Active Value Querying to Minimize Additive Approximate Error in Superadditive Set Function Learning

Anonymous Author(s)
Submission Id: «EasyChair submission id»

ABSTRACT

Superadditive set functions play a pivotal role in computational economics, combinatorial optimization or artificial intelligence applications such as interpretable machine learning. However, specifying a set function requires assigning values to an exponentially large number of subsets, a task that is often resource-intensive in practice, particularly when the values derive from external sources such as retraining of machine learning models. A simple omission of certain values introduces ambiguity that becomes especially significant when the incomplete set function has to be further optimized over. We study a problem of optimal querying of an unknown superadditive set function for unknown values with the overarching goal of efficiently closing the distance between minimal and maximal superadditive completions. The key contributions are threefold: (i) a thorough exploration of minimal and maximal completions of set functions with missing values and an analysis of their resulting distance, providing insights for more effective optimization; (ii) the development of methods to minimize this distance over classes of set functions with a known prior, achieved by disclosing values of additional subsets in both offline and online modes; and (iii) empirical demonstrations of the algorithms' performance in practical scenarios, accompanied by an investigation into the typical order of revealing subset values.

KEYWORDS

Function approximation, Superadditive functions, Online learning

ACM Reference Format:
Anonymous Author(s). 2024. Active Value Querying to Minimize Additive Approximate Error in Superadditive Set Function Learning. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 18 pages.

1 INTRODUCTION

Set function optimization provides a versatile framework for modeling subset selections where adding more inputs has (possibly) non-linear additional benefits depending on the other elements. Its far-reaching applications span diverse fields, including supply chain management [21], communication networks [25], logistics and resource allocation [17], or environmental agreements [13]. However, beneath its promising facade lies a fundamental challenge – specifying a set function for n items requires assigning a value to each possible subset, which can be a daunting process as the number of subsets is exponential in n.

In this paper, we study a scenario that often arises in practice, when the set function is not a priori known, and acquiring even a single value for a single subset can be a resource-intensive endeavor. Take, for instance, the realm of machine learning, where determining the value of a feature subset in the celebrated explainable approach SHAP [18] corresponds to retraining an entire model, consuming time and computational resources. What is more, these contributions can have ripple effects on subsequent financial outlays, such as acquiring new training samples. In the corporate world, estimating an employee's contribution to collective performance may facilitate their fair evaluation [20], but obtaining the value of a "subset" may involve the intricate process of rearranging teams of employees, incurring operational costs and potentially causing disruptions.

Yet, simply leaving the values of many subsets undetermined further compounds the problem by opening the doors to ambiguity. The space of potential completions of partial set functions could be large, which may negatively influence our abilities to optimize over it. For example, consider criterion functions that aim to estimate contributions of each element to the set function values. Such a scenario often arises in, e.g., cooperative game theory, where the criterion relates to feasible imputations. When the players lack precise information about subset (in this case – coalition) values, they may have inflated expectations. The individuals believe their contributions are more significant than they objectively are. This translates into unrealistic demands for a larger share of the grand, i.e. all-agents, coalition value, even to the point where the sum of individual claims surpasses the actual value of the grand coalition. For example, the companies might sometimes demand exorbitant prices for their data, just as employees may occasionally request wages that are unrealistically high. This discrepancy creates a critical gap between what the players expect and what is feasible within the game. More generally, we aim to minimize some notion of the size of the space of potential completions of a partial set function. In this work, we are able to identify two unique extremal points of this space, and refer to their distance as the set function divergence.

To narrow this distance, we assume the existence of an external principal. In the examples we mentioned, this role could be reserved for the company manager or the machine learning engineer. This principal possesses the unique ability to determine the sequence in which subset values are revealed. However, they are provided with only a limited number of opportunities to exercise this control. In this manner, the principal mitigates the ambiguity within the system, thereby diminishing the divergence as a consequence. Importantly, we assume that each revelation step carries a roughly equivalent cost, ensuring that there is no inherent preference in which subset value is unveiled next besides its effect on decreasing the divergence. The primary objective of the principal hence

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.

reduces solely to minimizing this distance under the budget constraint. Although we do not explicitly outline the methodology for resolving situations when a non-zero divergence exists, our underlying assumption is that a lower value is favorable.

1.1 Organization and Contributions

We begin by formally defining the framework of set functions. We then explain how to extend the framework to encompass incomplete functions, which feature missing values. Afterward, we delve into our main contributions. Central to our study is the introduction of superadditive extensions, resulting in the emergence of the incomplete superadditive set function divergence. We establish fundamental theoretical properties of the divergence, including its monotonicity, additivity, and circumstances under which it becomes zero. We also demonstrate instances where the divergence exhibits supermodularity.

Building upon these foundations, we formulate both offline and online problems for the principal aiming to minimize the divergence, offering a suite of heuristic and approximative algorithms for each scenario. Our empirical analysis provides valuable insights into algorithms' performance and the subset values typically revealed early in the online and offline setups. Importantly, our findings also illustrate the non-linear nature of the divergence's decrease with the number of revealed values, offering nuanced perspectives on what a principal can expect at various stages of the revelation process. Our results further indicate that for specific classes of monotone supermodular functions, the divergence can be nearly completely reduced by revealing just $O(n)$ values.

1.2 Related Work

Much of the research dedicated to learning set functions focuses on submodular or subadditive functions, largely motivated by the need to represent bidders' valuations in combinatorial auctions [14]. The emphasis on these specific classes arises because they describe auctions without complements, meaning goods do not gain added value in the presence of others, making them easier to solve and optimize. Additionally, the error is typically measured multiplicatively, bounding the unknown set function f from below by function g and from above by αg, rather than additively as in our case, where the general distance of the completions is used. Subsequent developments have focused on passive approximative learning from existing datasets [3–5, 11, 12]. It is worth noting that even in this specialized application of set functions, complements (i.e., superadditivity) are common in standard motivating applications for combinatorial auctions, such as spectrum license auctions [1].

Active querying of the values can be seen as an online construction of a compact function representation, an approach supported by promising results in the field of cooperative game theory [8, 9, 19, 30]. These results indicate the feasibility of such an approach, with some efforts demonstrating a substantial exponential reduction in the number of values required to represent superadditive functions. A comprehensive survey and detailed presentation of many of these findings can be found in Chalkiadakis' book [10]. While tailored representations have achieved significant reductions in specific cases [16], no general approach for constructing such representation when given a subclass of games has been identified.

Our work adopts an approach reminiscent of active learning [28], where an algorithm actively queries an oracle for labels to new data points to construct the most informative dataset, particularly in situations where labeling is resource-intensive. In our context, we seek values that minimize the utopian gap. To approximate the optimal querying strategy based on this concept, we employ reinforcement learning [29].

2 PRELIMINARIES

In this section, we expound upon set functions, defined as mappings that assign values to all possible subsets of a designated *ground set* $N = \{1, \dots n\}$. We delve into the categorization of these functions, placing particular emphasis on the notion of superadditivity. Then, going beyond the typical set function definition, we consider a scenario when the function values are known only for a selected subset of all subsets, defining an incomplete set function.

Definition 1. *A* set function $f : 2^N \to \mathbb{R}$ *assigns values to subsets of the ground set N. We say f is*

additive *if* $\forall S \subseteq N$,

$$f(S) = \sum_{i \in S} f(\{i\}), \tag{1}$$

superadditive *if* $\forall S, T \subseteq N, S \cap T = \emptyset$,

$$f(S) + f(T) \leq f(S \cup T), \tag{2}$$

supermodular *if* $\forall S \subseteq N \setminus \{i, j\}$,

$$v(S \cup \{j\}) - v(S) \leq v(S \cup \{i, j\}) - v(S \cup \{i\}). \tag{3}$$

By $\mathbb{S}^n$, we denote the set of superadditive functions on the ground set of size n.

In many applications, acquiring all values proves to be cost-prohibitive, and only selected values are known. To model such scenario, we introduce an *incomplete set function*, capturing only partial knowledge of the function. We use $\mathcal{K} \subseteq 2^N$ to denote the set of subsets where the values are known. Expanding the set $\mathcal{K}$ then models the acquisition of new information about the initially unknown (yet well-defined) values. The set $\mathcal{K}$ can be hence seen as a "masking set", acting as a filter applied to a complete function.

Definition 2. *An* incomplete set function *is an ordered pair $(f, \mathcal{K})$, where $f : 2^N \to \mathbb{R}$ is a set function, and $\mathcal{K} \subseteq 2^N$.*

Assuming additional properties of the underlying set function f, one can impose constraints on values of $S \in \mathcal{K}$, even in the absence of the exact knowledge. In this work, we assert the superadditivity of the underlying set function. With this assumption and relying on the partial knowledge encapsulated by $\mathcal{K}$, it becomes possible to delineate a set of possible *candidates* for the underlying function. This set comprises extensions of the partial set function that adhere to the superadditivity condition. To bound this set, it is necessary to know at least the *minimal information* $\mathcal{K}_0$, defined as

$$\mathcal{K}_0 \coloneqq \{\emptyset, N\} \cup \{\{\{i\} \mid i \in N\}\}. \tag{4}$$

Definition 3. *Let $(f, \mathcal{K})$, $\mathcal{K}_0 \subseteq \mathcal{K}$ be an incomplete set function. Then g is a $\mathbb{S}^n$-*extension *of $(f, \mathcal{K})$ if $g \in \mathbb{S}^n$ and*

$$\forall S \in \mathcal{K} : f(S) = g(S). \tag{5}$$

We say $(f, \mathcal{K})$ *is* $\mathcal{S}^n$-extendable *if it has a* $\mathcal{S}^n$*-extension and we denote the set of* $\mathcal{S}^n$*-extensions by* $\mathcal{S}^n(f, \mathcal{K})$.

Since the set of $\mathcal{S}^n$-extensions is given by a system of linear inequalities, it forms a convex polytope in $\mathbb{R}^{2^n}$. The set $\mathcal{S}^n(\mathcal{K}, v)$ can be tightly enclosed in a hyper-rectangle given by the so called *upper/lower functions*. Specifically, the lower function $\underline{f}_{\mathcal{K}}$ of $(f, \mathcal{K})$ is a (complete) set function given by

$$\underline{f}_{\mathcal{K}}(S) \triangleq \min_{\substack{S_1,\dots,S_k \in \mathcal{K} \\ \bigcup_i S_i = S \\ S_i \setminus S_j = \emptyset}} \sum_{i=1}^{k} f(S_i), \tag{6}$$

and the upper function $\overline{f}_{\mathcal{K}}$ of $(f, \mathcal{K})$ is

$$\overline{f}_{\mathcal{K}}(S) \triangleq \max_{T \in \mathcal{K} : S \supseteq T} f(T) - \overline{f}_{\mathcal{K}}(T \cap S). \tag{7}$$

The following result formally introduces the hyper-rectangle mentioned above.

Proposition 1. *Let* $(f, \mathcal{K})$ *be an* $\mathcal{S}^n$*-extendable incomplete set function. Then for every* $\mathcal{S}^n$*-extension* g *of* $(f, \mathcal{K})$ *it holds*

$$\underline{f}_{\mathcal{K}}(S) \leq g(S) \leq \overline{f}_{\mathcal{K}}(S), \qquad \forall S \subseteq N. \tag{8}$$

Further, $\forall S \subseteq \mathcal{K}$*, there are* $\mathcal{S}^n$*-extensions* g_*, g^* *such that*

$$g_*(S) = \underline{f}_{\mathcal{K}}(S) \quad \text{and} \quad g^*(S) = \overline{f}_{\mathcal{K}}(S).$$

Proof. The first part of the theorem is equivalent to Theorem 1 in [19] with a slight distinction that Theorem 1 deals with superadditivity instead of superadditivity. The second part follows from Theorem 3 in [19], which states that any set function defined for a non-empty set $T \subseteq N$ as

$$f^S(T) \triangleq \begin{cases} \overline{f}_{\mathcal{K}}(T) & S \supseteq T, \\ \underline{f}_{\mathcal{K}}(T) & S \not\supseteq T, \end{cases} \tag{9}$$

is an $\mathcal{S}^n$-extension of $(f, \mathcal{K})$. For $S \subseteq \mathcal{K}$, choose $g_* = f^N$ and $g^* = f^S$.

Note that the hyper-rectangle given by lower/upper functions might also contain non-superadditive set functions.

3 MINIMIZING SET FUNCTION AMBIGUITY

The "size" of the set of extensions $\mathcal{S}^n(f, \mathcal{K})$ measures the uncertainty arising from only knowing values of f in $\mathcal{K}$. As stated in Theorem 1, the upper and lower functions define a hyper-rectangle which tightly bounds this set. The divergence measures the distance of these two functions and ultimately gives a characterization on the amount of uncertainty.

