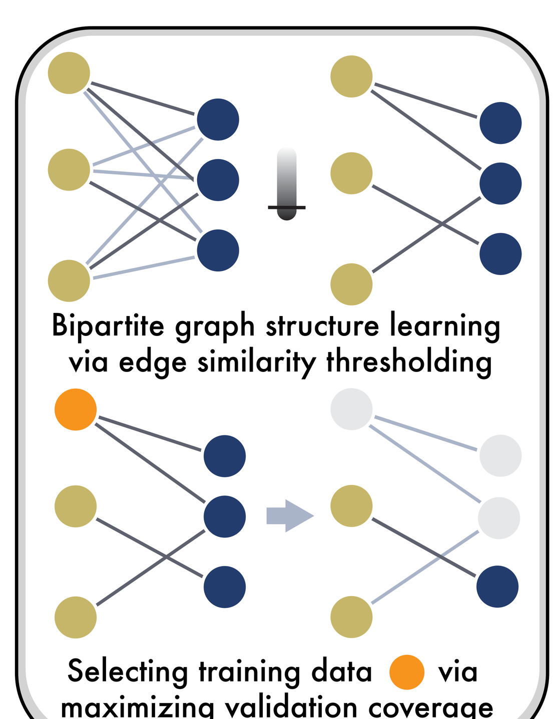
Prefix reward. Marginal gains telescope to $U(\mathcal{D})$ and lose ordering. Prefix utility keeps early, middle, and late budgets visible.

Approximate dynamic programming. ADP replaces exact Bellman recursion with a tractable value approximation and a greedy policy. With

$$\hat{U}(S) = \hat{b} + \sum_{i \in S} \theta_i.$$

one-step greedy selection ranks by θ_i. Shapley, Banzhaf, and BetaShap differ in how they weight sampled subsets while fitting this linear view.

Structured coverage.



Bipartite/BipCov surrogate. Use a sparse coverage graph instead of independent scores:

$$\hat{U}_{\mathrm{cov}}(S) = \frac{|\{v : \exists u \in S,\ (u,v) \in E\}|}{|X_{\mathrm{ref}}|}$$

Greedy selection adds the candidate that covers the most new validation or reference behavior.

- **Bipartite:** classical train candidates cover validation behavior measured through OpenML utility.
- **BipCov:** instruction candidates cover task references using Llama hidden-state similarity before LoRA fine-tuning.

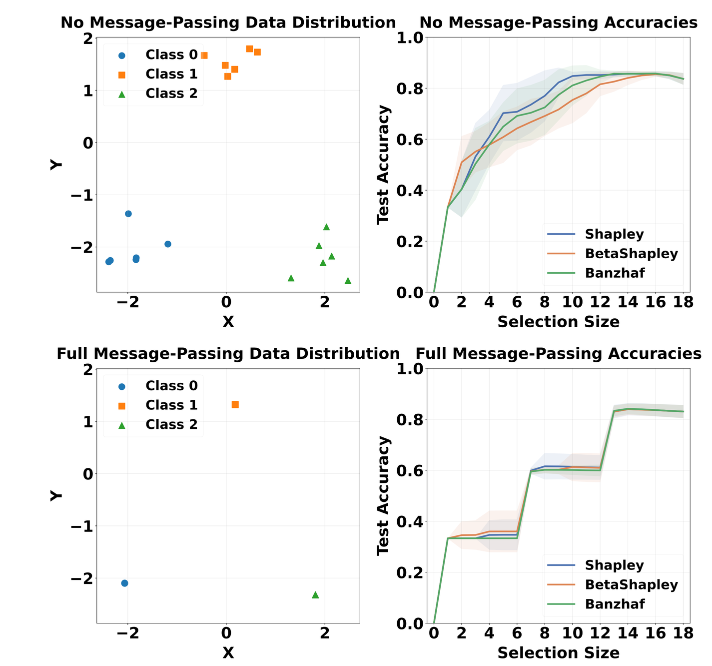
Coverage penalty. Once a region is covered, repeated neighbors add little value, so diversity emerges from the objective.

Theory

Curvature guarantee. For normalized, monotone, submodular utility with curvature c, semi-value rankings satisfy:

$$U(S_k^{v}) \quad \geq \quad (1 - c)^2 U(OPT_k^{\star}), \quad \forall k$$

- $c = 0$: linear utility; myopic data values are optimal.
- $c \to 1$: substitutability weakens static scores.