Definition 4. *Let* $f \in \mathcal{S}^n$ *and* $\|\cdot\|$ *be a set function norm. The* incomplete superadditive set function divergence *of* f *and* $\mathcal{K}$ *induced by* $\|\cdot\|$ *is a function* $\Delta_f : 2^{2^N \cap \mathcal{K}_0} \to \mathbb{R}$ *defined as*

$$\Delta_f(\mathcal{K}) \triangleq \left\| \overline{f}_{\mathcal{K}_0 \cap \mathcal{K}} - \underline{f}_{\mathcal{K}_0 \cap \mathcal{K}} \right\|. \tag{10}$$

It follows trivially from the properties of norms that the divergence is non-negative. Furthermore, it is zero if and only if $\forall S \subseteq N : \overline{f}_{\mathcal{K}}(S) = \underline{f}_{\mathcal{K}}(S)$, or equivalently, when there is just a single extension. However, even with a single missing value in the set function, divergence can be positive, indicating non-trivial extension sets, as explained in Appendix A.

Since the set functions on a ground set N can be regarded as elements of a vector space $\mathbb{R}^{2^n}$, we may use any vector norm as a divergence inducing norm. In the reminder of this text, we shall focus on divergences induced by absolute norms, i.e. those, where the norm of x is equal to the norm of $|x|$. These include all l_p-norms as well as many others. Since the divergence is non-negative, this restriction is without loss of generality.

Proposition 2. *Let* $f \in \mathcal{S}^n$*. Then the divergence* Δ_f *is*

(1) *monotonically non-increasing, i.e.,*

$$\Delta_f(\mathcal{K}) \geq \Delta_f(\mathcal{L}), \tag{11}$$

for $\mathcal{K} \subseteq \mathcal{L} \subseteq 2^N \cap \mathcal{K}_0$,

(2) *superadditive i.e.,*

$$\Delta_f(\mathcal{K}) + \Delta_f(\mathcal{L}) \leq \Delta_f(\mathcal{K} \cup \mathcal{L}) \tag{12}$$

for $\mathcal{K}, \mathcal{L} \subseteq 2^N \cap \mathcal{K}_0$ *such that* $\mathcal{K} \setminus \mathcal{L} \neq \emptyset$.

(3) *normalizable, i.e.*

$$\Delta_{\alpha f + \beta}(\mathcal{K}) = \alpha \Delta_f(\mathcal{K}) + \sum_{i \in 2N} \beta_i \tag{13}$$

for $\alpha > 0$, $(\alpha v + \beta)(S) \triangleq \alpha v(S) + \sum_{i \in 2N} \beta_i$ *and* $\beta_i \in \mathbb{R}$ *for* $i \in N$.

Proof. Let $\hat{\mathcal{K}} = \mathcal{K} \cup \mathcal{K}_0$, $\hat{\mathcal{L}} = \mathcal{L} \cup \mathcal{K}_0$ such that $\mathcal{K} \subseteq \mathcal{L}$. From the definition of the upper and the lower extension, for $T \subseteq N$, it follows

$$\overline{f}_{\hat{\mathcal{K}}}(T) \leq \overline{f}_{\hat{\mathcal{L}}}(T) \quad \text{and} \quad \underline{f}_{\hat{\mathcal{L}}}(T) \leq \underline{f}_{\hat{\mathcal{K}}}(T),$$

or equivalently $\overline{f}_{\hat{\mathcal{K}}} - \underline{f}_{\hat{\mathcal{K}}} \leq \overline{f}_{\hat{\mathcal{L}}} - \underline{f}_{\hat{\mathcal{L}}}$. Now (11) holds as long as the norm satisfies $0 \leq x \leq y \Rightarrow |x| \leq |y|$. In [6], it is showed this holds if $\|x\| = \||x|\|$. Further, from non-negativity and (11), superadditivity follows. Finally, we have

$$\begin{aligned} \Delta_{\alpha f + \beta}(\mathcal{K}) &= \left\| \overline{\alpha f + \beta}_{\mathcal{K}_0 \cap \mathcal{K}} - \underline{\alpha f + \beta}_{\mathcal{K}_0 \cap \mathcal{K}} \right\| \\ &= \alpha \left\| \overline{f}_{\mathcal{K}_0 \cap \mathcal{K}} - \underline{f}_{\mathcal{K}_0 \cap \mathcal{K}} \right\| = \alpha \Delta_f(\mathcal{K}). \end{aligned}$$

Finally, we show that l_1 induced divergence is concave in the underlying set function for a fixed set of known subsets.

Proposition 3. *Let* $\mathcal{K} \subseteq 2^N$, $\mathcal{K}_0 \subseteq \mathcal{K}$ *and* $f \in \mathcal{S}^n$*. Then the* l_1 *divergence is concave in the underlying set function* f.

Proof Sketch. Follows from concavity, resp. convexity of the upper, resp. lower function. See Appendix C for the proof.

3.1 Principal's Optimization Problems

A large divergence corresponds to a large uncertainty in the missing values. The more we know about the underlying function, the smaller the divergence gets, being zero if there is a single $\mathcal{S}^n$-extension. However, to obtain all the necessary unknown values might be too expensive, since there are exponentially many of them. We thus seek a way to minimize the divergence as much as possible within a limited number of known values.

To formulate the problem, we assume an existence of a *principal*. Her task is to choose which subsets of N should be investigated

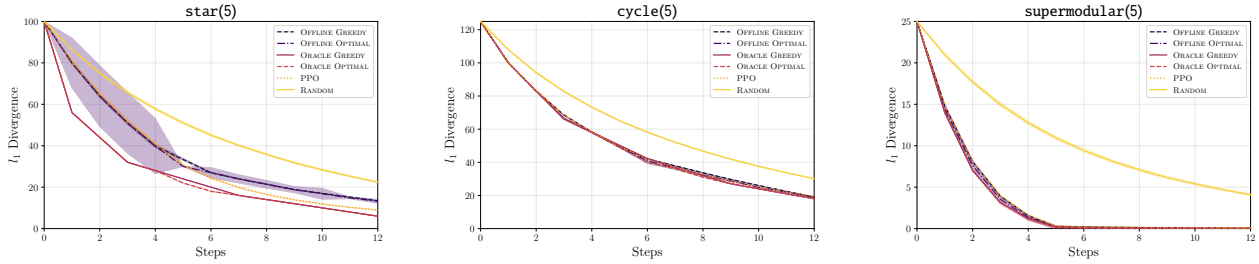

Figure 1: Comparison of divergence across algorithmic steps for various algorithms, showcasing `star(5)` (left), `cycle(5)` (center), and `supermodular(5)` (right) set function distributions. Notably, all algorithms significantly surpass the RANDOM benchmark, with greedy versions closely mirroring optimal performance.

to reduce the divergence the most. We further assume the principal holds a level of expertise that guides the selection process. This expertise is formalized as a prior distribution over a set of potential functions. For example, in a medical context, a doctor acting as the principal might seek to assess a patient's response to a combination of drugs, and base their selection on past clinical experience. Similarly, a machine learning engineer could rely on their prior knowledge of feature importance gained from previous problem-solving experiences. Consequently, we assume that each problem instance can be viewed as a sample drawn from a *known prior distribution*, denoted as F. To put bluntly, the principal is aware of the prior distribution, but not of the specific instance drawn from it.

There are two basic approaches to choosing which subsets to investigate, *online* and *offline*. In the online approach, the principal operates sequentially, utilizing information from previously revealed subsets.

Definition 5 (Online Principal's Problem). *Let $t \in \mathbb{N}$, F, supp $\mathsf{F} \subseteq \mathbb{S}^n$ be a distribution of superadditive set functions. Then $\mathsf{K}_t \subseteq 2^N \setminus \mathsf{K}_0$ is a solution of the online principal's problem of size t if*

$$\mathsf{K}_t \in \operatorname*{argmin}_{\mathsf{K}_t \subseteq 2^N \setminus \mathsf{K}_0, |\mathsf{K}_t| = t} \mathop{\mathbb{E}}_{f \sim \mathsf{F}} \left[\Delta_f(\mathsf{K}_t) \right], \tag{14}$$

where $\mathsf{K}_i = \{S_1, \ldots, S_i\}$, $S_i = \pi(f, \mathsf{K}_{i-1}) \setminus \mathsf{K}_0$ and π is a policy function, which chooses S_i based on the known values of f, i.e. values of $f(S_1), \ldots, f(S_{i-1})$.

In contrast, the offline approach entails a lack of such information.

Definition 6 (Offline Principal's Problem). *Let $t \in \mathbb{N}$, F, supp $\mathsf{F} \subseteq \mathbb{S}^n$ be a distribution of superadditive set functions. Then $\mathsf{K} \subseteq 2^N \setminus \mathsf{K}_0$ is a solution of the* offline principal's problem *of size t if*

$$\mathsf{K} \in \operatorname*{argmin}_{\mathsf{K} \subseteq 2^N \setminus \mathsf{K}_0, |\mathsf{K}| = t} \mathop{\mathbb{E}}_{f \sim \mathsf{F}} \left[\Delta_f(\mathsf{K}) \right]. \tag{15}$$

3.2 Algorithms Solving the Principal's Problems

In this section, we discuss various methods for finding (approximate) solutions to the principal's problems defined above. We defer further technical details about all algorithms to Appendix B.

3.2.1 Offline Algorithms. At each step t, the OFFLINE OPTIMAL algorithm chooses subsets $\{S_i\}_{i=1}^t$ which minimize the expected divergence under F. We estimate the expectation w.r.t. F in Eq. (15) by κ samples, see Algorithm 1.

Algorithm 1: OFFLINE OPTIMAL

Input: distribution of superadditive functions F, number of steps t, number of samples κ

$\overline{\mathsf{K}} \leftarrow 2^N \setminus \mathsf{K}_0$
$G \leftarrow \{\}$ // trajectories & their E[Δ]
for $S \in \overline{\mathsf{K}} : |S| = t$ **do** // each trajectory
  $\mu \leftarrow 0$
  for $j \in \{1, \ldots, \kappa\}$ **do** // approx. E[Δ]
    $f \sim \mathsf{F}$
    $\mu \leftarrow \mu + \Delta_f(S)$
  end
  $\mu \leftarrow \mu / \kappa$
  $G[S] \leftarrow \mu$
end
$\{S_i\}_{i=1}^t \leftarrow \operatorname{argmin}_{S \in \overline{\mathsf{K}}:|S|=t} G[S]$
return $\{S_i\}_{i=1}^t$

A computationally less demanding variant of the OFFLINE OPTIMAL is the OFFLINE GREEDY algorithm. It chooses the next subset S_t such that, given the previous trajectory $\{S_i\}_{i=1}^{t-1}$, it minimizes the expected divergence. Consequently, it can perform no better than the OFFLINE OPTIMAL. We again estimate the expectation in Eq. (15) by κ samples, see Algorithm 2.

3.2.2 Online Algorithm. In comparison to the offline problem, solving the online problem poses a significantly greater challenge. Intuitively, one key reason is that an algorithm for the online problem must compute (or approximate) a restriction of F that remains consistent with the values it has uncovered in prior steps. However, this can be particularly challenging, especially in case the only access to F is through sampling. To tackle the online problem and derive an approximate solution, we employ reinforcement learning [29], specifically, the proximal policy optimization (PPO) [26].

At each step $\tau < t$, PPO receives values of subsets $\mathsf{K}_{\tau-1} = \{S_i\}_{i=1}^{\tau-1}$ it uncovered in the past and chooses the next subset S_τ. To get a strategy which efficiently minimizes the divergence, we

Algorithm 2: Offline Greedy

Input: distribution of superadditive functions F, number of steps t, number of samples κ

if $t > 1$ **then**
  $\{S_i\}_{i=1}^{t-1}$ ← Offline Greedy($\mathsf{F}, t-1$)
end
$\overline{\mathsf{K}} \leftarrow 2^N$
$G \leftarrow \{\}$ // trajectories & their $\mathsf{E}[\Delta]$
for $S \in \overline{\mathsf{K}}$ **do** // each trajectory
  $\mu \leftarrow 0$
  for $j \in \{1, \ldots, \kappa\}$ **do** // approx. $\mathsf{E}[\Delta]$
    $f \sim \mathsf{F}$
    $\mu \leftarrow \mu + \Delta_f(S_i\}_{i=1}^{t-1} \cup \{S\})$
  end
  $\mu \leftarrow \mu/\kappa$
  $G[S] \leftarrow \mu$
end
$S_t \leftarrow \operatorname{argmin}_{S \in \overline{\mathsf{K}}} G[S]$
return $\{S_i\}_{i=1}^{t}$

define the reward (which is maximized by the PPO algorithm) as the negative expected divergence averaged over $\tau \in t$.

As previously mentioned, the greedy algorithm is significantly more computationally efficient, with a linear complexity in the number of subsets, while the optimal variant exhibits an exponential time complexity[1]. A natural question is under which conditions the local greedy search yields a "good" solution. It is known that, if the optimized function is supermodular, the locally optimal steps are guaranteed to yield a $(1 - 1/e)$-approximation of the global optimum [22, Proposition 3.4]. The divergence is one such function, as long as the size of the ground set is less or equal to four.

Proposition 4. *For $n \leq 4$, the l_1-divergence Δ_f is supermodular for every $f \in \mathsf{S}^n$.*

Proof Sketch. For $n = 3$, Δ_f is modular due to invariant bounds post-revelation. For $n = 4$, the supermodularity is demonstrated through a more technical case analysis.

When $n \geq 5$, however, the supermodularity of the divergence imposes a very restrictive condition on the underlying function, which is hardly satisfied in applications as n grows larger.

Proposition 5. *Let $n \geq 5$, $g \in \mathsf{S}^n$ have supermodular l_1-divergence. Then for $f \in \mathsf{S}^n$ defined as $f(S) = g(S) - \sum_{i \in S} g(\{i\})$, there are $i, j, k, l \in N$ such that $f(\{i, j\}) + f(\{j, k\}) < f(\{j\}) + f(\{k, l\})$, and*

$$f(\{i, j\}) \geq \frac{1}{2^{n-3}} f(\{k, l\}). \tag{16}$$

Proof Sketch. Let $\mathsf{K} = \mathsf{K}_0 \cup \{\{j, k\}\}$, $S = \{i, j\}$, and $Z = \{k, l\}$. Then the supermodularity constraint (3) implies the statement.

[1] Since the number of subsets is itself exponential in the size of the ground set, it makes the difference between the two even greater.

Superadditive functions with supermodular divergence are e.g. additive functions. As a corollary of Proposition 5, even supermodular functions f may have non-supermodular divergence. Proofs of both propositions can be found in Appendix D.

4 EMPIRICAL EVALUATION

Finally, we demonstrate the performance of our algorithms on practical examples. We outline the domains used for our evaluation, introduce baseline methods for comparison, detail the algorithmic setups, and conduct a comprehensive analysis of the gathered results. All the details that could not be accommodated within the main text have been addressed in the relevant appendices.

4.1 Experimental Domains

We conduct our experiments on two representative families of set functions. Their main difference is the degree to which the values of different subsets are correlated.

For the tightly correlated scenario, we use set functions defined over graphs[2] $G = (N, E)$ on $|N| = 2$ vertices. For a subset $S \subseteq N$, the value is $f(S) = |E(S)|$, where $E(S)$ is the set of edges in the induced subgraph $G[S]$. We define two classes of functions within this framework based on the graph representing the functions. First, `star`(n) denotes a class where G is a connected star, where the 'center' is chosen uniformly at random. Similarly, `cycle`(n) denotes a class where the graph is a single cycle on all vertices chosen uniformly at random.

The second family is the broad class of supermodular set functions. These are among the most studied set functions [15] – in some areas where set functions are applied, some authors consider this property so important they even impose it in the definition of the set function [23]. Incomplete supermodular set function have been already studied in a broader context [27]. In [7], the authors derived an efficient algorithm for uniform sampling of monotone supermodular functions. We refer to the distribution of corresponding set functions of size n as `supermodular`(n).

4.2 Benchmarks

We compare our algorithms introduced in the previous section to three baselines: a random algorithm and two oracle algorithms. The random algorithm selects the next subset uniformly at random. We refer to it in the results as Random. The oracle methods are not deployable in practice because they assume the knowledge of the underlying true set function. However, they provide an upper bound on what an optimal online algorithm could possibly achieve.

The Oracle Optimal algorithm operates similarly to Offline Optimal, but it additionally leverages an oracle to acquire values of the underlying set function f before making the subset selection. Consequently, Oracle Optimal can utilize complete knowledge of the underlying function to minimize the divergence.

Similarly, the Oracle Greedy algorithm selects next subset S_t such that, in combination with the previous trajectory $\{S_i\}_{i=1}^{t-1}$, it minimizes the divergence. It also uses the oracle to gather all information about the underlying function f. The pseudocodes for both Oracles can be found as Algorithms 3 and 4 in Appendix F.

[2] One natural generalization of these classes would be to assume weighted graphs.

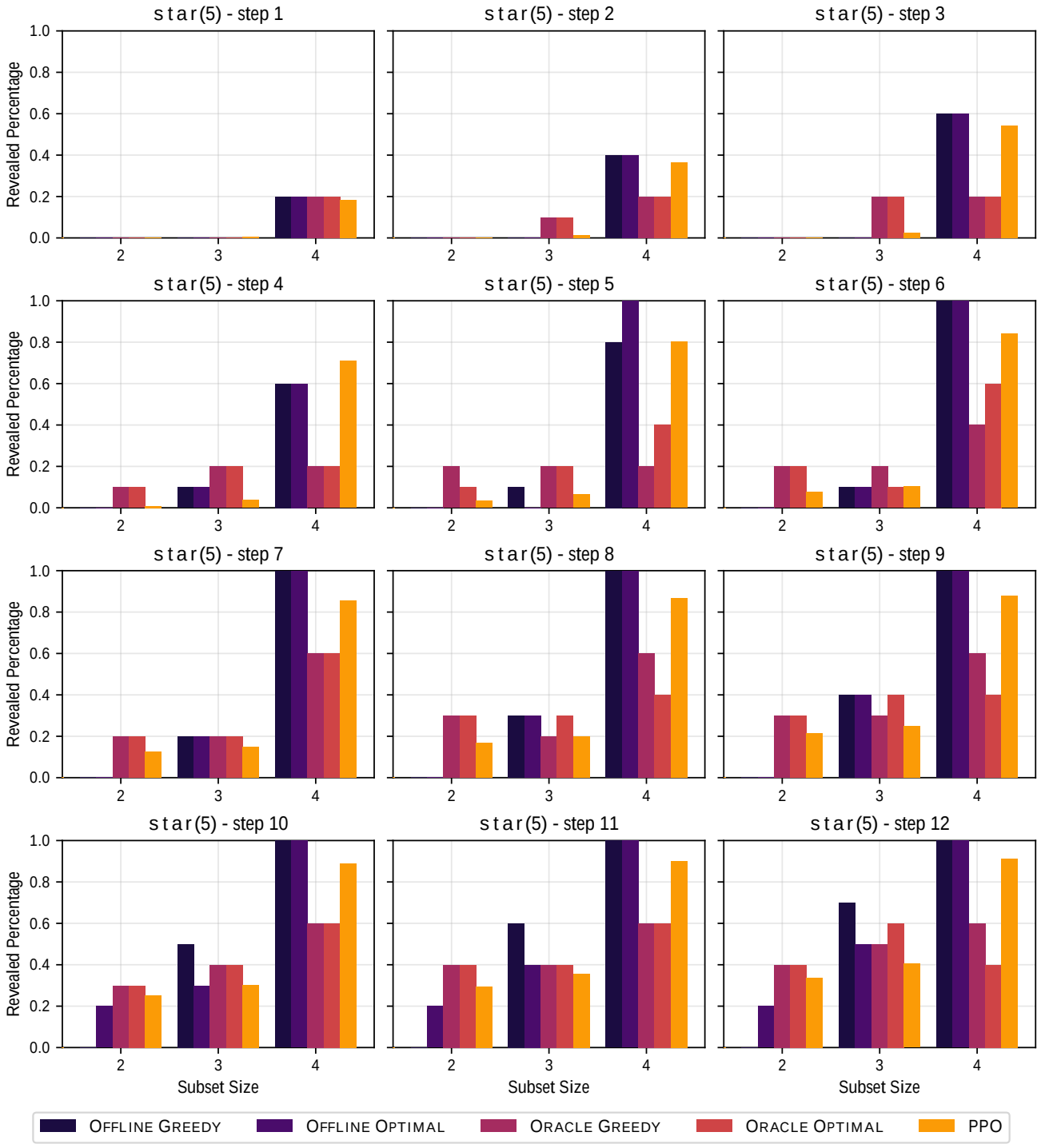

Figure 2: Proportion of subsets of the same size selected up to step twelve for `star(5)` and each algorithm. The results unveil a distinct inclination towards larger subsets, indicating their greater contribution to average set function information. The oracle algorithms exhibit an early preference for smaller subsets, suggesting that a tailored representation of a specific set function can efficiently incorporate even smaller subsets. PPO initially mirrors the behavior of the offline algorithms. However, leveraging previously obtained values, it later aligns its selections with the oracle methods. See also Figure 5 in Appendix G.

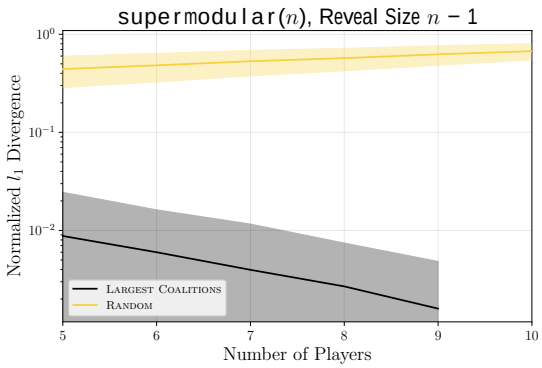

Figure 3: Comparison of normalized divergence for supermodular(n) as a function of the n = |N|, normalized by its minimal information value. We show the expected normalized divergence when n 1 subsets are randomly chosen with the scenario where all subsets of size n 1 are selected.

4.3 Experimental Setup

During the training phase, we execute the PPO algorithm for a total of 2,000,000 time-steps. In each iteration, we repeatedly collect 6 2048 trajectories by sampling batches from the distribution F. Following each iteration, we optimize the PPO surrogate objective for 10 epochs. To ensure uniformity, we normalize the input values f^1S^0 to a unit interval, as elaborated in Appendix H. For the two optimal algorithms, we employ $\kappa = 10$ samples to estimate the mean value in Eq. (15). See Appendix B for more details.

Among the proposed algorithms, both Offline Optimal and Oracle Optimal come with significantly higher computational costs (scaling as $\mathsf{O}^1 2^{2^n} {}^0$) when compared to their greedy counterparts (which scale as $\mathsf{O}^1 2^{n0}$). To provide some perspective, our evaluation for $n = 5$ with twelve steps and ten samples required approximately 10 hours to finish. Extrapolating, we estimate that completing the evaluations for the remaining steps in this setup, across all ten samples, would require roughly 300 hours. Furthermore, extending these experiments to $n = 6$ would demand over 100 years.

Hardware and Software. All experiments are conducted on a computational cluster with AMD EPYC 7532 CPUs running at 2.4 GHz. When running algorithms on set functions with ground set of size five, we utilize 15 cores and 12 GB of RAM. The code was implemented in Python 3.10 using pytorch 2.0, stable_baselines3 2.0, and gymnasium 0.28. Additional details are deferred to Appendix B.

4.4 Results

First, we study the star(5) and cycle(5) distributions. The dependency of the divergence on the number of revealed subsets is illustrated in Figure 1. As anticipated, the oracle algorithms exhibit superior performance, particularly in the initial stages. Notably, PPO initially aligns closely with the offline algorithms but leverages online information to converge toward comparable divergence values to the oracle methods. The greedy algorithms showcase performance akin to their optimal counterparts. Figure 2 then depicts the percentage of subsets chosen by each algorithm at each step. Offline algorithms prefer larger subsets, indicating perceived comprehensive information. Oracle algorithms show slight gains with smaller subsets, suggesting potential benefits from a tailored representation that includes smaller subsets for a fixed set function. For a detailed plot of individual subsets, refer to Figure 5 in Appendix G.

Turning our attention to the supermodular(5) distribution, Figure 1 shows the evolution of the divergence concerning the number of revealed subsets in this domain. The divergence experiences rapid deacrease, reaching about 99% reduction by step five. In each instance, the algorithms consistently favored subsets of size n 1, underscoring their significance in this distribution. To gauge the impact of the largest subsets, we examined the divergence as a function of n when all subsets of size n 1 are unveiled. Normalizing the divergence by the divergence of K_0, Figure 3 reveals that this simple heuristic significantly outperforms the Random strategy. Surprisingly, the resulting normalized divergence diminishes with the size of the ground set, indicating that the majority of information about a supermodular function can be captured by values of only $\mathsf{O}^1 n^0$ subsets. The reason is that, in expectation, the values of subset of size n 1 are several orders of magnitude smaller than f^1N^0. We suspect this is because of concentration of supermodular functions [2]. Note that this is not the case in general – see Appendix E.

5 CONCLUSION

In this paper, we study strategies for efficiently mitigating uncertainty within incomplete set functions. We introduce the concept of the "set function divergence", which quantifies the size of the set of possible extensions of a partial set function. We show fundamental properties of the set function divergence that enables us to compute it more efficiently. We focus on reducing the set function divergence through well-informed queries about the unknown values within the incomplete set function, effectively constructing a tailored representation – in both online and offline fashion. Our findings indicate that our approach significantly outperforms random queries and approaches optimality. Particularly noteworthy is our heuristic for supermodular set functions, which reduces the set function divergence by orders of magnitude while requiring only $\mathsf{O}^1 n^0$ queries – an amount logarithmic in the size of the domain.

Future Work. In our current exploration of set function divergence, we have specifically concentrated on the l_1-norm. However, we recognize the potential for more generalized insights by extending our focus to divergences induced by various norms, as hinted by the theorem on equivalence of norms. This approach holds promise, as demonstrated by Proposition 2, which is applicable not only to the l_1-norm but also to any l_p-norm.