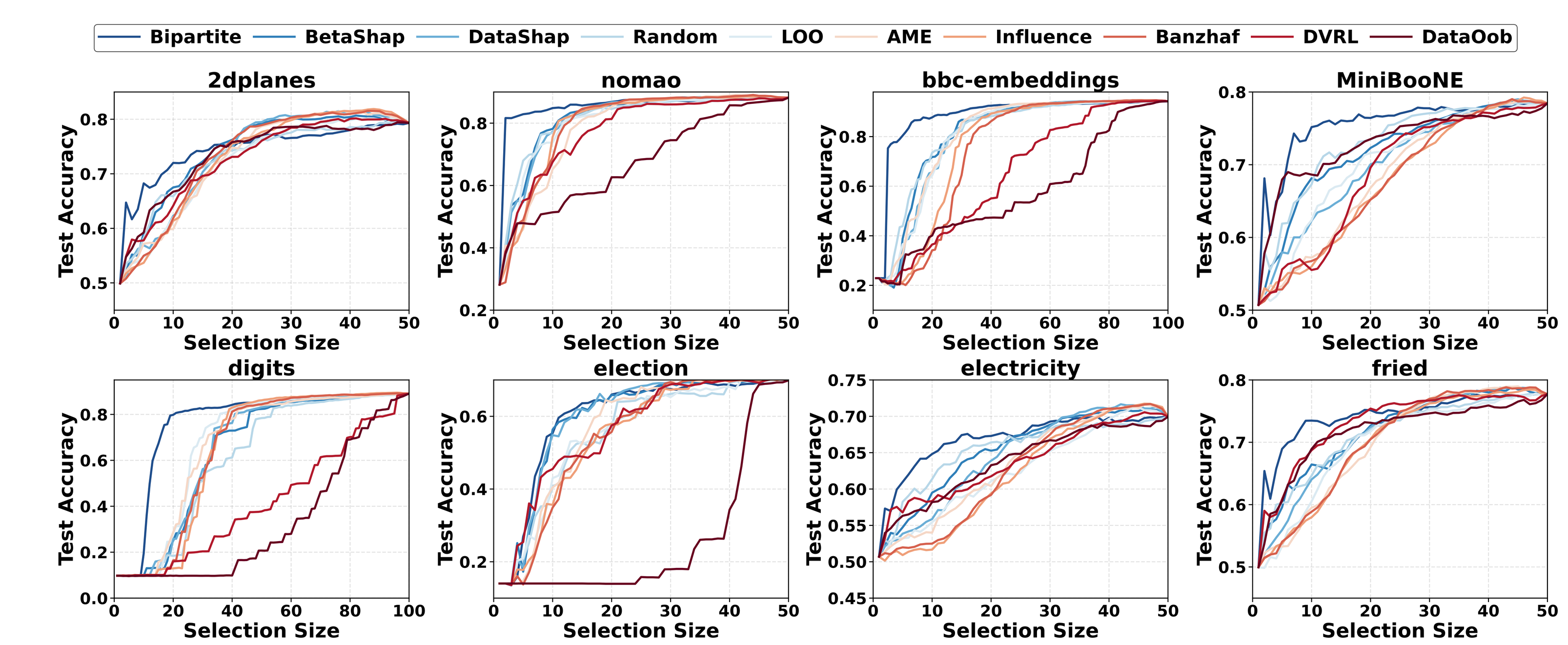
RQ2: When does curvature hurt static values? Controlled redundancy tests when a stateful selection policy becomes useful.

- Low curvature keeps semi-values close.
- Repeated behavior makes static marginals unstable.
- Coverage discounts candidates that repeat covered regions.

ADP lesson. Exact DP is a diagnostic reference; ADP is the scalable policy view. Substitutes make state matter: covered regions change the next useful point, so value-to-go must depend on the selected prefix.

Experiments

RQ3: Can Bipartite improve scalable selection? Tested by full curves, Budget-500 curve means, and utility-surrogate error.



Selection curves over 8 OpenML datasets, 20 seeds, 10 methods.

0.750	**0.828**	**0.019**
RQ3 mean vs BetaShap 0.713	Budget-500 vs BetaShap 0.821	utility MSE Linear/MLP 0.089/0.124

Study map. RQ1 = DP gap; RQ2 = curvature failure; RQ3 = scalable Bipartite; DATE-LM = LLM transfer.

Classical ML. 8 datasets $\times$ 20 seeds $\times$ 10 methods.
Budget-500: train_count=500; curve mean.
Bipartite first on 3 datasets; tied on 2 more.

DATE-LM. Select 10k from a 200k instruction pool.
LoRA fine-tune Llama-3.1-8B.
Evaluate MMLU, GSM8K, BBH over 3 seeds.

Method	Score	
BipCov	**63.89**$\pm$**0.34**	best
RDS+	62.88$\pm$0.25	baseline
Random	62.82$\pm$0.31	baseline

Mean over MMLU/GSM8K/BBH, 3 seeds.

Graph signal. BipCov uses Llama last-token states; coverage saturates after 2/1/5 examples for MMLU/GSM8K/BBH, then mean similarity fills the 10k order.

Fine-tuning view. Select before LoRA; evaluate after fine-tuning.

Dedication. In memory of Dimitri Bertsekas. His work on dynamic programming shaped this field, and this paper.

Takeaway. Optimize the full selection curve. Turn data values from static scores into policies for selection.

Paper Code Website