Moreover, our findings have already revealed that the divergence of functions on a ground set of size at least 5 may not necessarily be supermodular, even for supermodular set functions. Consequently, a crucial aspect of our future research will involve investigating the properties of set functions that yield supermodularity. This exploration particularly holds a lot of promise as, in many instances, our results indicate that the greedy approach closely approximates optimality, as if the divergences were supermodular.

Additionally, we aim to bolster our findings by providing guarantees on solution quality, which, we hope, can be achieved by incorporating regret minimization into our methodology.

REFERENCES

[1] Ittai Abraham, Moshe Babaioff, Shaddin Dughmi, and Tim Roughgarden. 2012. Combinatorial auctions with restricted complements. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. 3–16.

[2] Maria-Florina Balcan and Nicholas J. A. Harvey. 2010. Learning Submodular Functions. *CoRR* abs/1008.2159 (2010). arXiv:1008.2159 http://arxiv.org/abs/1008.2159

[3] Maria Florina Balcan, Florin Constantin, Satoru Iwata, and Lei Wang. 2012. Learning valuation functions. In *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 4–1.

[4] Maria-Florina Balcan and Nicholas JA Harvey. 2011. Learning submodular functions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*. 793–802.

[5] Maria-Florina Balcan and Nicholas JA Harvey. 2018. Submodular functions: Learnability, structure, and optimization. *SIAM J. Comput.* 47, 3 (2018), 703–754.

[6] F. L. Bauer, J. Stoer, and C. Witzgall. 1961. Absolute and monotonic norms. *Numer. Math.* 3 (1961), 257–264.

[7] Gleb Beliakov. 2022. On Random Generation of Supermodular Capacities. *IEEE TRANSACTIONS ON FUZZY SYSTEMS* 30, 1 (2022), 293–295. https://doi.org/10.1109/TFUZZ.2020.3036699

[8] Jan Bok and Martin Černý. 2023. 1-convex extensions of incomplete cooperative games and the average value. *Theory and Decisions* (2023). https://doi.org/10.1007/s11238-023-09946-8

[9] Martin Černý. 2023. Bounds on solution concepts of incomplete cooperative games. arXiv:arXiv:2212.04748 [cs.GT]

[10] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. 2012. *Computational Aspects of Cooperative Game Theory* (1 ed.). Springer Cham. XVI, 150 pages. https://doi.org/10.1007/978-3-031-01558-8

[11] Vitaly Feldman, Pravesh Kothari, and Jan Vondrák. 2020. Tight bounds on l1 approximation and learning of self-bounding functions. *Theoretical Computer Science* 808 (2020), 86–98.

[12] Vitaly Feldman and Jan Vondrák. 2014. Structure and learning of valuation functions. *ACM SIGecom Exchanges* 12, 2 (2014), 50–53.

[13] Michael Finus. 2008. Game theoretic research on the design of international environmental agreements: insights, critical remarks, and future challenges. *International Review of environmental and resource economics* 2, 1 (2008), 29–67.

[14] Michel X Goemans, Nicholas JA Harvey, Satoru Iwata, and Vahab Mirrokni. 2009. Approximating submodular functions everywhere. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 535–544.

[15] Michel Grabisch. 2016. *Set Functions, Games and Capacities in Decision Making*. Number 978-3-319-30690-2 in Theory and Decision Library C. Springer.

[16] Samuel Ieong and Yoav Shoham. 2005. Marginal Contribution Nets: A Compact Representation Scheme for Coalitional Games. https://www.cs.cmu.edu/~sandholm/cs15-892F15/MarginalContributionEC05.pdf

[17] Sebastián Lozano, Placido Moreno, Belarmino Adenso-Díaz, and Encarnacion Algaba. 2013. Cooperative game theory approach to allocating benefits of horizontal cooperation. *European Journal of Operational Research* 229, 2 (2013), 444–452.

[18] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[19] S. Masuya and M. Inuiguchi. 2016. A fundamental study for partially defined cooperative games. *Fuzzy Optimization Decision Making* 15, 1 (2016), 281–306.

[20] Kevin R Murphy and Jeanette N Cleveland. 1995. *Understanding performance appraisal: Social, organizational, and goal-based perspectives*. Sage.

[21] Mahesh Nagarajan and Greys Sošić. 2008. Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. *European journal of operational research* 187, 3 (2008), 719–745.

[22] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming* 14 (1978), 265–294.

[23] G. Owen. 2013. *Game Theory* (4th ed.). Emerald Group Publishing, Bingley, U.K.

[24] Bezalel Peleg and Peter Sudhölter. 2007. *Introduction to the Theory of Cooperative Games*. Springer. https://doi.org/10.1007/978-3-540-72945-7

[25] Walid Saad, Zhu Han, Mérouane Debbah, Are Hjorungnes, and Tamer Basar. 2009. Coalitional game theory for communication networks. *Ieee signal processing magazine* 26, 5 (2009), 77–97.

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). http://arxiv.org/abs/1707.06347

[27] C. Seshadhri and Jan Vondrák. 2014. Is Submodularity Testable? *Algorithmica* 69 (2014), 1–25. https://link.springer.com/article/10.1007/s00453-013-9847-6

[28] Burr Settles. 2009. Active Learning Literature Survey. *University of Wisconsin-Madison Department of Computer Sciences* (2009).

[29] Richard S. Sutton and Andrew G. Barto. 2014. *Reinforcement Learning: An Introduction*. https://inst.eecs.berkeley.edu/~cs188/sp20/assets/files/SuttonBartoIPRLBook2ndEd.pdf

[30] Martin Černý and Michel Grabisch. 2024. Incomplete cooperative games with player-centered information. *Discrete Applied Mathematics* 346 (2024), 62–79. https://doi.org/10.1016/j.dam.2023.12.007

A ZERO DIVERGENCE REQUIRES ALL VALUES

We remark throughout the paper that we do not aim to achieve a zero divergence, as this is impossible for most of the superadditive functions until all of their values are known. To see this, consider superadditive set function $(f, 2^N)$, for which all of the conditions of superadditivity hold with strict inequality and suppose all but one values are known. Formally, let $(f, 2^N)$ with $2^N = 2^N \setminus \{\hat{S}\}$, be such that

$$f(S) + f(T) < f(S \cup T)$$

for all $S, T \subseteq N, S \cap T = \emptyset$. Then as $f(\hat{S}) < f(\hat{S} \cup T) - f(T)$ for every $T \subseteq N \setminus \hat{S}$ and $f(X) + f(\hat{S} \cap X) < f(\hat{S})$ for every $X \subseteq \hat{S}$, it follows from the definition of the lower/upper functions that

$$\underline{f}_{2^N}(\hat{S}) < f(\hat{S}) < \overline{f}_{2^N}(\hat{S}).$$

This means there is more than one $\mathcal{S}^n$-extension, which implies $\Delta_f(2^N) > 0$.

B ALGORITHM SPECIFICATIONS

Oracle and Offline Algorithms. Greedy strategies are computationally straightforward compared to their optimal counterparts. A single step using a greedy approach demands $O(g \cdot 2^n)$ time for a single sample. Here, g represents the time required to compute the divergence.

On the other hand, optimal algorithms prove to be computationally intensive due to the necessity of examining every sequence of actions, that is, every subset of 2^N. For each sample, the time complexity amounts to $O(g \cdot 2^{2^n})$. Despite our efforts to parallelize the computation where possible, given our available resources, we were only able to compute the complete optimal strategies for set function on ground sets of size up to 5. The computation for ground set of size 5 took roughly 10 hours. We estimate, that to compute the optimal strategies for ground set of size 6, for all steps, would take over 100 years.

The online variants are easier to parallelize, since each sample can be computed and evaluated independently, in contrast to the offline variants, where all the samples need to be computed, put together, averaged, and then finally evaluated.

When estimating the expectation with respect to $\mathcal{F}$, we used 3,000 samples for ground sets of size 4. For ground sets of size 5, we ended up using only 10 samples.

Reinforcement Learning. We apply reinforcement leaning [29] to approximate the optimal strategy of the online principal's problem. Namely, we use the Proximal policy optimization (PPO) [26]. We want to find a strategy of the principal which efficiently minimizes the average divergence. As such, we train PPO to minimize the divergence

$$r(2^N_{\tau-1}, S_\tau) = \Delta_f(2^N_{\tau-1} \cup \{S_\tau\}),$$

at every step, which provides a stronger learning signal compared to the final reward. This is equivalent to training over a distribution of the online principal's problems with uniformly distributed size t.

In our implementation, we parametrize both actor and critic of the PPO algorithm with a two-layer fully-connected neural network with 64 hidden units and ReLU activation each. To optimize the surrogate PPO objective, we used the Adam optimizer. The rest of the hyperparameteres can be found in Table 1.

Parameter	Value	Description
α_a	$3 \cdot 10^{-4}$	Actor learning rate
α_c	$1.5 \cdot 10^{-4}$	Critic learning rate
β	0.1	Entropy regularization
γ	1	Reward discounting rate
λ	0.95	Generalized advantage estimate
ε	0.2	Surrogate clip range
B	$5 \cdot 10^4$	Rollout buffer size
M	0.5	Max gradient norm
n_e	10	Number of training epochs

Table 1: Hyperparameters used during training.

Random Algorithm. The RANDOM algorithm is computationally simple, requiring only $O(g)$ time to compute a single sample. As such, it took only a few minutes to compute for ground sets of size 5. We again used 3,000 samples to approximate the expectation and the standard deviation, for sizes of ground sets of 4 and 5.

C CONCAVITY OF DIVERGENCE

Proposition 3. *Let $\mathcal{K} \subseteq 2^N$, $\mathcal{K}_0 \subseteq \mathcal{K}$ and $f \in \mathcal{S}^n$. Then the l_1 divergence is concave in the underlying set function f.*

PROOF. Let $\alpha \in [0, 1]$ and denote $h = \alpha f + (1 - \alpha) g$. Then we want to show

$$\Delta_{\alpha f}(\mathcal{K}) + \Delta_{(1-\alpha) g}(\mathcal{K}) \geq \Delta_h(\mathcal{K}).$$

Using the l_1 definition of divergence, one can derive that

$$\sum_{S \in 2^N} \alpha \left(\overline{f}_{\mathcal{K}}(S) - f(S), f(S) - \underline{f}_{\mathcal{K}}(S) \right) + \sum_{S \in 2^N} (1 - \alpha) \left(\overline{g}_{\mathcal{K}}(S) - g(S), g(S) - \underline{g}_{\mathcal{K}}(S) \right)$$

has to be less or equal to

$$\sum_{T \in 2^N} \left(\overline{h}_{\mathcal{K}}(T) - h(T), h(T) - \underline{h}_{\mathcal{K}}(T) \right).$$

We show that a pair of stronger conditions hold, specifically that the inequality holds for each element of the sum, and for the lower/upper function separately, so

$$\alpha \underline{f}_{\mathcal{K}}(T) + (1 - \alpha) \underline{g}_{\mathcal{K}}(T) \geq \underline{h}_{\mathcal{K}}(T),$$

and

$$\alpha \overline{f}_{\mathcal{K}}(T) + (1 - \alpha) \overline{g}_{\mathcal{K}}(T) \leq \overline{h}_{\mathcal{K}}(T).$$

In other words, we require the lower function to be convex, and the upper function to be concave. Consider the lower function first. From its definition, it must hold

$$\max_{\substack{S_1, \ldots, S_k \in \mathcal{K} \\ \bigcup_i S_i = S \\ S_i \setminus S_j \neq \emptyset}} \sum_{i=1}^{k} \alpha f(S_i) + \max_{\substack{S_1, \ldots, S_k \in \mathcal{K} \\ \bigcup_i S_i = S \\ S_i \setminus S_j \neq \emptyset}} \sum_{i=1}^{k} (1 - \alpha) g(S_i)$$

is larger or equal to

$$\max_{\substack{S_1, \ldots, S_k \in \mathcal{K} \\ \bigcup_i S_i = S \\ S_i \setminus S_j \neq \emptyset}} \sum_{i=1}^{k} \alpha f(S_i) + (1 - \alpha) g(S_i).$$

This inequality clearly holds, as in the former expression, the two maxima can be attained for different subsets of K, while in the latter expression, a joint subset has to be chosen, which restricts the number of possibilities.

Regarding the upper function, the argument is similar. It holds that

$$\min_{\substack{T \subseteq \mathsf{K} \\ S \not\subseteq T}} h(S) - \underline{h}(T \cap S)$$

is from the convexity of the lower game larger or equal to

$$\min_{\substack{T \subseteq \mathsf{K} \\ S \not\subseteq T}} h(S) - \alpha \underline{f}(T \cap S) - (1 - \alpha) \underline{g}(S)$$

which, since $h(S) = \alpha f(S) + (1-\alpha) g(S)$, is larger or equal to

$$\min_{\substack{T \subseteq \mathsf{K} \\ S \not\subseteq T}} \alpha \left[f(S) - \underline{f}(T \cap S) \right] + \min_{\substack{T \subseteq \mathsf{K} \\ S \not\subseteq T}} (1-\alpha) \left[g(S) - \underline{g}(T \cap S) \right].$$

D SUPERMODULARITY OF DIVERGENCE

Throughout this section, we often abuse the notation in the following sense. We write i instead of $\{i\}$, ij instead of $\{i, j\}$ and S instead of $\{S\}$. Further, we implicitly assume that the set function $f : 2^N \to \mathsf{R}$ is normalized as described in Appendix H. This is without the loss of generality, as this preserves the value of the gap, see Observation 1. As a consequence, the lower function $\underline{f}_{\mathsf{K}}(S)$ is zero for $\mathsf{K} = \mathsf{K}_0$ for any $S \in 2^N \cap \mathsf{K}_0$. First, we prove several lemmas.

Lemma 1. *Let $\mathsf{K}_0 \subseteq \mathsf{K} \subseteq 2^N \cap S$ and $f \in \mathsf{S}^n$. Then $\forall T \in 2^N$*

$$\underline{f}_{\mathsf{K}}(T) \leq \underline{f}_{\mathsf{K} \cup S}(T) \quad \text{and} \quad \overline{f}_{\mathsf{K}}(T) \geq \overline{f}_{\mathsf{K} \cup S}(T).$$

Proof. By definition of the lower function, a larger set of known subsets can only increase the value of the lower function. Consequently, the upper function cannot increase if the set of known values becomes larger.

Lemma 2. *Let $\mathsf{K}_0 \subseteq \mathsf{K} \subseteq 2^N, S \in 2^N \cap \mathsf{K}$, and $T \in 2^N$. Then*

$$T \supseteq S \implies \underline{f}_{\mathsf{K}}(T) = \underline{f}_{\mathsf{K} \cup S}(T),$$

and

$$S \supseteq T \implies \overline{f}_{\mathsf{K}}(T) = \overline{f}_{\mathsf{K} \cup S}(T).$$

Proof. The value of T for $\underline{f}_{\mathsf{K}}$, resp. $\underline{f}_{\mathsf{K} \cup S}$ is given by maximal partition of T in K, resp. T in $\mathsf{K} \cup S$. As $T \supseteq S$, revealing $f(S)$ does not appear in any partition of T, thus $\underline{f}_{\mathsf{K}}(T) = \underline{f}_{\mathsf{K} \cup S}(T)$.

The value of T for $\overline{f}_{\mathsf{K}}$, resp. $\overline{f}_{\mathsf{K} \cup S}$ is given by considering minimum over $X \in \mathsf{K}, T \not\subseteq X$. As $S \supseteq T$, it does not appear as a possible X. The only possibility for the change of the upper value would be that by reavealing S, we get $\underline{f}_{\mathsf{K} \cup S}(X \cap T) < \underline{f}_{\mathsf{K}}(X \cap T)$ for some X. This means $S \not\subseteq X \cap T$, but this is not possible as $S \supseteq T$.

Lemma 3. *Let $\mathsf{K}_0 \subseteq \mathsf{K} \subseteq 2^N \cap S$ and $T \in 2^N$, such that $\emptyset < S \setminus T$, $S \supseteq T$ and $T \neq S$. Then*

$$\overline{f}_{\mathsf{K}}(T) = \overline{f}_{\mathsf{K} \cup S}(T), \quad \text{and} \quad \underline{f}_{\mathsf{K}}(T) = \underline{f}_{\mathsf{K} \cup S}(T).$$

Proof. Follows immediately from the definition of the upper/lower functions.

The divergence Δ_f of a superadditive function f is supermodular, if for every $\hat{\mathsf{K}} \subseteq 2^N \cap \mathsf{K}_0$ it holds

$$\Delta_f(\hat{\mathsf{K}} \cup S \cup Z) - \Delta_f(\hat{\mathsf{K}} \cup S) \geq \Delta_f(\hat{\mathsf{K}} \cup Z) - \Delta_f(\hat{\mathsf{K}}). \tag{17}$$

Denote $\mathsf{K} = \hat{\mathsf{K}} \cup \mathsf{K}_0$. By definition of the divergence and the fact that $|\overline{f}_{\mathsf{K}}(T) - \underline{f}_{\mathsf{K}}(T)| = \overline{f}_{\mathsf{K}}(T) - \underline{f}_{\mathsf{K}}(T)$, the supermodularity condition can be rewritten as

$$\tilde{\sum_{T \subseteq 2^N}} \overline{f}_{\mathsf{L} \cup Z}(T) - \underline{f}_{\mathsf{L} \cup Z}(T) - \overline{f}_{\mathsf{L}}(T) \geq \underline{f}_{\mathsf{L}}(T) - \tilde{\sum_{T \subseteq 2^N}} \overline{f}_{\mathsf{K} \cup Z}(T) - \underline{f}_{\mathsf{K} \cup Z}(T) - \overline{f}_{\mathsf{K}}(T) \geq \underline{f}_{\mathsf{K}}(T).$$

where $\mathsf{L} = \mathsf{K} \cup S$.

D.1 Proof of Proposition 3

To prove this proposition, we show an even a stronger condition holds, i.e., $\forall T \in 2^N$

$$\overline{f}_{\mathsf{K} \cup S \cup Z}(T) - \underline{f}_{\mathsf{K} \cup S \cup Z}(T) - \overline{f}_{\mathsf{K} \cup S}(T) \geq \underline{f}_{\mathsf{K} \cup S}(T) - \overline{f}_{\mathsf{K} \cup Z}(T) - \underline{f}_{\mathsf{K} \cup Z}(T) - \overline{f}_{\mathsf{K}}(T) \geq \underline{f}_{\mathsf{K}}(T). \tag{18}$$

For $T \in \mathsf{K} \cup S$, condition (18) holds for any superadditive $f : 2^N \to \mathsf{R}$ with N of arbitrary size. To see this, notice $\overline{f}_{\mathsf{K} \cup S \cup Z}(T) = \underline{f}_{\mathsf{K} \cup S \cup Z}(T)$ and $\overline{f}_{\mathsf{K} \cup S}(T) = \underline{f}_{\mathsf{K} \cup S}(T)$, thus (18) reduces to

$$\overline{f}_{\mathsf{K} \cup S}(T) - \underline{f}_{\mathsf{K} \cup S}(T) \geq \overline{f}_{\mathsf{K}}(T) - \underline{f}_{\mathsf{K}}(T).$$

This inequality follows from Lemma 1. For the sake of clarity, we split the rest of the proof into two lemmas based on the size of N.

Lemma 4. *For a superadditive set function $f : 2^N \to \mathsf{R}$ where $|N| = 3$, the l_1-divergence Δ_f is a supermodular set function.*

Proof. Since, only subsets of size 2 are possibly unknown, their lower and upper bounds do not change when another subset of size 2 is revealed. Thus, for $\mathsf{K}_0 \subseteq \mathsf{K} \subseteq 2^N \cap \{S, Z\}$ and $T \notin \mathsf{K} \cup S$, it holds

$$\overline{f}_{\mathsf{K} \cup S \cup Z}(T) - \underline{f}_{\mathsf{K} \cup S \cup Z}(T) = \overline{f}_{\mathsf{K} \cup S}(T) - \underline{f}_{\mathsf{K} \cup S}(T)$$

and similarly

$$\overline{f}_{\mathsf{K} \cup Z}(T) - \underline{f}_{\mathsf{K} \cup Z}(T) = \overline{f}_{\mathsf{K}}(T) - \underline{f}_{\mathsf{K}}(T),$$

which means (18) holds.

Lemma 5. *For a superadditive set function $f : 2^N \to \mathsf{R}$ where $|N| = 4$, the l_1-divergence Δ_f is a supermodular set function.*

Proof. To prove (18) for $\mathsf{K}_0 \subseteq \mathsf{K} \subseteq 2^N \cap \{S, Z\}$ and $T \notin \mathsf{K} \cup S$, we distinguish several cases based on the relation between Z and T. All of the cases follow a similar pattern.

(1) $Z \supseteq T$:
Since $|N| = 4$ and Z, T are unknown in $\mathsf{K} \cup S$, they must be of form $Z = \{i, j\}$ and $T = \{i, j, k\}$. By Lemma 2, $\overline{f}_{\mathsf{K} \cup Z}(T) = \overline{f}_{\mathsf{K}}(T)$ and $\overline{f}_{\mathsf{K} \cup S \cup Z}(T) = \overline{f}_{\mathsf{K} \cup S}(T)$, thus (18) reduces to

$$\underline{f}_{\mathsf{K} \cup S \cup Z}(T) - \underline{f}_{\mathsf{K} \cup S}(T) \geq \underline{f}_{\mathsf{K} \cup Z}(T) - \underline{f}_{\mathsf{K}}(T).$$

For a contradiction, suppose the converse holds. As, by Lemma 1, $\underline{f}_{\mathsf{K} \cup Z}(T) - \underline{f}_{\mathsf{K}}(T) \geq 0$, this means $\underline{f}_{\mathsf{K} \cup S \cup Z}(T) >$

$\underline{f}_{K \cup S}(T) = f(Z)$, $f(\{k\})$. Thus $\underline{f}_{K \cup S \cup Z}(T) = f(Z)$, $f(\{k\})$. As both Z and $\{k\}$ lie in $K \cup S$, it is one of the possible partition of T, thus it follow $\underline{f}_{K \cup S \cup Z}(T) \geq \underline{f}_{K \cup Z}(T)$. By Lemma 1, the opposite inequality holds. Thus, $\underline{f}_{K \cup S \cup Z}(T) = \underline{f}_{K \cup Z}(T)$ and the converse of (18) reduces to $\underline{f}_{K}(T) > \underline{f}_{K \cup S}(T)$, a contradiction with Lemma 1.

(2) $T \subset Z$: Similarly to the first case, we have $T = \{i, j\}$, $Z = \{i, j, k\}$ and by Lemma 2, (18) reduces to

$$\overline{f}_{K \cup S}(T) \geq \overline{f}_{K \cup S \cup Z}(T) \geq \overline{f}_{K}(T) \geq \overline{f}_{K \cup Z}(T).$$

For a contradiction, suppose the converse holds. As $Z \supseteq K \cup S, K \cup Z$ and $T \cap Z = \{k\} \in \mathcal{K}_0$, we have $\overline{f}_{K \cup S \cup Z}(T) = f(Z) \geq f(\{k\}) \geq \overline{f}_{K \cup Z}(T)$, which by Lemma 1 implies $\overline{f}_{K \cup S \cup Z}(T) = \overline{f}_{K \cup Z}(T)$, thus (18) reduces to $\overline{f}_{K \cup S}(T) > \overline{f}_{K}(T)$, a contradiction.

(3) $Z = T$: It holds $\overline{f}_{K \cup S \cup Z}(T) = \underline{f}_{K \cup S \cup Z}(T)$ and $\overline{f}_{K \cup Z}(T) = \underline{f}_{K \cup Z}(T)$, thus (18) reduces to

$$\overline{f}_{K \cup S}(T) \geq \underline{f}_{K \cup S}(T) \geq \overline{f}_{K}(T) \geq \underline{f}_{K}(T)$$

which holds by Lemma 1.

(4) $Z \setminus T = \emptyset$: In this case, $Z = \{i, j\}$, $T = \{k, \ell\}$ and (18) reduces to

$$\overline{f}_{K \cup S}(T) \geq \overline{f}_{K \cup S \cup Z}(T) \geq \overline{f}_{K}(T) \geq \overline{f}_{K \cup Z}(T)$$

as $\underline{f}_{K \cup S \cup Z}(T) = \underline{f}_{K \cup S}(T)$ and $\underline{f}_{K \cup Z}(T) = \underline{f}_{K}(T)$. For a contradiction, if $>$ holds, similarly to previous cases, $\overline{f}_{K \cup S \cup Z}(T) > \overline{f}_{K \cup S}(T)$. As Z is not superset of T, it must be the case that the lower bound of X such that $X \setminus T = \emptyset$ and $T \cup X \in K \cup S$ changed when Z was added. As $T = \{i, j\}$, only such X is Z, therefore $\overline{f}_{K \cup S \cup Z}(T) = v(N) \neq v(Z) = \overline{f}_{K \cup Z}(T)$, yielding $\overline{f}_{K \cup S \cup Z}(T) = \overline{f}_{K \cup Z}(T)$. This reduces the inequality to $\overline{f}_{K \cup S}(T) > \overline{f}_{K}(T)$, a contradiction.

(5) $Z \setminus T \subset \emptyset$ and $Z \not\supseteq T$ and $T \not\supseteq Z$: By Lemma 3, it holds $\underline{f}_{K \cup S}(T) = \underline{f}_{K}(T)$ and $\underline{f}_{L \cup S}(T) = \underline{f}_{L}(T)$, thus (18) reduces to

$$\overline{f}_{K \cup S}(T) \geq \overline{f}_{K \cup S \cup Z}(T) \geq \overline{f}_{K}(T) \geq \overline{f}_{K \cup Z}(T).$$

For a contradiction, let $>$ hold, thus $\overline{f}_{K \cup S}(T) > \overline{f}_{K \cup S \cup Z}(T)$. As Z is not superset of T, it must be the case that the lower bound of X such that $X \setminus T = \emptyset$ and $T \cup X \in L$ changed when Z was added. As $Z \setminus T \subset \emptyset$, no such X exists, a contradiction.

D.2 Proof of Proposition 5

We begin by stating a technical lemma which will be useful later.

Lemma 6. *For every superadditive function f on ground set $|N| \geq 5$, satisfying $f(x) = 0$ for every $x \in N$, there are $i, j, k, l \in N$ such that*

$$f(ij) \quad f(jk) \quad f(kl).$$

Proof. Let $\{x, y\}$ be a subset such that $f(xy)$ is the smallest value among all subsets of size 2. Further, let k_1, k_2, k_3 be elements different from x, y such that

$$f(xy) \quad f(yk_1) \quad f(yk_2) \quad f(yk_3).$$

Now either $f(yk_1) \quad f(k_1k_3)$, which gives us x, y, k_1, k_3 satisfying

$$f(xy) \quad f(yk_1) \quad f(k_1k_3)$$

or $f(k_1k_3) < f(yk_1)$ which gives us k_3, k_1, j, k_2 satisfying

$$f(k_3k_1) \quad f(k_1j) \quad f(jk_2).$$

Proposition 5. *For $n \geq 5$, let $g \in \mathbb{S}^n$ be a superadditive function with supermodular l_1-divergence. For $f \in \mathbb{S}^n$ defined as $f(S) = g(S) - \sum_{i \in S} g(i)$, there are $i, j, k, l \in N$ such that $f(ij) \quad f(jk) \quad f(kl)$, and*

$$f(ij) \quad \frac{1}{2^{n-3}} f(kl). \tag{19}$$

Proof. Let $g \in \mathbb{S}^n$ be the underlying set function with supermodular divergence. Let $f \in \mathbb{S}^n$ be the normalization of g s.t. the singletons have values zero, see Appendix H for more details. By Proposition 2, this preserves supermodularity of the divergence.

Consider distinct $i, j, k, l \in N$ such that $f(ij) \quad f(jk) \quad f(kl)$. This quadruplet always exists (see Lemma 6). Denote $f(ij) = f(jk) \quad \varepsilon$ and $f(kl) = f(jk) + \delta$ for some $\varepsilon, \delta \geq 0$. From supermodularity of the l_1 divergence, it holds

$$\sum_{T \in 2^N} \overline{f}_{L \cup Z}(T) \quad \underline{f}_{L \cup Z}(T) \quad \overline{f}_{L}(T) + \underline{f}_{L}(T)$$
$$\sum_{T \in 2^N} \overline{f}_{K \cup Z}(T) \quad \underline{f}_{K \cup Z}(T) \quad \overline{f}_{K}(T) + \underline{f}_{K}(T) .$$

for every $K \quad 2^N \cap \{S, Z\}$ and $L = K \cup S$. Consider $K = K_0 \cup \{\{j, k\}\}$, $S = \{i, j\}$, and $Z = \{k, l\}$ and compare sum members corresponding to every T, i.e.

$$L_T = \overline{f}_{K \cup S}(T) \quad \underline{f}_{K \cup S \cup Z}(T) \quad \overline{f}_{K \cup S}(T) + \underline{f}_{K \cup S}(T),$$
$$R_T = \overline{f}_{K \cup Z}(T) \quad \underline{f}_{K \cup Z}(T) \quad \overline{f}_{K}(T) + \underline{f}_{K}(T).$$

We shall show that for different T, either $L_T = R_T$, $L_T = R_T + \varepsilon \quad f(jk) = R_T \quad f(ij)$, or $L_T = R_T + f(kl)$. We will arrive at the criterion by considering the number of occurrences of each case.

In the reminder of the proof, we distinguish different cases based on the relation of $T \in 2^N \cap K$ and $\{i, j, k, l\}$.

(1) $i, j, k, l \in T$:
Using Lemma 2, we get

$$L_T = \underline{f}_{K \cup S}(T) \quad \underline{f}_{K \cup S \cup Z}(T),$$
$$R_T = \underline{f}_{K}(T) \quad \underline{f}_{K \cup Z}(T),$$

which can be expressed as

$$L_T = f(jk) \quad f(jk) + \varepsilon \quad f(jk) \quad \delta,$$
$$R_T = f(jk) \quad f(jk) \quad \delta.$$

It holds $L_T = R_T + \varepsilon \quad f(jk) = R_T \quad f(ij)$.

(2) $i, j, k, l \notin T$:
In this case, the lower bound on $f(T)$ is unaffected by the knowledge of S, Z, so

$$L_T = \overline{f}_{K \cup S \cup Z}(T) \quad \overline{f}_{K \cup S}(T),$$
$$R_T = \overline{f}_{K \cup Z}(T) \quad \overline{f}_{K}(T).$$

Since the only superset of T in $\mathsf{K} \setminus S \setminus Z$ is N, the upper bound (7) reduces to

$$\overline{f}_{\mathsf{L}}(T^{\circ}) = f(N^{\circ}) \quad \underline{f}_{\mathsf{L}}(N \setminus T^{\circ}) = 1 \quad \underline{f}_{\mathsf{L}}(N \setminus T^{\circ}),$$

for every $\mathsf{L} \supseteq \{\mathsf{K}, \mathsf{K} \setminus S, \mathsf{K} \setminus Z, \mathsf{K} \setminus S \setminus Z\}$. Thus

$$L_T = \underline{f}_{\mathsf{K} \setminus S}(N \setminus T^{\circ}) \quad \underline{f}_{\mathsf{K} \setminus S \setminus Z}(N \setminus T^{\circ}),$$
$$R_T = \underline{f}_{\mathsf{K}}(N \setminus T^{\circ}) \quad \underline{f}_{\mathsf{K} \setminus Z}(N \setminus T^{\circ}).$$

Further, $i, j, k, l \in N \setminus T$, which means

$$L_T = f(jk^{\circ}) \quad f(jk^{\circ}) + \varepsilon \quad f(jk^{\circ}) \quad \delta,$$
$$R_T = f(jk^{\circ}) \quad f(jk^{\circ}) \quad \delta.$$

As in the previous case, $L_T = R_T + \varepsilon \quad f(jk^{\circ}) = R_T \quad f(ij^{\circ})$.

(3) $S = T$:
In this case, since $\underline{f}_{\mathsf{L}}(T^{\circ}) = \overline{f}_{\mathsf{L}}(T^{\circ}) = f(T^{\circ})$ if $T \in \mathsf{L}$ and by Lemma 2, it holds

$$L_T = 0,$$
$$R_T = \overline{f}_{\mathsf{K} \setminus Z}(T^{\circ}) \quad \overline{f}_{\mathsf{K}}(T^{\circ}) = \underline{f}_{\mathsf{K}}(N \setminus T^{\circ}) \quad \underline{f}_{\mathsf{K} \setminus Z}(N \setminus T^{\circ}).$$

This means $R_T = 0 \quad f(jk^{\circ}) \quad \delta$, therefore it holds $L_T = R_T + f(jk^{\circ}) + \delta = R_T + f(kl^{\circ})$.

(4) $Z = T$:
In this case, by the fact that $\underline{f}_{\mathsf{L}}(T^{\circ}) = \overline{f}_{\mathsf{L}}(T^{\circ}) = f(T^{\circ})$ if $T \in \mathsf{L}$, it holds

$$L_T = \underline{f}_{\mathsf{K} \setminus S}(T^{\circ}) \quad \overline{f}_{\mathsf{K} \setminus S}(T^{\circ}),$$
$$R_T = \underline{f}_{\mathsf{K}}(T^{\circ}) \quad \overline{f}_{\mathsf{K}}(T^{\circ}).$$

Since $S \subseteq T$, it holds $\underline{f}_{\mathsf{K} \setminus S}(T^{\circ}) = \underline{f}_{\mathsf{K}}(T^{\circ})$, so we can rewrite

$$L_T = \underline{f}_{\mathsf{K} \setminus S}(N \setminus T^{\circ}) \quad f(N^{\circ}) = f(jk^{\circ}) \quad \varepsilon,$$
$$R_T = \underline{f}_{\mathsf{K}}(N \setminus T^{\circ}) \quad f(N^{\circ}) = \quad f(N^{\circ}).$$

Thus it holds $L_T = R_T + f(jk^{\circ}) \quad \varepsilon = R_T + f(ij^{\circ})$.

(5) $; \subset Z \setminus T, Z \subseteq T$, and $T \subseteq Z$ or
$; \subset S \setminus T, S \subseteq T$, and $T \subseteq S$: By Lemma 1, and Lemma 3, one immediately arrives at $L_T = R_T = 0$.

For supermodularity to hold, we want to have $\bigwedge_{T \in 2^N} L_T \quad R_T \geq 0$, which translates to $(c_1 + c_2 f(ij^{\circ}) + f(kl^{\circ}) + f(ij^{\circ}) \quad 0 \geq 0$, where $c_1 = (n^{\circ} = 2^n \quad 4 \quad 1$ and $c_2 = (n^{\circ} = 2^n \quad 4 \quad n + 3$ is the number of terms satisfying conditions 1 and 2, respectively. This yields

$$(2^n \quad 3 \quad n + 2^{\circ} f(ij^{\circ}) + f(kl^{\circ}) + f(ij^{\circ}) \quad 0 \geq 0,$$

or

$$f(ij^{\circ}) \quad \frac{1}{2^n \quad 3 \quad n + 2} f(kl^{\circ}) \quad \frac{1}{2^n \quad 3} f(kl^{\circ}).$$

Corollary 1. *Since the* supermodular(n) *set includes functions which have the same value after normalization, not even supermodularity of the underlying function guarantees supermodularity of it's divergence. However, the supermodularity constraints typically hold within floating point precision – see Appendix E for more details.*

E THE LARGEST SUBSETS STRATEGY

We have shown how the Largest Subsets strategy performs on the supermodular(n) distribution. The results show that this simple heuristic performs exceptionally well. In this section we argue that this simple heuristic does not work well in general. Figure 4 shows analogous results to Figure 3 from the main text. Clearly, the Largest Subsets heuristic's performance drops significantly compared to the supermodular(n) distribution.

The distributions shown in Figure 4 are from the family of graph functions, as described in Section 4.1. The cycle(n) is the same as in Section 4.1. The remaining distributions are based on an extension of the graph functions, where there is an additional weight function $w : E \to \mathbb{R}$, and

$$f(S^{\circ}) = \tilde{\sum}_{e \in E(S^{\circ})} w(e^{\circ}).$$

The remaining distributions are then defined on a clique, where for each $e \in E$, the value of $w(e^{\circ})$ is independently sampled from a distribution.

For the beta distribution, the weights are from the Beta distribution, with parameters $\alpha = 4$, $\beta = 5$. The increasing distribution uses weights from the interval $[0, 1]$, such that the probability density function is zero at zero, and then linearly increases towards one. The poiss distribution uses the Poisson distribution for its weights, with $\lambda = 0.01$. Note that due to the low value of λ, this distribution has high variance.

We hypothesize that the reason behind the exceptional performance of the Largest Subsets heuristic lies in concentration of supermodular functions. [2] This phenomenon demonstrates itself by large gap in values of subsets of size i and $i + 1$, often by orders of magnitude. As such, revealing all subsets of size $n \quad 1$ already upper-bounds the rest of the values exceptionally well. One side effect of this gap is that the set functions have almost supermodular divergence. In fact, the inequalities constraints (17) are typically satisfied within floating point precision.

F ORACLE ALGORITHMS

Following are the pseudocodes of Oracle algorithms discussed in Section 4.2.

Algorithm 3: Oracle Optimal

Input: characteristic function $f \in \mathbb{R}^{2^N}$, number of steps t

$\overline{\mathsf{K}} \quad 2^N \setminus \mathsf{K}_0$
$\{S_i\}_{i=1}^{t} \quad \operatorname{argmin}_{\mathsf{S}} \; \overline{\mathsf{K}}(|\mathsf{S}|=t) \; \Delta_f(\mathsf{K}_0 \setminus \mathsf{S}^{\circ})$
return $\{S_i\}_{i=1}^{t}$

G ADDITIONAL EXPERIMENTAL RESULTS

In this section, we present additional experimental results. Similar to Figure 2, we show the cumulative probability of a subset of a given size being selected by each algorithm. We do so for cycle(5) and supermodular(5) in Figures 6 and 8, respectively. We also present the probability per-subset in Figures 5, 7, and 9.

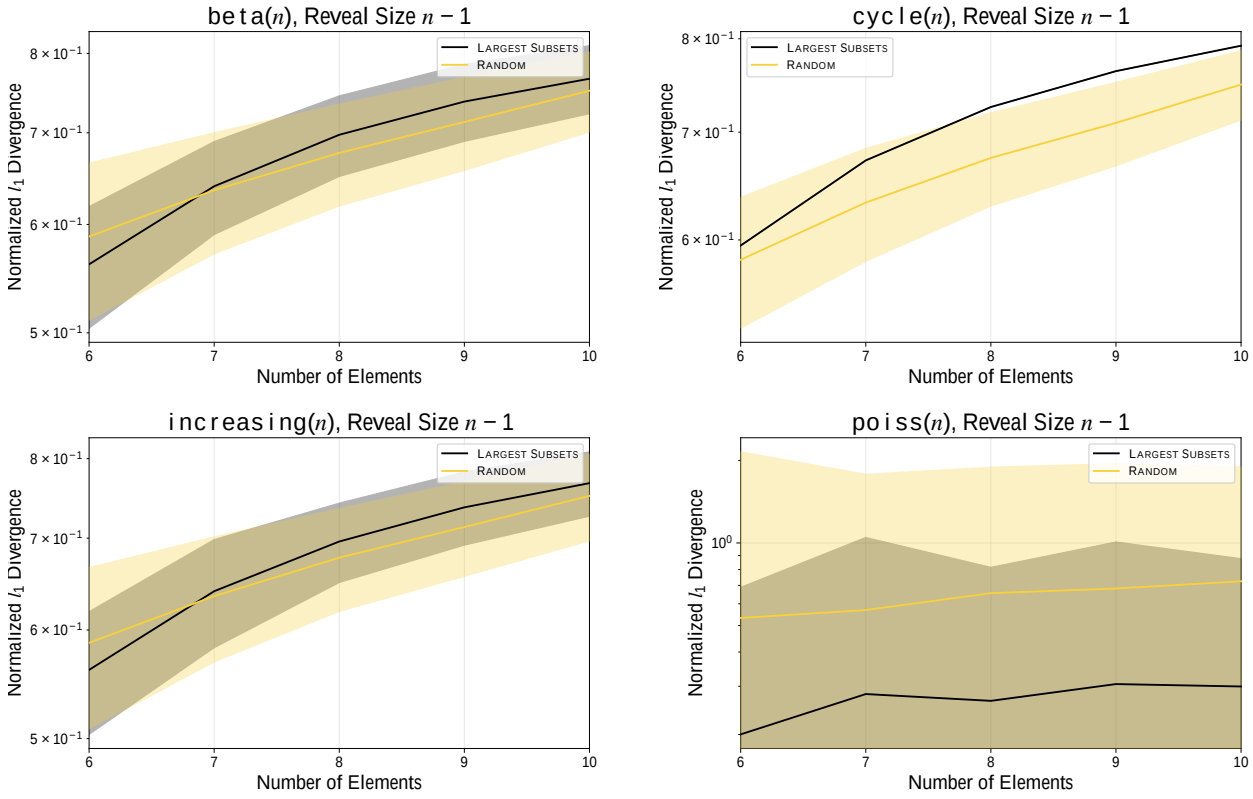

Figure 4: Comparison of the normalized divergence for various distributions of superadditive set functions as a function of the ground set size n. The figure contrasts the expected normalized divergence when subsets are randomly chosen with the scenario where all subsets of size n 1 are selected.

Algorithm 4: Oracle Greedy

Input: characteristic function f 2 S^n, number of steps t

if $t > 1$ **then**
  $\{S_i\}_{i=1}^{t-1}$ Oracle Greedy(f, t 1)
end
$\overline{K}$ 2^N n $^1K_0$ [$\{S_i\}_{i=1}^{t-1}$)
S_t argmin$_{S2\overline{K}}$ $\Delta_f{}^1K_0$ [$\{S_i\}_{i=1}^{t-1}$ [$\{S\}^o$
return $\{S_i\}_{i=1}^{t}$

H NORMALIZATION

In our application, it is convenient to transform a set function f by an affine mapping such that after the transformation, the values of the singletons are equal to 0 and the value of the ground set is equal to 1. After such transformation, the minimal information K_0 is trivial.

Formally, let $\alpha > 0$ and β 2 R^n such that

$$g^1S^o = \alpha \quad f^1S^o , \quad \tilde{O}_{i2S} \beta_i. \tag{20}$$

By considering

$$\beta_i = \frac{f^1\{i\}^o}{f^1N^o \quad \dot{I}_{i2N} f^1\{i\}^o} \quad \text{and} \quad \alpha = \frac{1}{f^1N^o},$$

we achieve the desired transformation. Proposition 2 describes the effect on the divergence when such a transformation is applied. We summarize it also here.

Observation 1. *The divergence* Δ *satisfies*

$$\Delta_{\alpha f , \beta}{}^1K^o = \alpha \quad \Delta_f{}^1K^o \quad \dot{I}_{i2N} \beta_i \tag{21}$$

where $\alpha > 0$, $^1\alpha v$, $\beta^o{}^1S^o$ B αv^1S^o , *and* β_i 2 R *for* i 2 N.

Proof. It is a standard result [24] that f 2 S^n =) αf , β 2 S^n. From these two results, we have

$$\begin{aligned}\Delta_{\alpha f , \beta}{}^1K^o &= \overline{\alpha f , \beta}_{K_0[K} \quad \underline{\alpha f , \beta}_{K_0[K} \\ &= \alpha \quad \overline{f}_{K_0[K} \quad \underline{f}_{K_0[K} \\ &= \alpha \quad \Delta_f{}^1K^o.\end{aligned}$$

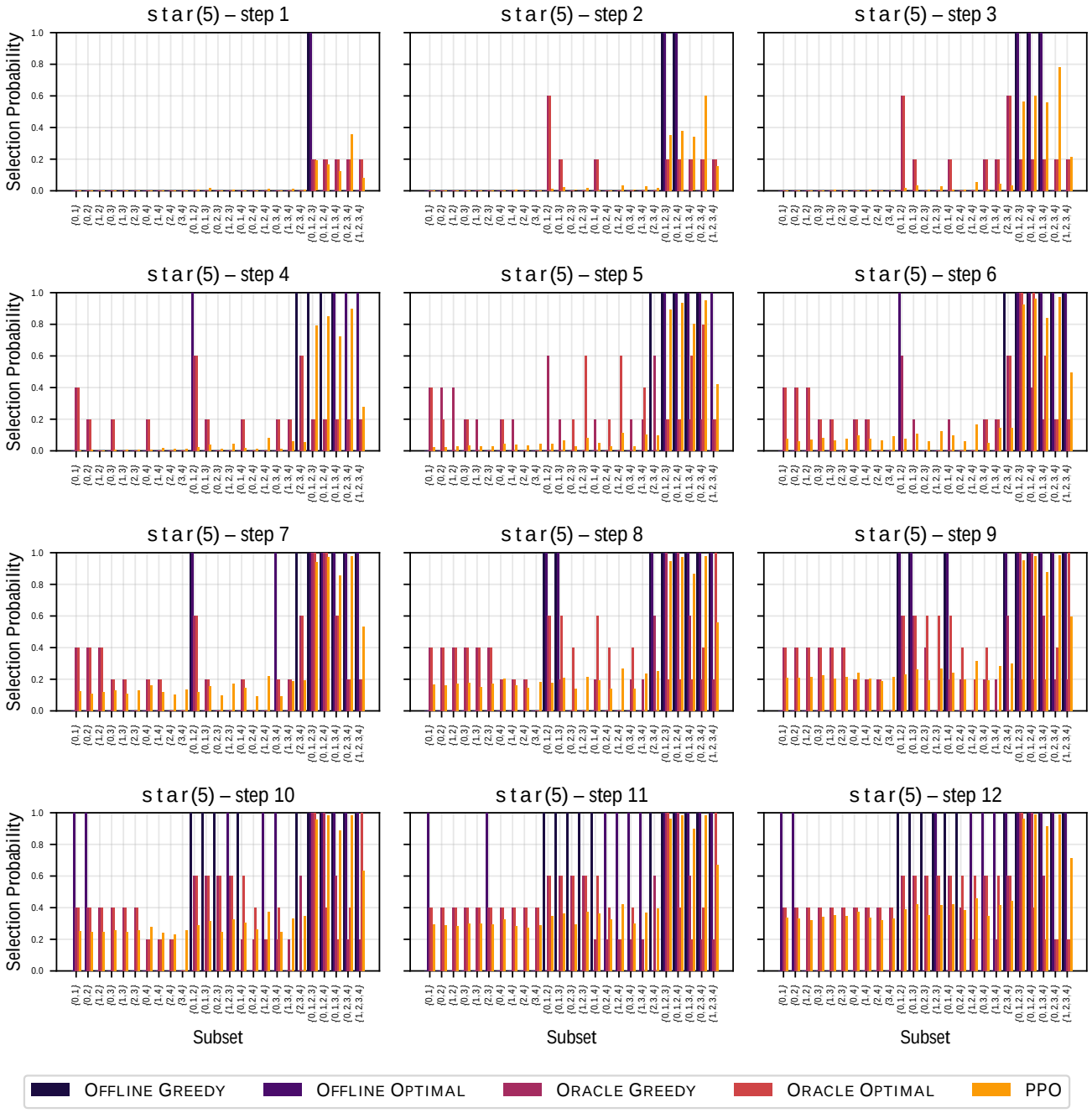

Figure 5: Percentage of subsets selected up to step twelve for `star(5)` and each algorithm. Results show clear preference for larger subsets, i.e. they contribute more information about the set functions on average. The oracle algorithms favor smaller subsets earlier, suggesting the representation of a specific function can efficiently use even smaller subsets. PPO **initially behaves similarly to the offline algorithms. At later steps, it uses the previously obtained values and its selections resemble the oracle methods. See Figure 2 for plot showing subsets of given size.**

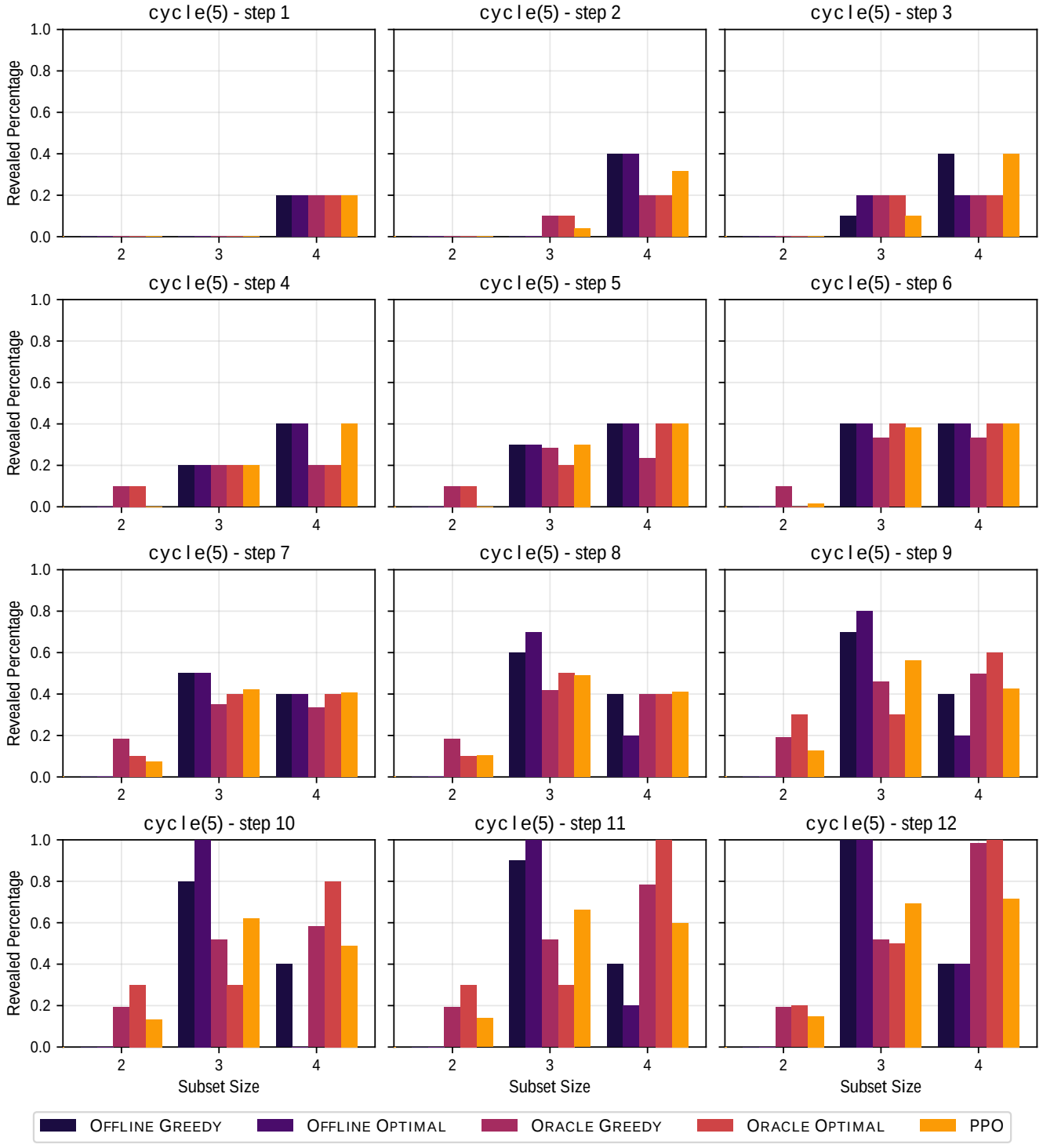

Figure 6: Percentage of subsets selected up to step twelve for `cycle(5)` and each algorithm. Results show clear preference for larger subsets, i.e. they contribute more information about the set function on average. The oracle algorithms favor smaller subsets earlier, suggesting the representation of a specific set function can efficiently use even smaller subsets. PPO **initially behaves similarly to the offline algorithms. At later steps, it uses the previously obtained values and its selections resemble the oracle methods. See Figure 7 for plot showing individual subsets.**

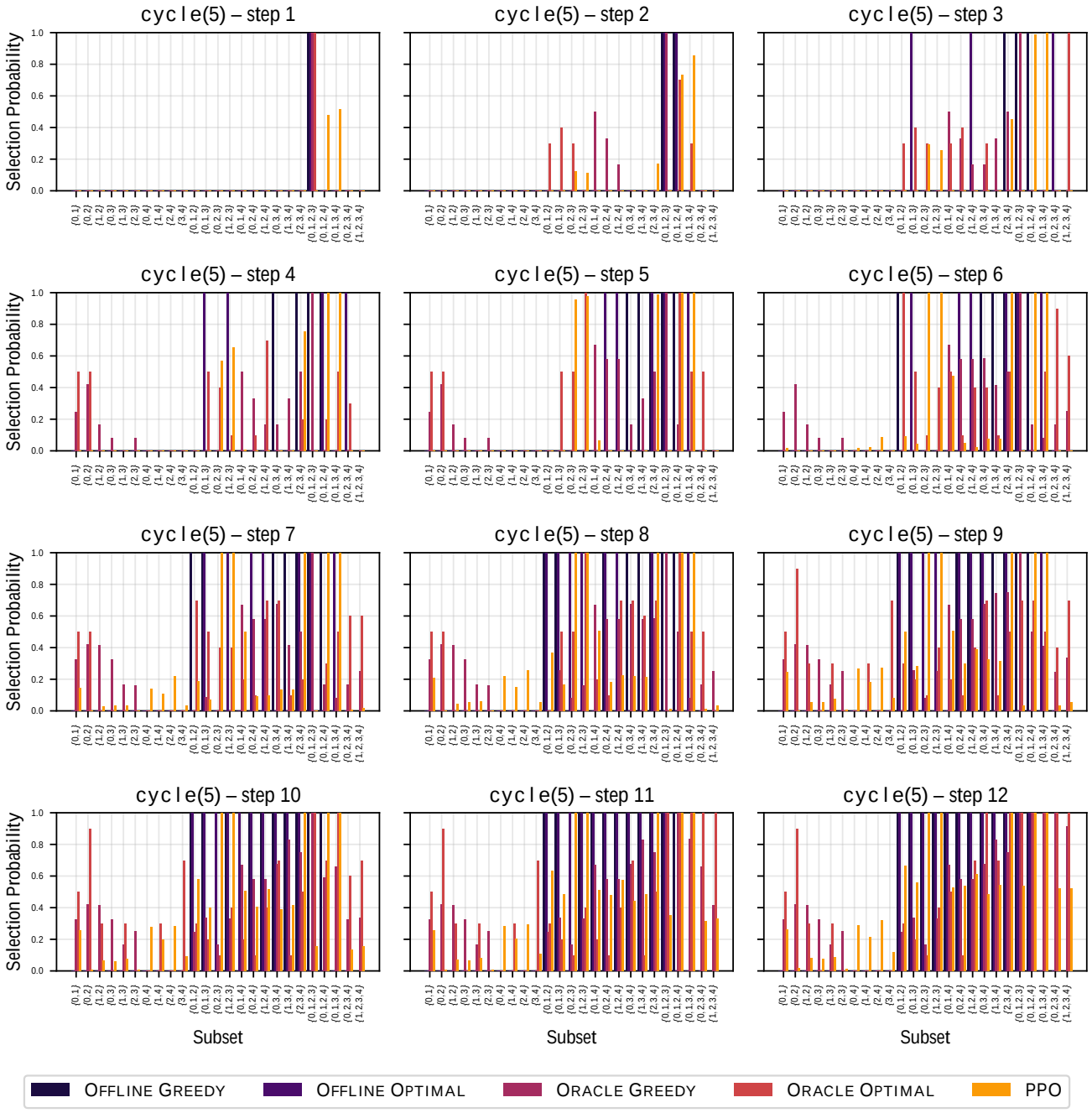

Figure 7: Percentage of subsets selected up to step twelve for `cycle`(5) and each algorithm. Results show clear preference for larger subsets, i.e. they contribute more information about the set function on average. The oracle algorithms favor smaller subsets earlier, suggesting the representation of a specific set function can efficiently use even smaller subsets. PPO **initially behaves similarly to the offline algorithms. At later steps, it uses the previously obtained values and its selections resemble the oracle methods. See Figure 6 for plot showing subsets of given size.**

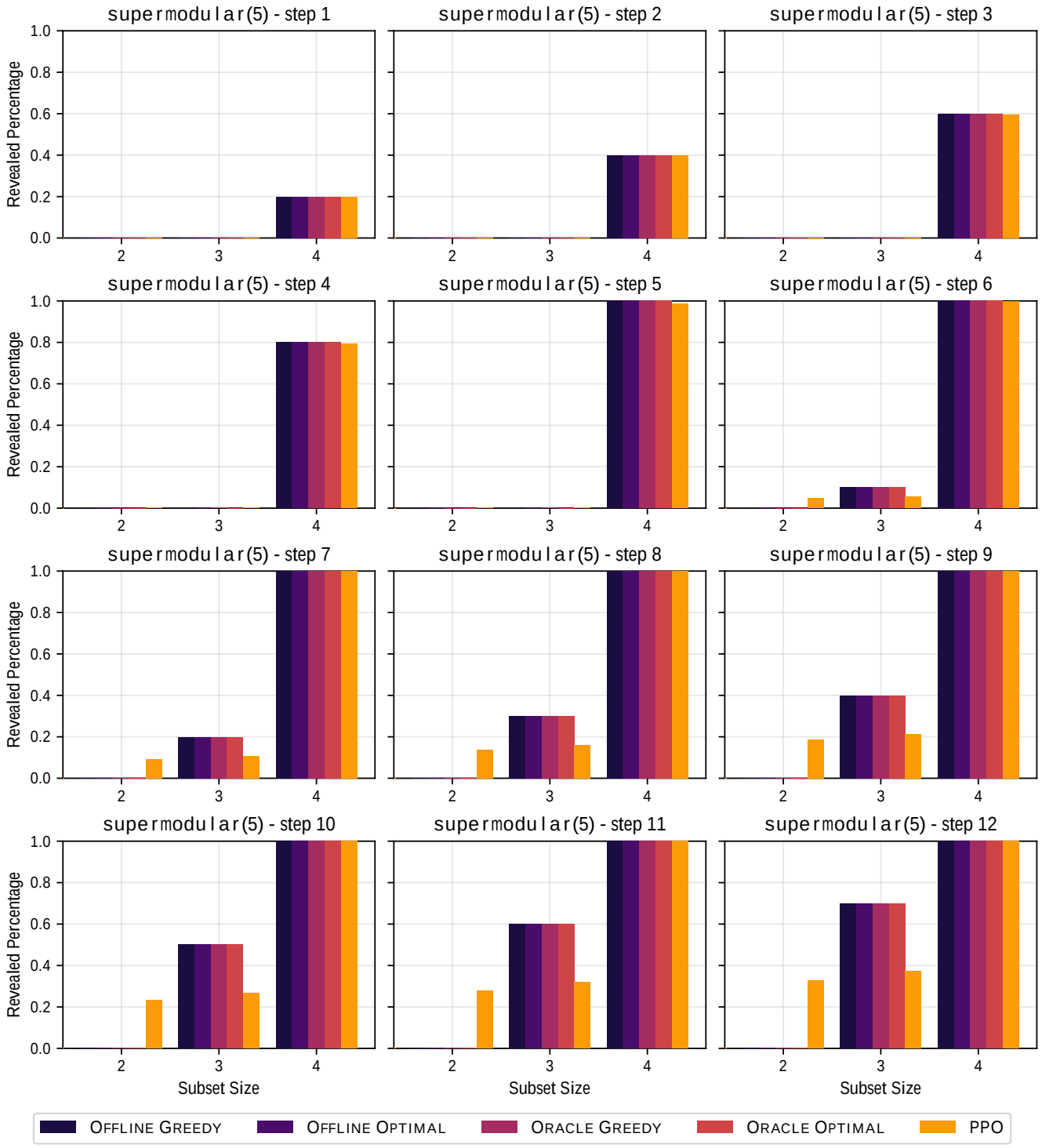

Figure 8: Percentage of subsets selected up to step twelve for `supermodular(5)` and each algorithm. Results show clear preference for larger subsets, i.e. they contribute more information about the set function on average. The oracle algorithms favor smaller subsets earlier, suggesting the representation of a specific set function can efficiently use even smaller subsets. PPO initially behaves similarly to the offline algorithms. At later steps, it uses the previously obtained values and its selections resemble the oracle methods. See Figure 9 for plot showing individual subsets.

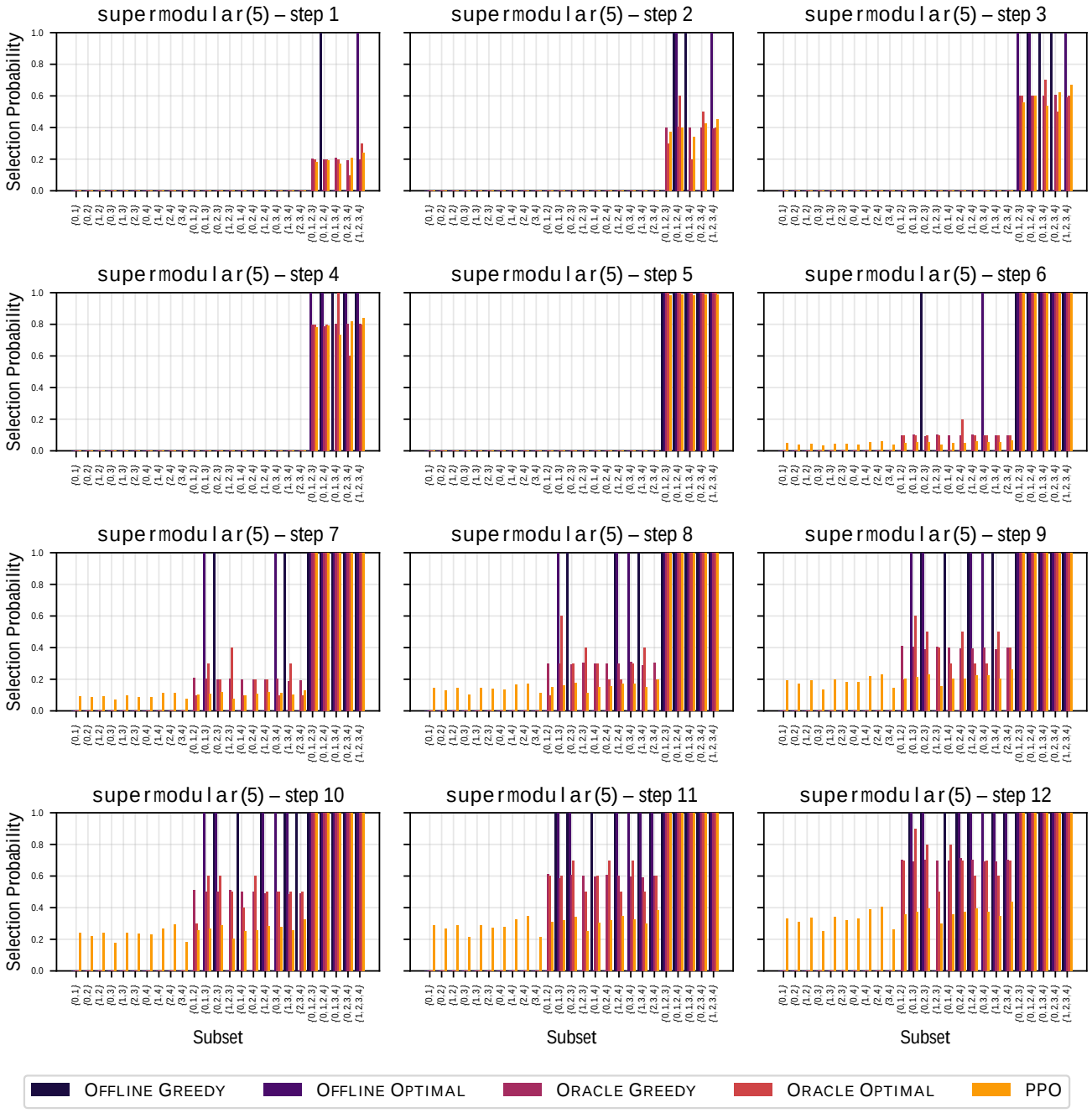

Figure 9: Percentage of subsets selected up to step twelve for `supermodular(5)` and each algorithm. Results show clear preference for larger subsets, i.e. they contribute more information about the set function on average. The oracle algorithms favor smaller subsets earlier, suggesting the representation of a specific set function can efficiently use even smaller subsets. PPO **initially behaves similarly to the offline algorithms. At later steps, it uses the previously obtained values and its selections resemble the oracle methods. See Figure 8 for plot showing subsets of the same size.**